

Irina Munteanu, Daniel Ioan
Universitatea "Politehnica" din București
Laboratorul de Metode Numerice

Utilizarea Instrumentelor Unix în Știință și Inginerie

1998

Cuprins

Introducere	9
1 Utilizarea sistemului de operare Unix	17
1.1 Intrarea în sistem	21
1.2 Lucrul cu ferestre XWindow	21
1.3 Ieșirea din sistem	24
1.4 Sesiunea de lucru și configurarea ei	24
1.5 Interpretorul de comenzi	26
1.6 Comenzi pentru lucrul cu fișiere și directoare	26
1.6.1 Structura de directoare și fișiere	26
1.6.2 Comenzi pentru lucrul cu fișiere	29
1.6.3 Comenzi pentru lucrul cu directoare	30
1.6.4 Schimbarea drepturilor asupra fișierelor/directoarelor	30
1.6.5 Căutări	31
1.6.6 Comparări de fișiere	31
1.7 Indirectări	32
1.8 Imprimarea fișierelor	32
1.9 Lansarea și oprirea comenzilor	33
1.10 Comenzi pentru controlul proceselor și utilizatorilor	33
1.11 Ajutor!	34
1.12 Transferul fișierelor între DOS și Unix	34
1.13 Diverse utilitare	34
1.14 Exerciții	35

2 Editarea textelor	
— vi și Emacs —	39
2.1 Editarea textelor cu vi	40
2.1.1 Apelare	40
2.1.2 Moduri de lucru	41
2.1.3 Modul comandă	42
2.1.4 Setare opțiuni	45
2.1.5 Câteva facilități ale interfeței grafice cu utilizatorul	45
2.2 Editarea textelor cu emacs	47
2.2.1 Lansare și ieșire din Emacs	47
2.2.2 Ecranul Emacs	47
2.2.3 Comenzi de deplasare a cursorului	49
2.2.4 Comenzi de ștergere	49
2.2.5 Comenzi de formatare-indentare	49
2.2.6 Comenzi pentru blocuri	50
2.2.7 Comenzi pentru fișiere (încărcare-salvare)	50
2.2.8 Căutare și înlocuire	50
2.2.9 Comenzi pentru ferestre	50
2.2.10 Lucrul cu macrocomenzi	50
2.2.11 Comenzi pentru dezvoltarea programelor	51
2.2.12 Diverse	51
2.3 Exerciții	52
3 Poșta electronică	
— elm —	55
Conceptele specifice poștei electronice	57
3.1 Apelare	59
3.2 Structura ecranului elm	59
3.3 Citirea mesajelor primite	59
3.4 Trimitere de mesaje	60
3.5 Pseudonime (aliasuri)	61

3.6	Ștergerea / mutarea mesajelor în altă casuță poștală	62
3.7	Modificarea opțiunilor	62
3.8	Ieșirea din <code>elm</code>	64
3.9	Exerciții	64
4	Editarea documentelor	
	— L^AT_EX —	67
4.1	Etapele utilizării L ^A T _E X	70
4.2	Structura unui fișier L ^A T _E X	70
4.3	Comenzi L ^A T _E X	74
4.4	Moduri de lucru L ^A T _E X	75
	4.4.1 Modul paragraf	75
	4.4.2 Modul matematic	77
	4.4.3 Modul LR	78
4.5	Cadru L ^A T _E X	79
4.6	Macrodefiniții L ^A T _E X	80
4.7	Aspecte speciale ale utilizării L ^A T _E X	81
	4.7.1 Caractere speciale	81
	4.7.2 Accente	81
	4.7.3 Despărțirea în silabe	82
4.8	Structura unei pagini	83
4.9	Fișiere rezultate în urma procesării cu L ^A T _E X	83
4.10	Câteva exemple	83
	4.10.1 Fișierul <code>latex_head.tex</code>	83
	4.10.2 Un exemplu de document	85
	4.10.3 Un exemplu de articol în stilul IEEE Trans. on Magnetics	92
4.11	Gestionarea referințelor bibliografice folosind BIB _T E _X	96
	4.11.1 Structura unui fișier BIB _T E _X	97
	4.11.2 Introducerea referințelor bibliografice în fișierele L ^A T _E X	99
	4.11.3 Fișiere utilizate în BIB _T E _X	100
	4.11.4 Exemple de utilizare	101

4.12	Configurarea \LaTeX de către utilizator	104
4.13	Realizarea planșelor de prezentare folosind stilul <code>seminar</code>	105
4.13.1	Comenzi specifice stilului <code>seminar</code>	106
4.13.2	Exemplu de utilizare	107
4.14	Folosirea programului utilitar <code>make</code>	109
4.15	Exerciții	110
5	Grafice și figuri	
	— <code>Gnuplot</code> , <code>XFig</code> , <code>XPaint</code> , <code>XV</code> —	121
5.1	<code>XFig</code>	125
5.1.1	Apelare	125
5.1.2	Obiecte <code>XFig</code>	125
5.1.3	Regiuni ale ecranului	126
5.1.4	Salvarea desenului în alte formate	128
5.2	<code>Gnuplot</code>	129
5.2.1	Apelare	129
5.2.2	Comenzi utile pentru început	129
5.2.3	Manualul pentru <code>gnuplot</code>	129
5.2.4	Ce poate face <code>gnuplot</code>	129
5.2.5	Comenzi <code>gnuplot</code>	130
5.2.6	Comenzi de bază: <code>plot</code> , <code>splot</code>	130
5.2.7	Alte comenzi utile	133
5.2.8	Salvarea graficelor în alte formate	135
5.2.9	Fișiere de comenzi <code>gnuplot</code>	136
5.3	<code>XPaint</code>	138
5.3.1	Apelare	138
5.3.2	Ferestre <code>xpaint</code>	138
5.3.3	Fereastra de instrumente	138
5.3.4	Fereastra de desenare	139
5.3.5	Salvarea desenului în diferite formate	140

5.4	XV	142
5.4.1	Apelare	142
5.4.2	Ferestre <code>xv</code>	142
5.4.3	Operațiuni în fereastra de imagine	142
5.4.4	Operațiuni în fereastra de control	143
5.5	Exerciții	145
6	Analiza numerică a modelelor matematice	
	— Scilab —	147
6.1	Apelare Scilab	150
6.2	Componente Scilab	150
6.3	Interfața grafică a programului Scilab	150
6.4	Câteva noțiuni introductive	151
6.4.1	Formatul comenzilor Scilab	151
6.4.2	Facilități	152
6.4.3	Demonstrație	152
6.5	Tipuri de date Scilab	152
6.5.1	Constante	152
6.5.2	Matrice de scalari	153
6.5.3	Matrice de șiruri de caractere	154
6.5.4	Matrice de polinoame și de funcții raționale	155
6.5.5	Matrice cu elemente de tip logic	155
6.5.6	Liste	156
6.6	Programarea în Scilab	156
6.6.1	Funcții Scilab	156
6.6.2	Funcții predefinite în Scilab	157
6.6.3	Tipuri de instrucțiuni Scilab	157
6.7	Fișiere de comenzi	158
6.8	Grafice	159
6.9	Exemple	160
6.10	Apelul unei rutine C sau FORTRAN din Scilab	162
6.11	Traducerea automată în FORTRAN a unei funcții Scilab	165
6.12	Exerciții	166

7 Medii de programare Unix	
— C, C++, FORTRAN —	169
7.1 Principalele opțiuni ale compilatoarelor <code>gcc</code> , <code>g++</code> , <code>g77</code>	172
7.2 Recomandări privind stilul de lucru profesional	173
7.3 Compilarea programelor folosind <code>make</code>	174
7.3.1 Structura unui fișier <code>makefile</code>	174
7.3.2 Modele de fișiere <code>makefile</code>	175
7.4 Utilizarea depanatorului <code>gdb</code>	176
7.5 Apelarea rutinelor FORTRAN dintr-un program scris în limbajul C . . .	179
7.6 Exerciții	182
8 Comunicația între sistemele Unix	
— Internet —	193
8.1 <code>telnet</code>	200
8.2 <code>write</code>	200
8.3 <code>talk</code>	201
8.4 <code>ftp</code>	203
8.5 Alte comenzi utile, legate de transferul de fișiere și de comunicație	204
8.6 Elemente de bază WWW	205
8.7 Structura limbajului HTML	207
8.8 Coduri de formatare HTML	208
8.8.1 Principalele perechi de coduri de formatare HTML	208
8.9 Un exemplu de document HTML	210
8.10 Exerciții	212
ANEXE	215
A Teste finale de laborator	215
B Regulamentul Laboratorului de Metode Numerice	221
C Evoluția Laboratorului de Metode Numerice – LMN	223

D Resursele Laboratorului de Metode Numerice	229
D.1 Resurse hardware	230
D.2 Resurse software	237
D.3 Interfața grafică a butoanelor mouse-ului	239
D.3.1 Meniuri de comenzi disponibile sub Linux	239
D.3.2 Meniuri de comenzi disponibile în rețeaua de stații grafice	242
Bibliografie și Webografie	247
Index	250

Introducere

Motto: “Cercetarea științifică înseamnă 1% inspirație și 99% folosirea eficientă a unor instrumente adaptate”.

De mai bine de două decenii, la clasicele “unelte de lucru” ale cercetătorului (teoria și experimentul) s-a mai adăugat unul: calculatorul. Viziunea modernă asupra cercetării, impusă nu fără obiecții și dificultăți, plasează metodele domeniului numit în SUA “*Computational Science and Engineering – CSE*” [54] pe același nivel cu abordările teoretice și cele experimentale (fig. I.1).

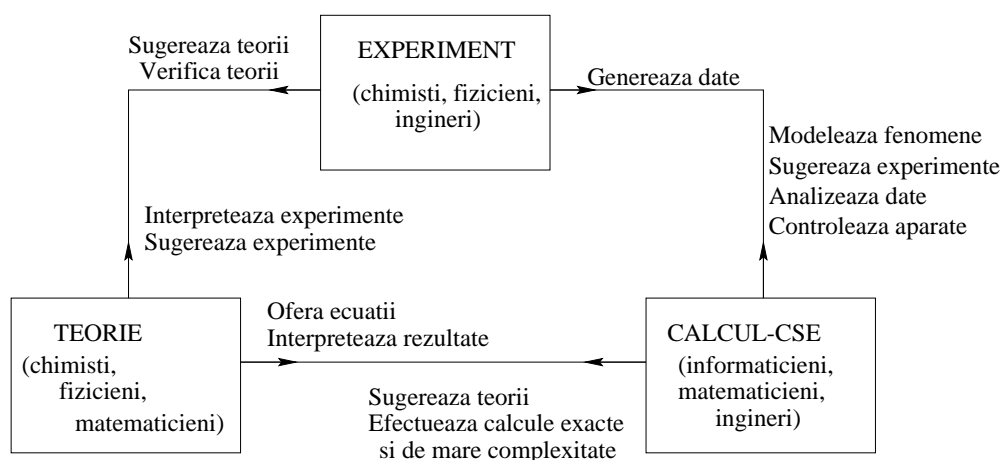


Fig. I.1: Poziția Științei și Ingineriei Computaționale (CSE) în tabloul științei

Dar lucrurile nu se opresc aici: cercetarea nu este importantă în sine, ci prin impactul ei asupra societății. Metoda uzuală de valorificare a rezultatelor cercetării este cea de diseminare a cunoștințelor descoperite, prin publicare. Prezența calculatoarelor în orice laborator științific a determinat o mutație și în acest domeniu: cercetătorul modern a devenit, practic, propriul său tehnoredactor și tipograf!

Mai mult, mediile de comunicare profesională migrează de la tiparul clasic pe hârtie spre formatul electronic, cu toate implicațiile acestuia: formele de reprezentare și de transmitere a cunoștințelor științifice se diversifică, migrând de la documentul scris alcătuit în cel mai complicat caz din text, tabele, formule matematice și chimice sau figuri, spre

“documente” electronice ce conțin în plus sunete, imagini 3D, animații sau secvențe video; interactivitatea începe să fie incorporată în structura intimă a producției, de la simpla structurare hipertextuală (mai corect *hipermedia*) până la interactivitate completă și puternică.

Potențate de dezvoltarea spectaculoasă a tehnologiei informației, CSE și noua filozofie a comunicării profesionale vor avea, cu certitudine, efecte sociale majore în secolul următor.

Modul în care ar trebui predat noul domeniu științific —CSE— studenților din universitățile americane a stârnit o extrem de interesantă discuție în câteva numere succesive ale *IEEE Computational Science and Engineering* (numerele 2-3 din 1996). Părerile exprimate au fost extrem de diverse, începând cu afirmația că totul ar trebui să constea de fapt în însușirea instrumentelor Unix, și sfârșind cu opinia că stăpânirea mediului *Mathematica* este suficientă și că specialiștii în calculatoare mai mult încurcă decât ajută la formarea inginerilor în utilizarea tehnologiilor CSE. Prezenta lucrare poate fi considerată ca o replică a Laboratorului de Metode Numerice din Politehnica Bucureștenă, în această dispută.

Revenind la motto-ul de la începutul introducerii, manualul de față nu se referă la acel procent de inspirație necesar cercetării creatoare (care de altfel nu poate fi învățată) și nici măcar la conținutul propriu-zis al cercetării, ci la **atelierul de lucru** al cercetătorului modern.

Firul conducător al acestui manual este de a descrie, într-o manieră progresivă care să permită înțelegerea și învățarea ușoară, folosirea instrumentelor necesare redactării documentelor științifice, în special ale celor din domeniul științei și ingineriei computaționale.

Inițial, prezenta lucrare a fost concepută ca un îndrumar de laborator pentru studenții anului VI (studii aprofundate) de la Facultatea de Electrotehnică din Universitatea “Politehnica” din București la direcțiile “Proiectarea Asistată de Calculator a Dispozitivelor Electromagnetice” și “Proiectarea microsystemelor”. Acești studenți își desfășoară activitatea în Laboratorul de Metode Numerice și a fost nevoie de o instruire specială în folosirea mediului hardware/software avansat care este disponibil în acest laborator. Primele materiale conținute în acest manual au fost elaborate începând din 1995 de Irina Munteanu și au fost puse la dispoziția studenților, în format electronic, în pagina de Web a laboratorului (<http://www.lmn.pub.ro>).

O analiză mai atentă ne-a făcut să înțelegem că experiența acumulată, mai ales în ultimii 5 ani, merită să fie distribuită într-un cadru mult mai larg. În consecință, am elaborat acest **manual, destinat tuturor celor interesați în folosirea profesională a calculatoarelor în activitatea științifică și inginerescă** (studenți, ingineri, cercetători în orice domeniu științific). Deși manualul nu se adresează în mod explicit studenților sau specialiștilor în calculatoare, experiența din laboratorul nostru, în care lucrează și astfel de persoane, ne-a demonstrat că o asemenea lucrare le poate fi extrem de utilă.

Această lucrare este o **premieră națională**, din mai multe puncte de vedere:

- în primul rând este un **bun început** pentru oricine are acces la un calculator personal cu un sistem de operare de tip Unix și aspiră să-l folosească pentru a obține rezultate profesionale de înalt nivel, folosind tehnicile cele mai avansate;
- lucrarea nu înlocuiește manualele de referință (dar **sintetizează** câțiva “metri” de raft de astfel de manuale, citate la bibliografie), pe care va trebui să le studiați dacă vreți să vă specializați în utilizarea unuia din instrumentele prezentate;
- lucrarea nu este un memorator sistematic și exhaustiv (gen “. . . *in a nut shell*”) ci este un **compendiu selectiv**, un memorator în care sunt descrise pe scurt comentariile care s-au dovedit a fi cele mai utilizate în activitatea curentă din laboratorul nostru;
- instrumentele Unix descrise alcătuiesc un **sistem coerent și complet** (“*toolbox*”) pentru realizarea unor cercetări științifice la nivel mondial, precum și pentru valorificarea rezultatelor obținute, prin publicarea și prezentarea lor, conform celor mai exigente standarde actuale;
- instrumentele prezentate fac parte din categoria **software cu licențe gratuite**, o categorie extrem de amplă și în dezvoltare, dar din care selecția pachetelor profesionale, cu performanțe peste nivelul produselor comerciale, nu este simplă;
- exemplele date în lucrare au fost testate, exercițiile au fost parcurse mai mulți ani la rând de studenți și **îmbunătățite în urma experienței câștigate** iar multe dintre “trucurile” prezentate au necesitat zile de muncă pentru a fi descoperite;
- manualul a fost realizat în întregime **folosind chiar instrumentele prezentate**, ceea ce validează încă o dată puterea mediului propus.

Citirea atentă a introducerilor, care conțin **descrierea** principalelor concepte (însoțite de trimiteri la bibliografie), completată cu parcurgerea conștiințioasă a exercițiilor reprezentată o cale posibilă (“calea regală”) de a **înțelege repede și corect** domeniul extins pe care acest manual încearcă să-l acopere.

Decizia curajoasă de a prefera calculatoare controlate de sisteme de operare Unix, aflat pe locul al doilea din punctul de vedere al numărului de utilizatori, a fost determinată de constatarea că sistemul MS-Windows deține supremația de vânzări mai ales datorită aplicațiilor casnice, jocuri și birotică și mai puțin prin aplicațiile lui profesional-științifice. Deoarece numele UNIX (cu majuscule) este marcă înregistrată, în acest manual se preferă utilizarea numelui Unix (cu litere mici!) pentru sistemele de operare *de tip* UNIX.

De fapt, Unix este mai mult decât o familie de sisteme de operare, este o **cultură** a unei întregi comunități care folosește rețeaua de calculatoare Internet și care adoptă un stil specific de programare și de rezolvare a problemelor, prin combinarea în structuri elegante a problemelor simple. Poate cea mai bună definiție a acestui concept este cea dată de Hahn [55]: “*Unix is a set of tools for smart people*”¹.

¹ *Unix este un set de instrumente pentru oameni inteligenți.*

Contribuția autorilor la lucrare este următoarea:

I. Munteanu (80%): Descrierea programelor și comenzilor (cu excepția Emacs), conceperea majorității exercițiilor, tehnoredactare, corectură (80%);

D. Ioan (20%): Introducerile capitolelor, introducerea lucrării, descrierea Emacs, anexele referitoare la istoria și la resursele Laboratorului de Metode Numerice, corectură.

Mulțumiri

Apariția acestui manual nu ar fi fost posibilă fără înființarea Laboratorului de Metode Numerice (LMN), dezvoltat în cadrul Catedrei de Electrotehnică în Politehnica bucureșteană. Faptul că această catedră este una din cele mai prestigioase din universitate se datorează tradiției ei și mai ales regretatului Academician *Remus Răduleț*, care a fost personalitate proeminentă și a cărui amprentă a rămas puternic imprimată în Catedră. Urmașii acestuia, profesorii *C.I. Mocanu*, *Alexandru Timotin* și *Andrei Țugulea*, ne-au învățat să muncim serios, să gândim riguros, dar și cum să împărtășim studenților cunoștințele acumulate.

Pentru înființarea LMN se cuvine să mulțumim în primul rând D-lui prof. *Constantin Nemoianu*, care a renunțat de bunăvoie la camera EB-206 unde a fost instalat pentru prima dată acest laborator. În perioada timpurie a LMN, un rol important l-au avut domnii *Alexandru Morega* (în prezent profesor la Catedra de Mașini Electrice din U.P.B.), *Costin Stancu* (în prezent “senior designer” la General Motor în California, USA) precum și *Mihai Georgian* (în prezent șeful unei echipe de programatori în Canada), toți trei absolvenți ai seriei 1980 ai Facultății de Electrotehnică.

Tot în acea perioadă ne-au sprijinit: *Nini Popovici*, inginer de sistem la în Centrul de Calcul al Politehnicii (în prezent Directorul *RoEduNet*), cu ajutorul căruia am instalat primele terminale la distanță (în LMN) ale sistemului FELIX C256; *Vlad Țepelea*, șeful unui colectiv din I.T.C., beneficiarul pachetului LOCAP (în prezent președintele Consiliului de Administrație RomTelecom); *Al. Nica*, directorul centrului de calcul CEPECA și care ne-a împrumutat modem-urile pentru conexiunea WANG, acordate cu mare îndemânare de dl. *Sorin Popescu*, șef de lucrări la Facultatea de Electronică; *Sorin Ghețan*, administratorul sistemului WANG; d-na *Julieta Florea*, care în calitate de Decan al Facultății de Electrotehnică și Energetică a acceptat în 1989 ca disciplina de Metode Numerice să fie predată de ingineri și nu de matematicieni; *Sven Kratochvil*, talentatul programator al pachetului FAP în prima sa versiune, dar și *Valentin Cotârță*, cu care în 21 decembrie 1989 ne strecuram printre barajele “antitero” pentru a aduce în Politehnică primul calculator compatibil IBM PC-AT, importat după un adevărat “război” cu autoritățile de atunci ale Ministerului Învățământului.

Mulțumiri se cuvin adresate și celor peste 50 de prestigioși specialiști din întreaga lume care ne-au răspuns la scrisorile de intenție privind colaborarea interuniversitară scrise încă din ultima săptămână a anului 1989. O bună parte dintre aceștia au devenit partenerii și prietenii noștri în cadrul proiectului *TEMPUS JEP 2717*, inițiat în 1990 și finanțat de *Comisia Europeană* prin *European Training Foundation* (ETF). Vizita la Politehnica din Torino, făcută ca urmare a invitației de a-l însoți pe profesorul *Andrei Țugulea* (pe atunci membru în Guvernul României), a constituit pasul decisiv în pregătirea unui astfel de proiect. Aici am întâlnit parteneri desăvârșiți ca prof. *Vicenzo Pozzolo*, președintele COREP, Rectorul *Rodolfo Zich*, *Aurelia Stepanescu* și *Michele Patrissi*. Aceștia, împreună cu *Alain Bossavit* (Electricité de France), *Jürgen Arnold* (MSC MacNeal-Schwendler, München), *Stavros Christodoulakis* (Technical University of Crete), *Dominic Collard* (ISEN, Lille), *Bernard Courtois* (Institut National Polytechnique de Grenoble), *Peter Kayes* (Thames Valley University), *Giorgio Molinari* (Uni-

versită di Genova), *Adel Razek* (Laboratoire de Génie Électrique de Paris), *Kurt Richter* (Technische Universität Graz), *David Rodger* (University of Bath), *Guglielmo Rubinacci* (Università di Cassino), *Jean-Claude Sabonnadière* (Institut National Polytechnique de Grenoble), *H. M. Schaedel* (Fachhochschule Köln), *John Tegopoulos* (National Technical University of Athens), au contribuit decisiv la succesul proiectului. Totuși, acest lucru n-ar fi fost posibil fără contribuția prof. *F.M.G. Tomescu*, membru în Comitetul Executiv al proiectului (în prezent director de studii aprofundate în catedra noastră), care a înțeles mai devreme decât alți colegi de catedră necesitatea extinderii laboratorului.

Importul primelor stații grafice cu procesor RISC nu a fost facil, datorită restricțiilor de export a tehnologiei strategice, iar pentru succesul acestei întreprinderi trebuie să le mulțumim reprezentanților producătorilor de stații grafice: *Mircea Zaharescu* (firma Hewlett Packard), *Bogdan Cocora* (firma DEC), *Trandafir Moisa* (firma Sun) și *Octavian Țuțurea* (firma SGI). Poate mai dificilă decât cumpărarea echipamentelor a fost configurarea și administrarea rețelei hibride în continuă creștere. Administratorii de rețea: *Mihai Lăzărescu* (în prezent în Italia), *Mihai Popescu*, *Alexandru Petrescu* (în prezent în Luxemburg), *Daniel Pandelea*, *Ștefan Pușcașu* și *Laurențiu Badea* (în prezent în SUA) au făcut minuni. Actualii administratori *Bogdan Brătucu*, *Victor Stănescu*, studenți la Facultatea de Calculatoare, fac eforturi laudabile pentru a întreține o configurație hardware–software extrem de complexă.

Un impact deosebit nu numai în LMN ci în întreaga comunitate Unix din România au avut-o: programul “*Free Unix for Romania*” (inițiat de *Marius Hâncu* din Canada și continuat de *Teodor Lungu* de la NASA-JPL-Pasadena) prin care a fost difuzat sistemul Linux, precum și organizarea de către *GURU* – “*Grupul Utilizatorilor Români de Unix*” a conferinței anuale *ROSE – Romanian Open System Event* (în care un rol hotărâtor l-au avut ing. *Alexandru Rotaru* și prof. *Irina Athanasiu*), ocazie cu care am fost vizitați de cele mai mari personalități din domeniu, cum sunt *Richard Stallman* (“Mr GNU”) și *Linus Torvald* (“Mr Linux”).

Interesul LMN pentru educația asistată de calculator a inginerilor datează din 1992, când Catedra de Electrotehnică a fost dotată de Ministerul Învățământului (condus în acea vreme de prof. *George Ștefan*) cu o rețea Novell cuprinzând un server PS/2 (386) și 12 calculatoarea PS/2 (286) cu ecran monocrom, pentru a fi folosită pentru studenții anului al II-lea la laboratorul asociat cursului de Metode Numerice. Într-un timp record (continuând preocupările începute în 1991), asistenții acestui curs: *Irina Munteanu*, *Bogdan Ionescu* (în prezent angajat la cea mai mare firmă pentru software de modelare electromagnetică, ANSOFT din SUA), *Radu Popa* (în prezent doctorand la Universitatea din Tokio), *Mihai Lăzărescu*, *Mihai Popescu*, dar la care s-au asociat ulterior *Gabriela Ciuprina*, *Mihai Platon*, *Mihai Rebican*, *Gabriel Preda* și *Cristina Bordeanu*, au dezvoltat un pachet de programe dedicat instruirii asistate în domeniul Metodelor Numerice în Ingineria Electrică. Acest pachet s-a bucurat nu numai de impact local, ci și de un deosebit succes internațional. În acest fel ne-am câștigat noi suporteri: *Jean Michel* (de la renumita Școală de Poduri și Șosele din Paris), *Eddy Forte* și *Guy de Lafontaine* (de la Școala Politehnică Federală din Lausanne).

Apariția tehnicilor multimedia și încercarea de a fi folosite în educație prin producerea de materialele didactice de către ingineri și studenți în inginerie nu a dat rezultate strălucite

datorită aspectului neprofesional al acestora. Acesta este motivul pentru care LMN s-a implicat din 1995 în proiectul *TEMPUS S-JEP 9122*, dedicat Comunicării Audiovizuale bazate pe Tehnologia Informației (cu un buget de peste 500.000 ECU) și în care, pe lângă U.P.B., au mai participat: Academia de Teatru și Film (prin Catedra de Multimedia reprezentată de coordonatorul proiectului – *Horea Murgu*, actualul Rector al Academiei – *Florin Mihăilescu*, iar în faza inițială *Victor Rebeniuc* și *Stere Gulea* – ulterior președintele interimar al Televiziunii Române), Academia de Arte Vizuale din Cluj (reprezentată de șeful secției de Foto-Video, *Karoly Feleky* și de Rectorul Academiei, *Cornel Nica*).

Împreună cu personalitățile din lumea artistică menționate și cu partenerii noștri de peste hotare: *Roy Ascott* (Newport, UK), *Yossi Bal* (Londra, UK), *Hans Froling* și *Peter van der Weyden* (Utrecht, NL), *Michèle Ganem* (Grenoble, F), *Yves Papin* și *Guy Bozard* (Nantes, F), Geneviève Toussaint (Bruxelles, B), am făcut un pas înainte în înțelegerea tainelor comunicării profesionale. CD-ROM-ul “București 2000”, realizat în LMN și primul curs de Comunicare Profesională adresat inginerilor din România, reprezintă dovezi incontestabile în acest sens. Interesul pentru comunicarea orală sau scrisă dar mai ales pentru comunicarea electronică (Internet, Web) reflectate în manualul de față reprezintă efecte ale acestui proiect.

De altfel, pe lângă apariția acestui manual, proiectul *TEMPUS S-JEP 9122* a mai contribuit și la dotarea laboratorului de comunicare profesională (EB-212). Tot în cadrul acestui proiect, partenerii noștri olandezi au donat Universității “Politehnica” din București rețeaua de stații grafice VAX din EB-211. Ne face plăcere să le mulțumim pentru generozitatea lor și pentru energia depusă în organizarea selecției și a transportului echipamentelor. Cu această ocazie au fost donate calculatoare și Centrului pentru Promovarea Comunicării coordonat de studenții Facultății de Electrotehnică din UPB. Caracterul absolut novator pentru România al acestui proiect a generat interes și sprijin atât din partea unor personalități precum *Andrei Marga* (Rectorul Universității Babeș Bolyai, în prezent Ministrul Învățământului), dar și unor organizații ca *Fundația Soros pentru o Societate Deschisă*, care a cofinanțat proiectul cu 120.000 USD.

Nu trebuie să uităm că toate aceste activități s-au desfășurat în Catedra de Electrotehnică (ce a fost condusă din 1990 de profesorii *Andrei Țugulea*, *Aurelian Panaitescu* și *Florea I. Hăntilă*), a Facultății de Electrotehnică (ce a fost condusă de Decanii *Aurelian Crăciunescu*, *Constantin Răduți* și *Mihai Popescu*), din U.P.B. (ce a fost condusă de Rectorii *V.N. Constantinescu* și *Gh. Zgură*). Tuturor le mulțumim pe măsura sprijinului acordat.

Nu mai puțin le mulțumim studenților care s-au perindat prin LMN și fără de care eforturile noastre nu ar fi avut nici un sens. Mulțumiri merită și personalul tehnic (*Cornel Nițulescu*, *Ion Ioniță*, *Mihai Sandu*, *Mary Klein*, *Nicolae Boboc*, *Flavian Colonel*, *Gabriel Ioan*), dar cu prisosință doamnei *Gabriela Iosif*, persoană a cărui devotament și dăruire pentru LMN este peste orice îndoială.

Suntem convinși că lista persoanelor care au contribuit într-un fel sau altul la dezvoltarea laboratorului și au făcut posibilă apariția acestui manual nu este completă.

Am lăsat la urmă familiile noastre care ne-au iertat pentru weekend-urile, serile și chiar nopțile pe care li le-am răpit.

Capitolul 1

Utilizarea sistemului de operare Unix

Sistemul de operare este cea mai importantă componentă software a unui sistem de calcul. El dă viață echipamentului (componentei hardware) asigurând următoarele **funcții principale**:

- gestiunea spațiului de **memorie** (alocări și eliberări de zone în memoria internă RAM);
- gestiunea **proceselor** (lucru extrem de important mai ales în sistemele care pot executa “simultan” mai multe programe);
- gestiunea spațiului de **disc** (prin intermediul sistemului de fișiere);
- controlul operațiilor de **intrare-ieșire** (operații elementare ca de exemplu accesul direct în memorie –DMA– sau operații mai complicate realizate prin intermediul unor componente software numite “drivere”, specifice fiecărui periferic: tastatură; mouse; ecran; imprimantă; unități de bandă; CD-ROM sau alte dispozitive de memorare externă; interfețe și plăci de comunicație, etc.).

Operațiile fundamentale ale sistemului de operare sunt incluse în “*nucleul*” lui și nu sunt de obicei accesibile direct de utilizator, din motive de securitate. **Interacțiunea** dintre nucleul sistemului de operare și utilizator se realizează printr-o altă componentă a sistemului, numită “*shell*”, un interpretor de comenzi care preia, interpretează și execută comenzile utilizatorului, introduse de obicei de la tastatură. Sistemele moderne mai au o componentă suprapusă peste shell, numită *interfață grafică cu utilizatorul* (*GUI – Graphic User Interface*), care permite interacțiunea calculator–utilizator prin intermediul unui **sistem de ferestre** pe ecran, însoțite de butoane și meniuri selectabile prin mouse sau de la tastatură.

Istoria evoluției calculatoarelor este strâns legată, pe de o parte, de evoluția tehnologiilor de realizare a componentei hardware (mai ales a componentei principale care este procesorul), iar pe de altă parte, de evoluția componentei software, în care sistemul de operare joacă un rol central. Datorită importanței lui, practic fiecare firmă producătoare

de calculatoare și-a dezvoltat și întreținut propriul sistem de operare. Odată cu apariția și răspândirea ca bun de larg consum a calculatorului personal *PC* (*IBM PC* și compatibile) situația s-a schimbat. Marea majoritate a PC-urilor folosesc în prezent sistemul de operare *MS-Windows*, dezvoltat de firma *Microsoft* pe baza unui sistem anterior numit *MS-DOS* (Microsoft Disk Operating System).

Pe al doilea loc, ca număr de utilizatori, se află sistemele de operare de tip Unix [65] [73]. Acestea au fost dezvoltate inițial pentru minicalculatoare (*DEC PDP*), dar au fost ulterior intens folosite de stațiile de lucru (sistem de calcul cu procesoare puternice *RISC*) produse de firmele *Sun*, *Silicon Graphics*, *Digital Hewlett Packard* sau *IBM*. Începând cu apariția procesoarelor *Intel 386*, *486* și *Pentium*, sistemele de tip Unix au migrat și în lumea PC-urilor. Trebuie remarcat faptul că sistemele Unix au fost permanent mai avansate conceptual și ca soliditate, decât sistemele DOS- Windows, fiind dedicate unor utilizări profesionale.

UNIX a devenit operațional ca sistem de operare pentru *DEC PDP 11* în februarie 1971. El a fost dezvoltat inițial de o echipă mică de programatori (D. Ritchie și K. Thomson) de la *Bell Laboratories (AT&T)*, prin efort personal (de circa 2 ani-om) și nu pe baza unor specificații predefinite, sarcini de servicii, cum s-a întâmplat cu toate sistemele anterioare. Bazați pe un sistem numit *Multics* (dezvoltat fără mare succes în anii '60 de *MIT*, *Bell* și *General Electric*) și nemulțumiți de ospitalitatea sistemelor existente, ai au decis să-și scrie propriul sistem de operare, mic ca dimensiuni, dar puternic, și care să implementeze cele mai fertile idei cunoscute în acel moment. Sursele sistemului de operare au fost distribuite gratis în comunitatea academică americană și au fost folosite în aplicații referitoare la educația în știința calculatoarelor (sisteme de operare, limbaje, rețele de calculatoare), dar și pentru editare texte și formatare de documente.

În perioada 1974 - 1979 UNIX a fost folosit intensiv și practic rescris din nou de studenții universității Berkley. Versiunea finală a fost *BSD 4.4 (Berkley Software Distribution)*. În paralel, cei de la Bell au ajuns la versiunea *Unix System V*. Prin reunirea practic a celor două versiuni, în 1985 s-a obținut produsul numit *UNIX SVR4 (System V Release 4)*. În acel moment, sistemul avea câteva mii de instalări și deținea un compilator de C, asamblor, editor de legături, depanator simbolic, editor de texte, sistem de formatare a documentelor, sute de instrumente diverse și compilatoare pentru mai multe limbaje incluzând *FORTRAN 77*, *Basic* și *Pascal*.

Experiența absolvenților Berkley a migrat odată cu ei la marile firme de calculatoare din Valea Siliciului, astfel încât au apărut pe piață o multitudine de sisteme de tip Unix: *HP-UX* (Hewlett Packard) [35], *Ulrix* (Digital) [34], *IRIX* (Silicon Graphics) [32], *Xenix* (Microsoft), *SCO* (filiala Microsoft: Santa Cruz Organisation) [47] și la compania Sun: *Sun OS* (de tip BSD) și *Solaris* (de tip SVR9) [38]. După ce a aparținut o vreme unui laborator nou creat în AT&T: *USL (UNIX System Lab.)*, marca **UNIX** a fost vândută companiei *Novel* (care distribuie *UNIXWare*) și în final a ajuns să aparțină unei organizații de standardizare numită *X/Open*.

Încercările de standardizare au început cu înființarea *Open Software Foundation* sprijinită de *Digital* (distribuitorul unei versiuni de sistem de operare numit chiar *OSF/1*) și *IBM*. În prezent, efortul de standardizare este realizat de *IEEE* care a elaborat primele două

capitole din standardul *IEEE 1003*, numit și *POSIX (Portable Operating System Interface)* și care în 1003.1 descrie funcțiile necesare programării în C, iar în 1003.2 descrie setul de comenzi utilizator [9].

Deoarece era convins că produsele software trebuie să fie gratuite, un tânăr și extrem de talentat programator, absolvent MIT, numit Richard Stallman, a înființat în 1990 o fundație numită *Free Software Foundation*. Principalul proiect al acestei fundații este numit *GNU (GNU = GNU is Not Unix)* [6] și urmărește realizarea unui nou sistem de operare care să satisfacă prevederile POSIX și a unui set complet de instrumente software de foarte bună calitate. Produsele GNU nu sunt plasate în domeniul public (“*public software*”) și nu sunt nici promoționale (“*shareware*”), ci sunt supuse unui contract de copiere numit *GPL (“General Public Licence”)*, care în esență stipulează că oricine are dreptul să vândă pachetul de software la orice preț (indiferent dacă l-a obținut gratis!), cu condiția să livreze și codul sursă. În ciuda reticenței cu care au fost privite FSF, GNU și GPL, se constată că proiectul reușește (aducând inclusiv avantaje financiare!).

Un alt eveniment important în lumea Unix are loc în 1991 când tânărul student finlandez Linus Torvald începe, practic de la zero, (și influențat de sistemul de școală *Minix* al lui Tanenbaum), dezvoltarea unui nou sistem de operare de tip Unix pentru PC 386, sistem pe care l-a numit *Linux* [2] [10] [11] [12] [16] [21] [40]. Istoria este interesantă deoarece sistemul Linux este dezvoltat și întreținut de o armată uriașă de programatori din întreaga lume, pe bază de voluntariat, folosind Internet-ul. În prezent, sistemul GNU Linux (cu clauza GPL) este cel mai utilizat sistem de tip Unix din lume, având milioane de instalări.

Caracteristicile sistemelor Unix, care explică succesul lor, sunt:

- **portabilitatea**, fiind accesibil în sursă C poate și este rulat pe o mare varietate de calculatoare;
- **puterea**, deoarece este acompaniat de o multitudine de instrumente și programe de aplicații, care permit creșterea productivității în cele mai diverse domenii, de la birotică la cercetare științifică;
- **multitasking**, permite executarea “simultană” (de fapt în timp partajat) a mai multor operații, ca de exemplu redactarea unui text și în background formatarea unui document sau/și rezolvarea unei ecuații;
- **multiuser**, permite accesul simultan la resurse valoroase (imprimante, calculatoare puternice sau servere de fișiere) a mai multor utilizatori;
- **conectivitate**, sistemul având suport nativ pentru conectarea calculatoarelor în rețea (networking).

În afara calităților menționate, utilizarea sistemelor Unix întâmpină și unele dificultăți legate de diversitatea variantelor și versiunilor (reflectată de altfel și în LMN, în care sunt instalate pe lângă sistemul Linux și variante comerciale ca HP-UX, SunOS, Ultrix

și Irix) și de faptul că sistemul de instrumente Unix nu este foarte coerent (apar dese suprapuneri ale funcțiilor, iar interfața lor cu utilizatorul nu este comună).

Începând încă din anii '70, firma *Rank Xerox* a finanțat un proiect de cercetare care să conducă la perfecționarea relației om-calculator. Ca urmare a acestui proiect (Xerox PARC) s-a inventat mouse-ul și conceptele specifice interfețelor grafice cu utilizatorul – GUI. Bazat pe aceste concepte s-a realizat la universitatea Berkley un sistem de ferestre (numit *W*), care a fost apoi preluat (în proiectul *Athena*) de MIT și care a condus în final la realizarea celei mai populare interfețe grafice pentru sistemele Unix, numită *XWindow* [49] [72]. Ultima versiune majoră a acestui sistem, numită pe scurt *X11*, a fost realizată în 1987 și este distribuită gratis, începând din 1988, de *MIT X Consortium*.

Cele mai importante caracteristici ale sistemului *XWindow* sunt:

- **portabilitatea**, fiind disponibil în sursă C;
- **transparența**, fiind dezvoltat în maniera client/server, el poate fi exploatat în rețeaua de calculatoare urmând ca să fie posibil ca un program să ruleze pe o mașină, folosind fișierele de pe altă mașină, iar rezultatele să le afișeze într-o fereastră de pe o a treia mașină;
- **flexibilitatea**, pot fi deschise oricâte ferestre, stilul ferestrelor nu este predefinit și suportă controlul unui număr nedefinit de managere de ferestre ca: *Athena*, *TWM*, *OpenLook* sau *Motif* (care în prezent este cel mai răspândit, dar nu este încă disponibil gratis).

Trebuie menționat că managerul de ferestre impune stilul (“look and feel”) ferestrelor generate.

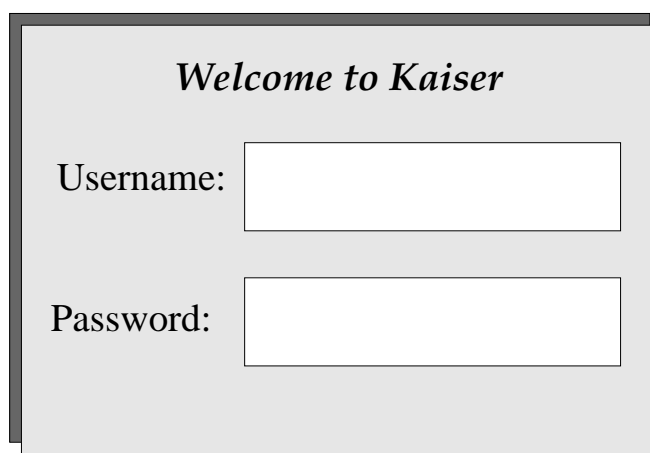
Scopul principal al capitolului constă în familiarizarea cu comenzile sistemului de operare Unix și a interfețelor *XWindow* privitoare la sesiunea de lucru (intrare, ieșire, securitate, configurație), sistemul de fișiere (creere, redenumire, ștergere, copiere, căutare, arhivare, protecție), vizualizare conținut fișiere, filtre și redirectări, imprimare și documentare.

1.1 Intrarea în sistem

Pentru a putea lucra în sistemul Unix trebuie să aveți un “cont” pe acel sistem. Fiecărui utilizator cu un astfel de cont i se acordă de către administratorul de sistem:

- un nume de cont (*username*);
- o parolă (*password*).

Inițial, pe ecranele calculatoarelor este afișată **fereastra de acces în rețea**, prin care se poate iniția o sesiune de lucru. Fereastra are aspectul:



The image shows a rectangular window with a light gray background and a dark gray border. At the top center, the text "Welcome to Kaiser" is displayed in a bold, italicized serif font. Below this, there are two input fields. The first is labeled "Username:" and the second is labeled "Password:". Both labels are in a bold serif font. The input fields are empty rectangular boxes with thin black borders.

În această fereastră se introduc numele de cont și parola, fiecare urmate de <↵> (Enter). Din motive de securitate, parola nu este afișată pe ecran.

Dacă numele și parola sunt recunoscute de sistemul de operare, după scurt timp pe ecran se deschide o fereastră XWindow.

1.2 Lucrul cu ferestre XWindow

La începutul unei sesiuni de lucru, sistemul afișează o fereastră de lucru (XTerm) și un grup (meniu) de butoane (să-l numim în continuare Toolbox) în colțul din dreapta jos a ecranului. Meniul Toolbox este descris în anexa D.3.1.

Mutarea cursorului mouse-ului în fereastra de lucru are ca efect vizibil schimbarea culorii barei superioare a ferestrei. Se marchează astfel faptul că fereastra a devenit **fereastra activă** de lucru. În acest moment se pot da comenzi la promptul sistemului, cum ar fi de exemplu `uname` dacă vreți să aflați ce variantă de Unix folosiți, sau `hostname` pentru a afla numele mașinii pe care lucrați (figura 1.1).

Butoanele mouse-ului au funcții diferite, după cum cursorul se află plasat într-o fereastră sau în afara oricărei ferestre.

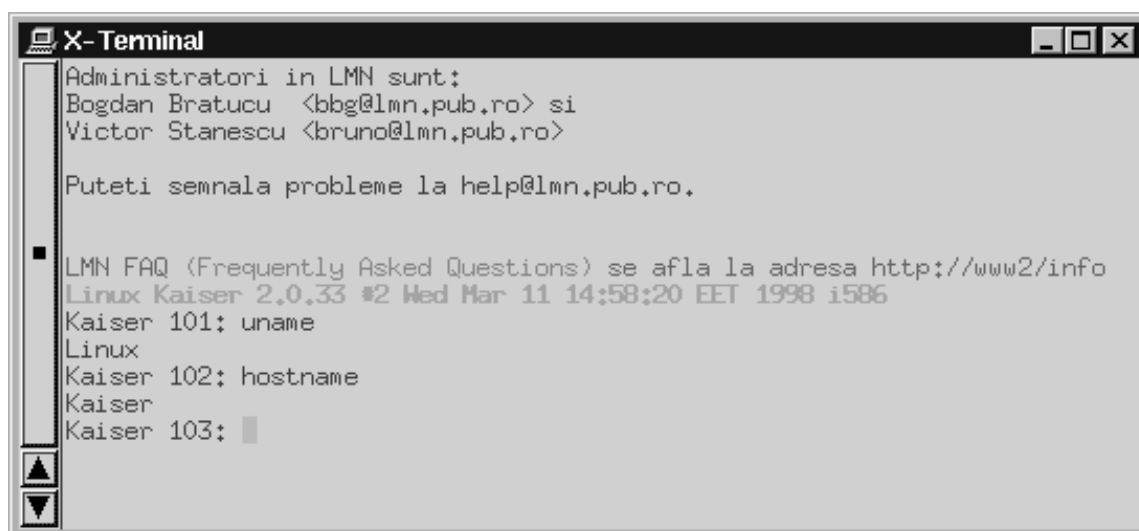


Fig. 1.1: Fereastră XWindow (XTerminal)

- **Când cursorul mouse-ului este plasat într-o fereastră**
 - Butonul din stânga: permite *selectarea* unui text, astfel: se plasează cursorul la începutul textului, se apasă butonul din stânga și se deplasează mouse-ul ținând butonul apăsat. Textul astfel selectat își schimbă culoarea.
 - Butonul din mijloc: permite *copierea* la promptul sistemului a textului selectat, ca și cum el ar fi introdus de utilizator de la tastatură literă cu literă.
 - Butonul din dreapta: se comportă similar cu butonul din stânga, dar permite *selectarea* unei porțiuni mai mari de text: se deplasează cursorul la începutul textului de selectat, se apasă scurt butonul din stânga (*click!*) fără a deplasa mouse-ul, după eliberarea butonului se deplasează mouse-ul la sfârșitul textului de selectat, se apasă acum butonul din dreapta.

- **Când cursorul mouse-ului este plasat pe bordura de sus a unei ferestre**
 - Butonul din stânga: permite *mutarea ferestrei*;
 - Butonul din dreapta: permite *ridicarea ferestrei* la suprafață (deasupra altor ferestre de pe ecran) sau *ascunderea ei* (sub celelalte ferestre).

- **Când cursorul mouse-ului este plasat în afara oricărei ferestre (pe fundalul ecranului)**
 - Butonul din stânga: afișează un **meni** din care se pot *lansa în execuție* diferite programe sau utilitare. Dintre acestea, deosebit de utile sunt: prima linie a meniului, care permite *deschiderea unei noi ferestre de lucru*; ultima linie a meniului, care permite *ieșirea din sistem* (nu încercați această opțiune chiar în acest moment!).

- Butonul din mijloc: permite efectuarea de operații cu ferestrele de lucru: schimbarea *dimensiunilor* (resize), *mutarea* (move) și *închiderea* (kill) ferestrelor.
 - Butonul din dreapta: afișează o listă a ferestrelor deschise, și permite selectarea unei ferestre și implicit deplasarea rapidă a cursorului în *fereastra activată*.
- **Când cursorul mouse-ului este plasat pe bordura laterală a unei ferestre**
 - Cu butonul din stânga se poate agăța un colț al ferestrei și să se *mărească* sau *micșoreze* dimensiunile ferestrei.
 - **Observație:** Schimbarea dimensiunii unei ferestre la dimensiunea maximă (cât tot ecranul) se poate face și “apăsând” (cu ajutorul butonului din stânga al mouse-ului) al doilea buton (pătrățelul mare) aflat în partea din dreapta sus a bordurii ferestrei.

Dacă la un moment dat sunt deschise mai multe ferestre, în fiecare dintre ele se pot da comenzi independente, ca și cum utilizatorul ar avea la dispoziție mai multe terminale (ecrane). Gestionarea unui număr mare de ferestre (care la un moment dat pot să se suprapună, unele dintre ele nefiind vizibile) este facilitată de:

- bara orizontală plasată în partea de jos a ecranului; ea conține câte un buton pentru fiecare fereastră deschisă; apăsarea unui astfel de buton are ca efect mutarea cursorului în fereastra corespunzătoare și selectarea acesteia ca fereastră activă;
- cele trei căsuțe intitulate **Desktop** din Toolbox (grupul de butoane plasate în colțul dreapta jos al ecranului, fig. 1.2); să notăm căsuțele cu numere de ordine: **Desktop0**, **Desktop1**, **Desktop2**. Ele pot fi privite ca porțiuni din “ecranul logic” pe care se pot plasa ferestre. Numai una din cele trei părți ale acestui “ecran logic” este afișată, la un moment dat, pe ecranul fizic (la început, **Desktop0**).



Fig. 1.2: Toolbox

1.3 Ieșirea din sistem

- Se închid pe rând toate ferestrele de lucru deschise, de exemplu folosind comanda `exit`; acest pas nu este obligatoriu, dar evită eventuala pierdere a unor date nesalvate;
- Se apasă butonul din stânga al mouse-ului (cursorul acestuia fiind plasat pe fundal, nu într-o fereastră) și se selectează ultima linie din meniu (`Exit Fvwm95`);
- La noul meniu apărut se răspunde (prin selectare cu mouse-ul) `Yes, really quit`.

Încheierea sesiunii de lucru este obligatorie când părăsiți laboratorul, în caz contrar riscați ca la următoarea ședință sa nu mai fiți recunoscut ca utilizator al sistemului!

1.4 Sesiunea de lucru și configurarea ei

- `login`

Permite accesul unui utilizator în sistem de la o consolă a sistemului. Se solicită introducerea numelui de utilizator și a parolei.

- `rlogin nume_mașină` (remote login), `ssh nume_mașină` (secure shell)

Permite accesul unui utilizator la un sistem aflat “la distanță” (în altă rețea decât rețeaua locală la care utilizatorul este conectat).

Parametrul *nume_mașină* reprezintă numele calculatorului aflat la distanță. Programul solicită introducerea parolei. În mod implicit, se consideră că numele de utilizator de pe mașina de la distanță este același cu numele de utilizator din rețeaua locală.

Dacă numele de utilizator pe mașina de la distanță este altul decât cel curent, trebuie adăugat încă un parametru, sub forma:

```
ssh nume_mașină -l nume_utilizator
```

Se recomandă folosirea comenzii `ssh` deoarece transmite parola criptată și nu în clar, fiind astfel mai sigură decât `rlogin`

- `exit`

Oprește execuția interpretorului de comenzi Unix și încheie sesiunea de lucru.

În cazul lucrului într-un sistem bazat pe ferestre, are ca efect închiderea ferestrei (într-un astfel de sistem se lansează câte un interpretor de comenzi în fiecare fereastră deschisă). În cazul lucrului cu o singură consolă de ieșire, are ca efect ieșirea din sistem.

- `passwd` – permite schimbarea parolei.

Se solicită introducerea vechii parole și apoi a celei noi (de două ori, pentru verificarea introducerii corecte a noii parole).

Parola trebuie să fie suficient de lungă (numărul minim de caractere variind de la sistem la sistem), să conțină atât litere cât și caractere speciale (numere, semne de punctuație).

Recomandări: Schimbați parola periodic. Alegeți-vă o parolă pe care să o știți numai Dvs., să fie ușor de memorat, deoarece dacă ați uitat-o nu mai puteți intra în sistem. Nu ignorați mesajul afișat la intrarea în sistem, care vă poate avertiza în legătură cu data de expirare a parolei. Nu comunicați nimănui parola și nu o notați.

- `lock` sau `xlock` – blochează terminalul până la introducerea parolei corecte a utilizatorului.

Vă recomandăm să folosiți această comandă de câte ori părăsiți locul de lucru din fața calculatorului, pentru a evita folosirea contului Dvs. de către persoane neautorizate.

În rețeaua din sala EB212, această comandă este apelată în mod automat dacă nu dați nici o comandă de la tastatură sau prin mouse, timp de câteva minute.

Țineți cont de faptul că în lipsa Dvs. alți utilizatori pot avea nevoie să lucreze în sistem, deci, dacă intenționați să lipsiți mai mult de cinci minute, închideți sesiunea de lucru (în caz contrar riscați penalizări din partea administratorului de sistem, cum ar fi suspendarea contului pe o perioadă).

- `hostname` – afișează numele mașinii pe care lucrați.
- `uname` – afișează numele sistemului de operare folosit.

Pentru utilizări avansate:

- `printenv` – afișează lista variabilelor de configurare a sesiunii de lucru.

Valorile acestor variabile se pot schimba cu:

- `setenv` *variabila valoare* (în shell-ul `csh`)

`set` *variabila=valoare*

`export` *\$variabila* (în shell-ul `sh`)

`declare` *variabila=valoare* (în shell-ul `bash`)

Exemplu: `declare -x TMP="/home/irina/tmp"`

Aceste comenzi sunt specifice **shell**-ului și nu sistemului de operare **Unix**!

Comenzile de configurare pot fi scrise în fișiere de configurare specifice shell-ului. Numele acestor fișiere începe cu `.` urmat de numele shell-ului, urmat de literele **rc**.

Exemplu: `.bashrc` – pentru shell-ul `bash`

`.cshrc` – pentru shell-ul `csh`

Printre variabilele predefinite se numără: **HOME**, **SHELL**, **TERM**, **LOG-NAME**, **PATH**, **MAIL**, **TZ**.

Odată definite, valoarea unei variabile se obține punând semnul `$` înaintea numelui acesteia.

Exemplu: `echo $TMP` – afișează valoarea curentă a variabilei de configurare `TMP`.

1.5 Interpretorul de comenzi

În sistemele Unix interpretoarele de comenzi ale utilizatorului (shell) sunt de fapt tratate de sistemul de operare ca și orice alt program executabil. În consecință, există o mare diversitate de astfel de interpretoare care se pot împărți în două categorii:

- cele specifice Unix System V (la care prompterul este `$`):
 - Bourne Shell: `sh`
 - Korn Shell: `ksh`
 - GNU Bourne Again Shell: `bash`
- cele specifice sistemelor de tip BSD (la care prompterul este `%`), cel mai cunoscut fiind Cshell: `csh`.

După cum ați văzut în paragraful anterior, `ssh` (Secure SHell) are un rol aparte, permițând conectarea de la distanță.

În mod normal, comenzile către shell sunt date de la tastatură, dar ele pot fi preluate și dintr-un fișier. Un **script** este un fișier care conține comenzi ale shell-ului, utilizat pentru a executa succesiuni lungi de comenzi Unix sau pentru a scrie scurte “programe” care sunt apoi apelate prin numele de fișier și sunt tratate la fel cu orice comandă Unix. În acest fel se obține, practic, o extindere de către utilizator a comenzilor recunoscute de sistem.

1.6 Comenzi pentru lucrul cu fișiere și directoare

1.6.1 Structura de directoare și fișiere

Ca orice sistem modern de operare, Unix operează cu două concepte legate de stocarea ierarhizată a informației:

- **fișiere:** entități care conțin informații (text, instrucțiuni de programare);
- **directoare:** entități care conțin fișiere și care permit gruparea și ordonarea acestora.

Folosind similitudinea cu o bibliotecă, fișierele se aseamănă cu cărțile, în timp ce directoarele se aseamănă cu rafturile în care acestea sunt așezate. La rândul lor, rafturile pot fi grupate în dulapuri de cărți, care în Unix sunt tot directoare. Fiecare “dulap de cărți”, raft sau carte au un nume. Pentru a preciza unde se află o carte, trebuie precizate toate informațiile care să permită găsirea cărții: dulapul, raftul, titlul. Tot astfel în Unix trebuie precizată toată *calea* pe care se poate ajunge la un director sau fișier. Calea reprezintă o succesiune de nume de directoare (eventual ultimul nume din succesiune putând fi un nume de fișier), separate prin caracterul / (în limba engleză *slash*).

Numele fișierelor și directoarelor pot conține litere, cifre, caractere speciale: `_` (underscore sau caracter de subliniere), `-` (minus) și `.` (punctul), etc. Spre deosebire de DOS, numele de fișiere pot conține oricâte puncte.

Exemplu:

```
program1, tema.pas.bak, tema-1, tema_2
```

sunt nume valide de fișiere sau de directoare.

Caracterul `*` înlocuiește oricâte caractere din numele unui fișier. Caracterul `?` înlocuiește un singur caracter.

Exemple:

```
te*a
```

se poate referi la oricare din următoarele nume de fișiere: `tema`, `tematica`, `teza`, `tt.c.bak.a`

```
te?a
```

se poate referi la nume de fișiere precum `tema`, `teza`, `te.a`, dar nu și la `tematica`, `tt.c.bak.a`

Un grup de caractere de tipul `[a-z]` selectează un singur caracter, cuprins (în lista codurilor ASCII) între `a` și `z`.

Exemplu:

```
[g-i]*
```

selectează toate fișierele al căror nume începe cu litera `g` sau `h` sau `i`.

Directoare “speciale”

- **directorul curent** – este simbolizat printr-un singur punct.

- **directorul precedent** în arborele de directoare – este simbolizat prin `..` (de două ori punct).
- **directorul rădăcină** – cel care conține toate celelalte directoare și fișiere ale sistemului, este simbolizat prin `/` (*slash* în limba engleză).
- **directorul home** – cel în care este plasat utilizatorul la intrarea în sistem, este simbolizat prin `$HOME` sau prin `~`. În principiu, un utilizator are drepturi depline (de citire și de scriere) doar în directorul home și în toate directoarele care se află în acesta.

În figura 1.3 se prezintă o parte din structura tipică a directoarelor unui sistem Unix.

Pe calculatoarele PC din sala EB212 este instalat atât Linux cât și Windows95 (intrarea în Windows făcându-se din Linux cu comanda `dos`). Aceasta este explicația prezenței directoarelor nestandard `/dos/c/` și `/dos/d/`, care vă permit accesul din Linux la fișierele create sub sistemele de operare DOS și Windows.

În sistemul prezentat în figura 1.3, dacă numele Dvs. de utilizator este `ioan`, directorul home este `/home/elth/an6/ioan`.

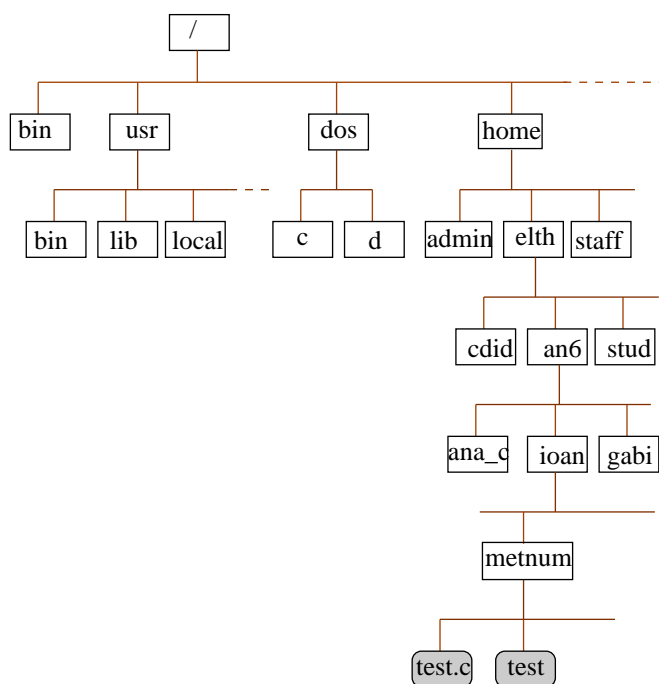


Fig. 1.3: Structura directoarelor unui sistem Unix

Vă recomandăm ca pentru fiecare din proiectele Dvs. să folosiți un director dedicat.

Pentru a avea acces la un fișier trebuie să specificați calea, ca în exemplele:

- absolut:

```
/home/elth/an6/ioan/metnum/test.c
```

- relativ (presupunând că vă aflați în directorul `/home/elth/an6/ioan/`):
`metnum/test.c`

1.6.2 Comenzi pentru lucrul cu fișiere

- `ls` – afișează conținutul (lista directorelor și fișierelor) directorului curent.

Comanda `ls` poate fi apelată cu parametri, de exemplu:

```
ls -l
```

pentru afișarea conținutului directorului curent, cu informații detaliate privitoare la drepturile de citire (**r**), scriere (**w**) sau execuție (**x**) asupra fișierului, data și ora ultimei modificări, dimensiunea fișierului, etc.

```
ls -a
```

pentru listarea tuturor fișierelor din directorul curent, inclusiv cele al căror nume începe cu punct (prin convenție, numele fișierelor “ascunse” ale sistemului de operare Unix încep cu punct).

- `cat nume_fișier` – listează conținutul unui fișier.
- `more nume_fișier` – listează conținutul unui fișier, cu oprirea afișării după fiecare pagină.

După lansarea `more`, sunt active următoarele comenzi:

- `spațiu` – avansare cu o pagină spre sfârșitul fișierului;
- `b` – avansare cu o pagină spre începutul fișierului;
- `<␣>` – avansare cu un rând înainte;
- `q` – ieșire din `more`.

- `cp nume_fișier_existent nume_nou` – copiază fișierul `nume_fișier_existent` sub numele `nume_nou`.
- `mv nume_fișier_existent nume_nou` – redenumeste fișierul `nume_fișier_existent`.
`mv nume_fișier_existent nume_director` – mută fișierul în directorul `nume_director`.
- `rm nume_fișier_existent` – șterge fișierul.

Comanda `rm` poate fi apelată cu parametri. De exemplu, un parametru deosebit de util este `-i`:

```
rm -i nume_fișier_existent
```

– șterge fișierul cu solicitarea unei confirmări.

1.6.3 Comenzi pentru lucrul cu directoare

- `cd nume_director` – schimbă directorul curent, în care *nume_director* este o cale absolută sau relativă către un director.

Exemplu: Efectul următoarelor trei comenzi:

```
cd /           schimbă directorul curent în directorul rădăcină
cd stiinta     schimbă directorul curent în /stiinta (deplasare relativă)
cd matematica
```

este echivalent cu al comenzii:

```
cd /stiinta/matematica
```

Comanda `cd` fără nici un nume de director are ca efect mutarea în directorul *home*.

- `pwd` – afișează numele directorului curent (calea absolută).
- `mkdir nume_director` – crează un nou director. Dacă numele directorului nu este precedat de o cale, noul director este creat în cel curent.

Exemplu:

`mkdir modelare` – crează directorul `modelare` în directorul curent

`mkdir /modelare` – crează directorul `modelare` în directorul rădăcină (de obicei, această comandă va genera un mesaj de eroare, deoarece nu aveți drepturi de scriere în directorul rădăcină).

- `rmdir nume_director` – șterge directorul.

Dacă directorul conține alte fișiere sau directoare, el nu poate fi șters cu comanda anterioară.

`rm -rf nume_director` – șterge recursiv (`-r`) și forțat (`-f`) toate fișierele și directoarele aflate în directorul *nume_director*.

Atenție! Comenzile `rm` și `rmdir` au efect definitiv! Un fișier sau director, odată șters, nu mai poate fi recuperat!

1.6.4 Schimbarea drepturilor asupra fișierelor/directoarelor

- `chmod abc nume`

Schimbă drepturile asupra fișierului sau directorului *nume*, conform cu parametrii *abc*.

Grupul celor trei litere semnifică: prima literă, al cui drept este modificat; a doua, în ce sens este modificat (acordat sau retras); a treia, care drept este modificat (de citire, de scriere sau de execuție).

a – poate lua valorile: *u* –utilizatorul; *g* –grupul din care face parte utilizatorul; *o* –toți ceilalți utilizatori.

b – poate lua valorile: **+** –acordarea dreptului; **-** –retragerea dreptului.

c – poate lua valorile: **r** –drept de citire; **w** –drept de scriere (modificare); **x** –drept de execuție.

Exemplu: `chmod u+x test1` – acordă drept de execuție proprietarului fișierului `test1` (`test1` devine un fișier executabil ca orice comandă Unix).

- `umask cod`

Schimbă drepturile implicite asupra fișierelor conform codului *cod*.

1.6.5 Căutări

- `locate nume`

găsește toate fișierele și directoarele al căror nume include *nume*. Căutarea se face în toate directoarele sistemului în care utilizatorul are drept de citire.

- `find cale opțiuni`

găsește toate fișierele care satisfac condițiile din *opțiuni*, care se află în directorul *cale* și în toate subdirectoarele acestuia.

Exemplu: `find /usr -name "test*" -print`

caută recursiv, în directorul `/usr` și în subdirectoarele acestuia, toate fișierele/directoarele al cărui nume începe cu `test` și afișează rezultatul pe ecran. Parametrul opțiunii `-name` (numele fișierului) trebuie inclus între ghilimele dacă el conține caractere speciale (de tipul `*`).

- `grep text nume_fișier`

Afișează toate liniile din *nume_fișier* care conțin *text*. Dacă textul conține blankuri sau caractere speciale, el trebuie pus între ghilimele. Numele *nume_fișier* poate conține caracterele `*` sau `?`

Exemple: `grep alias .bashrc`

`grep "am fost" *doc`

1.6.6 Comparări de fișiere

- `diff nume_fis1 nume_fis2`

Dacă fișierele sunt ASCII, afișează liniile care diferă în cele două fișiere.

- `sdiff nume_fis1 nume_fis2`

Afișează conținutul celor două fișiere, unul lângă altul, marcând cu `|` liniile care diferă și cu `<` sau cu `>` liniile care există doar într-unul din cele două fișiere.

1.7 Indirectări

- *comanda1* | *comanda2*

Execută comanda *comanda1* și transmite rezultatul ca intrare pentru comanda *comanda2*.

Exemple: `sdiff test1 test2 | more`

Afișează rezultatul comenzii `sdiff` pagină cu pagină (conform comenzii `more`).

`cat test* | grep "am fost"`

Afișează acele linii din fișierele al căror nume începe cu `test`, care conțin caracterele "am fost".

- *comanda1* > *nume_fișier* Execută comanda *comanda1* și scrie rezultatul în fișierul *nume_fișier* în loc să îl afișeze la consolă. Dacă fișierul există, el este rescris.

Exemplu: `ls -l > dir`

- *comanda1* >> *nume_fișier*

Execută comanda *comanda1* și scrie rezultatul la sfârșitul fișierului *nume_fișier* în loc să îl afișeze la consolă. Dacă fișierul nu există, el este creat.

- *comanda1* < *nume_fișier*

Execută comanda *comanda1*, preluând datele de intrare din fișierul *nume_fișier* în loc să le preia de la consolă.

1.8 Imprimarea fișierelor

- `lpr nume_fișier`

Trimite la imprimantă conținutul fișierului.

Trebuie notat că această comandă este specifică sistemelor Unix de tip SVR4, în timp ce la sistemele de tip BSD ea are forma:

`lp nume_fișier`

Acesta este cel mai simplu mod să vă dați seama ce tip de Unix utilizați!

Observație: În rețeaua de calculatoare Pentium, această comandă **nu funcționează** decât dacă fișierul este în format PostScript, acesta fiind singurul format recunoscut de imprimantă. De aceea, pentru a imprima un fișier text, dați următoarele două comenzi:

`a2ps nume_fișier > tmp.ps`

`ghostview tmp.ps &`

imprimare imprimați fișierul folosind butonul corespunzător al programului Ghostview. Înainte de a imprima un document mai lung, revedeți regulamentul de funcționare (anexa B).

- `lpq`

Afișează coada de așteptare a documentelor trimise la imprimantă de toți utilizatorii.

1.9 Lansarea și oprirea comenzilor

- *comandă* <↵> – Lansează comanda în execuție, în “foreground” (nu se mai pot da alte comenzi din fereastra în care a fost lansată, până la terminarea execuției comenzii).
- *comandă* & <↵> – Lansează comanda în execuție, în “background” (fără blocarea promptului).
- <Ctrl> Z – Suspendă execuția comenzii curente, lansată în foreground.
- `bg` – Repornește, în background, o comandă suspendată cu <Ctrl> Z.
- `fg` – Trece în foreground o comandă lansată în background.
- <Ctrl> C sau <Ctrl> D – oprește execuția unei comenzi.

1.10 Comenzi pentru controlul proceselor și utilizatorilor

- `who` – afișează lista utilizatorilor conectați în sistem
`w` – afișează lista utilizatorilor conectați în sistem și comanda pe care aceștia o execută;
`whoami` – “Cine sunt eu?” – afișează numele de cont al utilizatorului;
`finger nume` – “Cine este ...?” – afișează date despre toți utilizatorii al căror nume de familie, prenume sau nume de utilizator este *nume* (printre altele, se afișează numele de utilizator, numele și prenumele, precum și ultima dată când și-a citit poșta electronică);
- `ps` – lista proceselor.
 Sub Unix, mai multe comenzi sau programe pot fi rulate “în paralel” (de exemplu, dacă aveți două ferestre deschise puteți da comenzi care vor fi executate în mod independent). Fiecare comandă în curs de execuție se numește *proces* și este identificată printr-un număr, numit PID (Process IDentificator).
 Comanda `ps` fără nici un parametru afișează lista proceselor lansate de utilizatorul curent. Comanda poate fi lansată și cu parametri: de exemplu, `ps -aux` listează toate procesele în curs de execuție, cu informații despre utilizator, numărul procesului, timpul CPU ocupat, etc.
- `kill PID` – oprește definitiv execuția procesului cu numărul *PID*.

1.11 Ajutor!

Comanda care permite aflarea funcției și sintaxei unei comenzi, precum și multe alte informații despre comandă, este:

- `man nume_comandă`

După lansarea ei, sunt active comenzile de deplasare descrise la `more` (paragraful 1.6.2).

- `man -k cuvânt_cheie` sau `apropos cuvânt_cheie`

– caută cuvinte cheie în lista de comenzi

`whatis cuvânt_cheie` – caută cuvinte cheie în lista de comenzi Unix și afișează numai acele comenzi care conțin cuvântul ca atare (nu și dacă el este parte a altui cuvânt mai lung).

Pentru a observa diferența între `apropos` și `whatis`, încercați, de exemplu,

```
apropos dos și
whatis dos
```

1.12 Transferul fișierelor între DOS și Unix

Fișierele `text` Unix se deosebesc de cele create sub DOS prin aceea că sfârșitul de linie este marcat doar prin `line-feed` și nu prin combinația `carriage return + line-feed` ca în DOS. Pentru a transforma un format în altul se folosesc comenzile:

- `fromdos < nume_fis_DOS > nume_nou`
- `todos < nume_fis_Unix > nume_nou`

Pe diverse mașini, aceste comenzi se numesc diferit. Încercați `man -k dos`.

1.13 Diverse utilitare

- `date` – afișează data și ora;
- `cal` – afișează calendarul lunii curente;
`cal 08 1998` – afișează calendarul lunii august a anului 1998;
- `echo text` – afișează `text` pe ecran;
`echo $VAR` – afișează valoarea variabilei de configurare `VAR` pe ecran;

- `xclock` – deschide o fereastră în care se afișează ora, în mod implicit în format analogic;
`xclock -digital` – deschide o fereastră în care se afișează ora în format digital;
- `calc`, `xcalc` – emulează un calculator de buzunar;
- `sort nume_fisier` – sortează în ordine alfabetică liniile din fișier și afișează rezultatul pe ecran;
- `clear` – șterge conținutul ferestrei de lucru;
- `wc nume_fisier` – numără liniile, cuvintele și caracterele din fișier și afișează rezultatul pe ecran;
- `quota` – afișează spațiul de disc ocupat de utilizator și limita maximă alocată. Vă recomandăm să folosiți această comandă și să compactați fișierele pe care nu le utilizați în mod curent, pentru a nu depăși limita impusă.
- `xterm` – deschide o nouă fereastră X.

Poate avea diferiți parametri, printre care:

- T *nume_fer* sau -title *nume_fer* – setează numele ferestrei ca fiind *nume_fer* (implicit: XTerm)
- sb – fereastră cu scroll-bar;
- +sb – fereastră fără scroll-bar;
- e *comandă* – deschide fereastra și execută în ea comanda;

Exemplu:

```
xterm -T Conex_Heineken -sb -e ssh heineken &
```

1.14 Exerciții

EXERCITIUL 1

Inițiați sesiunea de lucru.

Aflați care este numele sistemului de operare, numele mașinii pe care lucrați și numele sub care vă cunoaște sistemul. Aflați ce alte informații deține sistemul despre Dvs.

Introduceți textul “EB-210” urmat de <↵> (Enter) la promptul din fereastra de lucru. Faceți aceeași operațiune folosind butoanele mouse-ului pentru a copia textul. Comentați utilitatea acestei ultime tehnici.

Cu excepția acestui exercițiu, NU IGNORAȚI NICIODATĂ MESAJELE DE EROARE afișate!!! (Ele sunt, la acest exercițiu, inofensive.) Dacă nu știți ce înseamnă, întrebați, dar NU LE IGNORAȚI!

EXERCITIUL 2

Deschideți o nouă fereastră.

Folosind butonul din mijloc, schimbați-i dimensiunile. Mutați fereastra în colțul din stânga jos al ecranului.

Lansați în această fereastră primele patru utilitare prezentate în paragraful 1.13.

Închideți fereastra nou creată.

EXERCITIUL 3

a) Deschideți încă două ferestre. Observați efectul asupra căsuței `Desktop0`.

b) Deplasați-vă din fereastră în fereastră (activând astfel una sau alta dintre ferestre), în două moduri:

- deplasând cursorul mouse-ului;
- folosind bara orizontală din partea de jos a ecranului.

c) Mutați o fereastră în `Desktop1` astfel:

- mutați fereastra spre dreapta până când o jumătate a ei nu mai este vizibilă (observați efectul asupra căsuței `Desktop1`);
- deplasați cursorul mouse-ului pe căsuța `Desktop1` și apăsați butonul din stânga al mouse-ului;
- poziționați fereastra în `Desktop1`.

Comentați utilitatea celor trei desktop-uri.

EXERCITIUL 4

a) Afișați lista fișierelor din directorul curent. Deschideți o altă fereastră și repetați comanda.

b) Afișați pe ecran, într-una din ferestrele de lucru, conținutul fișierului `/usr/local/man/whatis`.

EXERCITIUL 5

a) Copiați fișierul `.cshrc` sub numele `test`. Listați conținutul fișierului `test`.

Redenumiți `test` în `test1`.

Listați conținutul directorului curent. Ștergeți fișierul `test1`.

b) Fără a le executa, încercați să prevedeați efectul fiecăreia din următoarele instrucțiuni:

```
cp .bashrc temporar
mv temporar temp
rm temporar
rm temp
```

Executați aceste instrucțiuni și observați efectul lor.

EXERCITIUL 6

Mutați-vă în directorul Dvs. *home*:

```
cd <↵>
```

Listați conținutul acestui director.

Creați un nou director *metnum* în care veți lucra de acum înainte la laboratorul de “Metode numerice în ingineria electrică”. Listați noul conținut al directorului *home*.

În directorul *metnum*, creați un nou director, *tema1*.

Ați reușit?

Vă recomandăm să creați câte un director pentru fiecare temă (*tema2*, *tema3*, ...) și să lucrați de fiecare dată în directorul corespunzător.

În directorul Dvs. *home* nu trebuie să existe nici un fișier (cu excepția celor utilizate de sistemul de operare, al căror nume începe cu punct), ci numai directoare!

EXERCITIUL 7

Listați procesele în curs de execuție. Mutați-vă în directorul Dvs. *home*. Într-o fereastră, dați comanda:

```
more /usr/local/man/whatis
```

În altă fereastră, dați comanda *ps*

Observați diferențele față de rezultatul precedentei comenzi *ps*.

Lucrând în a doua fereastră, opriți execuția procesului care corespunde comenzii

```
more usr/local/man/whatis.
```

EXERCITIUL 8

a) Folosind comanda *man* aflați care este efectul comenzii *cal*.

b) Copiați fișierul *.bashrc* din directorul *home* în directorul *~/metnum/tema1/*.

Comparați efectul comenzilor

```
sort .bashrc
```

și respectiv

```
sort .bashrc > sortat
```

(Indicație: încercați `ls` și `more ...`).

EXERCITIUL 9

Comparați conținutul fișierelor `~.bashrc` și `~/metnum/tema1/.bashrc`, folosind `diff` și `sdiff`. Efectuați aceeași operație asupra fișierelor `~.bashrc` și `~/metnum/tema1/sortat`.

EXERCITIUL 10

Creați un fișier numit `dir` care să conțină lista fișierelor din directorul curent.

EXERCITIUL 11

Mutați-vă în directorul `Dvs. home`. Găsiți (cu `find`) fișierul numit `sortat`.

EXERCITIUL 12

Blocați accesul la terminal cu comanda `xlock`.

Deblocați accesul și încheiați sesiunea de lucru.

Intrați din nou în sistem și modificați-vă parola. Nu uitați noua parolă!

EXERCITIUL 13

Alcătuți o listă cu cele mai importante 10 comenzi Unix, în opinia Dvs. Aflați mai multe detalii despre aceste comenzi folosind comanda `man`.

Nu uitați să închideți sesiunea de lucru înainte de a părăsi laboratorul.

Dacă vreți să aflați mai multe despre Unix, vă recomandăm să consultați una din cărțile [55], [65] sau [73].

Capitolul 2

Editarea textelor

— vi și Emacs —

Textul conceput ca un șir (liniar!) de caractere (litere, cifre, semne de punctuație și caractere speciale, de exemplu <linie nouă>) reprezintă unul din conceptele fundamentale ale comunicării umane și în particular a celei tehnico-științifice. El reprezintă baza majorității documentelor, fie că acestea sunt rapoarte științifice sau programe de calculator, reprezentate pe hârtie, pe ecran sau electronic (pe suport magnetic sau în memoria internă a calculatoarelor).

Pentru a facilita operațiile de introducere în calculator, vizualizare și modificare ale textelor s-au dezvoltat mai multe programe capabile să manipuleze interactiv texte, numite *editoare*.

Editarea textelor este o operație esențialmente interactivă și ea trebuie deosebită de cea de *procesare* a textelor, care este în principiu neinteractivă. Procesarea textelor presupune prelucrarea automată a textelor, bazată pe căutări și substituiri condiționate (ca de exemplu înlocuirea diacriticelor românești cu literele simplificate corespunzătoare, sau invers!). Cu toate că pentru procesare s-au dezvoltat programe specializate, cum sunt **sed**, **awk**, **lex**, etc. [36], un editor de texte performant trebuie să ofere și el funcții de procesare (cel puțin “*search and replace all*”). De asemenea, editarea textelor nu trebuie confundată cu *editarea documentelor*, în care conceptele de *font* și așezare în pagină sunt esențiale.

Dacă sub sistemul de operare DOS editorul clasic este *WS (Word Star)*, iar sub Windows editorul clasic este *MS-Word*, în cazul sistemului Unix editorul “nativ” este **vi**. Cu toate că se spune despre acest editor că este dificil de învățat, cunoștințele elementare (care presupun memorarea a 10 maxim 20 de mnemonici formate dintr-o literă) vă sunt suficiente pentru a vă putea descurca în orice versiune de sistem de operare Unix.

Un alt editor celebru pentru Unix, extrem de performant și productiv, este **Emacs**. Și în acest caz, cunoașterea a 10-20 mnemonici (din păcate, altele decât în *vi*!) vă este suficientă pentru a putea efectua operații elementare de editarea textelor.

Acestea nu sunt singurele două soluții posibile. În LMN sunt instalate și alte editoare, mai puțin folosite, cum sunt *Joe* (editor ce emulează WS) și *Xedit* (editor simplu de tip Note-pad).

Indiferent ce editor veți folosi în viitor, vă recomandăm să vă familiarizați cel puțin cu două dintre acestea pentru a înțelege conceptele specifice editării de texte.

După ce v-ați ales editorul, exersați cât mai mult cu el. Productivitatea Dvs. în continuare (indiferent la ce veți folosi calculatorul în activitatea științifică și tehnică) va depinde esențial de viteza cu care reușiți să editați textele!

Scopul capitolului este de a deprinde operațiunile elementare de editare a textelor:

- lansarea editorului;
- introducerea unui text;
- modificarea unui text (ștergerea, adăugarea, copierea și mutarea unor părți din text, căutarea și înlocuirea);
- citirea și scrierea în fișiere;
- încheierea sesiunii.

2.1 Editarea textelor cu vi

Programul de editare a textelor *vi* [33] [52] a fost scris de *Bill Joy* când era student la Universitatea Berkley, ca o îmbunătățire a editorului de linie *ed* dezvoltat de *Bell Labs* (AT&T), în cadrul proiectului UNIX.

vi este un editor puternic, care permite editarea interactivă a textelor. Editorul *vi* are un avantaj esențial: este parte componentă a oricărui sistem Unix, indiferent de variantă. Dezavantajul principal al acestui editor constă în faptul că se bazează pe două moduri de lucru (text și comandă), ceea ce îl face greu de acceptat de cei care au experiența utilizării unui editor de documente (nu de texte!) de tip *WYSIWYG* (*What You See Is What You Get*) cum este MS-Word.

Varianta de *vi* instalată în laboratorul EB212 se numește *vim* (Vi IMproved) [27] și conține multe facilități suplimentare față de varianta clasică.

2.1.1 Apelare

- Lansarea variantei clasice:

```
vi fis
```

în care *fis* este opțional și reprezintă numele fișierului ce trebuie editat.

- Lansarea variantei cu interfață grafică (GUI) în altă fereastră X:

```
vi -g fis
```

În acest caz se deschide o nouă fereastră (figura 2.1), în care sunt disponibile toate comenzile variantei clasice (și multe altele, specifice **VIM**), în plus, o parte dintre operații putând fi efectuate folosind mouse-ul.



Fig. 2.1: Fereastră grafică a editorului VIM

Comenzile prezentate mai jos sunt cele de bază, disponibile în toate versiunile editorului vi.

2.1.2 Moduri de lucru

Există două moduri de lucru ale editorului vi:

- mod “comandă” (este cel inițial, după lansarea în execuție a editorului);
- mod “text” (modul în care introduceți text, dar în care intrați printr-o comandă).

Din modul de comandă se pot face următoarele operații:

- intrare în modul text;
- deplasarea cursorului;
- introducerea unor comenzi de editare;
- efectuarea unor căutări și înlocuiri de șiruri de caractere;
- salvarea fișierului și ieșirea din programul vi;

- altele (comenzi `ex`, Unix shell).

Din modul text se pot face următoarele operații:

- introducere text;
- trecere înapoi în mod comandă (cu `<ESC>`).

2.1.3 Modul comandă

Intrarea în mod text

Se poate face în mai multe variante:

- a - adăugare după cursor
- A - adăugare la sfârșit de linie
- C - modificarea restului liniei
- i - inserare înaintea cursorului
- I - inserare la începutul liniei
- o - linie nouă după linia curentă
- O - linie nouă înaintea liniei curente
- R - scriere "peste" (replace) – mai multe caractere
- s - înlocuire un caracter
- S - înlocuire întreaga linie

Comenzi de deplasare a cursorului

- l - dreapta
- h - stânga
- j - jos
- k - sus
- <blanc> - dreapta
- w, W - înainte cu un cuvânt
- b, B - înapoi cu un cuvânt
- e - sfârșit cuvânt
-), (- propoziție următoare, curentă
- }, { - paragraf următor, curent
- 0 - începutul liniei curente
- \$ - sfârșitul liniei curente
- ^ - primul caracter diferit de blanc din linia curentă
- +, - - primul caracter al liniei următoare, precedente
- <↵ > - primul caracter al liniei următoare
- H - început ecran
- M - mijloc ecran

L	- ultima linie ecran
Ctrl-F, Ctrl-B	- un ecran înainte, înapoi
Ctrl-D, Ctrl-U	- jumătate de ecran în jos, în sus
z	urmat de:
<↓> .	- linia cu cursorul plasată la început, mijloc, jos ecran
Ctrl-L, Ctrl-R	- redesenare ecran
G	- ultima linie din fișier
<n>G	- linia numărul <n> din fișier

Comenzi de editare

Au structura generală:

<operație><obiect>

Comenzi (operații) de bază:

c	- începutul unei modificări
d	- începutul unei ștergeri
y	- începutul unei copieri (yank)

Comenzile de editare acționează asupra unui obiect. Acesta poate fi:

- cuvânt = toate caracterele până la un blank sau un semn de punctuație
- propoziție = toate caracterele până la un ., ! sau ?, urmate de un spațiu
- paragraf = până la următoarea linie liberă
- secțiune = până la următorul titlu de secțiune, definit în mod corespunzător

Dacă obiectul este linia curentă, atunci numele obiectului este același cu numele operației:

cc	- modifică linia curentă
dd	- șterge linia curentă
yy	- copiază linia curentă

Comanda precedată de un număr n acționează asupra a n obiecte:

2dd - șterge două linii, începând cu cea curentă

Comenzi de editare

cw	- modificare cuvânt
cc	- modificare linie
C	- modificare de la cursor la sfârșit linie
dd	- ștergere linie
D	- ștergere rest linie
dw	- ștergere cuvânt

d}	- ștergere până la paragraf următor
d^	- ștergere înapoi până la început linie
rx	- înlocuire caracterul curent cu x
u	- undo ultima schimbare
U	- restaurare linie curentă
x	- ștergere caracter curent
X	- ștergere caracter la stânga
.	- repetă ultima modificare
p	- inserează ultimul text șters, după poziția cursorului
P	- inserează ultimul text șters, înainte de poziția cursorului
:r <fis>	- citește (include) fișierul <fis> în fișierul curent

Copiere și mutare

yy	- copiere linie curentă
"xyy	- copiere linie curentă în bufferul x
"xp	- pune conținutul bufferului x la poziția cursorului

Diverse

J - unește două linii

Căutări și înlocuiri

/text	- caută înainte text
?text	- caută înapoi text
/, sau n	- repetă ultima căutare, înainte
N	- repetă ultima căutare, înapoi
:s /<text>/<text1>/	- caută <text>, înlocuiește cu <text1>
:s /<text>/<text1>/g	- caută <text>, înlocuiește cu <text1>, în toate aparițiile lui <text> de pe linie
1,\$ s /<text>/<text1>/g	- același lucru, de la linia 1 la ultima linie (notată \$) din fișier

Salvare și ieșire

:x	- ieșire, cu scriere dacă s-au făcut modificări
:wq	- scriere și ieșire
:w	- scriere (salvarea fișierului curent)
:w <fis>	- salvarea fișierului curent sub numele <fis>
:q	- ieșire fără salvare
:q!	- ieșire fără salvare, chiar dacă s-au făcut modificări
:e <fis>	- editarea fișierului <fis>

2.1.4 Setare opțiuni

- în mod comandă:

```
:set <opțiune>
```

- în fișierul de configurare `.exrc`

```
set <opțiune>
```

<opțiune> e de forma:

```
<opt>      - setare  
no<opt>    - resetare  
x=val      - setează x la valoarea val
```

Exemple:

```
set ai - autoindent  
set noai - no autoindent  
set nu - numere de linii afișate pe ecran  
set tabstop=8 - numărul de caractere pentru un TAB  
set wrapmargin=72 - marginea din dreapta la 72 de caractere
```

2.1.5 Câteva facilități ale interfeței grafice cu utilizatorul

Interfața grafică a programului vi (în versiunea VIM) pune la dispoziția utilizatorului o serie de comenzi selectabile cu ajutorul mouse-ului.

Comenzile sunt grupate în șase meniuri (vezi și figura 2.1), după funcțiile lor: lucrul cu fișiere (**F**ile), comenzi de editare (**E**dit), lucrul cu ferestre (**W**indow), facilități pentru scrierea de programe (**I**DE), definirea tipului de fișier, pentru a permite verificarea automată a sintaxei (**S**yntax) și documentație (**H**elp).

Meniurile din care pot fi selectate diferitele comenzi din interfața grafică cu utilizatorul sunt prezentate în figura 2.1.5.

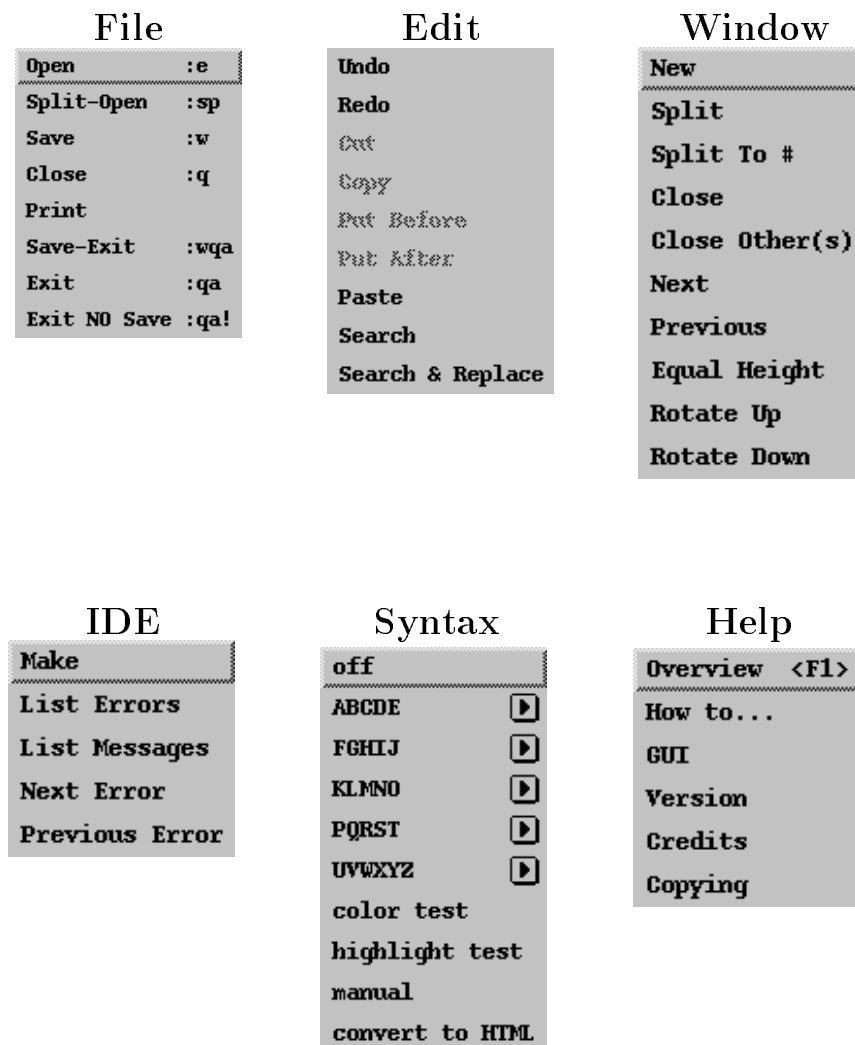


Fig. 2.2: Meniurile de comenzi ale interfeței grafice cu utilizatorul a programului VIM

2.2 Editarea textelor cu `emacs`

GNU Emacs [75] este un program extrem de puternic ce permite realizarea unei game largi de operații de editare de texte, dezvoltare de programe sau management de fișiere. Emacs (Editor MACroS) a fost dezvoltat de Richard Stallman și Guy Steele de la MIT și plasat în domeniul licenței gratuite GPL, domeniu gestionat de Free Software Foundation (înființată tot de Stallman), ca o componentă a proiectului GNU. Reamintim că acest proiect urmărește dezvoltarea unui set de componente și instrumente (software tools) care să fie alternative la sistemul UNIX clasic, dar să fie “free”, adică oferite gratis.

Emacs este nu numai o excelentă alternativă la editorul `vi` ci este un adevărat mediu eficient de gestiune a documentelor (texte ale unor documente, programe, mesaje electronice, etc.).

2.2.1 Lansare și ieșire din Emacs

- Lansare:

```
emacs
```

sau

```
emacs fisier
```

- Pentru a ieși din Emacs:

```
↑ x ↑ c
```

sau

```
↑ u ↑ x ↑ c
```

pentru a ieși fără salvare.

2.2.2 Ecranul Emacs

Ecranul Emacs (fig. 2.3) este alcătuit din (de sus în jos):

- zona de meniuri (File, Edit, ...) – disponibilă numai la varianta X a programului;
- zona de butoane (Open, DirEd, Save, ...) – disponibilă numai la varianta X a programului;
- spațiul de lucru (separat eventual în mai multe ferestre)
- linia de stare
- aria de ecou, în care sunt afișate mesajele Emacs
- opțional lista etichetelor cheilor de comandă (F1-F8)

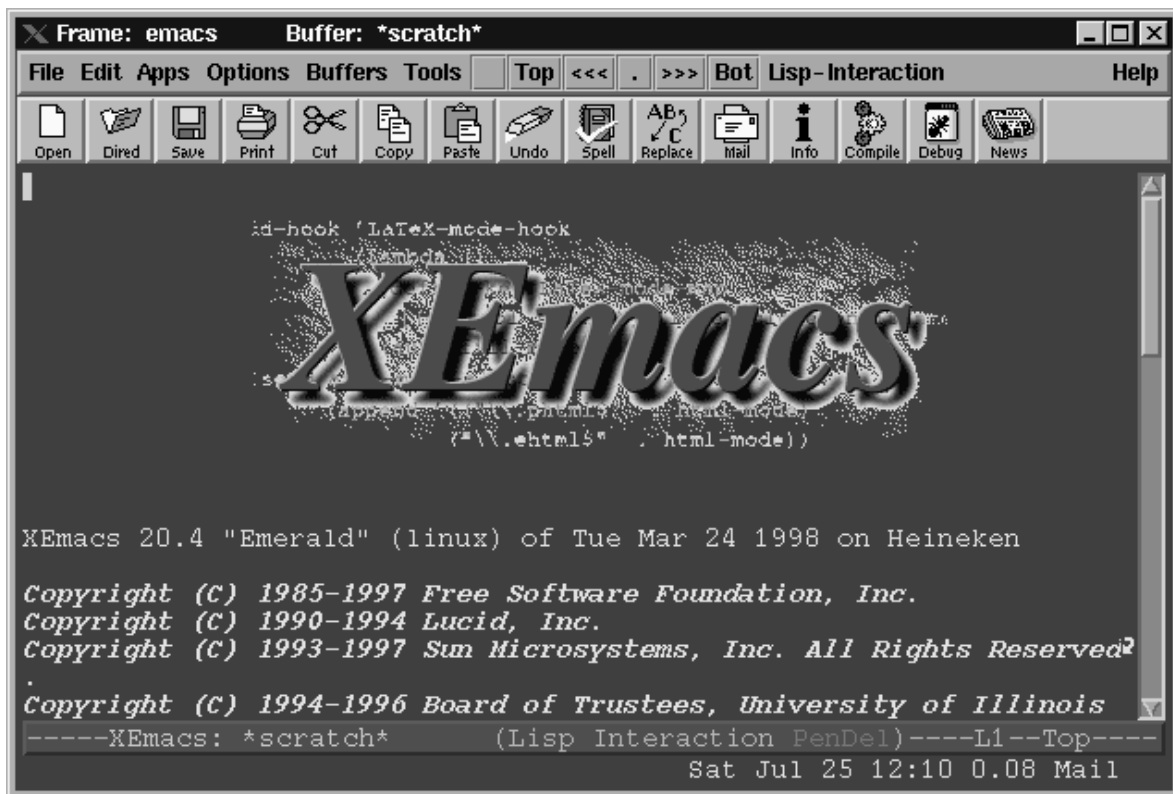


Fig. 2.3: Ecranul programului emacs

În orice ecran Emacs există un singur cursor la un moment dat, care marchează fereastra selectată.

Linia de stare conține în mod tipic:

- Starea (----- – text nemodificat, ----*** – text modificat);
- Sistem (Emacs, RMAIL, etc.);
- Nume buffer (fișier);
- Tipul și încărcarea mașinii (%);
- Mail – mesaj electronic necitit;
- Modul: Text, C, FORTRAN, etc. ;
- Poziția ferestrei în fișier.

Pentru a transmite comenzi în Emacs se folosesc două chei de comandă:

<Control> simbolizată ↑
 <Meta> simbolizată ↓

apăsate simultan cu caracterul următor. Tasta <Meta> este mapată pe tasta <Alt> sau pe <Esc> (caz în care nu trebuie apăsate simultan).

2.2.3 Comenzi de deplasare a cursorului

↑ <code>b</code>	- înapoi (back)
↑ <code>f</code>	- înainte (forward)
↑ <code>p</code>	- sus (previous)
↑ <code>n</code>	- jos (next)
↑ <code>a</code>	- început linie (mnemonica “alta”)
↑ <code>e</code>	- sfârșit linie (end)
↓ <code>b</code>	- înapoi cu un cuvânt
↓ <code>f</code>	- înainte cu un cuvânt
↓ <code>m</code>	- la primul caracter neblanc
↓ <code>a</code>	- propoziție înainte
↓ <code>e</code>	- propoziție înapoi
↓ <code>]</code>	- paragraf înainte
↓ <code>[</code>	- paragraf înapoi
↓ <code><</code>	- început fișier
↓ <code>></code>	- sfârșit fișier
↑ <code>v</code>	- scrie înainte (în jos <code>v</code>)
↓ <code>v</code>	- scrie înapoi (invers <code>v</code>)
↑ <code>x <</code>	- scroll la stânga
↑ <code>x ></code>	- scroll la dreapta
↑ <code>x]</code>	- pagină înainte
↑ <code>x [</code>	- pagină înapoi

2.2.4 Comenzi de ștergere

↑ <code>d</code>	- șterge caracter (delete)
↓ <code>d</code>	- șterge cuvânt
<code></code>	- șterge caracter anterior
↑ <code>k</code>	- șterge până la sfârșit de linie (kill)
↓ <code>k</code>	- șterge propoziție
↑ <code>x</code> ↑ <code>k</code> ↑ <code>k</code>	- șterge linia curentă (macrocomandă)
↑ <code>x</code> ↑ <code>o</code>	- șterge linie albă
↑ <code>x u</code>	- anulează ștergerea (extended undelete)

2.2.5 Comenzi de formatare-indentare

↑ <code>o</code>	- inserare linie albă
↓ <code>s</code>	- centrare linie
↓ <code>i</code>	- indent
↑ ↓	- indent la același nivel
↑ ↓ <code>o</code>	- rupe linie și indentează
↑ <code>^</code>	- unește linii indentate
↑ <code>x f</code>	- setează marginea din dreapta pe poziția cursorului
↑ <code>u N</code> ↑ <code>x f</code>	- setează marginea din dreapta la coloana <code>N</code>

2.2.6 Comenzi pentru blocuri

↑ <Spc> - setează marcarea bloc
 ↑ x↑ x - arată bloc
 ↑ w - șterge bloc ("cut")
 ↓ w - memorează bloc (pentru mutare)
 ↑ y - inserează copie bloc ("yank"-paste)
 ↑ x↑ u - înlocuiește litere din bloc cu majuscule
 ↑ x↑ l - înlocuiește litere din bloc cu minuscule

2.2.7 Comenzi pentru fișiere (încărcare-salvare)

↑ x↑ s - salvează fișier (extended save)
 ↑ x↑ w - scrie fișier cu redenumire (extended write)
 ↑ x↑ f - încarcă fișier (extended find)

2.2.8 Căutare și înlocuire

↑ s text ↑ s - căutare înainte (search)
 ↑ r text ↑ r - căutare înapoi
 <Esc> - ieșire din căutare
 ↑ g - abandon căutare
 ↑ % - căutare și înlocuire cu următoarele subcomenzi:
 <Spc> - înlocuiește și continuă
 - nu înlocui și continuă
 , - înlocuiește și oprește
 ! - înlocuiește toate aparițiile neinteractive

2.2.9 Comenzi pentru ferestre

↑ x2 - împarte fereastra vertical
 ↑ x5 - împarte fereastra orizontal
 ↑ x0 - elimină fereastra curentă
 ↑ x1 - elimină celelalte ferestre
 ↑ xo - mută cursor în altă fereastra

2.2.10 Lucrul cu macrocomenzi

↑ x (...↑ x) - definiție macrocomandă
 ↑ x e - lansează ultima macrocomandă definită
 ↑ u ↑ x (...↑ x) - adaugă noi comenzi la definiția ultimei macrocomenzi

2.2.11 Comenzi pentru dezvoltarea programelor

<code>↑ ↓ a</code> , <code>↑ ↓ e</code>	- deplasează cursor la început/sfârșit funcție
<code>↑ ↓ f</code> , <code>↑ ↓ b</code> , <code>↑ ↓ p</code> , <code>↑ ↓ n</code>	- deplasează cursor în cadrul unei expresii cu paranteze
<code>↑ ↓ h</code>	- definește bloc peste funcție
<code>↑ j</code>	- sfârșit linie cu indentare
<code>↑ ↓</code>	- reindentează liniile unui bloc
<code>↑ q <Tab></code>	- inserează Tab
<code>↑ (</code>	- inserează pereche de paranteze
<code>↓ ;</code>	- adaugă comentariu
<code>↑ ↓ j</code>	- sfârșit comentariu
<code>↑ u - ↑ x</code>	- șterge comentariu din linia sau blocul curent
<code>↓ x compile</code>	- compilează program
<code>↑ x`</code>	- mergi la următoarea linie a erorii de compilare

2.2.12 Diverse

<code>↑ g</code>	- abandonează comandă
<code>↑ h</code>	- acces la help
<code>↑ ht</code>	- execută tutorial "on-line" (ieșire cu <code>↑ x k <↵ ></code>)
<code>↓ \$</code>	- spelling cuvânt
<code>↓ x</code>	- spelling fișier
<code>↓ !</code>	- execută comandă UNIX
<code>↑ u ↓ !</code>	- execută comandă UNIX cu inserarea răspunsului în fișier
<code>↓ x tex-mode</code>	- invocare mod <code>TEX</code>
<code>↓ x latex-mode</code>	- invocare mod <code>L^ATEX</code>
<code>↑ xd</code>	- invocare editor de directoare pentru managementul fișierelor (extended DirEd), după care sunt disponibile comenzile: <code>c</code> = copiere, <code>f</code> = încărcare, <code>dx</code> = ștergere fișier

2.3 Exerciții

EXERCITIUL 1

Generați un fișier cu numele `vi.txt` (dacă folosiți editorul `vi`), respectiv `emacs.txt` (dacă folosiți editorul `emacs`) care conține descrierea comenzilor editorului folosit pe care le considerați indispensabile (nu mai mult de 10).

EXERCITIUL 2

Generați un fișier cu numele `cv.form` care conține formularul unui Curriculum vitae (Nume, Data și locul nașterii, Adresă, Ocupație actuală, Studii, Realizări, Preocupări, ...)

EXERCITIUL 3

Folosind fișierul `cv.form` generați propriul Dvs. CV în fișierul `cv.txt`.

EXERCITIUL 4

În directorul `~/tema2/`, generați un fișier cu numele `referat.form` care conține formularul unui referat de laborator (conform modelului de la pagina 112).

Fișierul va avea conține următoarele informații:

Antet: Universitatea "Politehnica" din București – Laboratorul de metode numerice

Titlu: Cuvântul Titlu

Autor: Numele Dvs.

Adresa de posta electronica, grupa, facultate

Instructor: Numele instructorului

Următoarele rânduri se introduc ca atare:

Introducere

Continut

Concluzii

Lista bibliografica

Anexe

Notă: adresa de poștă electronică se obține concatenând numele Dvs. de utilizator (obținut cu comanda `whoami`) cu șirul de caractere `@lmm.pub.ro`.

EXERCITIUL 5

Scrieți un fișier `~/tema1/unix.txt` ce conține descrierea principalelor comenzi Unix. Inserați în acest text părți din descrierea `man` a comenzilor selectate, folosind funcția de

marcare și copiere între ferestre.

EXERCITIUL 6

Scrieți, în directorul `~/tema1/`, un script Unix (par. 1.5) numit `azi` care:

- afișează textul “Hello” urmat de numele Dvs. de utilizator;
- Afișează data și ora;
- Afișează spațiul de disc ocupat și cel maxim alocat Dvs.;
- Afișează numele mașinii și shell-ul folosit.

Executați acest script prin invocarea numelui său. (Dacă nu reușiți, revedeți și paragraful 1.6.4).

Capitolul 3

Poșta electronică

— elm —

Poșta electronică (*electronic mail* sau pe scurt *e-mail*) este unul din cele mai importante servicii oferite de Unix. Capacitatea de a trimite mesaje la indivizi sau grupuri de persoane prin intermediul rețelei de calculatoare aparține tradițional Unix-ului, dar ulterior a fost preluată și de alte sisteme de operare (de exemplu Windows).

Poșta electronică (E) este o alternativă la serviciile tradiționale de poștă (P) sau telefon (T), dar cu trăsături specifice: ea este asincronă (ca serviciul P), are o viteză de transmitere mai mică decât T, dar incomparabil mai mare decât P, este un intermediar între *scripta manent* (P) și *verba volant* (T), în schimb are două avantaje majore față de serviciul telefonic: prețul mic și capacitatea de a comunica simultan cu grupuri mari.

Principalele **funcții** oferite de serviciul e-mail sunt:

- schimbul de mesaje între cunoscuți;
- transmiterea de documente în format electronic;
- abonamente la periodice electronice;
- transmiterea informațiilor care se pot reprezenta digital (programe executabile, sunete, grafică, animație, secvențe video, etc.).

Aceste funcții și caracteristici determină un nou mod de colaborare profesională în care rolul hârtiei se diminuează (*paperless office*) [73].

Tendința de globalizare a comunicațiilor digitale a condus la conceptul de *cyberspațiu*, care reunește mulțimea tuturor informațiilor reprezentate în calculatoarele conectate în rețeaua mondială. Actorii cibernațului sunt utilizatorii acestor calculatoare, fiecare identificat printr-o adresă de e-mail. Persoanele fără o astfel de adresă sunt echivalentul modern al celor care în viziunea tradițională nu au adresă poștală (“homeless”) sau nu au telefon. O adresă de e-mail are următoarea structură tipică [50]:

`Nume@Sistem.Subdomeniu.Domeniu`

în care:

- `Nume` este numele utilizatorului;
- `Sistem` este numele calculatorului sau rețelei locale de calculatoare;
- `Subdomeniu` este numele rețelei de calculatoare de nivel superior;
- `Domeniu` este domeniul național căruia îi aparține calculatorul, conform codificării:

`at` = Austria

`au` = Australia

`ch` = Elveția

`de` = Germania

`es` = Spania

`fr` = Franța

`gr` = Grecia

`jp` = Japonia

`ro` = România

`uk` = Marea Britanie

În cazul Statelor Unite, domeniile sunt:

`com` = comercial

`edu` = educațional

`gov` = guvernamental

`mil` = militar

`org` = organizații nonprofit

`net` = organizații ce gestionează rețele de calculatoare

`int` = organizații internaționale

Cât timp aveți un cont deschis în LMN, adresa Dvs. de e-mail este

`Nume@lmn.pub.ro`

`pub` fiind codul subdomeniului Universitatea “Politehnica” din București. Adresele e-mail nu sunt “case-sensitive”, dar pentru siguranță preferați literele mici.

Pentru a beneficia de serviciile de poștă electronică, trebuie să aveți propria adresă e-mail și să cunoașteți utilizarea unui program de e-mail. Tradiționalul program Unix de e-mail este `mail` [50], dar în versiunile recente s-au dezvoltat programe noi: `mailx` (SVR4) și `Mail` (BSD), virtual echivalente.

În prezent sunt populare programele `elm` [37] (mai ușor de învățat), `Pine` [18] (cu funcții puternice de transmitere a fișierelor binare), sau componentele e-mail din `NetScape` [15] sau `Emacs` [75].

Conceptele specifice poștei electronice

Un mesaj electronic este alcătuit din textul propriu-zis al mesajului precedat de un antet cu structura:

Date: data transmiterii

From: adresă expeditor

To: adresă destinatar

Subject: subiectul mesajului (o linie)

Cc: lista destinatarilor cărora li se transmit copii ale mesajului (“carbon copy”).

Fiecare mesaj are un identificator unic, iar dacă destinatarul nu a fost găsit, mesajul se întoarce la expeditor.

Mulțimea mesajelor primite se colectează într-o *căsuță poștală* a utilizatorului. După citire ele pot fi lăsate în căsuță (nerecomandat!), șterse, salvate într-un fișier, avansate altui destinatar (*forward*) sau mutate în altă căsuță poștală a aceluiași utilizator (clasificate).

În corespondența electronică se recomandă respectarea următoarelor **reguli de stil** [69]:

- folosiți mesaje scurte, cu paragrafe de max. 15 linii;
- lungimea liniei de max. 65 caractere;
- separați paragrafele cu linii albe;
- începeți mesajul cu numele real al destinatarului și încheiați cu numele Dvs. real;
- respectați regulile gramaticale în folosirea majusculelor;
- nu “bombardați” cu mesaje persoane cunoscute, dar mai ales necunoscute;

- folosiți un stil de scriere decent și politicos;
- nu utilizați caractere de control;
- când dați replici la un mesaj, includeți și textul receptat precedat de caracterul >, după modelul:

> Ne putem vedea astăzi?

Da, la ora 14:30.

- în încheierea mesajului puteți adăuga o “semnătură electronică”, după modelul

Nume, Poziție

Adresă,

Tel/Fax

Unii expeditori mai adaugă și o deviză personală sau un motto. Vă recomandăm să păstrați concizia și eleganța acestei semnături.

- în mesajele electronice se folosesc și elemente non-verbale (“smileys”):

: -) zâmbet (după glume);

: -(supărare;

: -o mirare;

; -) aluzie;

: * sărutări!

Vă puteți imagina că aceste semne reprezintă o față umană rotită cu 90°.

Obiectivele capitolului constau în:

- înțelegerea conceptelor specifice poștei electronice;
- căpătarea deprinderilor esențiale necesare utilizării poștei electronice: citirea mesajelor, transmiterea mesajelor, gestionarea listelor destinatarilor (aliasuri), gestionarea mesajelor primite (ștergerea, mutarea, salvarea).

În continuare se prezintă pe scurt programul **elm**, dezvoltat de Dave Taylor, care are avantajul principal că este distribuit gratuit [1].

3.1 Apelare

```
elm <↵ >
```

Programul de poștă electronică **elm** oferă facilități pentru:

- citire a mesajelor primite
- trimitere de mesaje
- menținerea listei de “alias”-uri personale
- ștergerea mesajelor primite sau mutarea lor în alte căsuțe poștale
- modificarea opțiunilor programului **elm**

Pentru documentare (“help”) se dă comanda: ? urmat de:

<nume comandă> – afișează informații despre această comandă
? – afișează lista tuturor comenzilor disponibile

3.2 Structura ecranului elm

Ecranul tipic al programului **elm** este prezentat în figura 3.1. El conține următoarele zone:

- Linia de informații – în care sunt afișate numele de fișier al căsuței poștale principale și numărul de mesaje pe care aceasta le conține;
- Lista de mesaje primite;
- Zona de help;
- Linia de comenzi.

3.3 Citirea mesajelor primite

Se selectează mesajul dorit cu una din comenzile următoare (implicit este selectat ultimul mesaj primit):

k sau ↑ - selectarea precedentului mesaj neșters
j sau ↓ - selectarea următorului mesaj neșters
K - selectarea mesajului anterior
J - selectarea mesajului următor
n - selectarea mesajului n

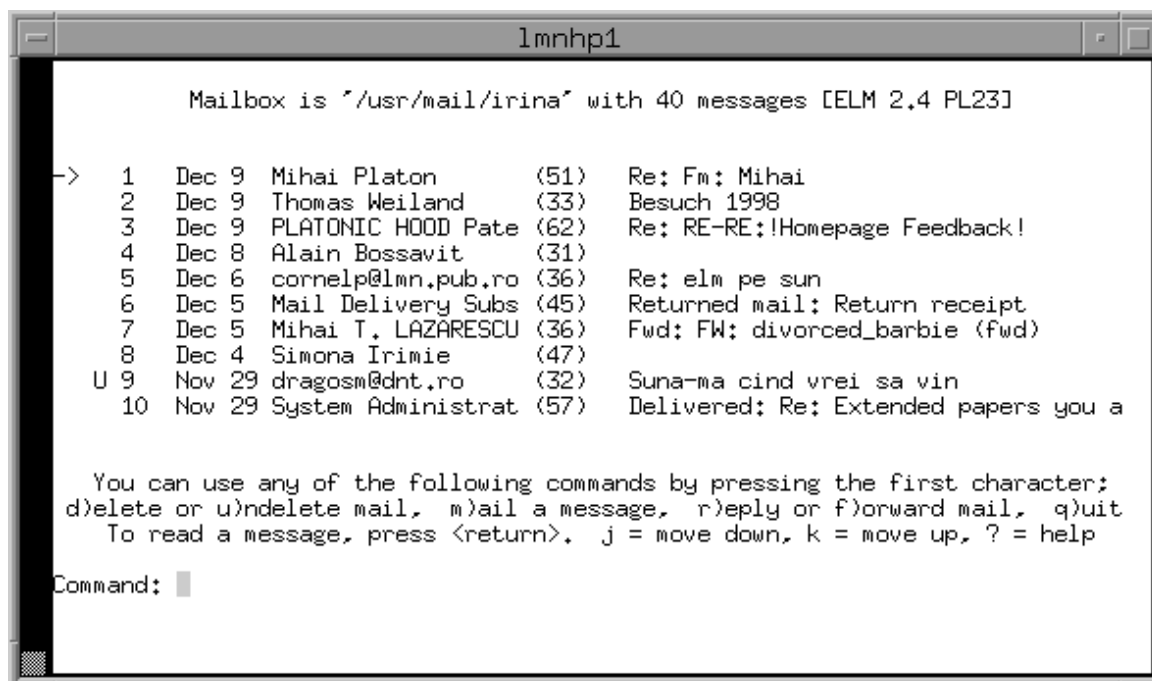


Fig. 3.1: Ecranul programului elm

- + - avans un ecran
- - înapoi un ecran

Pentru citirea mesajului selectat:

- <↵ > - citirea mesajului curent fără a afișa antetul (header-ul) mesajului
- h - citirea mesajului curent, cu afișarea antetului (header)

În cursul citirii unui mesaj:

- <blanc> - avansează un ecran
- <↵ > - avansează o linie
- ↑ - mesaj precedent
- ↓ - mesaj următor

3.4 Trimitere de mesaje

Se introduce una din comenzile:

- m - trimitere mesaj (mail)
- r - răspuns expeditorului mesajului curent (reply)
- g - răspuns expeditorului și tuturor recipientilor mesajului curent (global)

f - avansează mesajul curent unui alt destinatar (forward)

b - avansează mesajul curent unui alt destinatar, ca și cum ar fi fost trimis de expeditorul inițial (bounce)

Programul solicită:

- adresantul mesajului – listă de adrese sau aliasuri separate prin blancuri (fac excepție comenzile **f** și **g**)
- subiect
- cc (carbon copy) - listă de adrese sau aliasuri, separate prin blancuri

Se intră apoi într-un editor de text, în mod implicit **vi**. După comanda de scriere și salvare a mesajului, utilizatorul poate:

- s** - să trimită mesajul (send);
- e** - să reintre în editarea mesajului (edit);
- h** - să editeze header-ul mesajului (headers);
- f** - să renunțe la trimiterea mesajului (forget-it).

Pentru crearea unor headere personalizate, se folosește fișierul

```
$HOME/.elm/elmheaders
```

Printre altele, în el se poate introduce linia:

```
Return-Receipt-To: user@lmn.pub.ro
```

utilă pentru a avea confirmarea de primire a mesajelor trimise.

3.5 Pseudonime (aliasuri)

Se dă comanda:

a

În modul “alias” sunt disponibile comenzile:

n - creare alias nou (new)

a - creare alias din adresa expeditorului mesajului curent

Se solicită:

- alias (nume scurt)
- nume de familie (last name)
- prenume (first name)
- comentariu opțional
- adresă de e-mail (sau listă de adrese, pentru a crea un grup)

- ? - help pentru comenzile disponibile în “alias”
- d - ștergerea unui alias (delete)
- c - modificare alias curent (change)
- <↵ > - verificarea adresei de e-mail pentru alias-ul curent

3.6 Ștergerea / mutarea mesajelor în altă casuță poștală

- d - șterge mesaj curent (delete)
- s - salvează mesaj curent în altă casuță poștală și îl marchează ca șters
- C - salvează mesaj curent în altă casuță poștală fără să îl marcheze ca șters
- c - schimbarea casuței poștale (change)
 - fișier fis în directorul \$HOME: ~/fis
 - fișier fis în directorul Mail: =/fis
 - fișierul de mesaje primite: <
 - fișierul de mesaje trimise: >
 - casuță poștală “principală”: !

Sunt utile comenzile:

- t - marcare mesaj curent pentru operații ulterioare: salvare, ștergere, etc. (tag)
- Ctrl-t - marcarea tuturor mesajelor care îndeplinesc o condiție, în vederea unor operații ulterioare

3.7 Modificarea opțiunilor

- fiind în mod comandă se introduce:

- o

Sunt disponibile operațiile:

- Schimbarea nivelului de utilizator:

- u - 0 - începător; 1 - familiar; 2 - expert

- Afișare mail:

- d - implicit builtin sau usr/bin/more

- Editor:

- e - implicit /usr/bin/vi

- Folder (unde se țin mesajele primite):
 - `f` – implicit `$HOME/mbox`
- Criteriul de sortare a mesajelor:
 - `s` - implicit `received` (în ordinea primirii, cel mai recent primul)
Se poate folosi comanda
 - `R` – sortare în ordine inversă, implicit `reverse received` (în ordinea primirii, cel mai recent ultimul)
- Căsuța unde se salvează copiile mesajelor trimise:
 - `o` – implicit `$HOME/mbox`
Se recomandă folosirea unei căsute poștale dedicate, alta decât `mbox`, de exemplu `$HOME/Mail/outbox`. Pentru ca opțiunea să devină activă trebuie definit `copy = ON` în fișierul `elmr.c`.
- Programul de imprimare:
 - `p` – implicit `lp` sau `lpr`
Cum imprimanta din laboratorul EB212 nu funcționează în mod text, modificați această opțiune astfel: `/bin/cat %s | a2ps -c | lpr`
- Numele utilizatorului:
 - `y`
- Forma cursorului:
 - `a` – OFF (bară video-invers); ON (săgeată)
- Afișare numai numele persoanei în alias (Names Only):
 - `n` – OFF (afișează și adresa de Internet); ON (afișează numai numele)
- în fișierul `~/ .elm/elmr.c`:
 - Fișierul conține variabile grupate pe următoarele categorii:
 - formatarea afișărilor
 - dispozitive (“devices”) utilizate
 - directoare și fișiere de mesaje
 - opțiuni/răspunsuri implicite

Toate opțiunile sunt date prin setarea unor variabile. Instrucțiunile sunt de forma:
<variabilă> = <valoare>

Exemplu:

```
copy = ON
askcc = ON
editor = /usr/bin/vi
names = ON
```


3.8 Ieșirea din elm

x - ieșire fără modificarea căsuței poștale

q - ieșire cu modificarea căsuței poștale

3.9 Exerciții

EXERCITIUL 1

Folosind editorul de texte preferat, generați un fișier numit `logo` care să conțină numele Dvs., poziția (student în grupa ... la facultatea ...), adresa poștală, telefon, adresa electronică. Acest fișier poate să conțină și o deviză personală (motto) dar trebuie să fie formatat cât mai elegant (încadrat între două linii de marcaj alcătuite din semne speciale: *, = sau -).

Mutați fișierul `logo` în directorul `~/elm`

EXERCITIUL 2

Lansați `elm`, intrați în modificare opțiuni și setați ordinea mesajelor astfel încât cel mai recent primit mesaj să apară primul în listă. Salvați opțiunile.

EXERCITIUL 3

Editați fișierul `~/elm/elmrc`, pentru a seta salvarea mesajelor trimise și fișierul de “semnătură” (`localsignature`) ca fiind `~/elm/logo` (conform paragrafului 3.7).

EXERCITIUL 4

Scopul acestui exercițiu este de a vă permite să creați, evitând greșelile de tastare, o listă de adrese e-mail ale colegilor de grupă.

Trimiteți un scurt mesaj electronic colegului de grupă care vă urmează în lista alfabetică a numelor (ultimul de pe listă va trimite un mesaj primului).

Avansați (`forward`) copia tuturor mesajelor primite, colegului următor.

Salvați toate mesajele primite într-o nouă căsuță poștală, numită `grupa`, plasată în directorul `~Mail` (paragraful 3.6).

Constituiți un alias numit `grupa` care să conțină adresele colegilor de grupă. Folosiți facilitatea “cut and paste” (cu ajutorul mouse-ului) pentru a nu risca să introduceți eronat adresa.

Ștergeți mesajele din căsuța poștală grupă.

EXERCITIUL 5

Expediați o copie a fișierului `cv.txt` realizat în tema precedentă (propriul Dvs. CV) către toți colegii de grupă.

Creați un nou fișier care să conțină toate CV-urile primite. Editați acest fișier astfel încât să conțină doar lista numelor studenților și data nașterii.

Trimiteți acest mesaj colegului următor, și în copie (cc) instructorului. (*À propos: cum faceți să aflați adresa de poștă electronică a instructorului Dvs.?*)

Trimiteți un mesaj de felicitare colegilor născuți în luna curentă.

EXERCITIUL 6

Trimiteți un mesaj instructorului, în care explicați funcționarea comenzii Unix de imprimare a unui mesaj electronic, propusă în paragraful 3.7.

Ștergeți toate mesajele primite din căsuța poștală principală. Mutați-vă în căsuța poștală în care sunt salvate automat mesajele trimise de Dvs. și ștergeți primul scurt mesaj trimis (exercițiul 4).

Capitolul 4

Editarea documentelor

— L^AT_EX —

Rezultatul principal al activităților de cercetare îl constituie realizarea unor *documente tipărite*: cărți, articole în reviste de specialitate sau rapoarte științifice. Spre beneficiul întregii omeniri și, implicit, al oamenilor de știință, de cinci sute de ani, de la apariția tiparului, cunoștințele tipografice s-au perfecționat continuu, devenind o adevărată artă.

Spre deosebire de textele pure, *documentele* tipărite sunt caracterizate și prin fonturile folosite (tipurile și corpurile de literă) precum și de așezarea în pagină (inclusiv plasarea imaginilor și eventual a formulelor matematice și chimice).

Apariția calculatoarelor a avut impact major și asupra acestei activități. Realizarea cu calculatorul a unor documente tehnice și științifice de calitate, în vederea multiplicării lor se face în prezent în două moduri principial diferite:

- **într-un pas**, cu editoarele de documente (*word processing*) gen MS Word, la care efectul editării este vizibil imediat pe ecran (WYSIWYG - “What You See Is What You Get”) și
- **în doi pași**, folosind mai întâi un editor de texte pure (**vi**, **Emacs**, etc.) în care pe lângă textul documentului se adaugă specificațiile de formatare (folosind convențiile unui limbaj de tip “mark-up” care permite adăugarea indicațiilor pentru “tipograf”) și apoi un program neinteractiv de formatare, care interpretează fișierul text de descriere a documentului și produce un fișier de ieșire (codificat corespunzător, de obicei în format binar) reprezentabil pe ecran sau la imprimantă.

Aparent, prima metodă este ideală, dar se constată că în majoritatea cazurilor rezultatele obținute cu ea nu satisfac standardele profesionale. Se poate spune că primul mod este echivalentul modern al utilizării mașinii clasice de scris, pe când al doilea corespunde mașinii tipografice de cules.

În anul 1977, Donald Knuth, profesor la Universitatea Stanford și unul din cei mai mari specialiști în calculatoare din toate timpurile, a început lucrul la un proiect intitulat

T_EX (se pronunță *tehi*), care urmărea realizarea unui sistem de procesare a textelor care să permită realizarea unor “cărți frumoase”, conținând inclusiv formule matematice complexe [53]. Evident, el a ales al doilea mod de abordare (în doi pași) și în plus a definit un standard (*DVI = DeVice Independent*) pentru formatul fișierului de ieșire. Sistemul T_EX de formatare a documentelor, realizat de Knuth satisface cele mai exigente standarde privind:

- **calitatea** (absolut remarcabilă a) rezultatului;
- **portabilitatea** (poate rula pe diferite platforme UNIX, VMS, Windows, iar fișierele DVI pot fi reprezentate pe cele mai diverse dispozitive: ecrane, imprimante matriciale, imprimante cu laser, etc.);
- **disponibilitatea** (în ciuda efortului imens pentru realizarea lui, pachetul de software T_EX este disponibil fără plata licenței și distribuit de o organizație nonprofit numită TUG – TeX Users Group [24]);
- **stabilitatea** (Donald Knuth a anunțat “înghețarea” efortului de dezvoltare pentru a oferi stabilitate produsului).

Toate aceste calități au determinat succesul enorm al proiectului și l-a făcut pe Gordon Bell să afirme că “*T_EX este cea mai importantă invenție a secolului în domeniul tipografiei și poate fi pusă pe același nivel cu invenția tiparului de către Gutenberg*”.

Complexitatea și mai ales flexibilitatea sistemului T_EX îl fac totuși relativ greu de folosit. Din acest motiv Leslie Lamport a început în 1985 dezvoltarea unui sistem numit L^AT_EX (L^AYout TeX, pronunțat *leitehi*), bazat pe T_EX, dar care să permită utilizatorului (de obicei autorul documentului) să se poată concentra asupra structurii și organizării logice a documentului și nu pe detaliile tipografice.

L^AT_EX este un standard “de facto” pentru editarea documentelor științifice și tehnice de înaltă calitate (articole pentru publicații științifice, rapoarte tehnice, cărți, prezentări, dar și scrisori, broșuri sau lucrări umaniste) [63].

Spre deosebire de T_EX, care se poate considera ca un nivel de bază (*low-level*), L^AT_EX este un sistem de nivel înalt (*high-level*) și permite:

- structurarea documentelor mari;
- includerea de tabele, figuri, formule matematice complexe;
- cuprins, liste bibliografice, index generate automat.

Se poate spune că T_EX este “zețarul” iar L^AT_EX este “tehnoredactorul” [71].

Sistemul L^AT_EX realizează formatarea textelor conform celor mai exigente standarde fără să fie necesare prea multe eforturi din partea utilizatorului. Este adevărat că aspectul

general al documentului realizat cu \LaTeX se modifică greu însă rezultatul este profesional. Toate marile reviste de specialitate au definit stiluri \LaTeX obligatorii pentru autorii articolelor [25].

Proiectul \LaTeX a respectat toate caracteristicile \TeX , rezultatul fiind disponibil gratis prin rețeaua de calculatoare [26] sau la preț nominal, pe CD ROM; în plus el este relativ ușor de învățat și utilizat.

Spre deosebire de \TeX , dezvoltarea \LaTeX nu este înghețată: ea a fost preluată de proiectul **LaTeX3**, urmărindu-se dezvoltarea comenzilor existente (stil, tratare erori, interfață) dar și implementarea conceptului de *hipertext*. Aceasta va face posibilă utilizarea în viitor a sistemului și pentru generarea documentelor electronice “on-line” (cele care satisfac standardele SGML-HTML și PDF).

Obiectivele capitolului de față constau în:

- familiarizarea cu conceptele tehnoredactării profesionale (stil, elemente de structură, obiecte – inclusiv cele matematice, macrodefiniții, structura paginii, fonturi);
- însușirea deprinderilor necesare editării rapoartelor profesionale (inclusiv a rapoartelor de laborator LMN) și a articolelor în reviste internaționale (IEEE).

4.1 Etapele utilizării L^AT_EX

Actualmente, L^AT_EX este instalat în LMN pe mașinile: `lmnsun`, `lmnhp5` și pe toate mașinile Linux. Există două versiuni instalate: versiunea 2.09 (`lmnsun`) și respectiv 2ϵ (pe mașina `lmnhp5` și pe mașinile Linux).

Un model de fișier L^AT_EX cuprinzând definiții utile este fișierul `latex_head.tex`, al cărui conținut este listat în paragraful 4.10.1.

Etape:

1. Se scrie fișierul cuprinzând documentul, cu extensia `.tex`. Să presupunem că numele fișierului este `fis.tex`
2. Se dă comanda de procesare cu L^AT_EX a fișierului:

```
latex fis
```

Fișierul creat în urma acestei comenzi este `fis.dvi`.

Uneori, pentru a obține rezultatul dorit, comanda `latex fis` trebuie dată de mai multe ori (de exemplu, crearea unui cuprins necesită două treceri prin fișier, deci va fi nevoie să se mai execute o dată comanda).

3. Înaintea imprimării, este recomandabil să se facă vizualizarea fișierului `.dvi`, folosind comanda:

```
xdvi fis.dvi
```

4. După corectarea tuturor posibilelor erori, imprimarea se face astfel:

- se transformă fișierul `fis.dvi` într-un fișier PostScript cu comanda:

```
dvips -t a4 fis.dvi -o fis.ps
```

care crează fișierul în format PostScript `fis.ps`;

- se imprimă documentul cu comanda:

```
lpr fis.ps sau lp fis.ps
```

5. Fișierul în format PostScript poate fi și el vizualizat, folosind una din comenzile: `ghostview fis.ps` (pe majoritatea sistemelor) sau `xpsview fis.ps` (pe mașina `lmnsgi`).

Programul `ghostview` permite și imprimarea fișierului PostScript și se recomandă folosirea acestuia, și nu a comenzii `lpr`, datorită facilităților oferite (printre care, vizualizarea și imprimarea fișierelor comprimate, imprimarea separată a paginilor pare/impare sau a unor pagini selectate de utilizator).

4.2 Structura unui fișier L^AT_EX

Un fișier L^AT_EX cuprinde:

- Comentarii: încep cu % și se termină la sfârșitul liniei
- Comenzi L^AT_EX: încep în general cu caracterul \
- Conținutul documentului: text, figuri

Înainte de a scrie orice document, autorul trebuie să își clarifice:

- ce tip de document este
 - articol
 - carte
 - raport de cercetare
 - scrisoare, etc.
- titlul, cuprinsul aproximativ
- textul propriu-zis
- așezarea în pagină – în teorie, ea este stabilită de editorul lucrării (caz în care, pentru a evita erorile și efortul inutil, se recomandă să se obțină de la acesta un fișier de stil).

L^AT_EX admite comenzi care permit formatarea automată a fiecărei părți a documentului.

În cele ce urmează sunt prezentate principalele părți ale unui document și comenzile L^AT_EX care le “gestionează”. Parametrii unei comenzi sunt incluși între acolade, sau între paranteze drepte în cazul parametrilor opționali.

Singurele comenzi obligatorii sunt cele referitoare la *tipul de document*, *începutul* și *sfârșitul* documentului.

Tipul de document se definește cu comanda:

```
\documentstyle[opțiuni]{stil}
```

unde *opțiuni* reprezintă o listă de opțiuni separate prin virgule, care pot fi:

- | | |
|---|-----------------------------|
| • dimensiune font: | 10pt (implicit), 11pt, 12pt |
| • aliniere ecuații (implicit centrate): | fleqn (stânga) |
| • aliniere număr ecuații (implicit dreapta): | leqno (stânga) |
| • includerea instr. de test și ciclare: | ifthen |
| • obținerea automată a indexului: | makeidx |
| • coloane(implicit una): | twocolumn |
| • pagini pare/impare diferite: | twoside |
| • pagină separată de titlu în stilul <i>article</i> : | titlepage |

iar stilurile disponibile sunt:

- `article`
- `book`
- `report`
- `letter`

Părți document	Comenzi \LaTeX
Definire tip document	<code>\documentstyle</code>
Definire: titlu	<code>\title</code>
autori	<code>\author</code>
note de subsol	<code>\thanks,</code> <code>\footnote</code>
data scrierii doc.	<code>\date</code>
Începutul textului documentului	<code>\begin {document}</code>
Afișarea titlului	<code>\maketitle</code>
Rezumatul lucrării	<code>\abstract</code>
Partea I	<code>\part</code>
Capitolul 1	<code>\chapter</code>
Secțiunea 1.1	<code>\section</code>
Subsect. 1.1.1	<code>\subsection</code>
Subsubsecțiune	<code>\subsubsection</code>
.....	
Anexe	<code>\appendix</code>
Cuprins	<code>\tableofcontents</code>
Lista de figuri	<code>\listoffigures</code>
Lista de tablele	<code>\listoftables</code>
Sfârșit text document	<code>\end {document}</code>

Definirea paginii de titlu

- Titlu: `\title{text}`
`text` cuprinde `\\` acolo unde se dorește trecerea la rând nou
- Autori: `\author{text}`
- Data: `\date{text}`
 Ex: `\date{\today}`
- Trimiteri la note de subsol referitoare la titlu:
 - trimitere la o nouă notă de subsol: `\thanks{text}`
 se recomandă să fie introdusă în text imediat după cuvântul la care se referă
 - trimitere la o notă de subsol existentă: `\footnotemark[n]`,
 cu `n` = numărul notei existente
- Punerea în pagină a titlului: `\maketitle`
 e bine să apară IMEDIAT după `\begin{document}`
- Rezumat:
`\abstract text abstract \endabstract` sau

```
\begin{abstract} text_abstract \end{abstract}
```

- Secțiuni ale textului

- numerotate automat:

```
\part \chapter \section \subsection
\subsubsection
```

toate cu un parametru — textul titlului părții respective

Ex: `\part {Partea I - Definiții generale}`

- nenumerotate: `\part* \chapter*`

Obs. Numerotarea secțiunilor textului diferă în funcție de stilul ales.

- Anexe: `\appendix`

are ca efect schimbarea modului de numerotare automată a capitolelor

- Cuprins, lista de figuri, lista de tabele:

```
\tableofcontents
```

```
\listoffigures
```

```
\listoftables
```

- Referirea la un capitol:

etichetă: `\label{text_etichetă}`

referire la o etichetă:

```
\ref{text_etichetă}
```

```
\pageref{text_etichetă}
```

(pagina pe care se găsește plasat obiectul referit prin etichetă)

La crearea titlului, părților, listelor, cuprinsului, etc., \LaTeX folosește texte predefinite în limba engleză. Acestea sunt:

Comanda \LaTeX	Semnificație	Valoare implicită	Stilurile în care este disponibilă
<code>\partname</code>	nume parte	Part	book, report
<code>\chaptername</code>	nume capitol	Chapter	book, report
<code>\refname</code>	nume bibliografie	References	article
<code>\bibname</code>	nume bibliografie	Bibliography	book, report
<code>\appendixname</code>	nume anexe	Appendix	toate
<code>\abstractname</code>	nume rezumat	Abstract	toate
<code>\indexname</code>	nume index	Index	toate
<code>\contentsname</code>	nume cuprins	Contents	toate
<code>\figurename</code>	nume figură	Figure	toate
<code>\listfigurename</code>	nume listă figuri	List of Figures	toate
<code>\tablename</code>	nume tabel	Table	toate
<code>\listtablename</code>	nume listă tabele	List of Tables	toate

Schimbarea lor se poate face cu comanda:

```
\renewcommand <par1> <par2>
```

unde:

```
par1 = numele comenzii
par2 = noua definiție a comenzii
```

Ex: `\renewcommand {\chaptername}{Capitolul}`

4.3 Comenzi L^AT_EX

În general sunt reprezentate de: caracterul backslash (\) urmat de un cuvânt (o singură cifră, o literă sau mai multe). Există două tipuri de comenzi:

- Cu efect localizat - execută o acțiune, apoi se încheie, cum ar fi de exemplu comenzile de tipărire a unor secvențe de caractere;

Exemple:

```
\% - tipărește %, încheie comanda
\today - tipărește data de azi, încheie comanda
\LaTeX - tipărește LATEX, iese
\par - trece la paragraf nou, iese
```

- Cu efect global - efectul lor se încheie doar când se dă o altă comandă cu efect “contrar”;

Ex: comenzile de schimbare a fontului:

```
\tt - schimbă fontul în “typewriter”
\bf - schimbă fontul în “bold”
```

Există și comenzi care așteaptă o listă de parametri. Parametrii pot fi:

- un singur caracter care urmează imediat după numele comenzii, ca în exemplele:

```
\a - comanda “tipărește accent” (ea așteaptă un parametru)
\aa - tipărește accentul notat =
```

- un șir de caractere între acolade

Ex: `\hspace {1cm}` - lasă un spațiu orizontal de 1 cm

Pentru a localiza efectul unei comenzi cu efect global, se folosește gruparea - includerea între acolade a porțiunii pentru care se dorește modificarea prin comandă.

Ex: `Font normal {\bf font bold} font normal.`

Atenție! Tot acolade se folosesc, după cum s-a văzut mai sus, și pentru marcarea parametrilor! Parametrii, deși sunt puși între acolade, nu reprezintă grupuri, deci, dacă în interiorul unui parametru apare o comandă cu efect global, efectul ei va continua și după acolada închisă.

4.4 Moduri de lucru L^AT_EX

În orice punct al documentului L^AT_EX, ne putem afla în unul din trei moduri:

- modul paragraf - acesta este cel “implicit”, în el se poate scrie textul;
- modul matematic - permite scrierea de formule matematice;
- modul LR - analog cu paragraf, dar în acest mod L^AT_EX nu verifică încadrarea în spațiul disponibil, cum face în mod paragraf.

În fiecare mod de lucru se pot da comenzi specifice.

4.4.1 Modul paragraf

Este modul de lucru implicit.

Comenzi - grupate după efectul lor:

- setare font:

`\rm` - Roman

`\it` - *Italic*

`\bf` - **Bold**

`\sf` - Sans Serif

`\sc` - SMALL CAPITALS

`\sl` - *Caractere Incline (slanted)*

`\tt` - Typewriter

`\em` - trecerea de la roman la *italic* și invers

- setare dimensiune font:

`\tiny`, `\scriptsize`, `\footnotesize`, `\small`,

`\normalsize`, `\large`, `\Large`, `\LARGE`, `\huge`, `\Huge`.

cu efectul:

`\Font`, `Font`, `Font`, `Font`, `Font`, `Font`, `Font`, `Font`, `Font`, `Font`.

Dimensiunea propriu-zisă a acestor fonturi depinde de opțiunea (`10pt`, `11pt`, `12pt`) din `\documentstyle`.

- printare caractere speciale (vezi paragraful 4.7.1 și manualul de utilizare pentru lista completă)
- printare linii și spații de diverse dimensiuni:
 - - linie de despărțire în silabe
 - – linie din matematică
 - — linie (punctuație)

`\,` - spațiu mic, utilizat de exemplu între ghilimelele simple și cele duble
`\l` - spațiu (caracterul backslash urmat de un spațiu)
`\/` - spațiu pentru a nu permite ligatura unor cuvinte
`\@` - spațiu de sfârșit de propoziție, dacă următoarea propoziție începe cu literă mică
`\frenchspacing` - suprimă spațiul suplimentar după punctuație
`\nofrenchspacing` - elimină efectul comenzii `\frenchspacing`
`\noindent` - suprimă alineatul de început de paragraf
`\par` - linie liberă
`\hspace` - spațiu orizontal, uneori nu are efect la început și sfârșit de paragraf
`\hspace*` - ca mai sus, apare oriunde
`\vspace` - spațiu vertical
`\vspace*` - forma robustă a celei de mai sus
`\hfill` - spațiu orizontal de lungime variabilă (cât să completeze linia);
`\vfill` - spațiu vertical de dimensiune variabilă (cât să completeze pagina);
`\dotfill` - ca și `\hfill` dar produce puncte
`\hrulefill` - ca și `\hfill` dar produce o linie orizontală
`\smallskip`, `\medskip`, `\bigskip` - spații verticale (depind de stilul documentului)
`\addvspace` - adaugă un spațiu vertical

- printare note marginale:

`\marginpar`

- printare dreptunghiuri:

`\rule{dimx}{dimy}` sau

`\rule[dy]{dimx}{dimy}` (supraînălțat față de baza textului cu `dy`)

- setare opțiuni pentru rânduri:

`\\` - rând nou, restul liniei rămâne gol

`\newline` - echivalent cu `\\`

`\linebreak` - linie nouă, caracterele rămase se spațiază ca să ocupe întreaga lățime

`~` - interzice ruperea liniei în acel loc

`\nolinebreak` - interzice ruperea liniei în acel loc

- setare opțiuni pentru pagini:

`\flushbottom` - adăugare de spațiu pe verticală a.î. toate paginile să aibă aceeași lungime

`\raggedbottom` - efect invers
`\newpage` - trecerea la pagină nouă
`\pagebreak` - trecerea la pagină nouă
`\nopagebreak` - interzice trecerea la pagină nouă
`\samepage` - păstrarea unui set de linii pe aceeași pagină
`\clearpage` - trecerea la pagină nouă, cu punerea tabelor și figurilor rămase pe o pagină fără text

4.4.2 Modul matematic

Intrare în mod matematic:

- formulă în text:

`$ formulă $`

sau

`\begin{math} formulă \end{math}`

- formulă nenumerotată, în paragraf separat:

`$$ formulă $$`

sau

`\begin{displaymath} formulă \end{displaymath}`

sau

`\[formulă \]`

- formulă numerotată în paragraf separat – vezi paragraful 4.5 → ecuații.

Comenzi disponibile:

- indici: `_` `^` - indice inferior, superior
- fracție: `\frac{numărător}{numitor}`
- radical: `\sqrt{x}`, `\sqrt[n]{x}`
- funcții matematice: `\sin`, `\cos`, `\arccos`, `\lim`, `\min`, etc.
- modulo: `\bmod`, `\pmod`
- semne speciale:

`\sum` - sigma

`\int` - integrală

`\lim` - limită

`\cdots`, `\vdots`, `\ddots` - puncte centrate, verticale sau diagonale

`\ldots` - trei puncte la baza liniei

- litere grecești: `\alpha`, `\beta`, .. α, β, \dots

- setare font :

`\mathcal` - *CALIGRAFIC*, disponibil doar pentru literele mari

`\mathit` - *mathitalic* (implicit)

`\mathbf` - bold **bold**

`\mathrm` - Roman, drept

- delimitatori:

`\left(... \right)`

`\left{ ... \right}` sau

`\left{ ... \right.` pentru a nu mai afișa și paranteza din dreapta

- linii, acolade:

`\overline{text}` - linie deasupra: \overline{text}

`\underline` - subliniere: \underline{text}

`\overbrace` - acoladă deasupra: \overbrace{text}

`\underbrace` - acoladă dedesubt: \underbrace{text}

`\vec` - simbol de vector (sageată orizontală) deasupra: \vec{v}

`\imath` - i fără punct: i

`\jmath` - j fără punct: j

`\hat`, `\widehat` - căciuliță deasupra: \hat{v}, \widehat{v}

`\tilde`, `\widetilde` - tildă deasupra: \tilde{v}, \widetilde{v}

(vezi pag. 78-79 din [71]).

- spațiere:

`\;` - spațiu mare

`\!` - spațiu negativ îngust

`\quad` - spațiu egal cu de 6 ori `\`, (vezi paragraful 4.4.1)

`\qquad` - spațiu egal cu de 10 ori `\`, (vezi paragraful 4.4.1)

4.4.3 Modul LR

În acest mod, orice “obiect” \LaTeX (de exemplu: o literă, un fragment de text, o ecuație, etc.) sunt tratate ca niște “cutii” despre care \LaTeX cunoaște doar dimensiunile. Avantaj: comenzile care culeg text în acest mod pot apărea la rândul lor în orice mod (paragraf, matematic sau LR).

În acest mod se intră prin comenzi caracteristice:

- Culegere text într-o “cutie”:

```
\mbox
\makebox
\fbbox
\framebox
\raisebox
```

- Parametri ai stilului:

```
\fboxrule
\fboxsep
```

- Salvare cutie / citire “cutie” salvată:

```
\newsavebox
\sbox
\usebox
```

4.5 Cadru L^AT_EX

Delimitează o entitate (ecuație, tabel, figură, fragment de text, etc.) și reprezintă o parte a documentului care este așezată în pagină într-un mod special, caracteristic entităților respective.

Formatul general al comenzii `environment` (în traducere *cadru* este:

```
\begin {nume_environment} text \begin {nume_environment}
```

Aceste structuri pot fi imbricate, având grijă ca ultimul deschis să fie închis primul.

Cadruri disponibile (puse în loc de `nume_environment`):

- document: `document`
- rezumat: `abstract`
- citate:

```
quote
quotation
verse
```

- alinieri:

```
flushright
center
```

- păstrarea nemodificată a textului: `verbatim`
- liste:

```
itemize
```


- enumerate
 - description
 - list
- tabele:
 - tabbing
 - tabular
 - table (env. mobil)
- ecuații:
 - displaymath
 - equation
 - array
 - eqnarray
- desene:
 - picture
 - figure (env.mobil)
- “pagini” miniatură: minitext
- bibliografie: thebibliography
- index: theindex

4.6 Macrodefiniții \LaTeX

Utilizatorul poate extinde și configura \LaTeX , prin definirea de noi comenzi și cadruri în cadrul documentului și prin modificare celor existente.

- Comenzi noi
 - fără parametri: `\newcommand{\comandă_nouă} {definiție}`
 - unde `comandă_nouă` este: o cifră SAU mai multe litere
 - cu parametri: `\newcommand {\comandă_nouă}[n]{definiție}`
 - unde `n` este numărul de parametri, mai mic decât 9. Parametrul numărul `k` este utilizat în definiție sub forma: `#k`
- Redefinire comenzi:
 - `\renewcommand{\comandă_veche}[n]{definiție_nouă}`
- Environments noi
 - `\newenvironment ...`
- Facilitate scriere teoreme:
 - `\newtheorem ...`

4.7 Aspecte speciale ale utilizării L^AT_EX

4.7.1 Caractere speciale

Următoarele caractere au semnificații speciale în cadrul comenzilor L^AT_EX și nu vor apărea ca atare în text:

`$ & # _ { } ~ ^ " \ |`

iar la scrierea cu anumite fonturi și `<>`.

Uneori, nici `<CR>` nu apar cum ne-am aștepta:

- mai mult decât un `<CR>` = un `<CR>`
- un singur `<CR>` = `<CR>`
- mai mult de `2<CR>` = un `<CR>`

Cum se pot obține aceste caractere: unele dintre ele, precedându-le de un backslash `\`

- diverse `\$ \& \% \# _ \{ \} \blanc`
- linie nouă: `2 <CR>`
- backslash:

`\backslash` (în mod matematic, vezi moduri L^AT_EX mai sus)

`\tt\char 39` (în mod paragraf)

4.7.2 Accente

Se folosesc pentru obținerea literelor românești (sau din alte limbi). Pentru limba română:

`\^A, \^a` : Â â

`\uA, \ua` : Ă ă

`\cS, \cS` : Ș ș

`\cT, \ct` : Ț ț

Caz special: `\^i` ar da un rezultat nedorit: `î` (apare și punctul și căciulița). Accentul ar trebui aplicat literei `i` fără punct, definit ca `\i`. În consecință, `\^\i` va avea efectul dorit: `î`.

În rețeaua LMN este instalat stilul L^AT_EX `rom.sty` (realizat de I. Munteanu prin adaptarea modelului propus în [71]), stil care permite introducerea diacriticelor românești într-un mod mult mai simplu.

Pentru a folosi acest stil, în lista de opțiuni a comenzii `\documentstyle` trebuie introduse și opțiunile

`[romanian,rom]`

(în această ordine). Folosind stilul `rom`, diacriticele românești pot fi dactilografiate astfel:

```

´a – ă
´A – Ă
´i – î
´I – Î
´s – ș
´S – Ș
´t – ț
´T – Ț
´w – â

```

Se observă că ' (apostroful) devine în acest stil o *comandă* și nu un caracter obișnuit.

În același fișier de stil sunt definite două comenzi, care permit definirea caracterului ´ ca fiind o comandă pentru introducerea diacriticelor, respectiv un caracter obișnuit:

```

\rom – caracterul ´ devine o comandă
\norom – caracterul ´ devine un caracter afișabil.

```

Se recomandă, de exemplu, introducerea comenzii `\norom` înainte de comanda de generare a bibliografiei (în fișierele de bibliografie apostroful trebuie să fie un caracter afișabil), și a comenzii `\rom` imediat după comanda de generare a bibliografiei (deoarece, în documentele românești, și anexele conțin diacritice).

4.7.3 Despărțirea în silabe

Regulile de despărțire în silabe depind de limba utilizată în text. Aceasta se poate redefini cu comanda:

```
\language<număr_de_ordine>
```

Ex: `\language0 =` limba engleză

Pentru a despărți în silabe un cuvânt (efect local) se folosește comanda `\-`

```
ex: des\ -partire
```

Acest lucru ar putea fi necesar dacă în urma procesării documentului cu \LaTeX apare eroarea:

```
Overfull xxxpt too wide
```

adică, există un cuvânt prea lung, care nu poate fi trecut integral pe linia următoare fără a spația mult prea mult rândul curent.

4.8 Structura unei pagini

Pentru definirea lățimilor, lungimilor și distanțelor, \LaTeX folosește parametri de stil. Ei au valori implicite, dar pot fi modificați de către utilizator. Pentru o listă completă puteți consulta [71], pag. 89.

Modificarea lor se face cu comanda:

```
\setlength{\par_de_stil}{dimens}
```

unde `par_de_stil` poate fi numele unei dimensiuni: lungimea textului pe pagină (`\textheight`), lățimea textului pe pagină (`\textwidth`), distanța între paragrafe (`\parskip`), margini (dreapta, stânga, sus, jos), dimensiune header sau footer, etc.

4.9 Fișiere rezultate în urma procesării cu \LaTeX

- .tex - fișier “sursă”
- .dvi - fișier de ieșire
- .aux - fișier cu informații auxiliare (folosit la a doua procesare pentru a rezolva referințe încrucișate)
- .toc - fișier cu informații necesare alcătuirii cuprinsului
- .log - fișier conținând raportul asupra procesării fișierului .tex, erori, folosirea resurselor de calcul
- .lof - fișier asemănător cu .toc, dar pentru lista figurilor
- .lot - fișier asemănător cu .toc, dar pentru lista tabelor
- .idx - fișier cu informații necesare alcătuirii indexului
- .glo - fișier cu informații necesare alcătuirii glosarului de termeni

4.10 Câteva exemple

4.10.1 Fișierul latex_head.tex

Mai jos este prezentată structura unui fișier header, care conține comenzi de punere în pagină și care poate fi adaptat și inclus în orice document \LaTeX .

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FISIER latex_head.tex
% Contine comenzi de punere in pagina
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% -- stilul paginii --
\pagestyle{plain}

% -- spatierea paragrafelor --
```

```

\setlength{\parskip}{1.5ex plus0.5ex minus0.5ex}
\setlength{\parindent}{1em}
\renewcommand{\baselinestretch}{1}

% -- formatul paginii --
\raggedbottom
% flushbottom           % varianta la \raggedbottom

\setlength{\textwidth}{16cm}
\setlength{\textheight}{25cm}
\setlength{\topmargin}{0cm}
\setlength{\oddsidemargin}{1cm}
\setlength{\evensidemargin}{0cm}
\setlength{\headheight}{0cm}
\setlength{\headsep}{0cm}

% -- figuri --
\newcounter{myfig}           % contor pentru figuri

\newcommand{\mycaption}[2] { % comanda pentru numerotarea figurilor
    \refstepcounter{myfig} % creste contorul
    \label{#2}             % #2 eticheta pentru referiri ulterioare
    \begin{center}
    {\bf Fig. \themyfig. }{#1} % #1 titlul figurii
    \end{center}
}

\newcommand{\PSFIG}[5]      % figura ps numerotata,
{                             % plasata in pagina de LaTeX
    \begin{figure}{#5}      % #5 este una din literele h, t, b
    \centerline{\psfig{figure=#1,height=#2}}
                                % #1 este numele fisierului ps
                                % #2 -dimensiune
    \caption{#3}{#4}        % #3 - titlul #4 - eticheta
}

\newcommand{\PSFIGH}[4]    % figura ps numerotata,
{                             % plasata in pagina in punctul apelului
    \vspace{0.5cm}
    \par
    \centerline{\psfig{figure=#1,height=#2}}
    \par                      % #1 nume fisier ps;#2 -dimensiune
    \samepage
    \mycaption{#3}{#4}      % #3 - titlul #4 - eticheta
}

```

4.10.2 Un exemplu de document

Comenzi de formatare L^AT_EX

– *Exemple de utilizare* –

Nume Autor

28 iulie 1998

Cuprins

1	Intrare în mod matematic	1
2	Formule matematice	2
3	Tabele	3

1 Intrare în mod matematic

Formulele matematice pot fi scrise în cadrul textului sau pe linie separată.

1.1 Formule în text

Formulele se pot scrie în cadrul textului $a_1 = x^2$, anumite fonturi fiind în acest caz mai mici (selectate automat de L^AT_EX).

1.2 Formule pe linie separată

Alternativ, formulele se pot scrie pe linie separată:

$$a_1 = x^2.$$

Același rezultat se obține și astfel:

$$a_1 = x^2.$$

De multe ori e nevoie de ecuații numerotate:

$$a_1 = x^2. \tag{1}$$

2 Formule matematice

2.1 Integrale

$$\mathcal{I} = \int_a^{a+b} (x^2 + \sin(x) + e^x) dx.$$

2.2 Vectori

$$\text{rot } \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \quad \langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a} \cdot \mathbf{b}.$$

2.3 Sumă, produs

$$S = \sum_{k=1}^{\infty} a_k^2 = \sum_{k=1}^{\infty} \left(\frac{1}{2k+1} \right)^2, \quad P = \prod_{k=1}^m S_k.$$

2.4 Utilizarea parantezelor

Comparați următoarele două formule:

$$y = (x(2x^2 + x(a+c) + b^2) + \frac{b^2}{c^2})$$

$$y = \left(x \left(2x^2 + x(a+c) + b^2 \right) + \frac{b^2}{c^2} \right)$$

2.5 Funcții matematice

$$f(x) = \begin{cases} x, & x < 0 \\ x^2, & x \geq 0 \end{cases}$$

$$E = \sin(x) + \tan(x) + \ln(x).$$

$$\lim_{x \rightarrow \infty} \frac{x+1}{2x+7} = \frac{1}{2}, \quad x \in \mathbb{R}.$$

2.6 Matrice

$$A = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

Centrat	Centrat	Centrat
Centrat	Centrat	Centrat

Tabelul 1: O explicație pentru tabel

2.7 Sisteme de ecuații

$$x^2 + 2x + y = 0; \quad (2)$$

$$2y^2 + 2y + 3x + 7xy = 1. \quad (3)$$

sau pentru a numerota o singură dată:

$$x^2 + 2x + y = 0; \quad (4)$$

$$2y^2 + 2y + 3x + 7xy = 1.$$

$$\begin{aligned} x^2 + 2x + y &= 0; \\ 2y^2 + 2y + 3x + 7xy &= 1. \end{aligned} \quad (5)$$

Pentru a alinia ecuațiile la alt caracter decât semnul “=” se procedează ca în ecuațiile (6):

$$\begin{aligned} x^2 + \quad 2x + y &= 0; \\ 2y^2 + \quad 2y + 3x + 7xy &= 1 \end{aligned} \quad (6)$$

2.8 Diverse

Nu știm cum să facem litere grecești bold: φ φ .

Observați diferența dintre: aa...a și aa...a.

$$\|\mathbf{v}\|_2 \stackrel{def}{=} \sqrt{\sum_k v_k^2}$$

3 Tabele

Un tabel plasat în punctul unde este apelat:

Aliniat stânga	Centrat	Lățime 5cm	Aliniat dreapta
București	S	Aici putem scrie un text mai lung care va fi aliniat la ambele margini	1.000
Timișoara	V	Text	2.000
Cluj	E	Text	3.000

Se pot defini și tabele numerotate (tabelul 1), plasate automat de \LaTeX .

Și iată cum a fost realizat:

```

\documentstyle[psfig,twoside,12pt,a4,romanian,rom]{article}

\input{latex_head.tex}

% Definirea unor noi comenzi; pot fi puse in latex_head.tex
\newcommand{\mr}[1]{\mathrm{#1}}
\newcommand{\mb}[1]{\mathbf{#1}}
\newcommand{\mc}[1]{\mathcal{#1}}

\title{{\huge \bf Comenzi de formatare \LaTeX}\ \ \ %\newline
      {\Large\sl -- Exemple de utilizare --}}
\author{Nume Autor}
\date{\today}

%=====
%=====

\begin{document}          % inceputul documentului
%\large
\maketitle                % tipareste titlul, autorul 'si data

\setcounter{tocdepth}{1}
\tableofcontents          % tipareste cuprinsul lucrarii

\section{Intrare 'in mod matematic} % inceputul unei sectiuni
                                % sectiunile se numereaza automat

Formulele matematice pot fi scrise 'in cadrul textului
sau pe linie separat'a.

\subsection{Formule 'in text} % inceputul unei subsectiuni
Formulele se pot scrie 'in cadrul textului  $a_1 = x^2$ , anumite
fonturi fiind 'in acest caz mai mici (selectate automat de \LaTeX).

\subsection{Formule pe linie separat'a} % o noua subsectiune
Alternativ, formulele se pot scrie pe linie separat'a:  $\$a_1 = x^2.\$$ 

Acela'si rezultat se ob'tine 'si astfel:
\begin{displaymath}
a_1 = x^2.
\end{displaymath}

De multe ori e nevoie de ecua'tii numerotate:
\begin{equation}
a_1 = x^2.
\end{equation}

```

```

\section{Formule matematice}

\subsection{Integrale}
$$
{\mathcal{I}} = \int_a^b (x^2 + \sin(x) + e^x) dx.
$$

\subsection{Vectori}
$$ \operatorname{rot} \mathbf{E} = - \frac{\partial \mathbf{B}}{\partial t},
\quad \mathbf{a}, \mathbf{b} \rangle = \mathbf{a} \cdot \mathbf{b}.
$$

\subsection{Suma, produs}
$$ S = \sum_{k=1}^{\infty} a_k^2 =
\sum_{k=1}^{\infty} \left( \frac{1}{2k+1} \right)^2,
\quad P = \prod_{k=1}^m S_k.
$$

\subsection{Utilizarea parantezelor}
Comparați următoarele două formule:
$$
y = (x(2x^2 + x(a+c) + b^2) + \frac{b^2}{c^2})
$$

$$
y = \left( x \left( 2x^2 + x \left( a + c \right) + b^2 \right) + \frac{b^2}{c^2} \right)
$$

\subsection{Funcții matematice}
$$
f(x) = \left\{ \begin{array}{l} x, \quad x < 0 \\ x^2, \quad x \geq 0 \end{array} \right.
$$

$$
E = \sin(x) + \tan(x) + \ln(x).
$$

$$
\lim_{x \rightarrow \infty} \frac{x+1}{2x+7} = \frac{1}{2}, \quad x \in \mathbb{R}.
$$

```

```
\subsection{Matrice}
```

```
$$
```

```
A = \left[
```

```
\begin{array}{cccc}
```

```
1 & 0 & \cdots & 0 \\
```

```
0 & 1 & \cdots & 0 \\
```

```
\vdots & & \ddots & \vdots \\
```

```
0 & 0 & \cdots & 1
```

```
\end{array}
```

```
\right]
```

```
$$
```

```
\subsection{Sisteme de ecua'tii}
```

```
\begin{eqnarray}
```

```
x^2 + 2x + y = 0; \\
```

```
2 y^2 + 2y + 3x + 7xy = 1.
```

```
\end{eqnarray}
```

```
sau pentru a numerota o singura data:
```

```
\begin{eqnarray}
```

```
x^2 + 2x + y = 0; \\ \nonumber
```

```
2 y^2 + 2y + 3x + 7xy = 1.
```

```
\end{eqnarray}
```

```
% astfel reducem spatiul liber intre doua ecuatii
```

```
\begin{equation}
```

```
\begin{array}{l}
```

```
x^2 + 2x + y = 0; \\
```

```
2 y^2 + 2y + 3x + 7xy = 1.
```

```
\end{array}
```

```
\end{equation}
```

Pentru a alinia ecua'tiile la alt caracter decat semnul ``=''

se procedeaza ca in ecua'tiile (\ref{ec1}):

```
\begin{eqnarray}
```

```
\label{ec1}
```

```
x^2 + & 2x + y & = 0 ; \\ \nonumber
```

```
2 y^2 + & 2y + 3x + 7xy & = 1
```

```
\end{eqnarray}
```

```
\subsection{Diverse}
```

Nu stim cum sa facem litere grecesti bold:

```
$
```

```
\mb{\varphi} \\ \varphi
```

```
$.
```

Observa'ti diferen'ta dintre: aa\$\ldots\$a si aa\$\cdots\$a.

```

$$
|| \mb v ||_2 \stackrel{\text{def}}{=} \sqrt{\sum_k v_k^2}
$$

```

```
\section{Tabele}
```

Un tabel plasat în punctul unde este apelat:

```

\begin{tabular}{|l|c|p{5cm}|r|}
\hline
Aliniat st^anga & Centrat & L'a'time 5cm & Aliniat dreapta \\
\hline
Bucure'sti & S & Aici putem scrie un text mai lung
                care va fi aliniat la ambele margini & 1.000 \\
\hline
Timi'soara & V & Text & 2.000 \\
\hline
Cluj & E & Text & 3.000 \\
\hline
\end{tabular}

```

Se pot defini și tabele numerotate (tabelul `\ref{tabelcentrat}`), plasate automat de `\LaTeX`.

```

\begin{table}[t]
\begin{center}
\begin{tabular}{|ccc|}
\hline
Centrat & Centrat & Centrat \\
\hline
Centrat & Centrat & Centrat \\
\hline
\end{tabular}
\end{center}
\caption{O explica'tie pentru tabel}
\label{tabelcentrat}
\end{table}
\end{document}

```

4.10.3 Un exemplu de articol în stilul IEEE Trans. on Magnetics

Parameter Extraction for Microwave Devices Based on 4SID Techniques

Irina Munteanu and Daniel Ioan, members, IEEE

"POLITEHNICA" University of Bucharest, Numerical Methods Laboratory – Electrical Engineering Dept.
Spl. Independentei 313, 77206 Bucharest, Romania

Abstract—Microwave devices parameter extraction is achieved by a field analysis of the device, followed by a procedure capable to extract lumped circuit parameters from the time domain information. The proposed algorithm is based on the 4SID (Subspace-based State-Space System Identification) techniques, followed by an optimal order reduction.

Index terms—Microwave, parameter extraction, Hankel norm, 4SID technique.

I. INTRODUCTION

In order to reduce the simulation time, microwave devices are often represented by approximate lumped circuits. Subsequently, SPICE-like simulators can be used to analyze complex configurations.

For circuit parameter extraction, several techniques are currently used, among them the Krylov Subspace [1], Padé via Lanczos, and asymptotic waveform evaluation [2] techniques.

II. PROBLEM FORMULATION AND SOLUTION

Under certain boundary conditions [3], a microwave device can be treated as an Electromagnetic Circuit Element (ECE).

In its domain \mathcal{D} , linear media are considered:

$$\nabla \times \mathbf{H} = \sigma \mathbf{E} + \varepsilon \frac{\partial \mathbf{E}}{\partial t}, \quad \nu \nabla \times \mathbf{E} = -\frac{\partial \mathbf{H}}{\partial t}. \quad (1)$$

For each of the n terminals placed on the boundary $\partial \mathcal{D}$, a voltage and a current can be properly defined. As proven in the full paper, the solution of (1) with ECE boundary conditions can be expressed as a series of generalized eigenfunctions of the $\nabla \times (\nu \nabla \times \cdot)$ operator, which corresponds to an infinite RLC circuit.

What we look for, is an n -pole lumped circuit which is the best finite approximation for the device. It can be obtained by the following **algorithm**:

Step 1: The problem (1) together with the ECE boundary conditions is solved only once by a time-domain numerical method, at arbitrary excitation (the one used in device's real functioning is recommended);

Step 2: The 4SID algorithm [4] is used to generate a system S , equivalent to the device in what concerns the

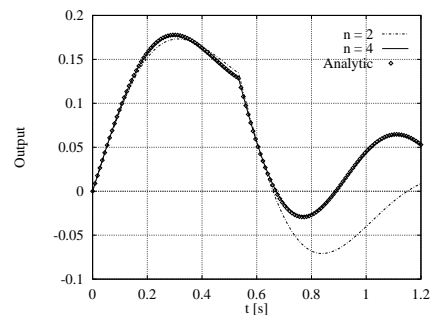


Fig. 1. Output of the initial and reduced systems

terminal behaviour. Using the best Hankel-norm approximation a reduced-order system S' is generated [3];

Step 3: The transfer model of the system S' , together with Foster synthesis provide the lumped circuit.

The parameter extraction algorithm was first applied to a problem with known analytical solution,

A bended strip, placed on a substrate and enclosed in a metallic box has also been analyzed. Numerical results will be presented in the full paper.

III. CONCLUSIONS

The proposed technique appears to be an efficient alternative for determining lumped circuit parameters for a class of general distributed-parameters devices. It can be used for extracting parameters valid over a wide range of frequencies with only one time-domain field simulation. Results obtained so far are very promising.

REFERENCES

- [1] M. Celik, A. C. Cangellaris, and A. Yaghmour, "An all-purpose transmission-line model for interconnect simulations in SPICE," *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 1857–1867, Oct. 1997.
- [2] J. E. Bracken and Z. J. Cendes, "Evaluation for S-domain solution of electromagnetic waves," in *Proc. 11th COMPUMAG Conf.*, (Rio de Janeiro, Brazil), pp. 207–208, Nov. 1997.
- [3] D. Ioan, I. Munteanu, and C.-G. Constantin, "The best approximation of the field effects in electric circuits coupled problems," in *Proc. 11th COMPUMAG Conf.*, (Rio de Janeiro, Brazil), pp. 747–748, Nov. 1997.
- [4] Y. Cho, G. Xu, and T. Kailath, "Fast identification of state-space models via exploitation of displacement structure," *IEEE Trans. AC-26*, no. 10, pp. 2004–2017, 1994.

Și iată cum a fost realizat:

```
% Rezumat 'in stilul IEEE Transactions on Magnetics
\documentstyle[IEEEemagn,psfig]{article}

% Definitii
\newcommand{\half}%
{\raisebox{2.5pt}{\scriptsize 1}{\small /}\raisebox{-1pt}{\scriptsize 2}}

\newcommand{\B}{\bf B}
\newcommand{\E}{\bf E}
\newcommand{\U}{\overline{u}}
\newcommand{\Fi}{\overline{\varphi}}
\newcommand{\Ep}{\overline{\varepsilon}}
\newcommand{\M}{\overline{\mu}}

\newcommand{\Be}{\mathbf B}
\newcommand{\Ee}{\mathbf E}
\newcommand{\He}{\mathbf H}
\newcommand{\De}{\mathbf D}
\newcommand{\Je}{\mathbf J}
\newcommand{\Do}{\mathcal D}

% -- keywords
\def\keywords#1{\vspace*{3.5ex}{\small \boldmath \indent
    $Index\ terms$\ /}---\ignorespaces {\small \bf #1}}

% -- figures --
\newcommand{\PSFIG}[4]
{
\begin{figure}
    \centerline{\psfig{figure=#1,height=#2}}
    \caption{#3}
    \label{#4}
\end{figure}
}

\begin{document}

\boldmath \title{Parameter Extraction for Microwave Devices Based on
4SID Techniques} \unboldmath

\author{Irina Munteanu and Daniel Ioan, members, IEEE\
{\small "POLITEHNICA" University of Bucharest, Numerical Methods Laboratory --
    Electrical Engineering Dept. }\
{\small Spl. Independentei 313, 77206 Bucharest, Romania} }

%% In articolul final apar si aceste linii
```

```

%%
%% \thanks{%
%% \newline \indent I.\ Munteanu, e-mail irina@lmn.pub.ro.
%% \newline \indent D.\ Ioan, e-mail daniel@lmn.pub.ro.
%% \newline \indent This work was partially supported by the Romanian Ministry
%% of Education under the CNCSU 106 grant.
%% }

\maketitle

\abstract{%
Microwave devices parameter extraction is achieved by a field analysis of the
device, followed by a procedure capable to extract lumped circuit parameters
from the time domain information. The proposed algorithm is based on the
4SID (Subspace--based State--Space System IDentification) techniques,
followed by an optimal order reduction.
}

\keywords{Microwave, parameter extraction, Hankel norm, 4SID technique.}

\section{Introduction}
In order to reduce the simulation time, microwave devices are often
represented by approximate lumped circuits. Subsequently, SPICE--like
simulators can be used to analyze complex configurations.

For circuit parameter extraction, several techniques are currently used,
among them the Krylov Subspace \cite{celik97a}, Pad\`{e} via
Lanczos, and asymptotic waveform evaluation \cite{bracken97a} techniques.

\section{Problem Formulation and Solution}

Under certain boundary conditions \cite{ioan97n}, a microwave device can be
treated as an {Electromagnetic Circuit Element} ({\bf ECE}).

In its domain \Do, linear media are considered:
%
\begin{equation}
\begin{array}{l}
\nabla\!\times\!\He = \sigma \Ee + \varepsilon\frac{\partial\!\Ee}{\partial t},\quad
\nu\nabla\!\times\!\Ee = -\frac{\partial\!\He}{\partial t}.
\end{array}
\end{equation}
\label{ec_gen}
%
For each of the $n$ terminals placed on the boundary $\partial\!\Do$, a voltage
and a current can be properly defined. As proven in the full paper, the
solution of (\ref{ec_gen}) with ECE boundary conditions can be expressed as a
series of generalized eigenfunctions of the $\nabla\!\times$ ($\nu\nabla\!\times$

```

$\mathbf{\dot{}}$ operator, which corresponds to an infinite RLC circuit.

What we look for, is an n -pole lumped circuit which is the best finite approximation for the device. It can be obtained by the following **algorithm**:

Step 1: \ \ \ The problem (\ref{ec_gen}) together with the ECE boundary conditions is solved only once by a time--domain numerical method, at arbitrary excitation (the one used in device's real functioning is recommended);

Step 2: \ \ \ The 4SID algorithm \cite{Cho} is used to generate a system S , equivalent to the device in what concerns the terminal behaviour. Using the best Hankel--norm approximation a reduced--order system S' is generated \cite{ioan97n};

Step 3: \ \ \ The transfer model of the system S' , together with Foster synthesis provide the lumped circuit.

The parameter extraction algorithm was first applied to a problem with known analytical solution.

\PSFIG{sol.eps}{4.7cm}{Output of the initial and reduced systems}{sol}

A bended strip, placed on a substrate and enclosed in a metallic box has also been analyzed. Numerical results will be presented in the full paper.

\section{Conclusions}

The proposed technique appears to be an efficient alternative for determining lumped circuit parameters for a class of general distributed--parameters devices.

It can be used for extracting parameters valid over a wide range of frequencies with only one time--domain field simulation. Results obtained so far are very promising.

\bibliographystyle{ieeetr}

\bibliography{rio97,ref}

\end{document}

4.11 Gestionarea referințelor bibliografice folosind BIBTEX

Referințele bibliografice se pot introduce direct, în interiorul fișierului `tex`. În acest fel însă, apare dezavantajul că o aceeași referință trebuie introdusă de mai multe ori, în fiecare fișier `.tex` în care este folosită. Cum fiecare editură are propriul său format al citărilor, aceeași informație (referința bibliografică) trebuie pusă în pagină în moduri diferite.

De aceea, se recomandă utilizare unor fișiere BIBTEX, care sunt de fapt baze de date cu referințe bibliografice și de unde LATEX își poate prelua numai acele referințe care sunt citate. În plus, o aceeași referință poate fi pusă în pagină în mod diferit, în diferite fișiere LATEX.

Un alt avantaj al utilizării BIBTEX constă în aceea că editarea bazei de date (adăugare, ștergere, modificare a unei referințe) se poate face eficient cu utilitarele **BibCard** și **BibView**, care au o interfață grafică prietenoasă cu utilizatorul (figura 4.1).

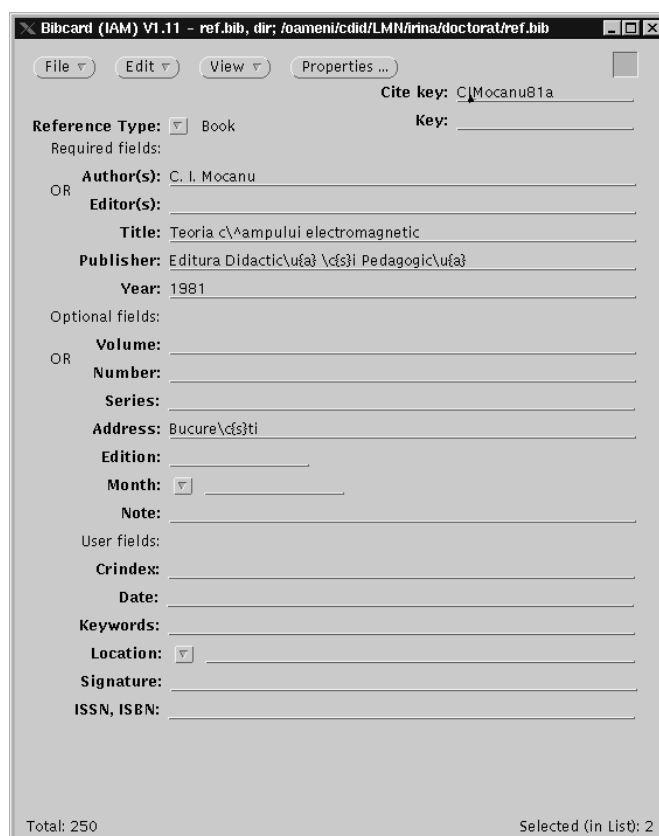


Fig. 4.1: Interfața grafică a programului **BibCard**

4.11.1 Structura unui fișier BIBTEX

Fișierele BIBTEX sunt fișiere text care conțin *înregistrări*, câte una pentru fiecare referință. Un astfel de fișier are extensia `.bib`.

O *înregistrare* BIBTEX are structura:

```
@tip {cheie_de_citare,
      câmp = valoare,
      câmp = valoare,
      :
}
```

După cum se constată, o *înregistrare* BIBTEX începe cu un cuvânt–cheie precedat de semnul `@`, care reprezintă **tipul** referinței. Acest cuvânt–cheie poate fi:

ARTICLE	articol
BOOK	carte
BOOKLET	broșură
CONFERENCE	lucrările unei conferințe
INBOOK	capitol dintr-o carte
INCOLLECTION	carte dintr-o colecție
INPROCEEDINGS	lucrare de la o conferință
MANUAL	manual
MASTERSTHESIS	teză de master
MISC	diverse
PHDTHESIS	teză de doctorat
PROCEEDINGS	lucrările unei conferințe
TECHREPORT	raport tehnic
UNPUBLISHED	lucrare nepublicată

Cheia de citare reprezintă o succesiune de litere și cifre, unică, ce va fi folosită pentru referirea la *înregistrare* în cadrul unui fișier L^AT_EX (folosind `\cite`).

Se recomandă folosirea unui sistem rațional de generare a cheilor de citare, de exemplu: inițiala prenumelui și numele de familie al primului autor, urmate de anul apariției și de o literă sau cifră, cum ar fi: `imunteanu97a`, `imunteanu97b`, ...

Chei de citare precum `Munteanu` nu sunt recomandabile: autorul Munteanu Ion nu va scrie probabil o singură operă pe parcursul carierei sale, iar cu acest sistem `Munteanu George` ar trebui să aibă aceeași cheie de citare! De asemenea, chei de citare conținând inițiale precum `Bsd22m` nu sunt recomandabile deoarece semnificația acestei combinații de litere se uită după scurt timp.

După cheia de citare urmează, între acolade, o listă de **câmpuri** ale *înregistrării*, separate prin virgulă. Câmpurile sunt obligatorii sau opționale. O parte dintre cele opționale sunt predefinite, dar utilizatorul poate defini oricâte noi câmpuri opționale.

Câmpurile **predefinite** ale unei *înregistrări* BIBTEX sunt:

- **KEY** (opțional) – poate conține, de exemplu, un cod al domeniului căruia îi aparține lucrarea, de exemplu: “matem”, “electromag”, “VLSI”, etc.
- **ADDRESS** (opțional) – adresa editorului sau a conferinței
- **AUTHOR** (obligatoriu) – lista autorilor, separați prin cuvântul **and**

Exemplu:

`AUTHOR = {Ion Popescu and George Ionescu and Virgil Georgescu}`

- **TITLE** (obligatoriu la tipul **ARTICLE**) – titlul lucrării.

În titlu se pot folosi comenzi \LaTeX , cum ar fi `\c{a}` pentru a obține litera **ă**, `L^2` pentru a obține L^2 , etc.

Majoritatea stilurilor de citare \LaTeX transformă literele mari din titlu în litere mici. De exemplu, titlul

`TITLE = {Ecuatiile lui Maxwell}`

va apărea în lista bibliografică sub forma

Ecuatiile lui maxwell

Dacă se dorește ca o parte a titlului să fie păstrată în lista bibliografică așa cum apare ea în fișierul $\text{BIB}\LaTeX$, acea parte trebuie inclusă între acolade:

`TITLE = {Ecuatiile lui {M}axwell}`

- **BOOKTITLE** (obligatoriu la tipul **BOOK** și **INPROCEEDINGS**) – titlul cărții
- **YEAR** (obligatoriu) – anul apariției
- **MONTH** (opțional) – luna apariției (sau luna în care a fost scrisă lucrarea, în cazul celor nepublicate). Pot fi folosite prescurtări (în limba engleză) ale lunilor anului: **jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec**.
- **JOURNAL** (obligatoriu la tipul **ARTICLE**) – revista în care a apărut lucrarea
- **VOLUME** (opțional) – numărul volumului
- **PAGES** – numerele de pagini, sub forma: **120** sau **120--140**
- **PUBLISHER** – Editura
- **EDITOR** – Numele editorilor (persoane, nu instituții!) în cazul cărților cu mai mulți autori
- **TYPE** (opțional) – Tipul raportului tehnic

Pot fi incluse și alte câmpuri definite de utilizator, cum ar fi: **ISBN, ISSN, ABSTRACT**, etc.

4.11.2 Introducerea referințelor bibliografice în fișierele L^AT_EX

Citările se fac în mod obișnuit, cu

```
\cite{cheie_de_citare}
```

sau

```
\nocite{cheie_de_citare},
```

unde `cheie_de_citare` este una din cheile dintr-un fișier `.bib`.

În locul unde se dorește bibliografia, este necesară introducerea, în fișierul `.tex`, a două comenzi L^AT_EX:

```
\bibliographystyle{stil}
```

```
\bibliography{listă_fiș_bib}
```

Stilul bibliografic

Parametrul *stil* este unul dintre **stilurile bibliografice** (a nu se confunda cu stilurile L^AT_EX) existente. Stilurile bibliografice sunt memorate în fișiere cu extensia `.bst`, care conțin regulile de punere în pagină a unei referințe, de exemplu: inițiala prenumelui urmată de numele de familie al autorului, titlul cu litere italice, numele revistei cu litere drepte, numărul volumului precedat de literele “vol.”, etc.

Cele mai des utilizate fișiere de stil bibliografic sunt:

- `plain.bst` (declarat cu `\bibliographystyle{plain}`), în care citările sunt numerotate între paranteze pătrate ([1]) și sunt sortate în ordinea alfabetică a numelor autorilor;
- `unsrt.bst`, asemănător cu `plain` dar referințele apar în listă în ordinea în care au fost citate în text;
- `alpha.bst`, asemănător cu `plain`, dar citările sunt alfa–numerice ([Pop97] pentru o lucrare din 1997 scrisă de autorul Popescu);
- `abbrv.bst`, asemănător cu `plain`, dar cu folosirea de prescurtări pentru prenume, lună și numele revistei;
- `plainrom.bst` și `unsrtrom.bst`, asemănătoare cu `plain`, respectiv `unsrt`, dar adaptate documentelor scrise în limba română (aceste stiluri nu sunt standard, ci au fost realizate de I. Munteanu folosind utilitarul `makebst`, pentru a fi utilizate în LMN).

Revistele de prestigiu au propriile lor stiluri bibliografice, de exemplu:

`ieee.bst` – la revista IEEE Transactions on Magnetics

`ieeetran.bst` – la celelalte reviste IEEE

`acm.bst` – la revistele ACM (Association of Computing Machinery)

Lista de fișiere de bibliografie

Parametrul `listă_fiș_bib` este o listă de fișiere `.bib`, fără extensie și fără blancuri între ele, separate prin virgule.

Exemplu: `\bibliography{carti,articole}`

unde `carti.bib` și `articole.bib` sunt două fișiere conținând înregistrări `BIBTEX`.

Comenzi pentru procesarea fișierelor `LATEX` cu includerea bibliografiei

Comanda care permite includerea listei de referințe citate în fișierul `LATEX` este `bibtex`.

O succesiune tipică de comenzi pentru realizarea fișierului imprimabil în cazul în care bibliografia este inclusă din fișiere `.bib` este (să presupunem că fișierul `LATEX` are numele `raport.tex`):

```
latex raport
bibtex raport
latex raport
latex raport
dvips -t a4 -o raport.ps raport
```

Se constată că după executarea comenzii `bibtex`, este necesară compilarea de cel puțin două ori cu `LATEX`, pentru ca citările să apară corect în text.

4.11.3 Fișiere utilizate în `BIBTEX`

- `.bst` - definiția stilului bibliografic
- `.bib` - conține referințe sub formă de înregistrări `BIBTEX`
- `.blg` - conține lista eventualelor erori apărute în urma executării `BIBTEX`
- `.bbl` - conține lista de referințe citate în fișierul `.tex`, formatate conform definiției din fișierul de stil (`.bst`) folosit.

4.11.4 Exemple de utilizare

În cele ce urmează este prezentat rezultatul folosirii diferitelor stiluri bibliografice, pentru același set de citări.

Fișierul `.tex` utilizat a avut structura:

```
\documentstyle[12pt,a4wide,romanian]{article}

\renewcommand{\refname}{\vspace*{-1cm}}
\pagestyle{empty}

\begin{document}
\section*{Stilul \tt\bfseries plain}
\nocite{reichard92a} \nocite{wolfram96a} \nocite{munteanu96b1} \nocite{ioan92a}
\nocite{ioan92c}

\bibliographystyle{plain}
\bibliography{ref,unix,controlsys,ref1,book}      % se inlocuieste pe rand cu:
                                                    % unsrt,alpha,abbrv,unsrtrom
\end{document}
```

Rezultatul procesării cu L^AT_EX și cu B_IB_TE_X este prezentat mai jos.

Stilul plain

- [1] D. C. Ioan and Irina Munteanu. Numerical methods in the electrical engineer's education. *European J. of Engineering Education*, 17(2):151–157, 1992.
 - [2] D. C. Ioan, Irina Munteanu, B. Ionescu, M. Popescu, and R. Popa. Computer aided learning of numerical methods for electrical engineers. In *3rd World Conf. on Engineering Education*, pages 15–20, Portsmouth, United Kingdom, September 20–25 1992.
 - [3] Irina Munteanu and F. M. G. Tomescu. Optimisation of a magnetic device based on symbolic analysis of the field. *IEEE Trans. Magn.*, 33(2):1840–1843, March 1997.
 - [4] K. Reichard and E. F. Johnson. *Teach Yourself UNIX*. MIS Press, New York, 1992.
 - [5] Stephen Wolfram. *The Mathematica Book*. Wolfram Media / Cambridge University Press, 3 edition, 1996.
-

Stilul unsrt

- [1] K. Reichard and E. F. Johnson. *Teach Yourself UNIX*. MIS Press, New York, 1992.
 - [2] Stephen Wolfram. *The Mathematica Book*. Wolfram Media / Cambridge University Press, 3 edition, 1996.
 - [3] Irina Munteanu and F. M. G. Tomescu. Optimisation of a magnetic device based on symbolic analysis of the field. *IEEE Trans. Magn.*, 33(2):1840–1843, March 1997.
 - [4] D. C. Ioan and Irina Munteanu. Numerical methods in the electrical engineer's education. *European J. of Engineering Education*, 17(2):151–157, 1992.
 - [5] D. C. Ioan, Irina Munteanu, B. Ionescu, M. Popescu, and R. Popa. Computer aided learning of numerical methods for electrical engineers. In *3rd World Conf. on Engineering Education*, pages 15–20, Portsmouth, United Kingdom, September 20–25 1992.
-

Stilul alpha

- [IM92] D. C. Ioan and Irina Munteanu. Numerical methods in the electrical engineer's education. *European J. of Engineering Education*, 17(2):151–157, 1992.
 - [IMI+92] D. C. Ioan, Irina Munteanu, B. Ionescu, M. Popescu, and R. Popa. Computer aided learning of numerical methods for electrical engineers. In *3rd World Conf. on Engineering Education*, pages 15–20, Portsmouth, United Kingdom, September 20–25 1992.
 - [MT97] Irina Munteanu and F. M. G. Tomescu. Optimisation of a magnetic device based on symbolic analysis of the field. *IEEE Trans. Magn.*, 33(2):1840–1843, March 1997.
 - [RJ92] K. Reichard and E. F. Johnson. *Teach Yourself UNIX*. MIS Press, New York, 1992.
 - [Wol96] Stephen Wolfram. *The Mathematica Book*. Wolfram Media / Cambridge University Press, 3 edition, 1996.
-

Stilul abbrev

- [1] D. C. Ioan and I. Munteanu. Numerical methods in the electrical engineer's education. *European J. of Engineering Education*, 17(2):151–157, 1992.
 - [2] D. C. Ioan, I. Munteanu, B. Ionescu, M. Popescu, and R. Popa. Computer aided learning of numerical methods for electrical engineers. In *3rd World Conf. on Engineering Education*, pages 15–20, Portsmouth, United Kingdom, Sept. 20–25 1992.
 - [3] I. Munteanu and F. M. G. Tomescu. Optimisation of a magnetic device based on symbolic analysis of the field. *IEEE Trans. Magn.*, 33(2):1840–1843, Mar. 1997.
 - [4] K. Reichard and E. F. Johnson. *Teach Yourself UNIX*. MIS Press, New York, 1992.
 - [5] S. Wolfram. *The Mathematica Book*. Wolfram Media / Cambridge University Press, 3 edition, 1996.
-

Stilul unsrtrom

- [1] K. Reichard și E. F. Johnson. *Teach Yourself UNIX*. MIS Press, New York, 1992.
 - [2] Stephen Wolfram. *The Mathematica Book*. Wolfram Media / Cambridge University Press, a treia ed., 1996.
 - [3] Irina Munteanu și F. M. G. Tomescu. Optimisation of a magnetic device based on symbolic analysis of the field. *IEEE Trans. Magn.*, vol. 33, nr. 2, pp. 1840–1843, mar. 1997.
 - [4] D. C. Ioan și Irina Munteanu. Numerical methods in the electrical engineer's education. *European J. of Engineering Education*, vol. 17, nr. 2, pp. 151–157, 1992.
 - [5] D. C. Ioan, Irina Munteanu, B. Ionescu, M. Popescu și R. Popa. Computer aided learning of numerical methods for electrical engineers. În *3rd World Conf. on Engineering Education*, pp. 15–20. Portsmouth, United Kingdom, sept. 20–25 1992.
-

4.12 Configurarea \LaTeX de către utilizator

Pentru ca \LaTeX să funcționeze fără probleme, el trebuie să “știe” unde sunt plasate fișierele de care are nevoie. Aceste fișiere sunt de tip:

- `.sty` – stiluri \LaTeX
- `.tex` – documente \LaTeX (generate de utilizator)
- `.pk` – fișiere care conțin definiții de fonturi
- `.bib` – fișiere care conțin liste de referințe $\text{BIB}\TeX$
- `.bst` – fișiere care conțin definiții de stiluri bibliografice

Aceste fișiere sunt căutate de \LaTeX și $\text{BIB}\TeX$, în mod implicit, în două directoare:

- directorul curent;
- directoarele predefinite de administratorul de sistem la instalarea \LaTeX .

Cum utilizatorul poate defini el însuși noi stiluri \LaTeX și $\text{BIB}\TeX$, sau poate prelua de pe Internet astfel de stiluri, va trebui să le salveze în directoare proprii (unde are drept de scriere) și nu în cele implicite. Pentru a evita existența în mai multe subdirectoare a acelorași fișiere `.sty`, `.bib` sau `.bst`, se recomandă:

- Crearea, în directorul `home`, a unui subdirector numit `latex`.
- Crearea, în subdirectorul `latex`, a trei alte directoare:
 - `sty`
 - `bst`
 - `bib`

în care utilizatorul va salva fișierele cu extensia corespunzătoare.

- Definirea următoarelor variabile de environment:

`TEXINPUTS` (se referă la fișierele `.sty`, `.tex`, etc.)

`TEXPKS` (se referă la fișierele `.pk`)

`BSTINPUTS` (se referă la fișierele `.bst`)

`BIBINPUTS` (se referă la fișierele `.bib`)

Cele patru variabile de environment pot fi definite din linia de comandă, sau direct în fișierul de configurare al shell-ului (de exemplu `.cshrc` sau `.bashrc`).

Aceste variabile trebuie să conțină liste de directoare (în care se găsesc fișierele utilizate de \LaTeX și de $\text{BIB}\TeX$), separate prin `:` (două puncte).

Exemplu de definire a variabilelor de configurare, în fișierul `.cshrc`

```
setenv TEXPKS /var/texfonts/pk/localfont
setenv TEXINPUTS ".:usr/lib/texmf/tex/latex/latex/:$HOME/latex/sty:\
/usr/lib/texmf/tex/latex/sty/:$HOME/docs/figuri/"
setenv BSTINPUTS $HOME/latex/bst
setenv BIBINPUTS $HOME/latex/bib
```

Exemplu de definire a variabilelor de configurare, în fișierul `.bashrc`

```
declare TEXPKS=/var/texfonts/pk/localfont
declare TEXINPUTS=".:usr/lib/texmf/tex/latex/latex/:$HOME/latex/sty:\
/usr/lib/texmf/tex/latex/sty/:$HOME/docs/figuri/"
declare BSTINPUTS=$HOME/latex/bst
declare BIBINPUTS=$HOME/latex/bib
```

4.13 Realizarea planșelor de prezentare folosind stilul `seminar`

Pentru a ușura realizarea planșelor sau transparentelor necesare unei prezentări orale a fost definit stilul \LaTeX `seminar`, a cărui principală caracteristică este aceea că folosește corpuri de litere mai mari.

Orice document \LaTeX poate fi **transformat într-o prezentare** prin următoarele trei modificări:

- se înlocuiește stilul \LaTeX folosit (`article`, `book` sau `report`) cu stilul `seminar`;
- se adaugă, după instrucțiunea \LaTeX `\begin{document}`, încă o linie:
`\begin{slide}`
- se adaugă, înainte de instrucțiunea \LaTeX `\end{document}`, încă o linie:
`\end{slide}`

Prelucrarea noului fișier cu `latex` se face în mod obișnuit.

O prezentare fără mari pretenții poate fi astfel realizată foarte rapid.

4.13.1 Comenzi specifice stilului seminar

Stilul `seminar` pune însă la dispoziție și alte **noi comenzi și cadruri** pentru gestionarea prezentărilor. Iată lista celor mai importante:

- `\begin{slide}`

⋮

`\end{slide}`

Cadru care definește o planșă de prezentare, așezată în pagină în format landscape.

- `\begin{slide*}`

⋮

`\end{slide*}`

Cadru care definește o planșă de prezentare, așezată în pagină în format portrait.

Observație: într-un fișier pot apărea oricâte cadruri `slide` și/sau `slide*`.

- `\newslide` – echivalent, în interiorul unui slide, comenzii `\newpage`.
- `\slideframe{chenar}` – comandă pentru definirea chenarului unei planșe. Dacă nu se dorește chenar, atunci parametrul `chenar` trebuie să aibă valoarea `none`. Valoarea implicită este `plain`.
Prin includerea, în lista de opțiuni ale comenzii `\documentstyle`, a opțiunii `fancybox`, parametrul `chenar` poate lua și valorile: `shadow`, `double`, `oval`, `Oval`.
- `\slidesmag{n}` – mărimea fonturilor în prezentare, cu n între -5 și 9.

Opțiuni ale comenzii `\documentstyle`, specifice stilului `seminar`

- `article` – planșele sunt micșorate și tipărite câte două pe o pagină (util în faza de pregătire a planșelor)
- `portrait` – imprimare pe hârtie așezată în format portrait.

În mod implicit, informația cuprinsă în fișierul `.dvi` despre tipul hârtiei este că aceasta este așezată în format landscape, deci pentru a imprima corect trebuie dată comanda `dvips` cu parametrul `landscape`:

```
dvips -t landscape fișier
```

La utilizare, în lista de opțiuni ale comenzii `\documentstyle`, a opțiunii `portrait`, se consideră hârtia așezată vertical (portrait).

Observație: această opțiune se referă numai la poziția hârtiei, nu și a textului în pagină (de așezarea textului în pagină se “ocupă” `slide` – landscape și respectiv `slide*` – portrait).

Pentru o listă completă a comenzilor, se recomandă consultarea documentației [44] și [82].

4.13.2 Exemplu de utilizare

Conclusions

- The proposed technique appears to be an efficient alternative for determining lumped circuit parameters for a class of general distributed-parameters devices.
- It can be used for extracting parameters valid over a wide range of frequencies with only one time-domain field simulation.
- Results obtained so far are promising.

Și iată cum a fost realizat:

```
% 0 prezentare cu o singura plansa slide*
\documentstyle[epsf,semrot,portrait,a4,fancybox]{seminar}

% -- page style --
\pagestyle{empty}
\slideframe{shadow}

\begin{document}

\begin{slide*}
\section*{Conclusions}
\begin{itemize}
\item The proposed technique appears to be an efficient alternative for
determining lumped circuit parameters for a class of general
distributed--parameters devices.
\item It can be used for extracting parameters valid over a wide range of
frequencies with only one time--domain field simulation.

\item Results obtained so far are promising.
\end{itemize}
\end{slide*}

\end{document}
```

4.14 Folosirea programului utilitar `make`

În vederea “automatizării” operațiilor necesare pentru a obține fișierul imprimabil, plecând de la un fișier-sursă \LaTeX , se poate folosi utilitarul `make`, descris mai pe larg în paragraful 7.3.

Pentru aceasta, în directorul în care se găsesc fișierele Dvs. \LaTeX trebuie să generați un fișier numit `Makefile`, a cărui structură poate fi următoarea:

```
# Numele fișierului .tex, fara extensie
FILE = referat

$(FILE).ps: $(FILE).dvi $(FILE).aux
    dvips -t a4 $(FILE) -o $(FILE).ps

cleanall:
    rm -f *.dvi *.aux *.log

clean:
    rm -f $(FILE).dvi $(FILE).aux $(FILE).log

$(FILE).dvi: $(FILE).tex
    latex $(FILE)

ghost: $(FILE).ps
    ghostview $(FILE).ps &
```

Comenzile din acest fișier (de genul `latex ...` sau `dvips ...`) vor fi indentate folosind `<Tab>` și nu spații.

Utilitarul se utilizează astfel:

- editați fișierul `Makefile` pentru a înlocui numele fișierului (definit prin `FILE =`) cu numele fișierului Dvs. \LaTeX , fără extensie (să presupunem că acesta este chiar `referat.tex`);
- pentru a genera fișierul imprimabil dați comanda:


```
make
```
- pentru a șterge fișierele `referat.aux`, `referat.dvi`, `referat.log` (e bine să faceți acest lucru după ce ați terminat de redactat și imprimat documentul):


```
make clean
```
- pentru a șterge toate fișierele cu extensia `.aux`, `.dvi` și `.log`:


```
make cleanall
```
- pentru a vizualiza a fișierului imprimabil PostScript:


```
make ghost
```

4.15 Exerciții

EXERCITIUL 1

Creați structura de directoare propusă în paragraful 4.12 (directoarele `latex`, `bst`, `sty`, `bib`).

În directorul `~/latex/bib`, creați fișierul `ref.bib` în care veți include lista Dvs. de referințe bibliografice. Pentru început, introduceți o singură referință, după modelul de mai jos:

```
@BOOK{CIMocanu81a,
      AUTHOR       = {C.~I.~Mocanu},
      ADDRESS      = {Bucure\c{s}ti},
      PUBLISHER    = {Editura Didactic\u{a} \c{s}i Pedagogic\u{a}},
      TITLE        = {Teoria c^ampului electromagnetic},
      YEAR         = {1981}
}
```

Observație: nu este recomandabil să folosiți, în fișiere `.bib`, prescurtări de tipul `ˆa` pentru a obține diacriticele românești.

EXERCITIUL 2

Editați fișierul `~/bashrc`, adăugând comenzile de configurare propuse în paragraful 4.12. Dați apoi comanda

```
source ~/.bashrc
```

pentru ca noile setări să devină active.

EXERCITIUL 3

În directorul `~/latex` creați fișierul `Makefile` cu structura descrisă în paragraful 4.14. Copiați apoi fișierul `Makefile` în directorul `~/metnum`.

EXERCITIUL 4

În directorul `metnum` scrieți un fișier numit `modelref.tex` care conține elementele principale ale unui referat de laborator. (Veți folosi acest model la toate laboratoarele de pe parcursul anului de studii aprofundate.)

O propunere de structură a fișierului `modelref.tex` este prezentată mai jos.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MODEL DE REFERAT DE LABORATOR %%%%%%%%%%
% Fisier: modelref.tex
% Se copiaza sub alt nume la fiecare nou referat
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\documentstyle[11pt,a4wide,psfig,rom,romanian]{article}

% Punere in pagina
\addtolength{\topmargin}{-1.5cm}
\addtolength{\textheight}{2cm}

% Definitii si noi comenzi
\newcommand{\instructor}[1]
    {\vspace*{-4ex}\centerline{{\bf Instructor}: #1}}

\newcommand{\antet}{{\normalsize
    U.P.B. \ \ Laboratorul de Metode Numerice --- LMN}
    \vspace*{3ex}}

\title{\antet\
\bf Titlul lucr'arii}          % Inlocuiti cu titlul real
\author{{\bf Ion Popescu}\ \ % Inlocuiti cu numele Dvs. si adresa email
    {\small {\tt ipopescu@lmn.pub.ro}, Grupa 161M,
    Facultatea Electrotehnic'a}}

\date{\today}

\begin{document}
\maketitle
\thispagestyle{empty}
\instructor{S.l. I. Munteanu} % Inlocuiti cu numele cadrului didactic

\tableofcontents              % Cuprinsul lucrarii
\vspace*{0.5cm}              % Spatiu vertical; la stilul report poate lipsi

%%% Aici incepe referatul de laborator propriu-zis
% Referatul este structurat cu sectiuni si subsectiuni

```



```

\section{Introducere}           % Numele acestei sectiuni este doar un exemplu
\section{Formulara problemei}
\section{Metoda de rezolvare}
\section{Rezultate ob'tinute}
\section{Concluzii}

\norom                          % Inhiba actiunea caracterului ´
\bibliographystyle{plain}
\bibliography{ref}              % Deocamdata foloseste doar ref.bib
\addcontentsline{toc}{section}{Bibliografie}

\rom                             % Caracterul ´ genereaza diacritice

% Referatul poate contine si anexe
\appendix
\section{Codul surs'a al programului} % Numele sectiunii este doar un exemplu

\end{document}

```

Compilați fișierul `modelref.tex` cu \LaTeX și eliminați eventualele erori. La vizualizarea cu `xdvi modelref`, rezultatul ar trebui să fie asemănător cu figura de pe pagina 112.

EXERCITIUL 5

După aceste pregătiri, veți începe acum crearea primului Dvs. “referat de laborator”. (Deoarece scopul acestei lucrări este de a vă deprinde cu utilizarea \LaTeX , veți constata că o parte din secțiunile standard cuprinse în `modelref.tex` vor lipsi. Vă recomandăm ca, atunci când va trebui să scrieți un adevărat referat de laborator să utilizați structura completă.)

Pentru siguranță, copiați modelul `modelref.tex` în directorul `~/latex`.

Copiați modelul `modelref.tex` sub numele `rc.tex` (evident, în directorul `metnum`).

Modificați fișierul `rc.tex` astfel încât documentul să reproducă identic (cu excepția numerelor și datei) modelul de pe pagina 112.

Notă: în realizarea modelului a fost folosită dimensiunea de 10pt a caracterelor.

Observație: NU scrieți niciodată numărul ecuației sau al referinței direct, ci folosiți comenzile de referire `\ref{ }`, respectiv de citare `\cite{ }`.

U.P.B. Laboratorul de Metode Numerice — LMN

Titlul lucrării

Ion Popescu

ipopescu@lmm.pub.ro, Grupa 161M, Facultatea Electrotehnică

28 iulie 1998

Instructor: Ș.l. I. Munteanu

Cuprins

1	Introducere	1
2	Formularea problemei	1
3	Metoda de rezolvare	1
4	Rezultate obținute	1
5	Concluzii	1
	Bibliografie	1
A	Codul sursă al programului	1

1	Introducere
2	Formularea problemei
3	Metoda de rezolvare
4	Rezultate obținute
5	Concluzii
A	Codul sursă al programului

U.P.B. Laboratorul de Metode Numerice — LMN

Studiul unui circuit RC serie în regim tranzitoriu

Ion Popescu

ipopescu@lmm.pub.ro, Grupa 161M, Facultatea Electrotehnică

19 noiembrie 1997

Instructor: Ș.l. I. Munteanu

Cuprins

1 Ecuatiile de funcționare ale rezistorului și condensatorului	1
1.1 Ecuatia de funcționare a rezistorului ideal	1
1.2 Ecuatia de funcționare a condensatorului ideal	1
Bibliografie	2

1 Ecuatiile de funcționare ale rezistorului și condensatorului

1.1 Ecuatia de funcționare a rezistorului ideal

Fie un fir conductor C dintr-un material izotrop, liniar și omogen, de conductivitate σ , parcurs de curent cu intensitatea i . Firul are lungimea l și aria secțiunii A , cu l mult mai mare decât dimensiunile secțiunii.

Tensiunea electrică u de-a lungul firului și intensitatea i prin orice secțiune S sunt prin definiție:

$$u \stackrel{D}{=} \int_C \mathbf{E} \cdot d\mathbf{s}; \quad i \stackrel{D}{=} \int_S \mathbf{J} \cdot n dA \quad (1)$$

Conform legii conducției electrice, în orice punct din conductor

$$\mathbf{E} = \rho \mathbf{J}, \quad (2)$$

cu $\rho = 1/\sigma$. Dacă se consideră densitatea de curent uniform repartizată pe orice secțiune ($J = i/A$), iar \mathbf{E} (\mathbf{J}) orientat de-a lungul conductorului, atunci relațiile (1) și (2) conduc la:

$$u = \int_C \rho \mathbf{J} \cdot d\mathbf{s} = \rho J l = \frac{\rho l}{A} i = Ri. \quad (3)$$

1.2 Ecuatia de funcționare a condensatorului ideal

Ecuatia de funcționare a condensatorului ideal poate fi dedusă folosind teorema conservării sarcinii electrice și proporționalitatea dintre sarcina electrică q a unei armături și tensiunea electrică u dintre armături. Această ecuație are forma [1]:

$$i = C \frac{du}{dt}, \quad (4)$$

unde C este capacitatea condensatorului, care nu depinde de sarcină și de tensiunea la borne, ci numai de geometria condensatorului și de permitivitatea ε a acestuia.

Bibliografie

- [1] C. I. Mocanu. *Teoria câmpului electromagnetic*. Editura Didactică și Pedagogică, București, 1981.

EXERCITIUL 6

Adăugați o nouă secțiune la referatul de laborator, care să reproducă modelul de mai jos. Completați formulele corecte ale ecuațiilor (5) și (6).

2 Ecuația tensiunii la bornele condensatorului

Fie un circuit RC serie, alimentat cu o sursă de tensiune electromotoare constantă E . Un contact K înseriat cu circuitul se închide la momentul $t = 0$, când tensiunea la bornele condensatorului este u_{C_0} .

A doua ecuație Kirchhoff are forma:

$$\sum_{k \in [o]} u_k = 0; \quad \text{sau} \quad u_R + u_C = E.$$

Folosind ecuațiile de funcționare (3) și (4) ale rezistorului și condensatorului, relația devine:

$$\text{Completați Dvs.} \tag{5}$$

Soluția ecuației diferențiale (5) are forma:

$$\text{Completați Dvs.} \tag{6}$$

EXERCITIUL 7

Adăugați încă o secțiune la referatul de laborator, numită **Comparație între circuitele RC serie și RL serie**, care să conțină următorul tabel, centrat orizontal și numerotat automat de L^AT_EX:

Tip	Variabilă de stare	Ecuație
R-L	$i (i_L)$	$L \frac{di}{dt} + Ri - E = 0$
R-C	$u (u_C)$	$RC \frac{du}{dt} + u - E = 0$

Tabelul 1: Comparație

Indicație. Includeți un cadru `tabular` într-un cadru `table`.

EXERCITIUL 8

Examinați documentul L^AT_EX de pe pagina 116.

Cum se pot realiza: • mărirea marginii din dreapta a paginii; • litera Σ de dimensiune mare; • alinierea la stânga a textului în coloana I a tabelului; • linia verticală lungă în ecuație; • indicele S_N din ecuație?

Ecuatia Poisson generalizata

Nume Autor

19 octombrie 1997

Cuprins

1 Formularea problemei	1
1.1 Ecuatia de rezolvat	1
2 Conditii de frontiera	1

1 Formularea problemei

1.1 Ecuatia de rezolvat

Ne propunem sa rezolvam o ecuatie de tip Poisson generalizata, care are forma:

$$\operatorname{div}(K \operatorname{grad} V) = f \quad (1)$$

in cazul unor domenii bidimensionale, intr-un domeniu de studiu D marginit de frontiera Σ .

Ecuatia (1) acopera o categorie larga de probleme de interes in inginerie. In tabelul de mai jos sunt prezentate semnificatia necunoscutei V precum si expresiile pe care le au constanta de material K si sursa F in cateva cazuri de interes:

Regimul	Necunoscuta V	Constanta de material K	Sursa f
Electrostatic	V	ε	$-\rho_v + \operatorname{div} \mathbf{P}_p$
Electrocinetice	V	σ	$\operatorname{div} \mathbf{J}_i$

2 Conditii de frontiera

Vom considera ca pe portiunea S_N a frontierei domeniului se cunosc valorile derivatei dupa normala a potentialului:

$$\left. \frac{dV}{dn} \right|_{S_N} = f_N.$$

⋮

Exercițiul următor este facultativ.

EXERCITIUL 9

Adăugați un index la referatul de laborator.

Pentru aceasta:

- Adăugați `makeidx` în lista de opțiuni ale comenzii `\documentstyle`.
- Adăugați comanda


```
\makeindex
```

 în fișierul `.tex`, înainte de `\begin{document}`.
- Adăugați comanda


```
\printindex
```

 în fișier, în locul unde doriți să apară indexul (de obicei la sfârșitul documentului, după bibliografie).
- Marcați în textul documentului intrările din index, după modelele:


```
. . . densitatea de curent \index{Densitate de curent}
```

 (cuvintele *Densitate de curent* vor apărea în index, urmate numărul paginii unde sunt plasate cuvintele `densitatea de curent`).


```
. . . Această ecuație are forma \index{Ecuații de funcționare!condensator}
```

 (astfel se definește un index pe două nivele, cuvântul *condensator* va apărea ca nivel inferior al intrării din index *Ecuații de funcționare*).
- Procesați fișierul pentru a obține forma sa imprimabilă:

```
latex rc
makeindex rc
latex rc
dvips -t a4 -o rc.ps rc
```

Observații.

Realizarea indexului este una din **cele mai anevoioase** activități în timpul pregătirii unui document. Este bine să vă ușurați munca, scriind câteva macroui speciale \LaTeX și `vi` (sau `emacs`).

De exemplu, indexul cărții de față a fost realizat astfel:

1. Au fost definite **trei noi comenzi** \LaTeX (preluate din [53], pag. 352):

- `\Index`, care permite introducerea unui text atât în document cât și în index; în acest fel se evită tastarea de două ori a aceluiași text;
- `\Indextt`, care funcționează la fel ca `\Index`, dar fontul textului este Courier (`\tt`) atât în document cât și în index;
- `\Com`, care permite introducerea în document și în index a numelui unei comenzi (care începe cu caracterul `\`).

Noile comenzi \LaTeX au fost definite astfel:

```
\newcommand{\Index}[1]{#1\index{#1}}
\newcommand{\Indextt}[1]{\tt #1\index{#1@\tt{#1}}}}
\newcommand{\bs}{\symbol{'134}}
\newcommand{\Com}[1]{\tt {\bs#1}\index{#1@\tt {\bs#1}}}}
```

2. Au fost adăugate **două noi macrouri** în fișierul de configurare a editorului `vi` (simbolul `↑` semnifică tasta `<Ctrl>`):

- `↑a` – aplică macroul \LaTeX `\Index` unui cuvânt care trebuie să apară în index; astfel, dacă inițial cuvântul se afla în fișierul `.tex` sub forma

`cuvant`

după ce se plasează cursorul pe începutul său și se dă comanda `↑a`, el se modifică astfel:

`\Index{cuvant}`

- `↑z` – aplică macroul \LaTeX `\Indextt` unui cuvânt.

Noile macrouri `vi` au fost adăugate în fișierul `.exerc` sub forma:

```
map ↑v↑a i<Bslash>Index↑v↑[ea↑v↑[
map ↑v↑a i<Bslash>Indextt↑v↑[ea↑v↑[
```

(`↑v` permite introducerea ca atare în fișier a unui caracter de control, cum este `↑a`, iar `↑[` are același efect ca și tasta `<ESC>`.)

3. A fost definit **un nou fișier de stil pentru index** (inspirat tot din [53]), numit `mystyle.ist`. (Extensia `.ist` este cea implicită pentru fișierele de stil al indexului.)

Structura fișierului este următoarea:

```
% Fisier de stil pentru MakeIndex -- mystyle.ist
%--- Litera inaintea fiecarui grup de intrari din index
heading_prefix "{\bfseries\hfil " % Insereaza in fata literei
heading_suffix "\hfil}\nopagebreak\n" % Adauga dupa litera
headings_flag 1 % Pune antet (litere mari)
%--- Delimitatori (linie punctata)
delim_0 "\dotfill "
delim_1 "\dotfill "
delim_2 "\dotfill "
```

4. Pentru a folosi noul fișier de stil, comanda de generare a indexului a fost (numele fișierului `.tex` fiind `carte.tex`):

```
makeindex -s mystyle.ist carte
```


Capitolul 5

Grafice și figuri

— Gnuplot, XFig, XPaint, XV —

Spre deosebire de literatura beletristică, orientată aproape exclusiv pe text, literatura științifică este caracterizată prin bogăția de reprezentări grafice pe care le conține. Acestea pot fi clasificate în:

- **grafice**
 - de una sau mai multe funcții de o variabilă
 - funcții de două variabile, suprafețe tridimensionale (3D)
 - alte tipuri de reprezentări ale datelor științifice cum sunt: contururi echipotențiale, spectre ale liniilor de câmp, etc.
- **desene și imagini de sinteză**
 - scheme bazate pe simboluri convenționale – abstracte (circuite, diagrame de stare, organigrame, scheme logice sau de fluentă, etc.)
 - desene ale unor obiecte (desene tehnice sau artistice, cu reprezentări 2D sau 3D, cu sau fără efecte de realism de tip lumini–umbre)
 - picturi (reprezentări concrete sau abstracte în care culoarea sau textura primează față de linii)
- **fotografii** care reprezintă imagini reale.

Imaginile permit pe de o parte vizualizarea datelor științifice obținute în urma experimentelor sau din calcul și pe de altă parte reprezentarea obiectelor reale sau virtuale. Utilizarea calculatorului în activitatea științifică presupune, în consecință, folosirea unor programe care să permită manipularea tuturor acestor tipuri de reprezentări grafice [42]. Aceste pachete au următoarele funcții tipice:

- **crearea**
- **memorarea**
- **editarea:**
 - transformarea (măritori, micșorări, translatarea, rotire)
 - ștergerea unor părți sau elemente
 - copierea unor elemente, eventual repetată
 - inserarea și suprapunerea (inclusiv adnotarea)
 - modificarea culorilor sau a contrastului
- **inserarea în documente sau producții**
 - vizualizare pe ecran de calculator sau TV
 - imprimare pe hârtie sau pe alt suport (de ex. transparent sau film).

În general, cantitatea de informație necesară reprezentării unei imagini este mult mai mare decât cea necesară textului. În consecință, memorarea imaginilor presupune adoptarea unor tehnici eficiente, care să conducă la compresia datelor și obținerea unor fișiere de dimensiuni cât mai mici.

Se deosebesc trei categorii fundamentale de astfel de tehnici:

- reprezentarea **vectorială**, specifică desenelor, în care se memorează secvențe de puncte, linii, curbe, texte, etc.;
- reprezentarea pixel (“**bitmaps**”), specifică picturilor și fotografiilor, în care se memorează culoarea (sau nuanța de gri) a fiecărei celule numită pixel din care este alcătuit rastrul imaginii;
- reprezentarea metafișier (“**metafile**”), specifică imaginilor complicate, numite *scene* și care presupune utilizarea tehnicilor vectoriale și bitmap în același fișier.

Diversitatea tipurilor de abordare și a pachetelor grafice a condus la apariția unei multitudini de formate pentru fișierele grafice, dintre care cele mai importante sunt [67]:

Format	Tip	Extensie	Inițiator
DXF – Drawing eXchange Format	vectorial	ADI	AutoDesk/AutoCAD
FIG – Figures	vectorial	FIG	XFig
HPGL – Hewlett Packard Graphic Language	vectorial	HPG	Hewlett Packard
GIF – Graphic Interchange Format	bitmap	GIF	Compuserve

TIFF	-Tag Image File Format	bitmap	TIF	Aldus
PCX	-PC Paintbrush Format	bitmap	PCX	MS Paint
BMP	-Windows Bitmap	bitmap	BMP	MS Windows
XBM	-Xwindow bitmap - mono	bitmap	XBM	Xwindow
XPM	-Xwindow Pixel Map - color	bitmap	XPM	Xwindow
PBM	-Portable Bitmap cu variantele PGM (gri), PPM (color)	bitmap	PBM, PGM, PPM	pbmplus
JPEG		bitmap	JPG	Joint Photographic Experts Group
PostScript		meta	PS, EPS	Adobe
PDF	-Portable Document Format	meta	PDF	Adobe
WMF	-Windows Meta Format	meta	WMF	MS Windows
POV	-Ray	meta	POV	Persistence of Vision Raytracer
VRML	-Virtual Reality Markup Language	meta	VRM	Virtual Reality

Diversitatea formatelor grafice face extrem de utilă funcția de **conversie a formatelor**, realizată de multe programe prin operații de import/export fișiere în formatul dorit. Operația de conversie nu dă întotdeauna rezultate satisfăcătoare, mai ales atunci când se modifică tipul de reprezentare, motiv pentru care se recomandă memorarea imaginilor într-un format grafic corespunzător: vectorial pentru grafice, desene și scheme și respectiv bitmap pentru picturi și fotografii.

Dintre pachetele de programe care permit generarea și prelucrarea imaginilor, următoarele sunt disponibile în LMN:

Nume program	Funcționalitate	Formate de fișiere acceptate	Citare bibliografică
gnuplot	permite realizarea de grafice 2D și 3D de funcții	DXF, FIG, HPGL, PBM și EPS	[7]
SciLab	permite realizarea de grafice 2D și 3D și vizualizări câmpuri 2D	FIG, PS	[22]
XFig	permite crearea și editarea de scheme și desene	FIG, EPS, XBM,	[29]

Nume program	Funcționalitate	Formate de fișiere acceptate	Citare bibliografică
AutoCAD	permite crearea și editarea de desene	DXF	[39]
mnm - Midnight Modeller	permite realizarea desenelor 3D	DXF, POV	[13]
Scen - Scene Editor	permite realizarea de scene realiste	VRML	[20]
gs - ghostscript	permite conversia din PS sau PDF în diverse formate bitmap	PS, PDF/TIFF, BMP, PCX, PBM, etc.	[5]
gv - ghostview	permite vizualizarea fișierelor PS sau PDF	PS, PDF	[3]
xv	permite capturi de ecrane, transformări, modificări și conversii de fișiere bitmap	GIF, JPEG, TIFF, PBM, PGM, PPM, XPM, PM, PCX și EPS (la ieșire)	[31]
XPaint	permite editarea picturilor reprezentate bitmap	XBM, XPM, PPM, GIF, TIFF și EPS (numai la ieșire)	[30]
GIMP - GNU Image Manipulation Program	program complex de editare grafică de tip Photoshop ce permite compoziția și retușarea fotografică	GIF, JPEG, PCX, TIFF, XPM și PS	[4]
Star Draw Star Image	programe de editare grafică sub Xwindow ce emulează MSDraw și MSPaint de sub Windows		[23]
Presto!	program sub MS Windows ce permite scanarea imaginilor	TIFF, JPEG, PCX, BMP	

Obiectivele capitolului constau în:

- familiarizarea cu principalele concepte ale editării grafice: formate de fișiere grafice, tipuri de imagini, obiecte grafice și transformări ale acestora, etc.
- căpătarea deprinderilor minimale necesare realizării și modificării de grafice, scheme, desene sau fotografii și inserarea lor ca figuri în documente.

5.1 XFig

5.1.1 Apelare

- În laboratorul EB212: `xfig <↵ >`
- În laboratorul EB213: cu butonul din stânga al mouse-ului - Graphics - Xfig sau cu comanda: `/comun/bin/GetXFig <↵ >`

În sala EB213 este instalat pe mașina `lmnsun` dar poate fi rulat și de pe alte mașini.

5.1.2 Obiecte XFig

XFig lucrează cu *obiecte grafice*, care pot fi obiecte *primitive* sau *compuse*.

Obiectele primitive

- arc
- cerc
- spline închis
- elipsă
- linie poligonală
- poligon
- fișier EPS (Encapsulated PostScript)
- dreptunghi
- dreptunghi cu colturile rotunjite
- spline
- text

Obiecte compuse

- combinații de obiecte primitive;
obiectele componente nu pot fi modificate individual.

Operații cu obiecte

- primitive:
 - mutare, rotire, reflectare (vertical, orizontal), scalare, copiere, aliniere (în cadrul unui obiect compus), ștergereExcepții: obiectele text nu pot fi reflectate; dacă sunt rotite, acest lucru nu se vede pe ecran dar, dacă se salvează figura în format PostScript (vezi secțiunea 5.1.4), textul va apărea rotit.
- Observație: Datorită unui bug existent, se pare, în unele versiuni `xfig`, comanda de rotire a textului poate produce setarea în mod eronat a dimensiunilor figurii la valori enorme, ceea ce face imposibilă includerea figurii în documente. Evitați această manevră, pe cât posibil.
- compuse:
 - mutare, rotire, reflectare, scalare, copiere, ștergereExcepții: dacă conțin dreptunghiuri, nu se pot roti decât cu multipli de 90 grade.

5.1.3 Regiuni ale ecranului

Ecranul programului XFig, prezentat în figura 5.1, conține următoarele regiuni:

- regiunea de comandă (meniuri)
- rigle (sus, lateral)
- butoane pentru desenare
- butoane pentru editare
- indicatorul funcțiilor mouse-ului
- zona desenului propriu-zis (“canava”)
- zona de setare a unor opțiuni: linii, culori, etc.

Regiunea de comandă conține meniurile:

Quit Undo Redraw Paste File Export Print

Zona de setare opțiuni (“indicator”) permite efectuarea operațiilor:

- zoom (lupă);

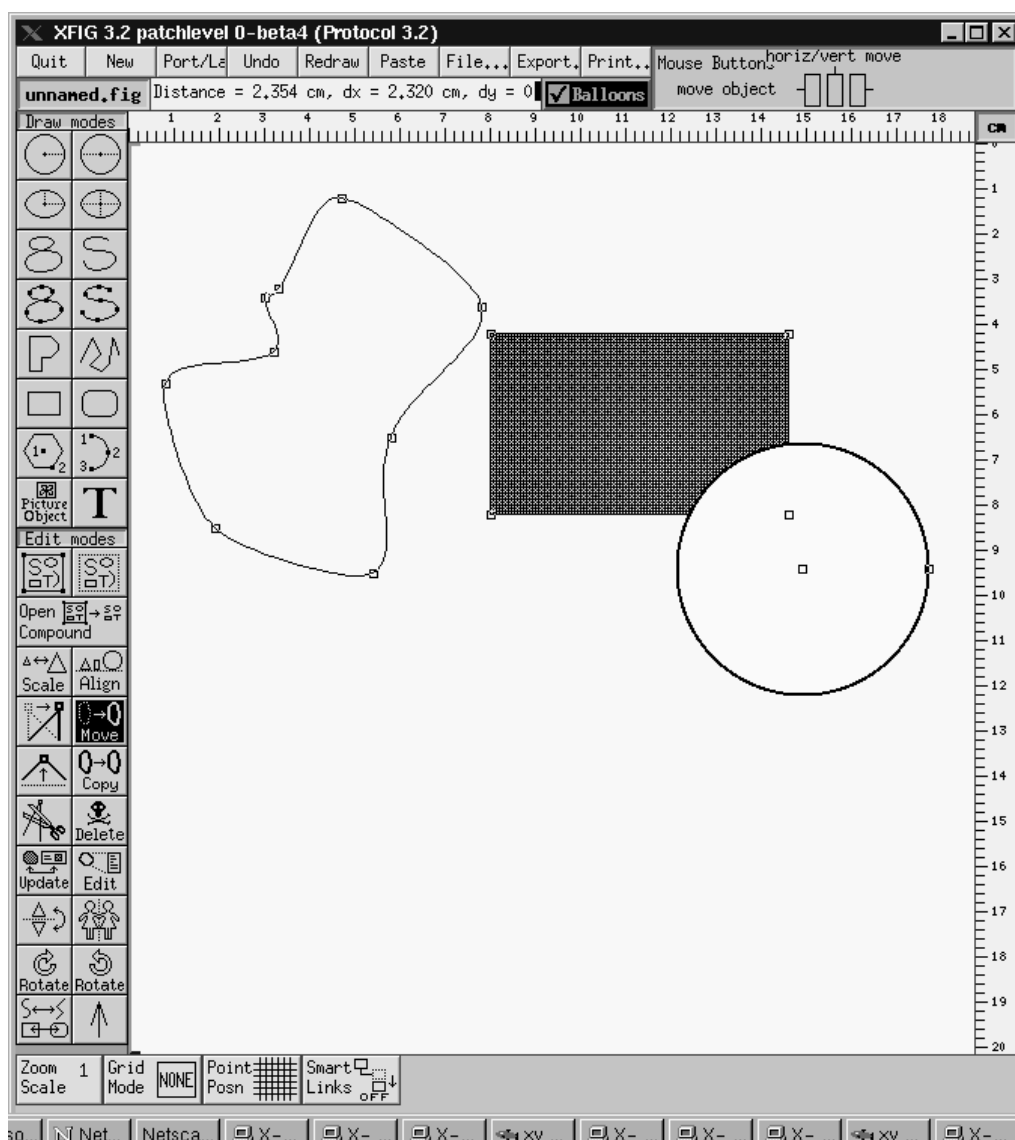


Fig. 5.1: Ecranul programului XFig

- grid (afișarea sau nu a unor linii de ghidare);
- coarseness (finețe);
- rotation angle (unghi de rotație).

Zona de indicare a funcțiilor mouse-ului:

- este plasată în zona din dreapta sus a ecranului;
- sunt afișate:
 - în stânga: acțiuni posibile pentru butonul din stânga;

deasupra: acțiuni posibile pentru butonul din mijloc;

în dreapta: acțiuni posibile pentru butonul din dreapta.

5.1.4 Salvarea desenului în alte formate

XFig oferă posibilitatea salvării desenului și în alte formate decât formatul său propriu. Aceasta este o opțiune extrem de utilă, care permite utilizarea desenului cu alt program de grafică, sau permite includerea acestuia într-un document.

La salvarea unei figuri, se crează un fișier în format XFig, cu extensia implicită `.fig`. Salvarea în alt format se face selectând opțiunea **Export** din regiunea de comandă. Se pot seta:

- așezarea paginii (Portrait, Landscape);
- formatul fișierului de ieșire (Encapsulated PostScript, de exemplu, este util pentru includerea figurii într-un fișier \LaTeX ; opțiunea `Latex` funcționează bine numai pentru forme simple);
- numele noului fișier;
- directorul în care va fi creat noul fișier.

5.2 Gnuplot

Funcționează pe toate mașinile din LMN, cu excepția `lmndec`. În sala EB213 este instalat pe masina `sun`.

5.2.1 Apelare

- în sala EB212: din meniurile plasate pe butonul din stânga sau cu comanda
`gnuplot <↵ >`
- în sala EB213: butonul din mijloc al mouse-ului - Mathematics - GNUplot sau
`/comun/bin/GetPlot <↵ >`

5.2.2 Comenzi utile pentru început

`help` – lista de comenzi `gnuplot`;

`quit` sau `exit` – ieșirea din program.

5.2.3 Manualul pentru `gnuplot`

- este disponibil în sala EB212:
`man gnuplot`

5.2.4 Ce poate face `gnuplot`

- **Trasare de grafice** pentru una sau mai multe funcții simultan, funcții care pot fi:
 - definite analitic în stilul limbajului C, cu ajutorul unor: operatori C, funcții din biblioteca C, operatori specifici `gnuplot`
 - definite prin date, prin citirea unor fișiere ASCII de date
- **Alte facilități:**
 - definirea intervalelor de plotare pe Ox și pe Oy
 - scalarea automată a axelor
 - marcarea axelor în diferite moduri
 - etichete pentru axe
 - funcții și variabile definite de utilizator
 - citirea / scrierea unui fișier de comenzi `gnuplot`

- redirectarea ieșirii
- calcule în complex
- setare opțiuni XWindow

5.2.5 Comenzi gnuplot

După lansarea programului, se afișează un antet cu informații despre program, se intră într-o fereastră dedicată, apare promptul `gnuplot>` și se pot lansa comenzi `gnuplot`.

Comenzile au următoarele caracteristici:

- sunt “case-sensitive”
- pot fi abreviate
- pot apărea mai multe comenzi pe o linie, separate prin `;`
- șirurile de caractere din cadrul unei comenzi sunt incluse între `"` sau ```
- o comandă lungă se poate extinde pe mai multe rânduri, punând câte un `\` la sfârșitul unui rând

În cele ce urmează sunt descrise principalele comenzi `gnuplot`. Se vor folosi următoarele convenții:

- între acolade `{ }`: parametri opționali
- cu litere drepte: cuvintele-cheie `gnuplot`
- cu litere italice: valori sau nume introduse de utilizator.

5.2.6 Comenzi de bază: `plot`, `splot`

Comenzile `plot` și `splot` permit realizarea graficelor 2D, respectiv 3D.

Structura lor:

- Grafic funcție definită analitic
`plot { domeniu } funcție {titlu} {stil}`
- Grafic funcție definită prin date numerice
`plot {domeniu} {"fișier_date"} {using ...} {titlu} {stil}`

Principalul parametru este fie funcția, fie numele unui fișier de date între ghilimele, fie orice combinație a lor. **Virgula** separă grupuri de parametri corespunzători unor funcții diferite.

Ex: `plot sin(x)`

`plot sin(x), cos(x)`

La rândul său, o funcție poate fi:

- o expresie matematică
- o pereche de expresii matematice pentru funcții parametrice 2D
- un triplet de expresii matematice pentru funcții parametrice 3D

În cele ce urmează sunt descriși parametrii și opțiunile comenzilor `plot` și `splot`.

Parametrul *fișier_date*

Fișierele de date au următoarea structură:

- Conțin câte o linie pentru fiecare punct de afișat:
 - 2 valori = x, y (2D)
 - 3 valori = x, y, z (3D)
 - opțional pentru grafice de erori:
 - * încă o valoare y_{delta} = eroarea absolută a lui y
 - * încă două valori $y_{\text{low}}, y_{\text{high}}$ = limitele inferioară și superioară ale lui y
- Valorile sunt separate în fișier prin blancuri sau tab-uri. La grafice 2D coloana x poate lipsi, la 3D coloanele x și y pot lipsi; în aceste cazuri, li se dau automat valori 0, 1, 2, ...
- O linie liberă în fișier va produce o întrerupere în grafic!

Opțiunea `using`

- Opțiunea `using` este folosită numai dacă graficul se face prin citirea unui fișier de date numerice.
- Poate avea ca parametru:

`ycol`

```

xcol:ycol
xcol:ycol:ydelta
xcol:ycol:ylo:yhigh
xcol:ycol:ylo:yhigh:boxwidth

```

În cazul comenzii `plot` mai apare și `zcol`.

`col` reprezintă coloana din fișierul de date care va fi considerată drept coloana x , y , z , eroare (`ydelta`), margini inferioară / superioară, dimensiunea “cutiei” pentru tipul de grafic `boxes`.

Excepții:

- în cazul 2D, dacă apare `<ycol>` singur, se consideră că prima coloană este coloana x -ilor
- în cazul 3D, dacă apare `<zcol>` singur, nu se va citi decât acea coloană
- dacă `xcol` este 0, se va plota y în funcție de numărul său de ordine 0, 1, 2, ...

Opțiunea [domeniu]

- Are sintaxa:

```
[ {var = } {xmin: xmax} ] {[ymin: ymax]}
```

unde `var` este un alt nume pentru variabila independentă.

Când se utilizează `ranges`, scalarea automată este dezactivată. Se poate utiliza

```
[ ] (domeniu nul)
```

pentru a dezactiva scalarea automată.

Ex: `plot sin(x)` - domeniul curent

```
plot [-10:30] sin(x) - x între -10 și 30
```

```
plot [t=-10:30] sin (t) - alt nume pentru variabila independentă
```

```
plot [-pi:pi] [-3:3] tan(x), 1/x
```

```
plot [:200] [-pi:] exp(sin(x)) - nu se precizează decât una din limite
```

Opțiunea stil

- Are sintaxa:

```
with <stil> {<tip linie> <tip punct>},
```

unde `<stil>` poate fi (figura 5.2):

- lines
- points

- linespoints
- impulses
- dots
- errorbars
- steps
- boxes
- boxerrorbars
- Stilul implicit este:
 - `lines` – la grafice de funcții definite analitic
 - `points` – la grafice de funcții definite prin date

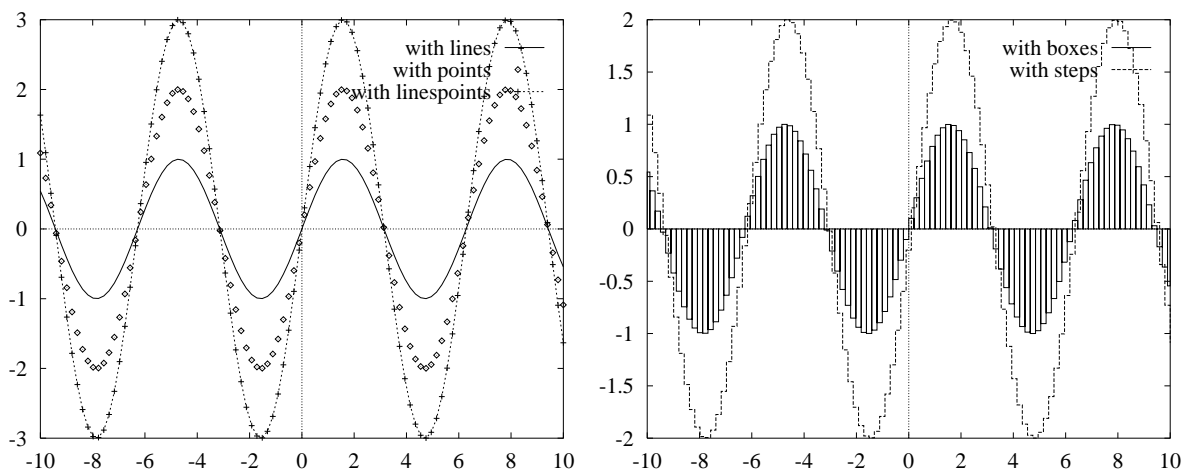


Fig. 5.2: Diferite stiluri de grafice gnuplot

Opțiunea title

- are rolul de a schimba “eticheta” unui grafic.

Ex: `plot sin(x) title "sinus"`

5.2.7 Alte comenzi utile

- comenzi Unix:

`cd`

`pwd`

``...`` – substituirea cu rezultatul comenzii aflate între apostroafele inverse

- lista funcțiilor de tip C disponibile se poate obține cu:

`help expressions` și selectare `functions`

- lista operatorilor disponibili se poate obține cu:

`help expressions` și selectare `operators`

- lista comenzilor editorului orientat pe linie al `gnuplot`:

`help line-editing`

- `save` – salvează în fișier:

- funcțiile definite de utilizator
- variabilele
- opțiunile setate
- ultima comandă `plot` sau `splot`

Ex: `save set var functions "work"`

Salvează în fișierul numit `work` opțiunile setate, variabilele și funcțiile definite de utilizator.

- `load` – citește de pe disc un fișier care conține comenzi `gnuplot`.

Fișierul poate fi unul creat printr-o comandă `save` anterioară, sau scris de utilizator. Efectul comenzii este citirea comenzilor din fișier, linie cu linie, ca și cum ar fi introduse de la consolă.

Ex: `load "work"`

- lista de erori `gnuplot` cunoscute:

`help bugs`

- numere complexe: au forma $\{re, im\}$

Ex: $\{1, 2\}$ pentru $1 + 2j$

Acoladele sunt obligatorii.

- `pause` – pauză, eventual cu afișarea unui mesaj

Util în fișierele de comenzi. Sintaxa:

`pause timp ["mesaj"]`

unde `timp` = constantă întreagă; valori particulare:

-1: așteaptă CR

0: nu așteaptă de loc (afișează mesajul și continuă)

>0: așteaptă numărul respectiv de secunde

- `reread` – citește de la început un fișier de comenzi. Astfel se pot crea bucle infinite.

- `set` – multe lucruri:
 - opțiuni de plotare: margine, scalare automată sau nu a axelor, etc.
 - tip axe
 - etichete axe, ... etc.
- `show parametru_set` – afișează valoarea pentru parametrul respectiv
`show all` – afișează toate setările
- fișier util: `$HOME/.gnuplot`
 poate conține comenzi `gnuplot` care să se execute la fiecare lansare în execuție a programului.
- variabile și funcții definite de utilizator:


```
var = const
funct(par1 {, par2 ...}) = expresie
```

 unde numărul de parametri este mai mic sau egal cu 5.
 Ex: `a = 5` # definirea variabilei a
`f(x) = a * sin(x)` # definirea funcției f

5.2.8 Salvarea graficelor în alte formate

Pentru a salva graficele `gnuplot` în format **PostScript** este necesară:

- Setarea tipului de terminal ca fiind “Encapsulated PostScript”
 Ex: `set term postscript eps mono "Times-Roman" 24`
 Definește terminalul de ieșire ca fiind de tip PostScript, cu opțiunile: Encapsulated, monocrom, font Times-Roman cu dimensiunea caracterelor de 24pt.
- Setarea ieșirii comenzilor `plot` spre un fișier:
`set output "fis.eps"`

Pe lângă formatul PostScript, `gnuplot` permite salvarea graficelor în diferite alte formate, printre care: `latex`, `pbm` (convertibil apoi cu ajutorul programului `pbmplus` în format GIFF, PCX, etc.), `table` (tablou de valori ASCII). O listă completă a formatelor se poate obține cu

```
help set term
```

Inițial (la lansarea `gnuplot`) terminalul este de tip X11, iar ieșirea este “STDOUT”. Pentru revenirea la aceste setări, se folosesc comenzile:

```
set term X11
set output
```


5.2.9 Fișiere de comenzi gnuplot

Comenzile `gnuplot` folosite pentru realizarea unui grafic pot fi scrise într-un fișier (se recomandă folosirea unei extensii specifice, de exemplu `.gnu`) și încărcate în `gnuplot` cu ajutorul comenzii `load`.

Comenzile lungi pot fi scrise pe două sau mai multe rânduri, caracterul de continuare fiind `\` (backslash). Pentru comentarii se folosește caracterul `#`.

Exemplu de fișier de comenzi:

```
# Grafic in scara logaritmica, o functie analitica si doua
#         definite prin date in fisierul "grafic.dat"

set logscale y 10
set grid
set function style linesp
set data style linesp
set samples 20      # pentru functii definite analitic
set key 2, 500      # plasarea etichetelor graficelor la x = 2, y = 500
                    # graficul se extinde pe x intre 0.5 si 5,
                    # iar pe y intre 0.01 si 1000
set title "Rezultatele analitice si experimentale"
set xlabel "x [m]"
set ylabel "f(x) [-]"

set term postscript eps mono "Times-Roman" 20
set output "graficx.eps"

f(x) = x**2
plot [0.1:5] \
    f(x) title "Analitic", \
    "grafic.dat" using 1:2 title "Experiment 1", \
    "grafic.dat" using 1:3 title "Experiment 2"

!mv "graficx.eps" "../figuri/"
```

Rezultatul executării acestui fișier de comenzi este prezentat în figura 5.3

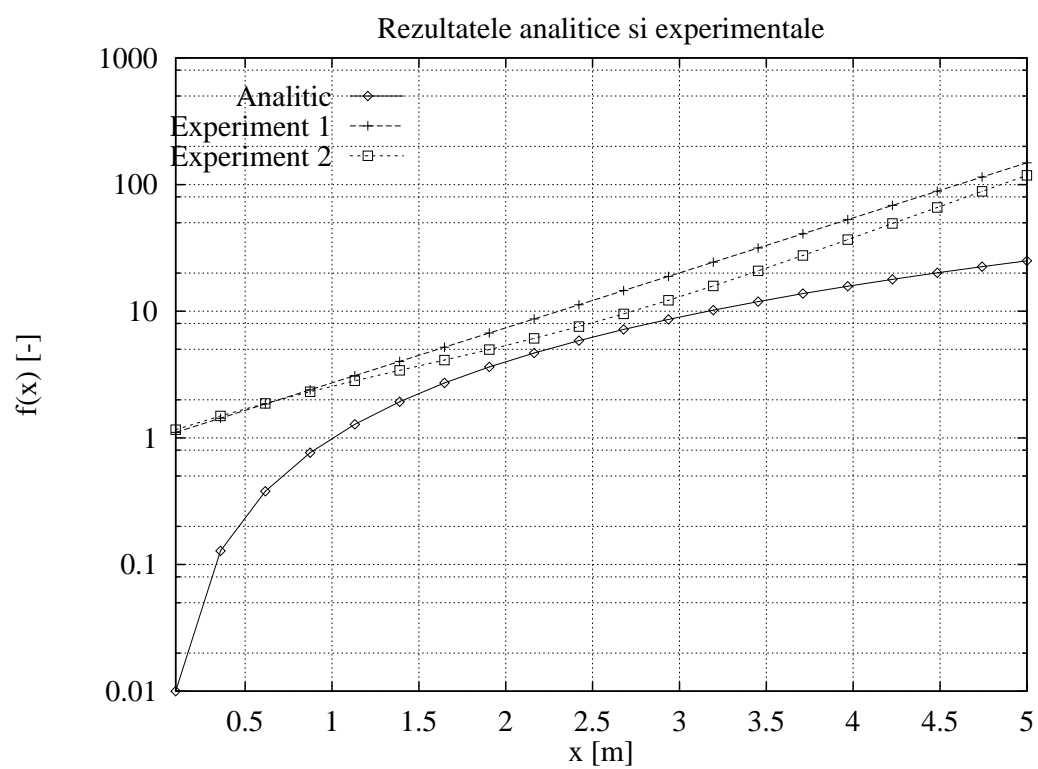


Fig. 5.3: Grafic realizat cu gnuplot

5.3 XPaint

5.3.1 Apelare

- În laboratorul EB212: `xpaint <↵ >`
- În laboratorul EB213: cu butonul din stânga al mouse-ului - Graphics - Xpaint sau cu comanda: `/comun/bin/GetXpaint <↵ >`

5.3.2 Ferestre xpaint

XPaint lucrează cu două ferestre:

- Fereastra de instrumente (toolbox window)
- Fereastra de desenare (painting window)

Fereastra de instrumente permite operații de intrare din fișier, deschiderea uneia sau mai multor ferestre de desenare, precum și selectarea instrumentelor de desenare.

Fereastra de desenare permite realizarea desenului propriu-zis, precum și operații de editare și salvare în fișier a desenului.

5.3.3 Fereastra de instrumente

Aspectul acestei ferestre este prezentat în figura 5.4.



Fig. 5.4: Fereastra de instrumente (toolbox window) a programului XPaint

Fereastra conține două zone:

- Zona de meniuri:
 - **File** – cu opțiunile de meniu: **New canvas** (nou desen cu dimensiuni predefinite); **New with size** (nou desen cu dimensiuni definite de utilizator); **Open**, **Load Clipboard** (deschiderea unui fișier existent); **Quit** (ieșire din XPaint);
 - **Line** – permite selectarea grosimii liniei de desenare;
 - **Font** – permite selectarea fontului și dimensiunii caracterelor
 - **Help** – deschide o fereastră cu informații despre XPaint sau despre modul de utilizare al acestuia.
- Zona de butoane – prin care se selectează instrumentul de desenare. Printre aceste instrumente se numără: desenare punct cu punct, ștergere, desen forme geometrice regulate sau neregulate, colorare, etc.

5.3.4 Fereastra de desenare

Aspectul acestei ferestre este prezentat în figura 5.5.

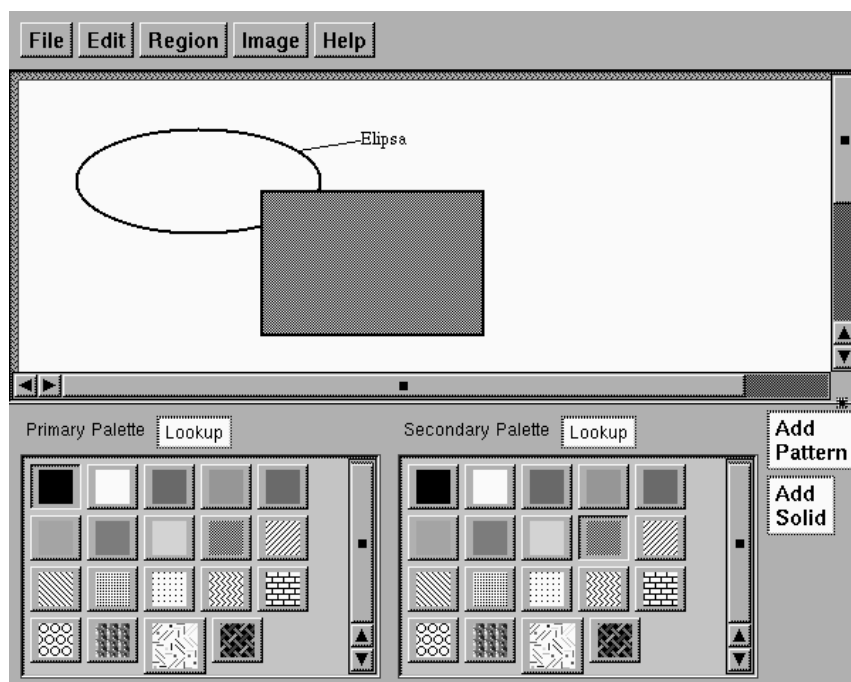


Fig. 5.5: Fereastra de desenare (painting window) a programului XPaint

Fereastra conține trei zone:

- Zona de meniuri: Opțiunile din aceste meniuri permit efectuarea unor operații asupra desenului din fereastra respectivă, precum și salvarea desenului în diferite formate.

- **File** – cu opțiuni pentru: salvare, salvare sub un alt nume, salvarea unei regiuni din desen, salvarea paletei de culori, încărcarea unei palete de culori creată anterior, ieșire din XPaint;
 - **Edit** – cu opțiuni pentru: anularea ultimei comenzi, ștergerea unei zone și memorarea ei în clipboard, copierea în clipboard a unei zone, copierea în desen a conținutului clipboard-ului;
 - **Region** – cu opțiuni pentru: reflectarea și rotirea imaginii, aplicarea unor algoritmi standard de prelucrare a imaginilor (inversarea culorilor, estomparea sau accentuarea contururilor, crearea efectelor de relief sau de pictură în ulei);
 - **Image** – cu opțiuni pentru: mărirea unei zone (lupă), setarea fineței caroiajului imaginar pe care este plasată imaginea, schimbarea culorii fondului, modificarea dimensiunii imaginii;
 - **Help** – afișarea instrucțiunilor de utilizare.
- Zona de desenare propriu-zisă;
 - Zona de selectare a culorii liniilor și respectiv a culorii de umplere a figurilor geometrice.

Atenție: versiunea XPaint instalată în LMN mai are, se pare, unele bug-uri, care se manifestă prin ieșirea bruscă din program atunci când se comandă anumite operații. Vă recomandăm să salvați desenul mai des decât ați face-o atunci când utilizați alte programe.

5.3.5 Salvarea desenului în diferite formate

XPaint oferă posibilitatea salvării desenului în 7 formate: TIFF, PPM, GIF, XBM, XPM, XWD și PostScript (primele 6 formate sunt recunoscute și ca formate de intrare).

Pentru a salva desenul în unul din aceste formate, se folosește meniul **File** al ferestrei de desenare și opțiunea **Save as** a acestui meniu, cu selectarea formatului dorit prin apăsarea butonului corespunzător (figura 5.6).

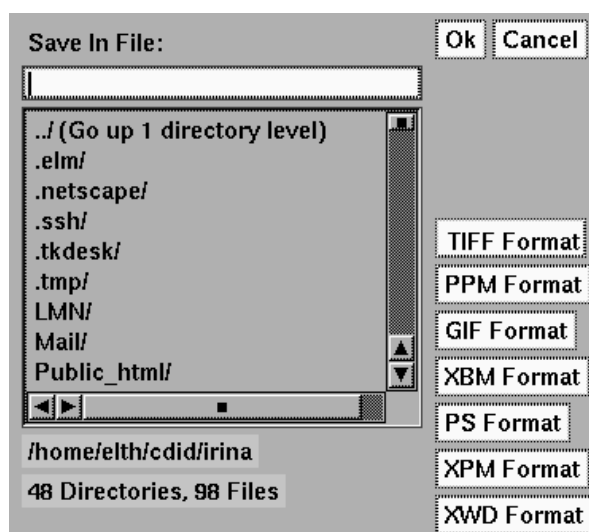


Fig. 5.6: Fereastra `xpaint` de salvare într-un fișier a desenului

5.4 XV

xv este un program interactiv care permite vizualizarea, modificarea și conversia în alt format a imaginilor.

Printre formatele de intrare și de ieșire se numără GIF, JPEG, TIFF, BMP, etc. Programul **xv** poate de asemenea genera fișiere PostScript.

Atenție! Spre deosebire de celelalte programe prezentate până acum, **xv** nu este distribuit gratuit (*freeware*), ci este un produs promoțional (*shareware*), iar pentru folosirea sa profesională trebuie să cumpărați licența de utilizare. Pentru mai multe informații, consultați paginile de Help ale programului.

5.4.1 Apelare

- Cu butonul din stânga al mouse-ului – Graphics – xv
- Cu comanda
`xv <↵ >`

5.4.2 Ferestre xv

Programul **xv** lucrează cu două ferestre:

- Fereastra de imagine – este cea care apare la lansarea **xv**;
- Fereastra de control – poate fi deschisă sau închisă prin click cu butonul din dreapta al mouse-ului în fereastra de imagine.

În fereastra de imagine se află imaginea în curs de vizualizare/modificare. Comenzile de modificare a imaginii sunt disponibile în fereastra de control.

5.4.3 Operațiuni în fereastra de imagine

În această fereastră **xv** sunt posibile trei operații:

- **Apăsând butonul din mijloc al mouse-ului** – Afișarea informațiilor referitoare la imagine: poziția cursorului în pixeli, culorile și saturația pixel-ului curent;
- **Apăsând butonul din dreapta al mouse-ului** – deschiderea / închiderea ferestrei de control;
- **Apăsând butonul din stânga și deplasând mouse-ul** – Selectarea unei porțiuni din imagine, asupra căreia vor acționa prelucrările ulterioare.

5.4.4 Operațiuni în fereastra de control

Fereastra de control **xv** are aspectul prezentat în figura 5.7.

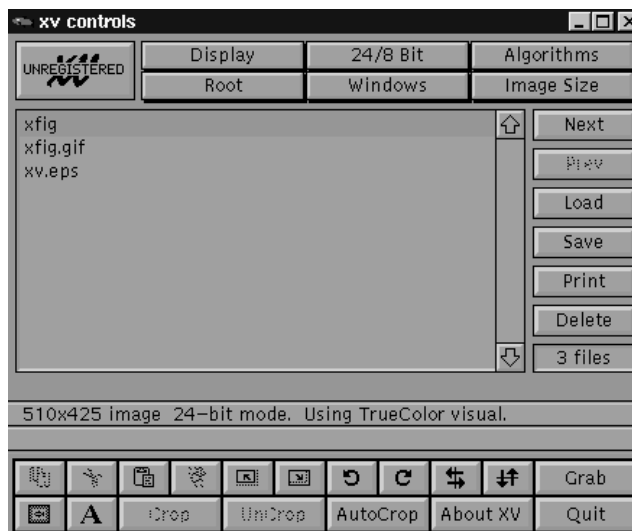


Fig. 5.7: Fereastra de control a programului **xv**

Fereastra de control cuprinde o zonă în care este afișată **lista fișierelor**—imagine deschise sau salvate în sesiunea curentă, precum și **trei zone de butoane și meniuri**.

Zona superioară conține **meniuri** pentru efectuarea unor operații asupra imaginii:

- **Display** – permite schimbarea aparenței imaginii pe ecran, fără modificarea acesteia; meniul conține opțiuni precum Raw (forma inițială); Smooth (forma “netezită”, compensând erorile de rotunjire care intervin la mărirea/micșorarea imaginii); Dithered (culori cât mai apropiate de cele originale); etc.
- **Root** – permite schimbarea fondului întregului ecran
- **24/8 Bit** – trece imaginea din formatul cu 24 biți/pixel în cel cu 8 biți/pixel și invers.
- **Windows** – deschide și alte ferestre, precum: Visual Schnauzer (un manager de fișiere–imagine); Color Editor (editor de culori); Text View (vizualizarea conținutului fișierului imagine, în format ASCII); etc.
- **Algorithms** – permite aplicarea unor algoritmi standard de prelucrare a imaginilor, precum: Blur (estomparea conturilor), Sharpen (accentuarea conturilor), Oil Painting (efect de pictură în ulei), Copy/Clear Rotate (rotirea imaginii), etc.
- **Image size** – permite modificarea dimensiunii imaginii.

Zona din dreapta este dedicată **lucrului cu fișiere** și conține butoanele:

- **Next** – deschide următorul fișier din lista de fișiere ale sesiunii curente;
- **Prev** – deschide precedentul fișier din lista de fișiere ale sesiunii curente;
- **Load** – încarcă un nou fișier de pe disc;
- **Save** – salvează fișierul curent în formatul original sau în alt format;
- **Print** – imprimă un fișier imagine;
- **Delete** – șterge un fișier, fie din lista de fișiere ale sesiunii, fie de pe disc.

Zona inferioară a ferestrei de control conține **butoane** pentru modificarea imaginii. O parte dintre butoane sunt active numai dacă a fost selectată o zonă din imagine. Butoanele din zona inferioară au următoarele funcții:

- **Rândul de sus:**

1. memorarea în clipboard a unei copii a zonei selectate;
2. ștergerea zonei selectate (colorarea ei în culoarea fondului) și copierea în clipboard a zonei șterse;
3. copierea (“Paste”) a imaginii din clipboard; butonul trebuie apăsat de două ori: prima dată pentru a selecta zona în care să se facă copierea, și a doua oară pentru a realiza copierea propriu-zisă;
4. ștergerea zonei selectate;
5. micșorarea întregii imagini cu 10%;
6. mărirea întregii imagini cu 10%;
7. rotirea la stânga cu 90 grade;
8. rotirea la dreapta cu 90 grade;
9. reflectarea imaginii față de axa verticală;
10. reflectarea imaginii față de axa orizontală;
11. “capturarea” unei imagini de pe ecran și afișarea ei în fereastra de imagine, cu variantele: Grab (capturează o fereastră cu butonul din stânga, o zonă dreptunghiulară cu butonul din mijloc); AutoGrab (după un interval prestabilit, capturează fereastra în care se află cursorul mouse-ului)

- **Rândul de jos**

1. adaugă o margine imaginii curente, cu dimensiuni și culoare selectabile de utilizator;
2. adaugă un text;
3. elimină marginea din jurul zonei selectate (Crop);
4. anulează efectul comenzilor Crop și AutoCrop;

5. elimină marginea din jurul zonei de interes a imaginii (AutoCrop);
6. afișează un text despre `xv`;
7. iese din `xv`.

5.5 Exerciții

EXERCITIUL 1

Folosind `gnuplot`, realizați graficul funcției $\sin x/x$, pe intervalul $[-15, 15]$. Adăugați titlu și etichete pentru axe.

EXERCITIUL 2

Realizați cu `gnuplot` graficul funcției $f : [-10, 10] \times [-1, 3] \rightarrow \mathbb{R}$, $f(x, y) = \sin(x) * \cos(y)$, reprezentat în figura 5.8. Adăugați toate elementele care apar în figura 5.8 (titlu, etichete, săgeată, etichete axe).

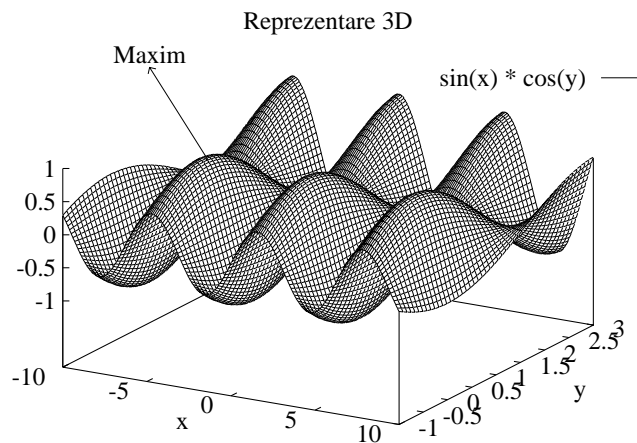


Fig. 5.8: Graficul funcției $\sin(x) * \cos(y)$

Indicație: folosiți parametrii `isosamples`, `xlabel`, `title`, `arrow`, `label`.

EXERCITIUL 3

Completați paragraful 2 al referatului de laborator realizat la exercitiul 6 din capitolul 4 cu încă o ecuație: formula intensității curentului prin circuitul RC serie.

Folosind `gnuplot` reprezentați pe același grafic tensiunea la bornele condensatorului și intensitatea curentului, pentru valori arbitrare ale parametrilor R , C și E și pentru un interval de timp convenabil ales.

Adăugați la paragraful 2 al referatului o figură cu acest grafic, precizând într-o frază valorile parametrilor și făcând o referire la figură. (Ca orice figură dintr-un text științific, și aceasta va fi numerotată și va avea o scurtă explicație.)

EXERCITIUL 4

Realizați cu XFig schema circuitului RC serie descris în referatul de laborator (fișierul `rc.tex`).

Includeți schema în paragraful 2, având grijă ca figura să aibă un număr, un titlu, și să fie citată în textul referatului.

EXERCITIUL 5

Cu ajutorul programului `xv`, realizați o așa-numită “captură de ecran” a ferestrei `xfig` în care ați realizat exercițiul precedent.

Opțional: Editați această imagine cu `xpaint`, indicând prin săgeți și nume câteva din regiunile ecranului XFig prezentate în paragraful 5.1.3.

Încercați diferitele comenzi `xv` (algoritmi, culori, modificare dimensiuni, etc.) și observați efectul acestora.

Când rezultatul este satisfăcător, salvați imaginea în format PostScript, încercând să obțineți un fișier cât mai mic dar și o bună calitate a imaginii (încercați diferite variante: culori, dimensiuni ale imaginii). Atenție, dacă imaginea nu se vede bine pe ecran, nu înseamnă că același lucru se va întâmpla și pe hârtie!

Adăugați o primă anexă la referatul Dvs: **Programul folosit pentru desenarea schemei electrice**, care va conține un scurt comentariu urmat de figura ecranului XFig.

Capitolul 6

Analiza numerică a modelelor matematice — Scilab —

Specific activității de cercetare științifică este faptul că se urmărește caracterizarea cantitativă a sistemelor și fenomenelor studiate. Pentru a realiza acest deziderat, în inginerie se folosește modelarea abstractă, bazată pe următoarele etape:

- **modelarea fizică**, în care se identifică fenomenele esențiale, ipotezele simplificatoare și se stabilesc mărimile fizice caracteristice studiului respectiv;
- **modelarea matematică**, în care relațiile stabilite anterior sunt aduse într-o formă matematică și se formulează corect o problemă care, în esență, presupune rezolvarea unui sistem de ecuații algebrice liniare sau neliniare, diferențiale, integrale sau cu derivate parțiale;
- **modelarea numerică**, urmărește rezolvarea problemei matematice formulate anterior, până la obținerea rezultatelor numerice susceptibile de a fi comparate cu cele experimentale sau determinate pe altă cale.

Deoarece problemele actuale au o complexitate tot mai mare, pentru a putea obține rezultate de nivel înalt în cercetarea științifică și dezvoltarea tehnologică este absolut necesară utilizarea tehnicii moderne de calcul în studiul situațiilor reale, întâlnite în practică. Din acest motiv s-a conturat un nou domeniu al științei intitulat “*Computational Science and Engineering – CSE*”, care complementează abordările teoretice și cele experimentale.

Încă de acum două decenii, cercetătorii s-au confruntat cu dificultatea lucrului în echipe interdisciplinare (care conțin matematicieni și informaticieni) sau cu necesitatea căpătării unor cunoștințe profunde de metode numerice și de programare a calculatoarelor. În acest context au fost create instrumente software specializate (cum sunt *Mathematica*

[81] și *MATLAB* [51]), care permit cercetătorului să se concentreze cu mare productivitate asupra problemei supusă studiului și care ascund detaliile matematice și de programare. Aceste programe sunt sisteme software complexe (pachete polifuncționale integrate) bazate pe un interpretor (asemănător BASIC-ului), care recunoaște comenzi ce permit:

- **calcul numerice**, evaluări de expresii matematice;
- **vizualizarea** datelor, reprezentări grafice de funcții;
- realizarea și execuția unor **programe de calcul** într-un limbaj specializat, de nivel înalt;
- o **interfațare** grafică prietenoasă cu utilizatorul, interfațarea cu alte programe (controlul programelor și proceselor externe, controlul din exterior al proceselor interne, importul și exportul de date care să conducă la realizarea de documente profesionale PostScript sau \TeX).

Unele pachete oferă și funcția de calcul simbolic (implicit în *Mathematica* și opțional în *MATLAB*).

Utilizarea unor astfel de pachete specializate sporește incomparabil productivitatea cercetătorului, deoarece pe de o parte algoritmi matematici folosiți de ele sunt cei mai performanți, iar pe de altă parte, limbajul de programare este simplu (nu necesită declarații de date).

De exemplu, în *MATLAB*:

- obiectul fundamental este matricea de numere complexe, care, dacă are o singură coloană devine vector, iar dacă are un singur element devine scalar, urmând ca prin anularea părții imaginare, el să devină număr real;
- dimensiunile matricelor nu sunt declarate explicit, ci alocarea memoriei se efectuează dinamic pe măsura necesităților, iar operația de eliberare a memoriei (“*garbage collection*”) este tot automată;
- limbajul de programare *MATLAB* oferă toate structurile de control specifice unui limbaj de programare modern;
- expresiile admise de limbaj au mult mai multă putere decât la un limbaj universal: operanzii sunt matrice, iar operatorii pot realiza operația de înmulțire, inversare (sau de pseudoinversă) sau funcții de matrice;
- operațiile simple (calcul aritmetic, reprezentări grafice) se realizează practic instantaneu și cu un efort minim de tastare din partea utilizatorului.

În laboratorul LMN, pe lângă *Mathematica* și *MATLAB*, este instalat un astfel de mediu numit **Scilab** (se citește *silab*), realizat și distribuit gratis de INRIA – *Institut National de Recherche en Informatique et Automatique* [22]. Acest mediu emulează și extinde funcțiile *MATLAB* permițând și manipularea matricelor ale căror elemente sunt:

- șiruri de caractere;
- polinoame cu coeficienți eventual complecși;
- funcții raționale (raport de polinoame);
- liste cu obiecte de tipuri diferite,

cea ce permite, în final, manipularea simbolică a sistemelor liniare în sensul teoriei sistemelor. Programul permite interfațarea cu limbaje ca FORTRAN și C sau cu programe precum MAPLE V [56] pentru realizarea calculelor simbolice și conține și interfețe pentru descrierea grafică și simularea sistemelor dinamice (*SCICOS* — mediu asemănător *SIMULINK*-ului).

Obiectivele capitolului sunt:

- familiarizarea cu funcțiile și conceptele unui mediu de înaltă productivitate de tip “laborator științific” sau “laborator matematic”;
- însușirea deprinderilor necesare analizei numerice a unui sistem simplu (de exemplu un circuit electric) și redactarea unui raport profesional folosind funcțiile “laboratorului științific”.

6.1 Apelare Scilab

- În laboratorul EB212: `scilab <↵ >`
- În laboratorul EB213:
butonul din mijloc - Mathematics - Scilab
sau
`/comun/bin/GetScilab`

În LMN este instalat pe mașinile `lmnhp1`, `lmnhp4` și pe toate mașinile Linux; necesită mediul X11 pentru a putea fi executat.

6.2 Componente Scilab

- Interpretor – permite introducerea în mod interactiv a unor comenzi de la consolă și executarea imediată a acestora
- Biblioteci de macrouri - proceduri Scilab
- Biblioteci de rutine C și FORTRAN - externe programului Scilab, apelabile de acesta

6.3 Interfața grafică a programului Scilab

În figura 6.1 este prezentată interfața grafică a programului Scilab.

Pe lângă zona de comandă, în care utilizatorul poate introduce de la consolă comenzi și instrucțiuni Scilab, interfața mai conține butoane pentru realizarea următoarelor funcții:

- Operații cu fișiere și directoare (schimbarea directorului curent, încărcarea unui fișier conținând un program Scilab, încărcarea unui fișier conținând definițiile unor funcții Scilab, oprirea execuției programului Scilab, etc.)
- Operații de control al execuției (oprirea temporară sau definitivă a comenzii în curs de execuție, continuarea execuției comenzii)
- Lansarea programelor demonstrative
- Operații cu ferestrele grafice (creare, închidere, definirea ferestrei curente)
- Lansarea ferestrei de Help

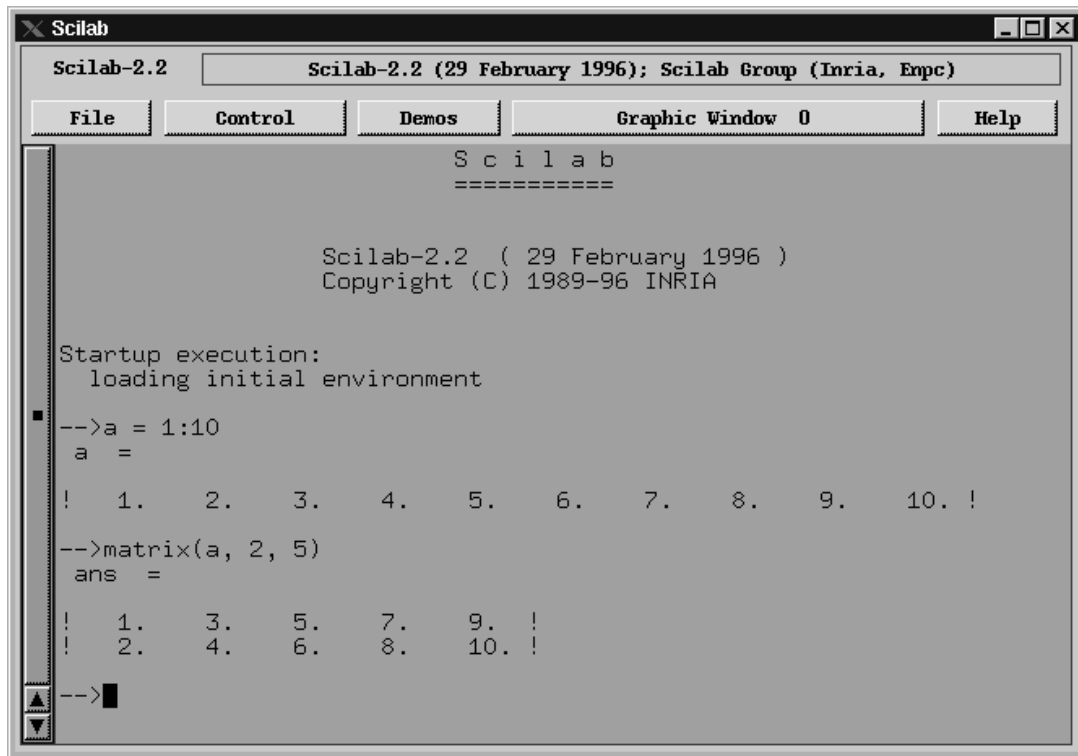


Fig. 6.1: Interfața grafică a programului Scilab

6.4 Câteva noțiuni introductive

6.4.1 Formatul comenzilor Scilab

- Comentarii: tot ceea ce urmează pe o linie după caracterele `//`
- În mod normal, după ce se dă o comandă, se afișează rezultatul imediat după apăsarea tastei `<↵>`. Pentru a se executa comanda fără afișarea rezultatului, se pune `;` (punct și virgulă) la sfârșitul instrucțiunii.
- Se pot da mai multe comenzi pe o linie, separate prin `;`
- Dacă nu se precizează cărei variabile să i se atribuie rezultatul, printr-o instrucțiune de forma:

```
var = expresie
```

ci se invocă doar `expresie`, atunci valoarea acesteia este memorată în variabila implicită `ans`.

- În fișierul de configurare `~/.scilab` se pot scrie comenzi care să se execute de fiecare dată la pornirea Scilab (de exemplu comenzi de tip `getf`, vezi paragraful 6.6.1).

6.4.2 Facilități

- manipulare de matrice cu elemente reale sau complexe de tip:
 - scalari
 - vectori / matrice
 - polinoame
 - șiruri de caractere
 - liste
- calcule simbolice cu polinoame și cu matrice de polinoame
- calcule simbolice cu sisteme liniare
- simulări, optimizări
- interfață cu programe scrise în C și în FORTRAN.

Tipurile de date sunt, așadar, variate, flexibile. Sintaxa comenzilor este naturală și ușor de utilizat. Există un număr mare de primitive standard și se pot adăuga altele noi de către utilizator.

6.4.3 Demonstrație

Selectând cu mouse-ul butonul “Demos” din meniu se lansează un set de programe demonstrative pe care vă recomandăm să le executați pentru a înțelege rapid filozofia Scilab. Sunt vizualizate instrucțiunile Scilab utilizate în demonstrație precum și rezultatele lor.

6.5 Tipuri de date Scilab

6.5.1 Constante

Există constante predefinite. Numele lor incepe cu %:

- `%i` = $\sqrt{-1}$;
- `%pi` = π ;
- `%e` = $e = 2.7182818\dots$;
- `%eps` – precizia masinii, numărul care satisface egalitatea numerică $1 + \%eps = 1$;
- `%inf` – infinit;

- `%nan` = “not a number”

Constante de tip boolean:

- `%t` – true (rezultatul comparației `1 == 1`);
- `%f` – false.

6.5.2 Matrice de scalari

Tipul “matrice” este cel mai important tip de date în Scilab. Orice obiect de tip: scalar, vector, matrice în sens clasic, face parte din clasa “matrice” iar operațiile cu ele se efectuează în mod unitar, ca operații cu matrice. În particular, un scalar este o matrice de dimensiune 1×1 .

1. Scalari

Ex: `a = 2`
`b = 1 + 3* %i // 1 + 3j`
`B = -1 + 3* %i // B și b sunt două variabile diferite`

Se pot aduna, scădea, împărți, ridica la putere (cu operatorul `^`, de exemplu: `x^2`).

2. Vectori

- Incluși între paranteze pătrate,
 - cu blank sau virgulă între elemente: vectori linie
 Ex: `v = [1 2 1+%i]` sau
`v1 = [1, 2, 1+%i]`
 - cu `;` între elemente: vectori coloană
- Operații:
 - `+ - * /` - în măsura în care dimensiunile vectorilor sunt compatibile
 - `'` - transpunere, elementele complexe conjugate prin transpunere
 - `v1 / v2` - echivalent cu `v1 * pinv(v2)`, unde `pinv(v2)` este pseudoinversa (`v2*pinv(v2) = 1`)
 - `.*` - înmulțire element cu element
 - `./` - împărțire element cu element
- Crearea automată a unui vector:
 - `v = val_in : pas : val_fin`
`pas` poate lipsi; în acest caz el este considerat implicit egal cu 1.

3. Matrice

- Definite astfel: elementele de pe o linie sunt separate prin blank sau prin virgulă, iar între linii se pune caracterul ;
Ex: `a = [2 3; 4 5]` // Prima linie: 2, 3; A doua linie: 4, 5
- Operații disponibile:
 - + - * - operații matriceale, dimensiunile operanzilor trebuie să fie compatibile
 - .* ./ - înmulțire, respectiv împărțire element cu element (rezultă o matrice de aceeași dimensiune)
 - b / A - are semnificația: soluția sistemului de ecuații $Ax = b$
- Se pot forma matrice din vectori sau din alte matrice (concatenare).
Ex: `a = 1:10`
`b = [a, a, a]`
`b1 = [a; a; a]`
- Se pot reorganiza elementele existente într-o matrice astfel încât să se modifice numărul de linii / coloane.
Ex: `c = matrix (b, 3, 4)` // o nouă matrice cu 3 linii și 4 coloane
- Transpunere: `a'`
- Determinant: `det(a)`

6.5.3 Matrice de șiruri de caractere

- Șir de caractere: inclus între apostroafe (')
- Operații cu șiruri:
 - `string(nr)` - valoarea numerică `nr` scrisă ca șir
 - `addf(sir1, sir2)` - crează șirul `sir1+sir2`
 - `sir1 + sir2` - concatenează șirurile `sir1` și `sir2`
 - ..., etc.
- Matricele de șiruri de caractere permit:
 - crearea și manipularea funcțiilor utilizator
 - manipularea simbolică a unor obiecte matematice

Exemplu

```
a = 2; b = 1; c = 3; d = 5
```

```
b1 = 1; b2 = 2;
```

```
A = ['a' 'b'; 'c' 'd']
```

```

B = ['b1'; 'b2']
evstr(A)           // evalueaza 'in a, b, c, d
                  // matricea de siruri A
At = trianfml(A)   // triangularizeaza matricea A
                  // -- formula, nu numeric
[x] = trisolve(At, B) // rezolva simbolic
                  // sistemul At x = B

```

6.5.4 Matrice de polinoame și de funcții raționale

- Polinoamele se definesc:

– specificând rădăcinile:

```

polinom = poly([listă de rădăcini], 'var', 'roots'),
unde 'roots' poate lipsi, fiind implicit.

```

Ex: `p1 = poly([1 2], 's')`

– specificând coeficienții: `polinom = poly([listă de coeficienți], 'var', 'coeff')`.

– în forma clasică, după definirea anterioară a variabilei polinomului, ca fiind un alt polinom cu o singură rădăcină zero:

```

z = poly(0, 'z')
p2 = 1 + 2*z + 5*z^2 + ...

```

- Funcție rațională: raportul a două polinoame

Ex: `r = p1/p2`

- Se pot construi matrice din polinoame și din funcții raționale.

```

Ex: a = [1, p; p+3, p^2]
     b = [1, p; 1/p, (p+1)/p]

```

Se recomandă să se folosească forma cu `,` (virgulă) pentru a separa elementele de pe o linie.

6.5.5 Matrice cu elemente de tip logic

Conțin elemente de tipul `%t`, `%f` sau variabile logice. Asupra lor se pot efectua operațiile obișnuite cu matrice: concatenare, transpunere, operațiile `or` și `and`.

6.5.6 Liste

Matricele sunt colecții de obiecte de același tip, pe când **listele** sunt colecții de obiecte de tipuri diferite. Listele sunt folosite pentru a defini date structurate (structuri sau înregistrări).

Printre listele predefinite un rol important îl joacă sistemele liniare în sensul teoriei sistemelor.

- Forma generală:

```
l = list(a1, a2, ..., an)
```

Fiecare a_i este un obiect Scilab (matrice, listă, ..)

- Operații cu liste:

inserare: $l(i) = a$

ștergere: $l(i) = \text{null}()$

extragerea unor obiecte: $[o1, o2, \dots] = l(v)$, unde:

v este un vector de indici iar

$o1, o2, \dots$ sunt elementele din listă, corespunzătoare indicilor.

6.6 Programarea în Scilab

6.6.1 Funcții Scilab

Funcțiile Scilab (numite și macrouri) pot fi definite într-un fișier sau on-line.

- În fișier:

```
function [y1, y2, ..., yn] = nume_fct(x1, x2, ..., xm)
    // y1 .. = rezultate; x1 .. = parametri
    instrucțiuni
end
```

- On-line:

```
deff ('[y1, ..., yn] = nume_fct(x1, ..., xm)', ...
    'instrucțiuni')
```

- Citirea unui fișier conținând macrouri: `getf('nume_fis')`

6.6.2 Funcții predefinite în Scilab

Printre funcțiile predefinite se numără:

- Funcțiile matematice clasice (`sin`, `cos`, `log`, ...)
- Rutine specializate pentru rezolvarea unor probleme matematice, de tipul:
 - ecuații diferențiale (`ode`, `impl`, `dassl`, ...)
 - optimizare (`optim`)
 - funcții de algebră liniară (`svd`, `qr`, `det`, `lu`, `schur`, ...)
 - integrare numerică (`integrate`)
 - ... și multe altele. Puteți obține lista completă a comenzilor folosind **Help**.
- Funcții pentru supraîncărcarea (*overloading*) unor operatori (precum `+`, `-`, `*`), care trebuie să acționeze în mod diferit asupra unor tipuri diferite de operanzi (realizându-se astfel polimorfismul acestor operatori). Numele lor începe cu `%`.

Fișierele care conțin funcții predefinite pot fi atât în format cod-sursă (cu extensia `.sci`), fie compilate, în format binar (cu extensia `.bin`).

6.6.3 Tipuri de instrucțiuni Scilab

Sunt disponibile (lista nu e completă însă) următoarele tipuri de instrucțiuni:

- Expresii cu operatori matematici: `+` `-` `*` `/` `\`
- Expresii cu operatori de relație (pentru comparații):
 - `==` sau `=` pentru testarea egalității,
 - `<`, `>`, `<=`, `>=` pentru testarea ordinii,
 - `<>` sau `~=` pentru testarea inegalității
- Operații de intrare/ieșire (`print`, `printf`, `read`, `scanf`)
- Decizii:
 - simple:


```
if condiție then, ..., else, ..., end
```
 - multiple:


```
select nume_var,
  case val1, instrucțiuni,
  case val2, instrucțiuni,
```

```

    . . .
    else, instrucțiuni
end

```

- Cicluri

- cu contor:

```

    for contor = vmin:vmax, instrucțiuni, end

```

- sau:

```

    v = [lista elemente]

```

```

    for contor = v, instrucțiuni, end

```

- cu test inițial:

```

    while condiție, instrucțiuni, end

```

- Instrucțiuni de oprire a ciclurilor

<code>pause</code>	- se intră în alt nivel (alt bloc)
<code>return, return(val)</code>	- se revine la nivelul anterior
<code>resume</code>	- se revine la nivelul anterior
<code>abort</code>	- oprește execuția funcției
<code>setbpt</code>	- setare breakpoint
<code>delbpt</code>	- ștergere breakpoint
<code>disbpt</code>	- afișare breakpoint

- Apeluri de funcții, predefinite sau definite de utilizator (vezi paragrafele anterioare)
 - se realizează prin invocarea numelui funcției urmat de parametrii actuali.

6.7 Fișiere de comenzi

Fișierele de comenzi se pot scrie off-line cu un editor de text, sau prin salvarea definițiilor sesiunii curente (variabile, funcții, etc.). Se recomandă folosirea extensiei implicite `.sci`.

- Afișarea variabilelor definite: `who`
- Lista de comenzi și funcții sistem definite: `what`
- Help: click pe “Help” din meniu sau cu comanda `help cuvânt_cheie`
- Salvarea variabilelor în fișier extern (binar): `save`
- Citirea unui fișier de variabile: `load`
- Scrierea unei variabile (obiect Scilab) într-un fișier, în format FORTRAN:

```

write ( 'nume_fis', var)

```

- Citirea :

```
var = read ( 'nume_fis', format) unde
    "format" e de tipul: nr. linii, nr. coloane
```

6.8 Grafice

Graficele se pot afișa în ferestre X și se pot salva pe disc în trei formate, corespunzătoare driverelor: **X11**, **Pos** (PostScript) și **Xfig**. Mai există și driverul **Rec**, care afișează graficul în fereastra X și îl și salvează în fișier.

- Selectarea unui tip de driver:

```
driver ('nume_driver')
```

- Comanda `plot..(...)` afișează graficul în fereastra 0; dacă driverul selectat este **rec**, scrie și instrucțiuni PostScript în fișierul **foo.ps**
- Pentru a selecta un fișier PostScript, altul decât **foo.ps** se folosește comanda:

```
xinit('fis.ps')
```

- Comenzi de desenare de grafice 2D:

plot: grafic a 2 vectori, y ca funcție de x

```
plot(x, y, <et_x, et_y, titlu>)
```

plot2d - plot multiplu: parametrii sunt matrice de aceeași dimensiune; se iau coloanele pe rând și se face câte un grafic.

```
plot2d(x, y, <stil linie>, "pqr", "etx@ety@titlu" ...)
```

p = 0, 1 (afișare sau nu etichete)

q = 0 .. 4 (scalare axe)

r = 1, 2 (frontieră grafic – border)

Suprapunerea mai multor curbe pe același grafic se face dând pe rând comenzile de realizare a fiecărui grafic, fără ștergerea ferestrei grafice:

parametrul al patrulea: "p0r" (nu șterge fereastra grafică)

plot2d1 - permite și axe logaritmice

plot2d2 - funcții constante pe porțiuni

plot2d3 - grafic cu bare verticale

...

- Comenzi de desenare de grafice 3D:

`plot3d` - grafice tridimensionale

Ex: `plot3d(x, y, z, teta, alfa, "etx@ety@etz", <...>)`

`plot3d1` - nivele de gri

`plot3di` - modificarea interactivă a poziției de vizualizare cu butonul din stânga, oprire cu butonul din dreapta

- Se pot face grafice în mai multe ferestre sau se pot trimite în fișiere cu nume selectat de utilizator.
- Se pot afișa mai multe grafice independente în aceeași fereastră cu comanda `xsetech`

Observații:

1) Salvarea unui grafic într-un fișier, în formatul dorit, se poate face și direct din fereastra X în care este afișat graficul, cu butonul File → Export.

2) Dacă formatul de ieșire dorit este EPS și vreți să realizați mai multe grafice în aceeași sesiune Scilab, trebuie să introduceți comanda

```
xbasc()
```

(ștergere fereastră grafică) după fiecare grafic, altfel, în fișier se vor salva toate comenzile de grafică introduse de la începutul sesiunii Scilab.

3) O variantă robustă și care permite realizarea cu ușurință a unor grafice cu aspect profesional este salvarea graficelor în format XFig, urmată de editarea și salvarea lor în format EPS, din XFig.

6.9 Exemple

În figura 6.2 este prezentat graficul din figura 5.3, realizat de astă dată cu Scilab. Instrucțiunile Scilab sunt următoarele:

```
x = [0.1:0.2:5]; x = [x, 5];
```

```
f = x^2;
```

```
y = read("grafic.dat", -1, 3);
```

```
xa = y(:, 1);
```

```
f1 = y(:, 2);
```

```
f2 = y(:, 3);
```

```
xs = [x' xa' f1'];
```

```
fx = [f' f2' f3'];
```

```
xset("use color", 0)
```

```
xset("dashes")
```

```

xset("font",2,3)
xbasec()
plot2d1('gnl', xs, fs, [1 2 3]) // cu puncte
plot2d1('gnl', xs, fs, [-1 -2 -3], '102', 'Analitic@Experiment1@Experiment2')
// cu linii si etichete grafice
xgrid()
xtitle("Rezultatele analitice si experimentale")

```

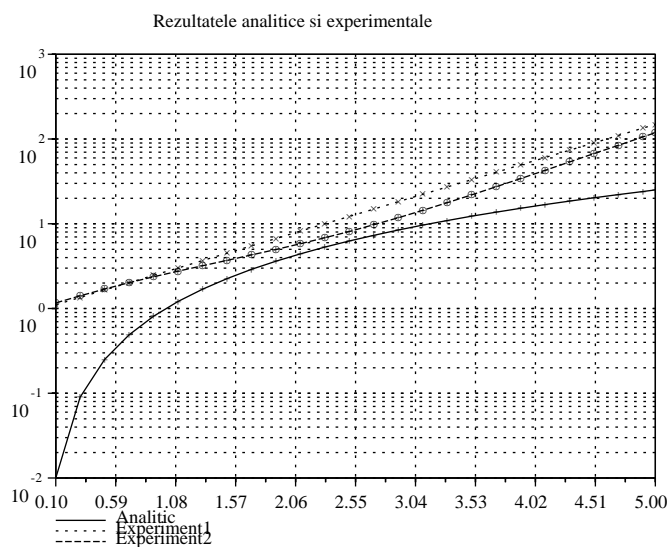


Fig. 6.2: Graficul din fig. 5.3, realizat cu Scilab

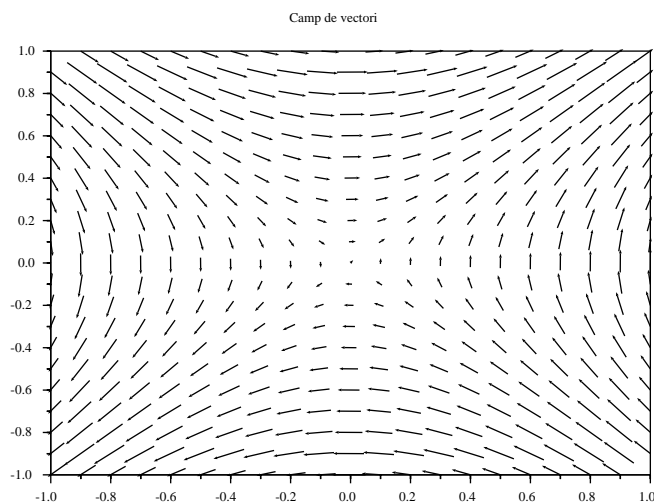


Fig. 6.3: Câmp de vectori reprezentat grafic cu Scilab

În figura 6.3 este prezentat un grafic realizat cu Scilab, folosind următoarele instrucțiuni:

```
x = [-1:0.1:1]; y=x; u = ones(x);
fx = x .* u'; fy = u .* y';
xbasec();
champ(x, y, fx, fy);
xtitle('Camp de vectori')
```

6.10 Apelul unei rutine C sau FORTRAN din Scilab

Deși Scilab dispune de un număr mare de funcții predefinite, acestea nu acoperă toată gama de probleme matematice de care un cercetător are nevoie în activitatea sa. Pentru a evita necesitatea de a “traduce” în limbajul Scilab o rutină de calitate deja existentă, dar scrisă în FORTRAN sau în C, Scilab oferă facilitatea de a apela rutine externe din bibliotecile matematice.

Pentru ca o rutină externă să poată fi apelată din Scilab, aceasta trebuie să îndeplinească câteva condiții:

- Să fie deja compilată, în format obiect (cu alte cuvinte, să fie conținută într-un fișier obiect, de obicei cu extensia `.o` sau `.a`).
- Să fie o subrutină (în cazul rutinelor FORTRAN), respectiv o funcție de tip `void` (în cazul rutinelor C).
În cazul funcțiilor FORTRAN, sau funcțiilor C care întorc o variabilă de orice tip cu excepția `void`, apelul poate fi făcut din Scilab, dar nu se va putea accesa valoarea întoarsă de funcție.
- În cazul funcțiilor C, toți parametrii trebuie să fie transmiși prin adresă și nu prin valoare.

Aceste condiții sunt impuse de faptul că tipul implicit al rutinelor externe apelate din Scilab este “FORTRAN”, deci aceste funcții trebuie să îndeplinească toate cerințele compatibilității C–FORTRAN (vezi paragraful 7.5).

Pentru a **extinde lista de rutine din Scilab** cu o rutină externă și pentru a apela o astfel de rutină trebuie parcurse următoarele **etape**:

- Se definește legătura (“link”) la rutina externă:

```
ll = link('cale/bibl.o', 'rutina', 'F')
```

în cazul rutinelor FORTRAN, respectiv:

```
ll = link('cale/bibl.o', 'rutina', 'C')
```

în cazul rutinelor C.

În comenzile de mai sus, `cale` reprezintă o cale spre directorul în care se găsește fișierul obiect `bibl.o`, iar `rutina` reprezintă numele rutinei.

Pentru a afișa lista tuturor rutinelor externe, se folosește aceeași comandă, fără parametri:

```
link()
```

Pentru a elimina o rutină din lista de rutine externe se folosește comanda (disponibilă în versiunile mai recente decât 2.2):

```
ulink(l1)
```

- Se apelează rutina cu o comandă de forma:

```
[y1,y2, ...] = fort('rutina', intrari, 'out', iesiri)
```

în care:

- `y1`, `y2`, ... reprezintă numele unor variabile Scilab în care vor fi memorate ieșirile rutinei
- `fort` este numele comenzii Scilab de apel al unei rutine externe
- `rutina` este numele rutinei externe
- `intrari` reprezintă descrierea, într-un format specific, a parametrilor de intrare ai rutinei. Pentru fiecare parametru trebuie precizate trei informații:
 - * numele variabilei Scilab în care este memorată
 - * poziția (numărul de ordine) în lista de parametri ai rutinei
 - * tipul: `'i'` – tip întreg sau caracter; `'r'` – tip real (simplă precizie); `'d'` – tip real (dublă precizie)
- `'out'` reprezintă separatorul între lista de variabile de intrare și lista de variabile de ieșire
- `iesiri` reprezintă descrierea, într-un format specific, a parametrilor de ieșire ai rutinei. Pentru fiecare parametru de ieșire trebuie precizate trei informații:
 - * dimensiunile matricei, sub forma:


```
[lin, col]
```

 cu `lin` numărul de linii și `col` numărul de coloane.
 - * poziția (numărul de ordine) în lista de parametri ai rutinei
 - * tipul: `'i'` – tip întreg sau caracter; `'r'` – tip real (simplă precizie); `'d'` – tip real (dublă precizie)

Se constată că instrucțiunea de apel al unei rutine externe are o sintaxă relativ complicată, de aceea este de preferat ca pentru fiecare rutină externă să fie scrisă și o funcție Scilab care să o apeleze. Numele funcției Scilab poate fi același cu cel al rutinei externe.

Exemplu de apel al unor rutine externe

Să presupunem că rutinele externe, scrise în limbajul C, sunt următoarele:

- `suma` – calculează în variabila `c` suma a două numere reale, `a` și `b`; funcția este declarată sub forma:

```
void suma(float *a, float *b, float *c);
```

- `sumav` – calculează în variabila `w` suma, element cu element, a vectorilor `u` și `v`; funcția este declarată sub forma:

```
void sumav(int *n, float *u, float *v, float *w);
```

Să presupunem de asemenea că cele două rutine se găsesc în fișierul `test.o` plasat în directorul curent.

Următorul program Scilab realizează link-*area* rutinelor la Scilab și apelul lor în Scilab.

```
// Apel functii suma, suma de vectori, scrise in C
ll = link('./test.o', 'suma', 'C');
ll = link('./test.o', 'sumav', 'C');

// Verificare: au fost linkate?
link()

// Citeste definitiile functiilor de apel numite la
// fel ca si rutinele externe: suma, sumav
getf('functiiapel.sci');

// Apeleaza functiile
a = 1; b = 2;
c = suma(a,b)

u = 1:5; v = 1:5;
w = sumav(5,u,v)
```

Fișierul `functiiapel.sci` conține definițiile funcțiilor Scilab care apelează cele două rutine externe (observați folosirea caracterelor de continuare a comenzii `...` pentru sporirea clarității):

```
function[c] = suma(a,b)
c = fort('suma', ... //nume rutina externa
        a,    1, 'r', ... //intrari: nume a, par. 1, real
        b,    2, 'r', ...
        'out', ... //iesire: dimensiuni 1x1, par.3, real
        [1,1],3, 'r')
return

function[c] = sumav(n,a,b)
dim = size(a);
c = fort('sumav', ... //nume rutina externa
```

```

        n, 1, 'i', ... //intrari
        a, 2, 'r', ...
        b, 3, 'r', ...
'out',      ... //iesire
        dim,4, 'r')
return

```

6.11 Traducerea automată în FORTRAN a unei funcții Scilab

După ce “prototipul” unui program a fost implementat și testat în Scilab, de obicei, codul programului este tradus într-un alt limbaj de nivel înalt, cum ar fi C sau FORTRAN, în scopul sporirii eficienței programului și, dacă este cazul, al păstrării confidențialității codului.

Scilab oferă facilitatea de a **traduce în mod automat** în FORTRAN o funcție scrisă în Scilab.

Pentru aceasta trebuie parcurse următoarele etape (să presupunem că numele funcției este `func`):

- Se compilează funcția:

```
comp(func)
```

- Se definește o listă cu un număr de componente egal cu numărul parametrilor de **intrare** ai funcției:

```
l = list()
l(1) = list('tip', 'nrlin', 'nrcol')
. . .
```

unde fiecare componentă a listei caracterizează un parametru (în cazul general, acest parametru este o matrice) și este la rândul său o listă cu trei elemente:

- Tipul variabilei `tip`: 0 – întreg; 1 – real în dublă precizie; 10 – șir de caractere;
- Numărul de linii `nrlin`: este un **nume formal** asociat numărului de linii ale variabilei, de exemplu `m` sau `nl`
- Numărul de coloane `nrcol`: este un **nume formal** asociat numărului de coloane ale variabilei, de exemplu `n` sau `nc`

- Se apelează funcția de traducere în FORTRAN:

```
mac2for( macr2lst(func), l)
```

Rezultatul apelului este un vector de șiruri de caractere, fiecare element al vectorului reprezentând o linie a programului FORTRAN.

Exemplu de utilizare a comenzii `mac2for`

Să presupunem că funcția care trebuie tradusă în FORTRAN se găsește în fișierul `sumaf.sci` și are următoarea formă:

```
function[c] = sumaf(a,b)
// o functie simpla, pe care vrem sa o traducem in FORTRAN
// calculeaza suma matricelor (reale) a si b
c = a+b
```

Următorul program Scilab realizează traducerea automată în FORTRAN a funcției `sumaf` și scrie rezultatul (codul sursă al rutinei FORTRAN) în fișierul `sumaf.f`. Acest cod sursă poate fi ulterior compilat cu compilatorul FORTRAN și linkeditat cu alte programe C, C++ sau FORTRAN.

```
// Citeste o functie Scilab si o traduce in FORTRAN
getf ('sumaf.sci')

// Functia sumaf are 2 par de intrare -- matrice de reali, a si b

l = list();
l(1) = list('1', 'm', 'n');
l(2) = list('1', 'm', 'n');

comp(sumaf)
txt = mac2for(mac2lst(sumaf), l);

// Scrie in fisier rezultatul
fis = file('open', 'sumaf.f', 'unknown')
write(fis, txt)
file('close', fis)
```

6.12 Exerciții

EXERCITIUL 1

Scrieți un program Scilab care să reprezinte grafic intensitatea curentului și tensiunea la bornele condensatorului, în circuitul RC serie descris în Referatul Dvs. de laborator realizat la tema 4.

Programul va permite introducerea interactivă de la consolă a datelor de intrare – parametrilor elementelor de circuit (R, C, E), va determina intervalul de timp și pasul de

discretizare în timp adaptate valorilor parametrilor, și va reprezenta grafic u_C și i dați de formulele analitice (paragraful 2 al referatului de laborator).

EXERCITIUL 2

Opțional: Salvați graficul în format XFig și editați acest grafic, indicând prin săgeți cele două grafice și eventual adăugând alte elemente grafice pe care le considerați necesare pentru sporirea clarității graficului.

EXERCITIUL 3

a) Adăugați o nouă secțiune, **Rezultate numerice**, în referatul de laborator. În cadrul subsecțiunii **Implementarea în Scilab a formulelor analitice** descrieți pe scurt funcțiile programului Scilab scris la exercițiul 1 și includeți graficul realizat.

b) Adăugați încă o secțiune, **Codul sursă al programelor Scilab**, în partea de anexe. În subsecțiunea **Implementarea în Scilab a formulelor analitice** includeți codul sursă al programului Dvs.

EXERCITIUL 4

a) Scrieți un program Scilab care preia datele de intrare ale circuitului RC serie, stabilește intervalul de timp adaptat datelor de intrare și apoi folosește funcțiile Scilab de rezolvare a ecuațiilor diferențiale ordinare pentru a determina soluția (tensiunea la bornele condensatorului). Programul va determina de asemenea intensitatea curentului în circuit, folosind tehnici de derivare numerică.

Refaceți calculele pentru aceleași valori R , C și E , dar pentru diverse valori ale parametrilor funcțiilor Scilab, respectiv pentru diverse valori ale pasului de discretizare temporală. Calculați, pentru fiecare set de valori ale parametrilor, eroarea relativă în norma euclidiană și în norma Cebîșev între soluțiile numerice și cele obținute folosind formulele analitice. Salvați aceste valori într-un fișier ASCII, pentru a putea să le includeți în referatul de laborator.

b) Pentru un set de valori ale parametrilor, reprezentați grafic în funcție de timp tensiunea la bornele condensatorului (obținută folosind formula analitică și respectiv tehnica numerică). Reprezentați grafic intensitatea curentului (obținută folosind formula analitică și respectiv tehnica numerică).

c) Adăugați o nouă subsecțiune la secțiunea de **Rezultate numerice** a referatului de laborator. Includeți: o scurtă descriere a funcțiilor programului Scilab și a parametrilor acestuia, un tabel care să cuprindă erorile pentru diverse valori ale parametrilor, precum și graficele curentului și tensiunii.

d) Adăugați o nouă subsecțiune la anexa **Codul sursă al programelor Scilab**, numită **Programul de rezolvare cu Scilab a ecuației diferențiale ordinare**, în care includeți codul sursă al programului scris.

Capitolul 7

Medii de programare Unix — C, C++, FORTRAN —

Oricât de performant și de flexibil ar fi un pachet integrat specializat pentru aplicații științifice, matematice sau ingineresti, sunt situații în care cercetătorul este nevoit să facă apel la un limbaj universal de programare. Acest lucru se întâmplă atunci când:

- este necesară o mare **viteză** de execuție (se poate spera cel puțin la înjumătățirea timpului de calcul);
- este necesar un program **executabil** independent, care să fie distribuit în format binar (eventual păstrând confidențial codul sursă);
- trebuie folosit un cod preexistent, eventual preluat dintr-o **bibliotecă matematică** standard.

În acest fel, pachetul integrat specializat devine un mediu pentru dezvoltarea (producerea și testarea) rapidă a prototipurilor, fiind necesară ulterior traducerea manuală sau automată din limbajul specializat (de exemplu MATLAB) în limbajul universal.

Începând din anii '50, problema definirii unui limbaj universal pentru programarea calculatoarelor a generat pasionante studii și discuții în lumea specialiștilor în calculatoare. Cu toate că au fost definite sute sau poate chiar mii de limbaje, doar câteva au reușit să se impună.

Dintre acestea, limbajul FORTRAN, lansat de firma IBM și destinat inițial calculelor matematice, reprezintă un exemplu uimitor de vitalitate. În ciuda vechimii sa, el este încă intens folosit de comunitatea științifică datorită cantității enorme de cod sursă FORTRAN de excepțională calitate (cum sunt bibliotecile matematice), și datorită efortului de standardizare care urmărește permanent implementarea în limbaj a conceptelor moderne. Versiunile FORTRAN 77 [57] și Fortran 90 [62] (apărute în 1977 și respectiv 1990) au multe elemente diferite, progresul fiind evident.

Chiar dacă programele scrise în FORTRAN sunt relativ eficiente în execuție, instrucțiunea de salt necondiționat (`GOTO`) din acest limbaj a permis scrierea unor programe încâlcite (“spaghete”), greu de depanat și întreținut. Din acest motiv încă din anii '70 a apărut un curent care pledează pentru folosirea “**programării structurate**”, bazată pe un număr redus de instrucțiuni de control (decizia și ciclul).

S-a impus necesitatea utilizării unor tipuri de date clar definite (pentru a permite implementarea eficientă a “tipurilor abstracte de date”) și chiar apariția unor limbaje de programare care să reflecte această viziune. Profesorul vienez Niklaus Wirth, autorul limbajului Pascal, a scris o carte fundamentală în domeniu, al cărei titlu “*Algorithms + Data Structures = Programs*” este ilustrativ în acest sens.

În paralel, în SUA, la laboratoarele Bell care aparțineau de AT&T, Dennis Ritchie a definit un nou limbaj universal numit de el **C** [61]. Acest limbaj, în ciuda aspectului său criptic (accesibil la prima vedere doar specialiștilor) s-a impus devenind limbajul oficial al sistemului de operare UNIX. De altfel, însuși sistemul de operare UNIX este scris în limbajul **C** exceptând o parte ne semnificativă dependentă de mașină. Ca dovadă a succesului limbajului **C** s-a scris și un translator automat din **FORTRAN** în **C**, numit `f2c`.

În afara modelului imperativ de programare structurată, cercetările de inteligență artificială au condus la definiția unor limbaje funcționale gen *Lisp*, potrivite pentru calcule simbolice, dar nu și pentru cele numerice.

Folosirea intensivă a modelului imperativ a condus, la mijlocul anilor '80, la apariția conceptului de programare orientată pe obiecte – **OOP**, concept fundamental pentru realizarea eficientă a programelor de mari dimensiuni, precum și dezvoltarea de către Bjarne Stroustrup a unui limbaj numit **C++** [77]. Acesta este o extensie a limbajului **C** care permite implementarea caracteristicilor *OOP*: *încapsularea*, *moștenirea* și *polimorfismul*.

Trebuie remarcat că realizarea unor programe de calitate este mai mult o problemă de stil de programare decât de limbaj. Se pot da exemple remarcabile [17] de coduri scrise în **C** sau chiar **FORTRAN** care satisfac cele mai exigente cerințe, inclusiv cele ale orientării pe obiecte. Realizarea unor programe performante nu depinde nici măcar de calculatorul avut la dispoziție, ci de calitățile programatorului. Nu trebuie uitat că sistemul de operare UNIX a fost dezvoltat pe un sistem DEC PDP care avea performanțe mai slabe decât orice calculator personal actual.

În consecință, vă recomandăm să vă perfecționați stilul de programare, iar cea mai bună metodă este să studiați cu atenție exemplele ilustre.

Compilatoarele sistemului Unix au proprietatea remarcabilă de **interoperabilitate**, ceea ce înseamnă că indiferent în ce limbaj a fost scris un program el poate apela rutine scrise în alt limbaj. În acest fel programatorii **C** pot exploata tezaurul de cunoștințe cuprins în bibliotecile matematice FORTRAN [74].

Cel mai mare depozit de software de calitate cu aplicații matematice și științifice, disponibil în sursă (majoritatea FORTRAN) este NetLib [14], din care am selectat numele câtorva pachete celebre:

- `a` - algoritmi de aproximare
- `blas` - sistem algebric liniar de bază [45]
- `eispack` - vectori și valori proprii pentru matrice [80]
- `fftpack` - transformanta Fourier rapidă
- `itpack` - rezolvarea iterativă a sistemelor liniare [48]
- `lapack` - algebră liniară
- `lanz` - valori proprii pentru matrice rare de mari dimensiuni
- `linpack` - sisteme de ecuații liniare
- `minpack` - sisteme de ecuații neliniare
- `odepack` - sisteme de ecuații diferențiale ordinare
- `pppack` - subrutine de interpolare, spline
- `quadpack` - integrare numerică
- `sparspack` - algebră liniară cu matrice rare
- `specfun` - funcții speciale
- `voronoi` - triangularizări Delaunay și diagrame Voronoi

În acest de depozit se află sursele unor biblioteci complete cum sunt `sciport` (CRAY), `slatec` (Boeing) sau `toms` (ACM), precum și pachete de funcții ca `PVM`, `MPI`, `HENCE`, `PETSc` [17] ce permit dezvoltarea de algoritmi paraleli și distribuiți, care sunt executați simultan pe mai multe procesoare sau calculatoare cuplate în rețea.

Utilizarea în activitatea de cercetare a acestui tezaur de cunoștințe vă va permite să eliminați greșeala de a reinventa roata și să ridicați nivelul cercetărilor Dvs. la standardul mondial. Efortul de a învăța să folosiți ce au realizat alții va fi răsplătit însutit!

Obiectivul capitolului de față nu îl constituie învățarea unui limbaj de programare (de altfel nici nu ar fi posibil în câteva ore!) ci să învățați folosirea mediului de dezvoltare sub UNIX specific LMN (editare, compilare, linkeditare, depanare); presupunând că stăpâniți un limbaj de programare de nivel înalt (FORTRAN, C sau C++). În plus, se urmărește însușirea deprinderilor minimale necesare folosirii unei subrutine din biblioteca matematică FORTRAN, cu apel din C. În acest fel vi se deschid porțile unui univers extrem de bogat, cuprins în biblioteci de renume.

Pentru informații suplimentare referitoare la limbajul C, se pot consulta referințele [46], [70] și [79] pp. 315–333, pentru traducerea din pseudocod în C, C++ și FORTRAN vă recomandăm [59] pp. 261–322, iar pentru limbajul FORTRAN vă recomandăm [57] și [62].

7.1 Principalele opțiuni ale compilatoarelor gcc, g++, g77

Compilatoarele `gcc`, `g++`, `g77` sunt versiuni GNU ale compilatoarelor C, C++ și respectiv FORTRAN 77 în mediul Unix.

De fapt, compilatoarele sunt integrate, orice fișier sursă putând fi compilat cu compilatorul `gcc` care identifică tipul fișierului corespunzător extensiei (`.c`, `.cc` și respectiv `.f`) și interpretează corespunzător instrucțiunile din fișierul sursă.

În consecință, opțiunile de compilare de bază sunt identice pentru toate cele trei compilatoare.

- o *fis_ex* – fișierul executabil creat de compilator să aibă numele *fis_ex* (implicit: `a.out`)
- g – fișierul executabil să conțină informații necesare pentru depanarea programului (nume variabile și plasarea lor în memorie, adresele liniilor de cod sursă)
- c – nu parcurge decât etapa de compilare nu și cea de linkeditare; rezultatul este un fișier obiect (cu extensia `.o`)
- On – compilează cu optimizare; numărul *n* este opțional și poate lua valorile 0 (fără optimizare), 1 (implicit), 2 sau 3
- p fișierul executabil să includă instrucțiuni suplimentare pentru generarea, după rularea sa, a unei liste de informații despre program (*program profile*), cum ar fi: număr de apeluri ale funcțiilor, timpul consumat cu execuția fiecărei funcții, etc.
- lname – linkeditare cu biblioteca matematică `libname`.

Observație: În mediul Unix, apelul compilatorului determină atât compilarea fișierelor sursă cât și linkeditarea fișierelor cu alte fișiere obiect sau cu biblioteci standard.

De exemplu:

```
gcc -g -o test -p test.c -lm
```

are ca efect compilarea sursei `test.c` cu includerea: informațiilor pentru depanare (`-g`), a informațiilor de “profile” (`-p`), scrierea fișierului executabil sub numele `test` și linkeditarea cu biblioteca matematică `libm.a`.

Pentru a nu trebui să introduceți de fiecare dată o comandă de compilare complicată, puteți crea un script care să conțină instrucțiunea de compilare. În cazul programelor ale căror surse se află în mai multe fișiere, este recomandabil să folosiți comanda `make` (paragraful 7.3).

Extensii fișiere:

- `.c` – fișiere sursă în limbaj C
- `.cc` – fișiere sursă în limbaj C++
- `.f` – fișiere sursă în limbaj FORTRAN77
- `.h` – fișiere de tip “include”
- `.o` – fișiere obiect (rezultate în urma compilării, înaintea creării fișierului executabil)

7.2 Recomandări privind stilul de lucru profesional

Productivitatea și eficiența programării în mediul de programare Unix depind în mod esențial de folosirea “inteligentă” a facilităților existente.

Deși fiecare programator are propriile tehnici de creștere a eficienței, atunci când sunteți la început puteți folosi sugestiile de mai jos.

- Cum lucrați într-un mediu care permite deschiderea mai multor ferestre, folosiți această facilitare!

Utilizați o fereastră pentru editarea programului, altă fereastră pentru compilări și, de preferință, o a treia fereastră pentru execuția programului.

După ce ați terminat de editat programul, salvați-l, dar nu ieșiți din editor: în cazul în care programul Dvs. ar avea o eroare pe care ar trebui să o corectați, veți economisi astfel timpul de lansare, încă o dată, a editorului.

- Utilizați de preferință editorul Emacs pentru editarea programelor. Acest editor are facilitarea de a lansa comanda de compilare, fără măcar să trebuiască să mutați mouse-ul în altă fereastră!

La programe lungi, evitarea chiar și a unei simple operații precum mutarea în altă fereastră va contribui la micșorarea timpului de editare și depanare a programului.

- Dacă aveți de realizat un program asemănător unuia existent, nu îl scrieți de la zero! Copiați programul existent sub numele dorit pentru programul Dvs. și modificați-l. Această tehnică, de folosire a unor *modele* (patterns) preexistente (preluate din fișierele de exemple, tutorials) vă poate reduce substanțial timpul de introducere a programului și numărul de erori de programare.
- Rareori vi se va întâmpla să scrieți un program, să îl depanați și să îl executați, iar apoi să nu îl mai “atingeți” niciodată. De cele mai multe ori, un program odată scris trece prin multe modificări, unele dintre ele făcute la mult timp după data primei creări. Uneori, nu programatorul inițial este cel care modifică programul, ci un alt membru al echipei.

Pentru a putea reduce sensibil efortul de întreținere, documentați programul încă de la început. Altfel veți irosi un timp enorm de descifrare atunci când ulterior trebuie să faceți o mică modificare.

Comentați programele scrise: începeți cu un comentariu privind scopul programului, numele programatorului, data la care a fost realizat.

Începeți fiecare rutină cu un comentariu privind scopul funcției și semnificațiile parametrilor.

Adăugați comentarii și la acele linii de program care codifică o metodă neobișnuită de realizare a unei operații, la acelea foarte “concentrate”, care conțin diferite artificii de programare, etc.

După ce ați terminat de scris și de depanat programul, scrieți și o documentație a acestuia, de tip “Manualul utilizatorului” și chiar “Manualul programatorului” în care descrieți tehnicile de programare folosite (algoritmi și structuri de date). În 90% din cazuri, după un timp vă veți felicita că ați făcut acest efort.

7.3 Compilarea programelor folosind make

Un program de mari dimensiuni, al cărui cod sursă se află în mai multe fișiere separate, poate necesita un timp foarte lung pentru compilare și editarea legăturilor necesare constituirii programului executabil.

Programul utilitar **make** determină automat care părți ale unui astfel de program s-au modificat și trebuie recompilate, “comandă” recompilarea selectivă și apoi linkeditarea.

Pentru a putea utiliza comanda **make** utilizatorul trebuie să creeze un fișier numit **Makefile** sau **makefile**, în care sunt “descrise”, într-un format specific, relațiile dintre fișierele care compun programul precum și comenzile care realizează compilarea.

7.3.1 Structura unui fișier makefile

Un fișier **makefile** conține următoarele tipuri de linii:

- comentarii: linii care încep cu caracterul **#**
- definiții de variabile: **VARIABILA = valoare**

Ex: **CFLAGS = -g -c**

O variabilă definită poate fi folosită în liniile următoare ale fișierului **makefile**, sub forma

\$(VARIABILA)

- definiții de relații:

```
fișier: listă_fișiere
```

în care, în cazul nostru, `fișier` este numele unui fișier obiect (cu extensia `.o`) iar `listă_fișiere` este o listă de fișiere de tip `.c` și `.h` de care “depinde” fișierul obiect. De exemplu, dacă un modul al programului se numește `test.c` și el include fișierul header `test.h` (conține instrucțiunea `#include "test.h"`) atunci: fișierul obiect `test.o` trebuie re-generat dacă se modifică `test.c` sau se modifică `test.h`. Această dependență va apărea în fișierul `makefile` sub forma:

```
test.o: test.c test.h
```

- instrucțiuni de generare a principalului “rezultat”

Sunt linii care încep cu Tab, plasate imediat sub o linie de tip “definire de relație”, care conțin instrucțiunea care “generează” obiectul.

Exemplu:

```
test.o: test.c test.h}
    gcc $(CFLAGS) test.c
```

7.3.2 Modele de fișiere `makefile`

Iată mai jos un **model simplu** de fișier `makefile`. Acest model presupune următoarele:

- Codul sursă al programului se află în două fișiere: `inout.c` și `main.c`.
Fișierul `inout.c` include fișierul header `inout.h`, iar fișierul `main.c` include fișierele header `inout.h` și `main.h`.
- Fișierul executabil trebuie salvat sub numele `calcul`.

```
main: main.o inout.o
    gcc -g -p main.o inout.o -o calcul -lm
```

```
# Dependencies
```

```
main.o: main.c inout.h main.h
    gcc -g -p -c main.c -o main.o
inout.o: inout.c inout.h
    gcc -g -p -c inout.c -o inout.o
```

Modelul de mai sus, extrem de simplu, poate fi dezvoltat astfel încât programul scris să fie ușor de întreținut, prin salvarea tuturor fișierelor în directoare dedicate.

Vă propunem un **model** de fișier `makefile` pe care vă recomandăm să îl folosiți, adaptându-l programului concret pe care îl scrieți.

Acest model presupune următoarele:

- În directorul curent există patru subdirectoare: `include` (conține fișierele de tip `.h`); `src` (conține fișierele de tip `.c`); `obj` (conține fișierele de tip `.o`); `bin` (conține fișierele executabile).
- Codul sursă al programului se află în două fișiere: `inout.c` și `main.c`, plasate în subdirectorul `src`.
Fișierul `inout.c` include fișierul header `inout.h`, iar fișierul `main.c` include fișierele header `inout.h` și `main.h`. Fișierele header sunt plasate în subdirectorul `include`.
- Pentru a păstra “ordinea” în directoarele `include` și `src`, se dorește salvarea fișierelor obiect în directorul `obj`.
- Fișierul executabil trebuie salvat în directorul `bin` sub numele `calcul`.

O posibilă structură a fișierului `makefile` este următoarea:

```
CC = gcc
CFLAGS = -g -p
INCLUDEFLAGS = -Iinclude

INCLUDE_DIR = include
SRC_DIR = src
OBJ_DIR = obj
BIN_DIR = bin
PROGNAME = calcul
LIBS = -lm

OBJECTS = $(OBJ_DIR)/main.o \
          $(OBJ_DIR)/inout.o

main: $(OBJECTS)
      $(CC) $(CFLAGS) $(OBJECTS) $(LIBS) -o $(BIN_DIR)/$(PROGNAME)

# Dependencies
$(OBJ_DIR)/main.o: $(SRC_DIR)/main.c $(INCLUDE_DIR)/inout.h $(INCLUDE_DIR)/main.h
      $(CC) $(CFLAGS) $(INCLUDEFLAGS) -c $(SRC_DIR)/main.c -o $(OBJ_DIR)/main.o
$(OBJ_DIR)/inout.o: $(SRC_DIR)/inout.c $(INCLUDE_DIR)/inout.h
      $(CC) $(CFLAGS) $(INCLUDEFLAGS) -c $(SRC_DIR)/inout.c -o $(OBJ_DIR)/inout.o
```

7.4 Utilizarea depanatorului gdb

Lansare:

```
gdb <↵ >
```

sau, pentru versiunea X:

```
xxgdb <↵ >
```

- `file nume`: încărcarea fișierului **executabil** *nume*
- `set args argumente`: definirea argumentelor programului (cele care ar apărea în linia de comandă, dacă programul ar fi lansat de la consolă și nu din debugger)
- `break numar`: setare breakpoint (oprire) pe linia cu numărul *numar* din fișierul **sursă** (*.c)
- `display var`: afișarea valorii variabilei *var*
- `watch var`: afișarea valorii variabilei *var* automat, de câte ori aceasta își schimbă valoarea.
- `run`: rularea programului
- `continue`: continuarea execuției programului după o oprire
- `step`: rularea programului pas cu pas
- `next`: rularea programului pas cu pas, fără a intra în alte funcții
- `kill`: oprirea execuției programului
- `quit`: ieșirea din gdb
- `list`: afișarea a 10 linii din program

Observație: `xxgdb` este o versiune a `gdb` bazată pe sistemul de ferestre grafice XWindow. Toate comenzile `gdb` sunt valabile și în `xxgdb`. În plus, `xxgdb` permite selectarea majorității comenzilor `gdb` și cu ajutorul mouse-ului.

Fereastra programului `xxgdb` este prezentată în figura 7.1.

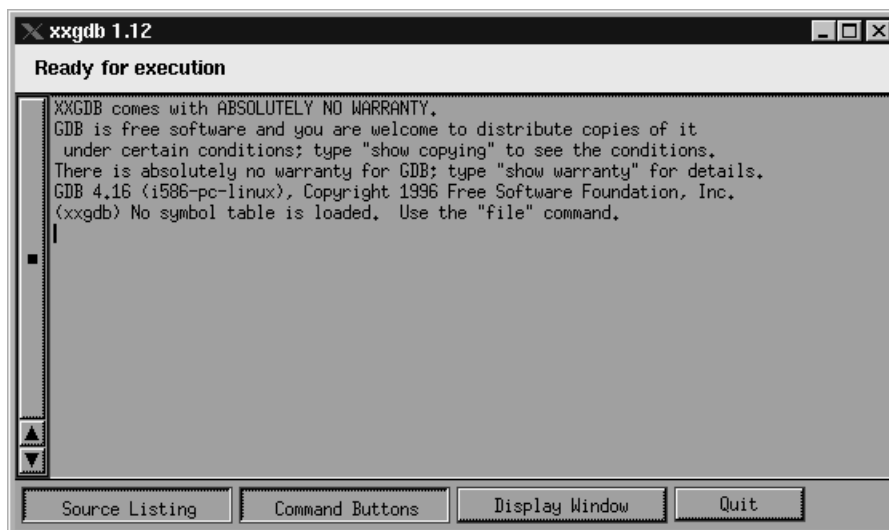


Fig. 7.1: Fereastra principală a programului `xxgdb`

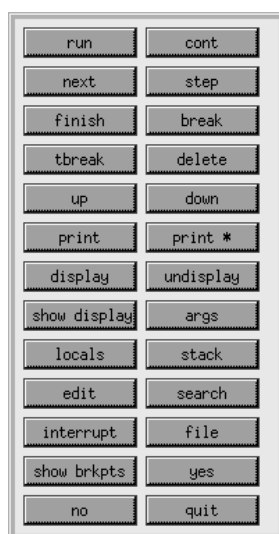


Fig. 7.2: Butoanele de comenzi ale programului `xxgdb`

După apăsarea butonului “Command Buttons” al ferestrei principale, se afișează fereastra de butoane, al cărei aspect este prezentat în figura 7.2.

Pentru informații suplimentare referitoare la debugger-ele `gdb` și `xxgdb`, se pot consulta referințele [58] pp. 489–497 și [79] pp. 355–371.

7.5 Apelarea rutinelor FORTRAN dintr-un program scris în limbajul C

Pentru a apela o rutină FORTRAN dintr-un program C, trebuie să se țină cont de următoarele observații:

- Toți parametrii unei rutine FORTRAN sunt transmiși prin adresă. În consecință: tablourile (vectori și matrice), șirurile de caractere și funcțiile se transmit ca parametri spre o rutină FORTRAN la fel ca și în C. În schimb, scalarii trebuie transmiși prin adresă.

De exemplu: dacă `i` este o variabilă declarată prin instrucțiunea

```
int i;
```

ea va fi transmisă ca parametru unei funcții FORTRAN sub forma

```
&i
```

- Stocarea în memorie a matricelor este făcută în mod diferit de modul de stocare C, și anume: întâi pe linii și apoi pe coloane (în ordinea: $a_{11}, a_{21}, \dots, a_{n1}, a_{12}, a_{22}, \dots$

De aceea, o matrice care urmează să fie transmisă ca parametru unei rutine FORTRAN trebuie permutată înainte de a apela funcția FORTRAN.

- Transmiterea unei variabile de tip complex se face astfel:

- Variabila se declară sub forma:

```
struct complex {float re; float im;} varc;
```

- Transmiterea se face prin adresa variabilei `varc`:

```
&varc
```

Exemplu

Se consideră următorul antet al unei funcții FORTRAN:

```
c FUNCTIA    calcul
c Parametri de intrare:
c   f: subrutina, definita ca:
c       subroutine f(a,b,c,n)
c       real b(10),c(10), a(10,10)
c       integer n
c   g, h: variabile de tip real
c   titlu: sir de 80 caractere
c   a: matrice cu 8 linii si 10 coloane
c   b, c: vectori cu 10 componente
c   n: intreg
```

```

c Parametri de iesire:
c   gh: variabila de tip real
c   ierr: intreg
c   cx: complex
c Functia intoarce:
c   o variabila de tip intreg

integer function calcul(f,g,h,gh,titlu,a,b,c,n,ierr,cx)

external f
real g, h, gh
character*80 titlu
real b(10), c(10), a(8,10)
complex cx

```

Această funcție poate fi apelată din FORTRAN, folosind următorul program principal:

```

c Program FORTRAN de apel al rutinei "calcul"
program main

external f
real g, h, gh
character*80 titlu
real b(10), c(10), a(10,8)
integer ii
complex cx;

titlu = 'da'
n = 2
a(1,1) = 1
a(2,1) = 2
a(1,2) = 3
a(2,2) = 4

b(1) = 1
b(2) = 2

g = 7
h = 8
ii = calcul(f,g,h,gh,titlu,a,b,c,n,ierr,cx)

write(*,*) 'c = ', c
write(*,*) 'gh = ', gh
write(*,*) 'titlu = ', titlu
write(*,*) 'ierr = ', ierr
write(*,*) 'ii = ', ii
write(*,*) 'cx = ', cx
end

```

Programul C care apelează această funcție are forma:

```

/* Program C de apel al functiei "calcul" */
#include <stdlib.h>
#include <string.h>

struct complex {float re; float im;};

void f(float a[8][10], float b[10],float c[10],int n);
int calcul(void f(float a[8][10], float b[10],float c[10],int n),
          float * g, float * h, float * gh,
          char *titlu, float a[10][8], float b[10],float c[10],int * n,
          int * ierr, struct complex * cx );

main() {
    float g, h, gh;
    char titlu[80];
    float b[10], c[10], a[8][10];
    int ii, ierr, n, len;
    struct complex cx;

    strcpy(titlu, "da");

    /* Matrice cu elementele:
       | 1 3 |
       | 2 4 |
    transpusa pentru a fi transmisa ca parametru */
    n = 2;
    a[0][0] = 1; a[0][1] = 2;
    a[1][0] = 3; a[1][1] = 4;

    b[0] = 1; b[1] = 2;
    g = 7.0; h = 8.0;

    ii = calcul(f,&g,&h,&gh,titlu,a,b,c,&n,&ierr,&cx);

    printf("c = %f %f %f \n", c[0], c[1], c[2]);
    printf("gh = %f \n", gh);
    printf("titlu = %s \n", titlu);
    printf("ierr = %d \n", ierr);
    printf("ii = %d \n", ii);
    printf("cx = %f %f \n", cx.re, cx.im);
} /* End main */

```

7.6 Exerciții

EXERCITIUL 1

Primul program

1. Creați un director numit `~/programe`
2. Folosind editorul `vi` sau `emacs` introduceți următorul program (în directorul `~/programe`):

```
main()
{
float a, b, c;

a = 2.;
b = 3.;

c = a + b;
printf("Suma este: %f.\n", c);

c = a * b;
printf("Produsul este: %d.\n", c);

}      /* end main */
```

3. Salvați fișierul sub numele `test.c`.

Vă recomandăm să NU ieșiți din editor, ci să deschideți o nouă fereastră în care veți da comenzile de compilare și veți executa programul.

4. Compilați programul folosind comanda:

```
gcc -g test.c
```

5. Apelând `ls`, veți constata că s-a creat fișierul executabil `a.out`.

6. Executați programul, cu comanda:

```
a.out
```

7. Observați efectul formatului `%d` (cu semnificația: valoare de tip întreg) asupra rezultatului afișat.

Note:

- Orice program C trebuie să conțină cel puțin o funcție, funcția `main`.

- comentariile sunt delimitate de caracterele `/* */`
- Principalele tipuri de date sunt: caracter (`char`), întreg (`int`), real (`float`).
- Orice instrucțiune se încheie cu `;` (punct și virgulă)
- Funcția `printf` afișează pe ecran un text.

Instrucțiunea: `printf("Suma este: %f.\n", c);` are următorul efect: se afișează pe ecran textul `Suma este:` , urmat de valoarea unei variabile de tip "float" (formatul "float" este desemnat prin `%f`), urmat de punct și de trecerea la linie nouă `\n`. Șirul de caractere delimitat de `"` este primul parametru al funcției `printf`. Următorii parametri (în cazul de față unul singur) reprezintă variabilele ale căror valori vor fi afișate, conform formatului specificat în primul parametru.

EXERCITIUL 2

Primul program, extins

1. Lucrând în fereastra de editare a programului, modificați programul `test.c` astfel încât să preia de la consolă valorile parametrilor `a`, `b`. Pentru aceasta:
 - Ștergeți cele două linii care setează valorile parametrilor `a` și `b`.
 - Introduceți în locul lor liniile:

```
printf("Introduceti primul numar: ");
scanf("%d", &a );

printf("Introduceti al doilea numar: ");
scanf("%f", &a );
```

2. Folosind fereastra pe care ați dedicat-o operațiunilor de compilare și execuție a programului, compilați din nou și executați programul, dând valorile $a = 1.5$ și $b = 2.0$. Veti constata că rezultatul nu este cel așteptat (deci, există o greșeală de logică în program). Va trebui, dacă nu vă dați seama direct, să utilizați un "depanator" (debugger) pentru a depista eroarea.

EXERCITIUL 3

Depanarea folosind debugger-ul `xxgdb`

1. Lansați debugger-ul folosind butonul din stânga al mouse-ului și selectând `Applications` → `xxgdb`. Alternativ, îl puteți lansa din linia de comandă, tastând:

```
xxgdb <Enter>
```


2. Deschideți fereastra pentru codul sursă, fereastra de butoane și fereastra "Display".
3. Selectați "File" din fereastra de butoane și apoi alegeți din lista de fișiere afișată fișierul executabil `a.out`.

4. Setați un "breakpoint" (punct de oprire) pe linia

```
printf("Introduceți al doilea număr: ");
```

astfel:

- În fereastra conținând textul sursă, deplasați cursorul pe linia respectivă și selectați "Break" din fereastra de butoane

sau

- În fereastra inițială, dați comanda:

```
b <număr>
```

unde *număr* este numărul liniei pe care se dorește oprirea, afișat în fereastra conținând textul sursă în partea din dreapta sus.

5. Selectați "run" (sau dați comanda `r`). Programul se va executa până la linia marcată ca "breakpoint".

6. Dați comanda de afișare a valorii variabilei `a`:

```
p a
```

Alternativ, cu comanda `display a`, variabila se va afișa în fereastra "Display".

7. Încercați să vă dați seama unde poate fi greșeala din linia anterioară celei pe care rularea s-a oprit.

8. Setați un alt breakpoint pe linia:

```
c = a + b;
```

9. Continuați rularea (cu una din comenzile: Step, Run sau Continue) și după oprire verificați valoarea variabilei `b`. Încercați să depistați cea de a doua eroare.

10. În fine, setați un alt breakpoint pe linia:

```
printf("Produsul este: %d.\n", c);
```

și încercați să depistați din ce motiv rezultatul afișat (care ar trebui să fie produsul `a * b`) nu este corect.

EXERCITIUL 4

Citiți cu atenție următorul cod cu scopul de a înțelege fiecare instrucțiune. Ce se va afișa la consolă?

```
float v[50], u[50];      /* Doi vectori */
int n=10;                /* Un intreg initializat cu 10 */

for (i=0, j=0; i<n; i++, j=j+2)
{
    v[i] = i;
    u[i] = j;
    printf("v[%d] = %f, u[%d] = %f \n", i, v[i], i, u[i]);
}
```

EXERCITIUL 5

Scrieți un program care să calculeze produsul scalar a doi vectori.

- Dimensiunea maximă a vectorilor: 20
- Dimensiunea actuală, precum și componentele vectorilor vor fi introduse de la consolă.
- Programul va conține două funcții:
 1. `main`: preia valorile de intrare, apelează `prod_s`, afișează rezultatul;
 2. `prod_s`: primește ca parametri doi vectori și returnează rezultatul (produsul scalar) în cel de al treilea parametru.
- Cele două funcții vor fi scrise în fișiere separate, `main.c` și respectiv `prods.c`.
- Folosiți utilitarul `make` pentru compilare și editarea legăturilor.

EXERCITIUL 6

Folosind facilitatea Scilab de a apela rutine externe (paragraful 6.10), incorporați în Scilab, executați și verificați în mediul Scilab rutina de calcul al produsului scalar.

EXERCITIUL 7

Folosind facilitatea Scilab `mac2for` (paragraful 6.11) generați automat codul FORTRAN al următoarei funcții Scilab:

```
function[b,err] = permuta(a)
// permuta elementele vectorului a, astfel incat:
//  ultimul devine primul, penultimul devine al doilea, etc.
[l,c] = size(a);
if l ~= 1 then
```

```

    err = 1;
    b = a;
    return;
end
b = a(c:-1:1);
err = 0

```

Compilați și executați acest cod, atât în Scilab cât și în FORTRAN. Pentru a testa codul FORTRAN, scrieți un program principal în C care să preia parametrii de intrare și să apeleze rutina `permuta`.

EXERCITIUL 8

Folosind comanda `f2c`, traduceți automat în C programul FORTRAN generat la exercițiul 7. Compilați programul rezultat și verificați rezultatul.

EXERCITIUL 9

Scrieți un raport (referat de laborator) care să descrie modul de rezolvare a exercițiilor anterioare și rezultatele obținute. Comentați dificultățile apărute. Anexați codul sursă al tuturor programelor scrise, precum și conținutul fișierelor `Makefile` folosite.

Următoarele exerciții sunt facultative. Vă recomandăm să le faceți, deoarece scopul lor este de a vă familiariza cu o facilitate extrem de utilă pentru productivitatea muncii Dvs. de cercetare: utilizarea rutinelor din bibliotecile matematice profesionale.

EXERCITIUL 10

Scrieți un program principal în limbajul C care apelează o rutină din biblioteca matematică FORTRAN.

Rutina apelată este `QAGI` și face parte din pachetul de rutine `QUADPACK` dedicat integrării numerice [19]. Codul sursă al acestui pachet este depozitat în biblioteca matematică Netlib [14].

Antetul rutinei `QAGI` este:

```

    subroutine qagi(f,bound,inf,epsabs,epsrel,result,abserr,neval,
    *   ier,limit,lenw,last,iwork,work)
c***begin prologue  qagi
c***date written   800101   (yymmdd)
c***revision date  830518   (yymmdd)
c***category no.   h2a3a1,h2a4a1
c***keywords   automatic integrator, infinite intervals,

```

```
c          general-purpose, transformation, extrapolation,
c          globally adaptive
c***author  piessens,robert,appl. math. & progr. div. - k.u.leuven
c          de doncker,elise,appl. math. & progr. div. -k.u.leuven
c***purpose the routine calculates an approximation result to a given
c          integral  i = integral of f over (bound,+infinity)
c                  or i = integral of f over (-infinity,bound)
c                  or i = integral of f over (-infinity,+infinity)
c          hopefully satisfying following claim for accuracy
c          abs(i-result).le.max(epsabs,epsrel*abs(i)).
c***description
c
c          integration over infinite intervals
c          standard fortran subroutine
c
c          parameters
c          on entry
c          f          - real
c                    function subprogram defining the integrand
c                    function f(x). the actual name for f needs to be
c                    declared e x t e r n a l in the driver program.
c
c          bound - real
c                    finite bound of integration range
c                    (has no meaning if interval is doubly-infinite)
c
c          inf      - integer
c                    indicating the kind of integration range involved
c                    inf = 1 corresponds to (bound,+infinity),
c                    inf = -1           to (-infinity,bound),
c                    inf = 2           to (-infinity,+infinity).
c
c          epsabs - real
c                    absolute accuracy requested
c          epsrel - real
c                    relative accuracy requested
c                    if epsabs.le.0
c                    and epsrel.lt.max(50*rel.mach.acc.,0.5d-28),
c                    the routine will end with ier = 6.
c
c          on return
c          result - real
c                    approximation to the integral
c
c          abserr - real
c                    estimate of the modulus of the absolute error,
c                    which should equal or exceed abs(i-result)
```

```
c
c      neval - integer
c              number of integrand evaluations
c
c      ier   - integer
c              ier = 0 normal and reliable termination of the
c                  routine. it is assumed that the requested
c                  accuracy has been achieved.
c              - ier.gt.0 abnormal termination of the routine. the
c                  estimates for result and error are less
c                  reliable. it is assumed that the requested
c                  accuracy has not been achieved.
c
c      error messages
c          ier = 1 maximum number of subdivisions allowed
c                  has been achieved. one can allow more
c                  subdivisions by increasing the value of
c                  limit (and taking the according dimension
c                  adjustments into account). however, if
c                  this yields no improvement it is advised
c                  to analyze the integrand in order to
c                  determine the integration difficulties. if
c                  the position of a local difficulty can be
c                  determined (e.g. singularity,
c                  discontinuity within the interval) one
c                  will probably gain from splitting up the
c                  interval at this point and calling the
c                  integrator on the subranges. if possible,
c                  an appropriate special-purpose integrator
c                  should be used, which is designed for
c                  handling the type of difficulty involved.
c          = 2 the occurrence of roundoff error is
c                  detected, which prevents the requested
c                  tolerance from being achieved.
c                  the error may be under-estimated.
c          = 3 extremely bad integrand behaviour occurs
c                  at some points of the integration
c                  interval.
c          = 4 the algorithm does not converge.
c                  roundoff error is detected in the
c                  extrapolation table.
c                  it is assumed that the requested tolerance
c                  cannot be achieved, and that the returned
c                  result is the best which can be obtained.
c          = 5 the integral is probably divergent, or
c                  slowly convergent. it must be noted that
c                  divergence can occur with any other value
c                  of ier.
c          = 6 the input is invalid, because
```

```
c          (epsabs.le.0 and
c          epsrel.lt.max(50*rel.mach.acc.,0.5d-28))
c          or limit.lt.1 or leniw.lt.limit*4.
c          result, abserr, neval, last are set to
c          zero. except when limit or leniw is
c          invalid, iwork(1), work(limit*2+1) and
c          work(limit*3+1) are set to zero, work(1)
c          is set to a and work(limit+1) to b.
c
c          dimensioning parameters
c          limit - integer
c          dimensioning parameter for iwork
c          limit determines the maximum number of subintervals
c          in the partition of the given integration interval
c          (a,b), limit.ge.1.
c          if limit.lt.1, the routine will end with ier = 6.
c
c          lenw - integer
c          dimensioning parameter for work
c          lenw must be at least limit*4.
c          if lenw.lt.limit*4, the routine will end
c          with ier = 6.
c
c          last - integer
c          on return, last equals the number of subintervals
c          produced in the subdivision process, which
c          determines the number of significant elements
c          actually in the work arrays.
c
c          work arrays
c          iwork - integer
c          vector of dimension at least limit, the first
c          k elements of which contain pointers
c          to the error estimates over the subintervals,
c          such that work(limit*3+iwork(1)),... ,
c          work(limit*3+iwork(k)) form a decreasing
c          sequence, with k = last if last.le.(limit/2+2), and
c          k = limit+1-last otherwise
c
c          work - real
c          vector of dimension at least lenw
c          on return
c          work(1), ..., work(last) contain the left
c          end points of the subintervals in the
c          partition of (a,b),
c          work(limit+1), ..., work(limit+last) contain
c          the right end points,
c          work(limit*2+1), ...,work(limit*2+last) contain the
```

```

c          integral approximations over the subintervals,
c          work(limit*3+1), ..., work(limit*3)
c          contain the error estimates.
c***references (none)
c***routines called  qagie,xerror
c***end prologue  qagi

```

Programul scris de Dvs. trebuie să fie traducerea în limbajul C a programului de test FORTRAN:

```

C 4.5. Calling Program for QAGI
C  -----
C
C          REAL ABSERR,BOUN,EPSABS,EPSREL,F,RESULT,WORK
C          INTEGER IER,INF,IWORK,LAST,LENW,LIMIT,NEVAL
C          DIMENSION IWORK(100),WORK(400)
C          EXTERNAL F
C          BOUN = 0.0E0
C          INF = 1
C          EPSABS = 0.0E0
C          EPSREL = 1.0E-3
C          LIMIT = 100
C          LENW = LIMIT*4
C          CALL QAGI(F,BOUN,INF,EPSABS,EPSREL,RESULT,ABSERR,NEVAL,
C          * IER,LIMIT,LENW,LAST,IWORK,WORK)
C  C  INCLUDE WRITE STATEMENTS
C          STOP
C          END
C
C          C
C          REAL FUNCTION F(X)
C          REAL X
C          F = 0.0E0
C          IF(X.GT.0.0E0) F = SQRT(X)*ALOG(X)/
C          *          ((X+1.0E0)*(X+2.0E0))
C          RETURN
C          END
C

```

Programul de mai sus realizează calculul numeric al integralei:

$$\int_0^{\infty} \frac{\sqrt{x} \ln x}{(x+1)(x+2)} = \pi\sqrt{2} \ln 2 \approx 3.0795717821424 \quad (7.1)$$

De această dată veți primi prin poștă electronică de la instructor codul sursă al subrutinei, dar în viitor va trebui să vă familiarizați cu explorarea Netlib și cu teleîncărcarea prin Internet a subrutinei potrivite.

Indicații

- Declarațiile datelor vor fi traduse în C astfel:

INTEGER → int

REAL → float

DIMENSION → tipul variabilei este stabilit printr-o declarație de tip anterioară, instrucțiunea DIMENSION precizând doar faptul că datele sunt vectori sau matrice și dimensiunile acestora.

- Eventualele variabile nedefinite în sursele FORTRAN se consideră declarate implicit prin prima literă a numelui lor, indiferent că sunt nume de variabile, tablouri sau funcții.

Convenția de declarare implicită este următoarea: variabilele al căror nume începe cu una din literele *i*, *j*, *k*, *l*, *m*, *n* sunt de tip întreg, toate celelalte sunt implicit de tip real.

- Cuvântul cheie EXTERNAL reprezintă o declarație de funcție. În FORTRAN nu este necesar să se precizeze în declarația unei funcții EXTERNAL și parametrii ei.

În C, nu există un cuvânt cheie corespunzător lui EXTERNAL, în schimb, va trebui să declarați funcția (tipul, numele, lista de parametri cu tipurile lor).

- Cuvântul cheie CALL este folosit în FORTRAN pentru apelul subrutinelor.

În C, nu există un cuvânt cheie special, ci apelul unei rutine care nu întoarce o valoare (funcție C de tip void) este făcut doar prin invocarea numelui urmat de o listă a parametrilor actuali.

- Atenție la definirea parametrilor funcției! Înainte de a o scrie, revedeți paragraful 7.5.
- Cuvintele cheie STOP și END din FORTRAN semnifică sfârșitul programului principal, și sunt echivalente acoladei închise (}) de sfârșit al funcției main din programul C.
- Spre deosebire de C, variabilele și instrucțiunile FORTRAN nu sunt “case sensitive” (CALL și call sunt echivalente).

EXERCITIUL 11

Folosind facilitatea Scilab de a apela rutine externe (paragraful 6.10), incorporați în Scilab, executați și verificați în mediul Scilab rutina de calcul al integralei definite pe intervale nemărginite, `qagi`.

EXERCITIUL 12

Scrieți un raport (referat de laborator) care să descrie modul de rezolvare a exercițiilor de utilizare a unei rutine din biblioteca matematică `QUADPACK`.

Descrieți scopul subrutinei `qagi`. Comentați dificultățile apărute. Anexați codul sursă al programelor scrise, precum și conținutul fișierelor `Makefile` folosite.

Capitolul 8

Comunicația între sistemele Unix — Internet —

O caracteristică esențială a sistemelor Unix este faptul că ele oferă implicit suportul software pentru comunicația între calculatoarele conectate. Evident că transmiterea de date între calculatoare depinde și de caracteristicile hardware ale acestora, dar efortul enorm de standardizare și normare (privind nivelul logic sau cel fizic: conectori, nivele de semnale, etc.) permite în prezent realizarea de rețele hibride în care sunt incluse cele mai diverse echipamente (calculatoare, servere de fișiere, echipamente de conectare, imprimante și alte echipamente periferice), provenite de la producători diferiți, pe baza modelului client/server. LMN este primul laborator din țară care a realizat în 1992 o rețea hibridă de stații grafice aplicând acest model.

Rețelele de calculatoare sunt structurate ierarhic pe nivele:

- **LAN** (Local Area Network) – rețeaua dintr-un laborator (cum este `lmn.pub.ro`), departament sau instituție cu dimensiuni cuprinse între 5 m și 30 km;
- **MAN** (Metropolitan Area Network) – rețeaua dintr-un oraș, obținută prin conectarea mai multor LAN-uri, cu dimensiuni între 10 și 100 km;
- **WAN** (Wide Area Network) – rețeaua de la nivelul unei țări (cum este *RoEduNet*) sau chiar continent;
- **GAN** (Global Area Network) – rețeaua la nivel mondial (cum este *Internet*).

Cel mai simplu mod de conectare a două calculatoare este cel *punct-la-punct*, de exemplu folosind interfața serială (comuna majorității sistemelor de calcul) și două fire conductoare prin care datele se transmit serializat (bit cu bit) cu viteze cuprinse între 4 kb/s până la 128 kb/s. Eventual, pentru a folosi liniile telefonice obișnuite trebuie utilizate două echipamente specializate numite *modemuri*.

Rețelele reale au topologii ale conexiunilor mai complicate, ca de exemplu:

- topologie **magistrală**, un fir pe care se pot conecta mai multe calculatoare
- topologie **inel**, la care conexiunea în buclă închisă elimină reflexiile;
- topologie **stea**, în care fiecare sistem este conectat direct la un echipament central numit “*hub*”;
- topologie de **graf**, la care în fiecare nod se conectează un echipament de comunicație numit “*router*” și care stabilește căile prin care va circula mesajul de la sursă la destinație.

Pentru a realiza comunicația este nevoie de un canal de transmisie a datelor realizat fizic pe baza unui suport (mediu) de transmisie. Acesta poate fi:

- pereche de **fire torsadate** (*UTP= Unshielded Twisted Pair*) folosite la LAN-uri;
- **cablu coaxial**, de obicei cu impedanța caracteristică de 50Ω sau cablu TV, folosit la LAN-uri și respectiv MAN-uri;
- cablu **optic**, folosit de la MAN-uri până la GAN-uri;
- unde **radio și microunde**, folosite la MAN-uri;
- unde **infraroșii**, folosite la LAN-uri;
- linii de **telefon** (comutate sau dedicate), folosite la toate nivelele;
- telefonie **celulară** (analogică sau GSM), folosite la toate nivelele;
- **sateliți** (geostaționari sau cu orbită fixă *LEO – Low Earth Orbit*), folosiți la WAN și GAN.

În funcție de suportul folosit, viteza de transmisie a datelor variază de la câțiva kb/s până la Gb/s. În LMN vitezele sunt de 10 Mb/s la rețelele bazate pe cablu coaxial din EB210 și EB213 și de 100 Mb/s la rețeaua cu UTP, în arhitectură stea, din EB212. Universitatea “Politehnica” din București, ca nod central al rețelei Ministerului Educației Naționale – *RoEduNet*, are o rețea bazată pe cablu coaxial galben și fibră optică între clădiri (cca 10 km), conexiunea cu exteriorul realizându-se prin radiomodem (2 Mb/s), antenă de satelit VSAT (1.5 Mb/s), cablu TV (5 Mb/s) și prin modem special de conectare la linie telefonică dedicată (128 kb/s).

Istoria rețelelor de calculatoare Internet începe în perioada războiului rece, în 1969 când *ARPA (Defense Advanced Research Projects Agency)* a finanțat un proiect de conectare sigură a calculatoarelor în rețea pentru transmiterea de informații. Rețeaua *ARPAnet* lega în 1969 trei calculatoare ale unor universități californiene (UCLA, USC și Stanford) cu Universitatea Utah. Rețeaua s-a extins progresiv acoperind tot teritoriul SUA: 8 sisteme în 1970, 15 sisteme în 1971, 35 sisteme în 1972, iar în 1973 a căpătat dimensiune transatlantică prin conexiunile cu Marea Britanie și Norvegia.

Un rol important în dezvoltarea rețelelor de calculatoare l-a avut programul finanțat de *NSF (National Science Foundation)* care a permis realizarea, începând din 1979, a unei rețele dedicate cercetătorilor *NSFNet*. Începând din 1991 (prin legea numită *High Performance Computing – HPC Act*) dezvoltarea acestei rețele a condus la realizarea rețelei naționale americane de cercetare și educație (*National Research and Education Network – NREN*) la care sunt conectate cele mai mari supercalculatoare americane și are o coloană vertebrală (“*backbone*”) de mare viteză numită și “*autostradă informatică*”.

Creșterea numărului de calculatoare conectate a avut loc exponențial (213 în 1981, 535 500 în 1991 și peste 3 milioane în 1994) și continuă și în prezent.

În paralel s-au dezvoltat pe toate continentele rețele dedicate activităților nonprofit sau celor comerciale și respectiv guvernamentale. Serviciile oferite utilizatorilor s-au diversificat continuu: poșta electronică **e-mail** (din 1977), agenție de știri - **Usenet** (1979), hipertext **Web** (1992 la CERN în Elveția) și **browserul Mosaic** (1993 la NCSA în SUA).

Numele Internet s-a impus după 1983, când a fost realizat protocolul **IP** (*Internet Protocol*), care împreună cu **TCP** (*Transport Protocol*) permite conectarea în rețea nu numai a calculatoarelor simple ci și a rețelelor locale de calculatoare (LAN). În acest fel Internet-ul este de fapt nu o rețea de calculatoare ci o rețea de rețele de calculatoare, fiecare având adresă IP proprie (pe 32 biți, deci un total de cca 4 miliarde de adrese cuprinse între 0.0.0.0 și 255.255.255.255). De exemplu adresele calculatoarelor din LMN sunt 141.85.34.xxx. În 1996 a apărut o versiune nouă a protocolului **IPv6**, care adaugă protecții suplimentare de securitate și care extinde lungimea adresei la 128 biți astfel încât practic pe fiecare metru pătrat de pe suprafața locuită a planetei să se poată alocă mii de adrese.

Pentru a asigura flexibilitatea și robustețea necesară unei astfel de dezvoltări s-au făcut eforturi de stabilire a unor standarde și norme internaționale, în special de Organizația Internațională de Standarde *ISO*. Cel mai cunoscut este modelul de transmisie a datelor *OSI (Open System Interconnection)* prin care mesajul este prelucrat succesiv de 7 nivele (fiecare introducând eventual câte un antet special) și prezentate în figura 8.1. În prezent comunicările se realizează folosind mai puține nivele (4 în cazul Internet clasic).

Ideea centrală a transmisiei Internet (TCP/IP) este de a sparge mesajul în pachete de dimensiuni mai mici fiecare având câte un antet în care sunt codificate adresa sursă și cea destinație ale mesajului. Pachetele unui mesaj pot în general circula pe rute diferite iar la destinatar ele se assemblează în ordinea corectă [78].

Un alt aspect este legat de accesul multiplu la un canal de transmisie (bandă radio sau magistrală). Primele cercetări au fost realizate în anul 1970 în cadrul proiectului *ALOHA* de comunicație prin radio între insulele Hawaii. Rezultatele au fost preluate de Xerox PARC, unde în 1976 s-au conectat la un cablu de 1 km lungime peste 150 de stații de lucru personale (la 3 Mb/s) într-o rețea locală numită *Ethernet*. Ulterior protocolul și specificațiile mecanice pentru conectori au fost cuprinse în standardul IEEE 802.3 cu două variante: *Ethernet* (10 Mb/s) și *Fast Ethernet* (100 Mb/s), ultima apărută în 1995.

În tabelul 8.1 se prezintă soluțiile permise de acest standard.

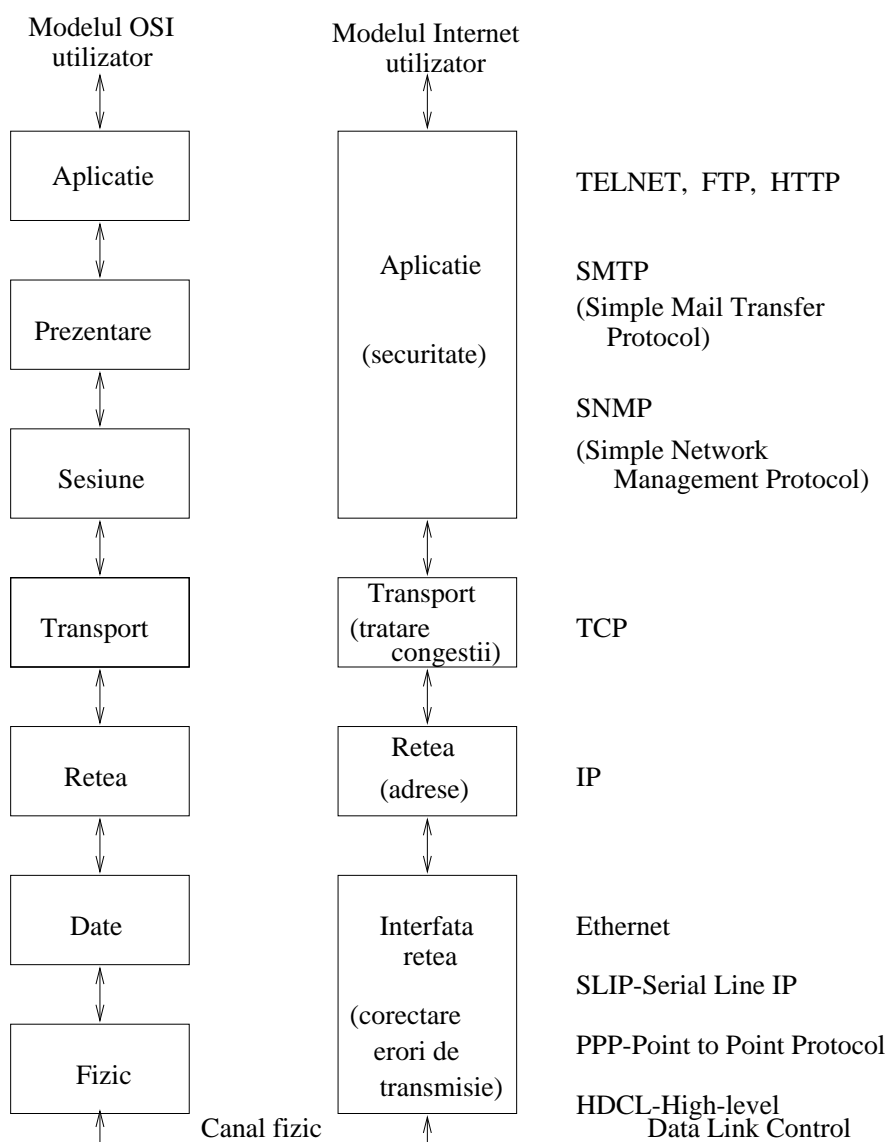


Fig. 8.1: Comparație între modelele OSI și Internet

Orice rețea, indiferent de extensie, are câteva caracteristici definitorii, ca cele descrise în tabelul 8.2.

Dezvoltarea fără precedent a rețelei de calculatoare a impus pentru realizarea interoperabilității adoptarea unor norme tehnice comune. În acest sens, s-a creat în 1992 *Internet Society*, care are o procedură deschisă de redactare a proiectelor de norme, numită *RFC (Request For Comments)* [8]. Aici se mențin definițiile diferitelor protocoale, cum sunt: IP, TCP, FTP, TELNET, SMTP, DNS (Domain Name Server, ce standardizează alocarea numelor de domenii și subdomenii) sau NFS (Network File Sistem, ce permite distribuirea în rețea a unui sistem de fișiere fără ca utilizatorul să fie preocupat de poziționarea fizică a unui fișier).

În România primele încercări de conectare la Internet au avut loc la *Catedra de Calculatoare* din UPB în iulie 1990 (transmisie de poștă electronică pe linie comutată cu Uni-

Tabelul 8.1: Standardul Ethernet IEEE 802.3

Nume	Cablu	Lungime max. segment	Nr. max. noduri / segment	Avantaje
10 Base 5	cablu coaxial gros (galben)	500 m	100	Backbone
10 Base 2	cablu coaxial subțire (negru)	200 m	30	Ieftin
10 Base T	Torsadat	100 m	1024	Ușor de menținut
10 Base F	cablu optic	2000 m	1024	Între clădiri
100 Base T4	Torsadat	100 m	1024	UTP 3
100 Base-TX	Torsadat	100 m	1024	UTP 5
100 Base-F	fibre optice	2500 m	1024	Lungime

Tabelul 8.2: Caracteristicile principale ale unei rețele

Caracteristica	Posibilități
Serviciu	cu sau fără conexiune fixă
Protocol	IP, IPX
Adresare	plată (802) sau ierarhică (IP)
Multitasking	prezent sau absent
Mărime pachete	scurte sau lungi
Calitate	sub control sau nu
Corector erori	da sau nu
Transmisie	cu controlul vitezei sau nu
Controlul congestiei	cu diverse tehnici
Securitate	cu sau fără encriptare
Contabilitate	după timp, cantitate de informație, nr. pachete sau fără

versitatea Tehnică din Darmstadt – THD). În februarie 1993 s-a dat în folosință rețeaua locală a UPB, una din cele mai mari din Europa, care în iunie 1993 a fost conectată prin linie telefonică dedicată la 9.6 kb/s cu Berlin. Domeniul *.ro* a apărut pe harta Internet în anul 1993 prin eforturile *Institutului Central de Informatică* [43], [60].

În 1994 rețeaua UPB a deschis accesul la Internet și altor universități din țară. În prezent rețeaua *RoEduNet* are arhitectură stea cu nodul central în UPB și alți cinci Internet Service Providers în Craiova, Timișoara, Cluj, Tg.Mureș și Iași, oferind servicii pentru practic întreaga comunitate universitară din țară.

Dintre aceste servicii, cel care a contribuit cel mai mult la impactul Internet-ului (până la a se confunda cu acesta) este cel de *Web* [64]. Acest serviciu se bazează pe conceptul de hipertext și dă posibilitatea de a vizualiza documente răspândite în întreaga lume cu subiectele cele mai diverse, dar “legate” logic între ele prin “ancore” (cuvinte sensibile, marcate în document prin culoare sau subliniere și prin a căror indicare se realizează automat transferul la documentul cu care acest cuvânt este legat). Hipertextele sunt scrise într-un limbaj specializat numit *HTML (Hypertext Mark-up Language)* [68].

În acest fel se realizează o bibliotecă electronică cu o topologie asemănătoare unei pânze de păianjen (*Web*), distribuită pe întreg globul (*WWW = World Wide Web*). Această bibliotecă sporește continuu, deoarece orice utilizator are posibilitatea să afișeze informație în acest ciber spațiu. Sistemele pe care sunt memorate documentele se numesc *Web-servere*, iar programul client care permite vizualizarea lor de la distanță se numește *browser* [76]. Operațiunile pe care utilizatorul *Web* le poate realiza sunt:

- navigația
- explorarea
- producția (folosind următoarele tehnici: editare *off-line*, *WYSIWYG* sau prin traducere din alt limbaj)

Documentele pot conține nu numai text ci și imagini (*GIF* și *JPEG*), iar în prezent chiar elemente multimedia: sunete, animații și secvențe video, realizându-se ceea ce se numește “*hipermedia*”. Mai mult, nu toate “documentele” sunt statice, ci unele permit interacțiunea cu utilizatorul (descrișă prin intermediul unor limbaje de programare specializate, cum sunt: *CGI*, *Pearl*, *Java*, *JavaScript*, etc.) [66].

Normele și standardele privind *Web*-ul sunt menținute de *W3C – World Wide Web Consortium* [28], înființat în acest scop în 1994.

Perspectivile deschise de rețeaua de calculatoare Internet și de abordarea client-server au făcut ca filozofia privind utilizarea calculatoarelor să fie total modificată în ultimii ani, iar în perspectivă este așteptată o adevărată revoluție cu mare impact social [41]. Nu întâmplător firma Sun și-a ales deviza “*The Network is the Computer*”.

Obiectivul capitolului constă în însușirea deprinderilor necesare folosirii principalelor servicii Internet, realizate de programe Unix care permit comunicarea directă cu alți

utilizatori, conectarea la un calculator aflat la distanță și respectiv transferul de fișiere de la / spre un calculator aflat la distanță și conectat la Internet:

- Comunicare directă cu alți utilizatori:
 - `elm` — primire și transmitere de mesaje (poștă electronică), care a fost tratat deja în capitolul 3;
 - `write` — comunicare unidirecțională cu alți utilizatori;
 - `talk` — comunicare interactivă (bidirecțională) cu alți utilizatori;
- Conectare (login) pe alte calculatoare:
 - `telnet`
- Transfer de fișiere de pe alte calculatoare:
 - `ftp`
- Alte comenzi utile
 - `mesg`
 - `uuencode`, `uudecode`
 - `gzip`, `compress`
 - `tar`
 - `ping`
 - `finger`
- World Wide Web
 - elemente de bază WWW
 - limbajul HTML
 - coduri HTML

8.1 telnet

Acest utilitar permite conectarea la o altă mașină, folosind un terminal alfa-numeric.

Pentru a testa această comandă, procedați astfel:

- Tastați:

```
telnet mașină
```

unde *mașină* este “numele” unei mașini pe care aveți cont, sau pe care există un cont cu acces public (de exemplu, numele altei mașini decât cea pe care sunteți conectat în laborator). Va urma o procedură standard de login pe acea mașină.

- Din acest moment, sunteți conectat la acea mașină și puteți da comenzi ca și cum v-ați afla în fața unui terminal al acelei mașini (cu observația ca, lucrând pe un terminal alfanumeric, comenzile care implică deschiderea de ferestre grafice sunt fie inaccesibile, fie necesită proceduri speciale).
- Pentru ieșire, tastați

```
logout <↵ >
```

sau

```
exit <↵ >.
```

Veți reveni la promptul mașinii locale.

- Comanda `telnet` fără parametri are ca efect intrarea în programul `telnet`, afișarea promptului specific (`telnet >`) și așteptarea unei comenzi. Puteți folosi comenzile:

```
open mașină: conectarea la mașină
```

```
close: întreruperea legăturii la mașină
```

```
quit: ieșirea din telnet
```

8.2 write

Utilitarul Unix `write` permite comunicarea unidirecțională cu alți utilizatori conectați pe aceeași mașină.

Pentru a testa modul de funcționare a acestei comenzi, procedați astfel:

- Verificați că puteți primi mesaje, cu comanda `mesg <↵ >`. Dacă rezultatul acestei comenzi este mesajul `Is n`, setați `mesg y`.
- Stabiliți un “partener de discuție” și conectați-vă la mașina lui (folosind, de exemplu, `ssh`).

- Verificați dacă utilizatorul este conectat:

```
finger nume
```

sau

```
who
```

- Dacă utilizatorul are deschis cel puțin un terminal pe mașina respectivă, rezultatul este similar cu:

```
irina    ttyp0    Oct 30 10:48    (:0.0)
irina    ttyp1    Oct 30 10:06    (:0.0)
irina    ttyp2    Oct 30 10:06    (:0.0)
```

“Numele” terminalelor pe care lucrează utilizatorul *irina* este *ttyp*.

- Introduceți comanda:

```
write nume terminal
```

(*terminal* poate lipsi), apoi introduceți textul mesajului, de exemplu:

```
test write <↵ >
```

- În acest moment puteți:

- încheia transmiterea mesajului: Ctrl-D sau Ctrl-C
- continua cu scrierea altei linii, terminate cu <↵ >.

Avantajul utilizării *write* în locul poștei electronice (e-mail) este acela că destinatarul primește imediat mesajul, care devine vizibil pe propriul terminal (metoda este sincronă și nu asincronă, nefiind nevoie ca destinatarul să facă vreo acțiune pentru a primi mesajul). Această metodă este mai “agresivă” decât e-mail și se recomandă să o folosiți doar în cazurile în care sunteți convins că destinatarul nu este deranjat. Imaginați-vă ce se întâmplă dacă destinatarul primește 10 mesaje simultan, într-un moment în care este preocupat de rezolvarea unei probleme ce nu suportă amânare.

8.3 talk

Utilitarul *talk* permite comunicarea interactivă (bidirecțională) cu alți utilizatori conectați pe aceeași mașină sau pe mașini diferite.

Pentru a iniția o sesiune *talk* procedați astfel:

- Verificați că puteți primi mesaje, cu comanda *mesg* <↵ >. Dacă rezultatul acestei comenzi este mesajul *Is n*, setați *mesg y*.

- Stabiliți un “partener de discuție” și verificați la ce mașină este conectat, prin comanda:

```
rwho (comandă non-standard, disponibilă în laboratorul EB-212)
```

sau

```
finger nume@mașină
```

- Introduceți comanda:

```
talk nume@mașină terminal
```

(*terminal* poate lipsi). Veți vedea pe ecran mesajul:

```
[Waiting for your party to respond]
```

Dacă corespondentul nu răspunde, va apărea de câteva ori mesajul:

```
[Ringing your party again]
```

după care se renunță.

- În momentul în care corespondentul a acceptat să converseze, ecranul se va împărți în două, în partea de sus apărând textul tastat de Dvs, în partea de jos textul introdus de cealaltă persoană.
- Când doriți să încheiați conversația, tastați:

```
bye
```

```
Ctrl-C
```

Pentru a răspunde la o “invitație” `talk` procedați astfel:

- Pe ecran va apărea un mesaj de tipul:

```
Message from Talk_Daemon@lmnsun at 12:52 ...
```

```
talk: connection requested by irina@lmnsun.lmn.pub.ro.
```

```
talk: respond with: talk irina@lmnsun.lmn.pub.ro
```

- Răspundeți cu comanda indicată pe ultima linie a mesajului:

```
talk irina@lmnsun.lmn.pub.ro
```

(Atenție, `irina@lmnsun.lmn.pub.ro` se înlocuiește cu adresa corespondentului Dvs.)

- Puteți lua inițiativa încheierii conversației (deși nu este politicos, deoarece nu ați fost cea/cel care a lansat-o!). Pentru aceasta tastați:

```
bye
```

```
Ctrl-C
```

8.4 ftp

Utilitarul `ftp` (File Transfer Protocol) permite conectarea la o altă mașină cu scopul de a transfera fișiere spre / de la acea mașină.

- Introduceți comanda:

```
ftp ftp.lmn.pub.ro
```

Observație: `ftp.lmn.pub.ro` poate fi înlocuit cu orice altă adresă cu facilități de `ftp`.

- Veți vedea un mesaj similar cu:

```
220 lmnroute FTP server (Version wu-2.4(11) Mon Jul 10 19:14:15 GMT+0300
1995) ready.
```

```
Name (ftp.lmn.pub.ro:irina):
```

Introduceți numele de utilizator `anonymous` și ca parolă, adresa Dvs. de poștă electronică (`nume@lmn.pub.ro`).

De la caz la caz, numele de login dedicat accesului public poate fi `anonymous`, `guest`, sau alt nume. În general, vi se solicită o parolă, dar există locații `ftp` unde nu este nevoie de parolă.

- Din acest moment aveți la dispoziție o serie de comenzi pentru listarea conținutului directorilor, deplasarea în alte directoare, transferul de fișiere, etc. O listă selectivă a acestor comenzi este prezentată mai jos.
- Ieșiți din `ftp` cu una din comenzile: `bye` sau `quit`

Listă selectivă de comenzi ftp

<code>ascii</code>	- setează tipul de transfer de fișiere ca "ASCII"
<code>binary</code>	- setează tipul de transfer de fișiere ca "binar"
<code>cd director</code>	- schimbarea directorului curent, pe mașina de la distanță ("remote")
<code>lcd director</code>	- schimbarea directorului curent, pe mașina locală ("local")
<code>dir director</code> sau	- afișare conținut director pe mașina de la distanță
<code>ls director</code>	
<code>put fis_local</code>	- copiază fișierul local pe mașina de la distanță
<code>get fis_dist</code>	- copiază fișierul de pe mașina de la distanță pe mașina locală
<code>mput fis_local</code>	- asemănător cu <code>put</code> dar pentru mai multe fișiere. De exemplu: <code>mput *.c</code>
<code>mget fis_dist</code>	- asemănător cu <code>get</code> dar pentru mai multe fișiere
<code>hash</code>	- afișează câte un caracter <code>#</code> pentru fiecare bloc de date transferat; permite astfel vizualizarea evoluției transferului
<code>open</code>	- inițierea unei sesiuni <code>ftp</code> , de la promptul <code>ftp</code>

<code>close</code>	- încheie sesiunea cu mașina de la distanță, fără să părăsească <code>ftp</code>
<code>user</code>	- setează numele de utilizator <code>ftp</code> pe mașina de la distanță (util în cazul în care ați greșit numele de utilizator sau parola)
<code>bye</code>	- încheie sesiunea <code>ftp</code>

8.5 Alte comenzi utile, legate de transferul de fișiere și de comunicație

- `rlogin` *mașină* sau `ssh` *mașină*

Permite conectarea la o altă mașină, folosind un terminal alfa-numeric. Se recomandă utilizarea comenzii `ssh` care este mai sigură (encripțează parola).

- `mesg`

setarea acceptării / neacceptării mesajelor din exterior

- `gzip`, `gunzip`:

compactarea/decompactarea fișierelor. Fișierele compactate rezultate au extensia `.gz`

- `compress`, `uncompress`:

compactarea/decompactarea fișierelor. Fișierele compactate rezultate au extensia `.Z`. Comanda `compress` este mai puțin eficientă (din punctul de vedere al reducerii dimensiunii fișierului) decât `gzip`.

- `uuencode`, `uudecode`:

transformarea fișierelor binare în fișiere ASCII și respectiv revenirea la forma binară a fișierelor. Se folosesc pentru transmiterea unor fișiere pe “căi” care nu acceptă decât formatul text (ASCII), cum ar fi poșta electronică.

- `tar`:

arhivare fișiere, directoare, etc. Cu ajutorul acestei comenzi, un sistem structurat format din mai multe fișiere este arhivat într-un singur fișier care include toate informațiile pentru a putea reconstitui structura inițială la dezarhivare.

Exemple:

```
- tar cvf arhiva.tar test*
```

Crează fișierul `arhiva.tar`, care conține arhiva tuturor fișierelor și directoarelor al căror nume începe cu `test`, plasate în directorul curent.

Parametrii au semnificațiile:

`c` (“create”) – crearea unei noi arhive.

`v` (“verbose”) – determină afișarea numelor fișierelor incluse, pe măsură ce ele sunt incluse în arhivă.

f (“file”) – indică faptul că următorul parametru este numele fișierului–arhivă.

– **tar tvf arhiva.tar**

Afișează (“list”) conținutul arhivei plasată în fișierul **arhiva.tar**.

– **tar xvf arhiva.tar**

Extrage (“extract”) fișierele cuprinse în arhiva **arhiva.tar**, reconstituind structura de fișiere și directoare.

8.6 Elemente de bază WWW

Pentru a accesa serviciile *Internet–World Wide Web* (*WWW*) este necesar un “browser” — un program care citește și afișează pe ecranul calculatorului Dvs. informațiile aflate pe alte calculatoare conectate la Internet.

Browselele cele mai folosite sunt *Netscape* (figura 8.2) și *Lynx* (cel din urmă, dedicat terminalelor alfa–numerice, deci neoferind facilități grafice).

Orice informație pe Internet este accesibilă prin “adresa” sa, numită *URL* (*Uniform Resource Locator*).

Un URL este compus din trei părți:

protocol://server/cale

protocol indică browser–ului cum să interpreteze informația găsită pe mașina ***server*** în fișierul indicat prin ***cale***. Dacă parametrul ***cale*** reprezintă un director și nu un fișier, atunci browserul va prelua în mod implicit fișierul ***Index.html*** plasat în directorul indicat prin ***cale***.

protocol poate fi:

http — hiperdocument în format HTML

gopher — pentru lansarea unei sesiuni **gopher**

ftp — pentru lansarea unei sesiuni **ftp**

file — pentru accesarea unui fișier care poate fi în format HTML sau nu.

Exemplu de URL (adresa paginii **WWW** și a celei **ftp** a Laboratorului de Metode Numerice):

http://www.lmn.pub.ro/

http://archive.lmn.pub.ro/

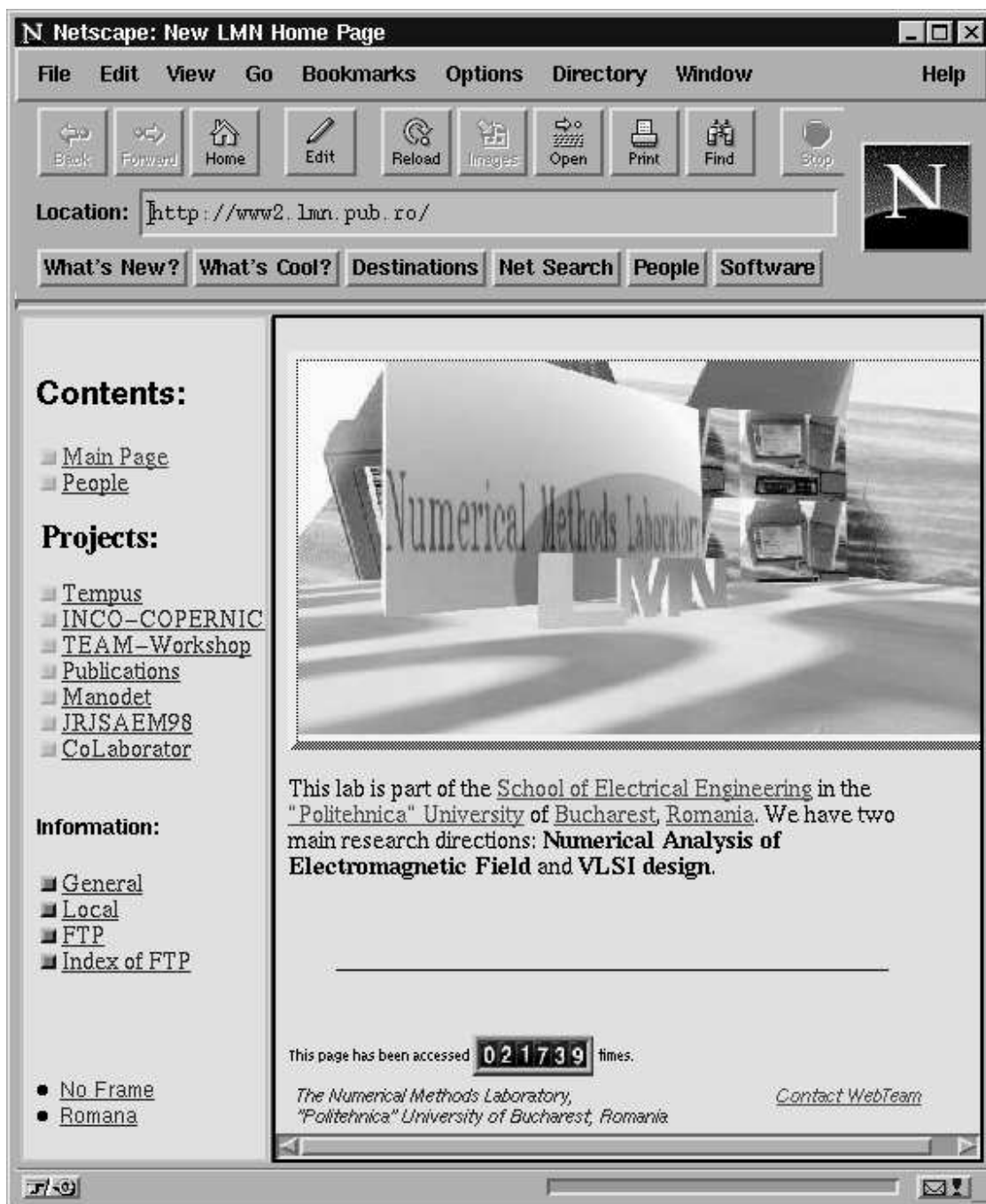


Fig. 8.2: Ecranul browser-ului Netscape

8.7 Structura limbajului HTML

Documentele pe Internet au un format special, numit *HTML* (*HyperText Markup Language* = limbaj bazat pe marcaje pentru hipertext). Noțiunea de hipertext desemnează o structură a informației care nu este liniară, ca de exemplu într-un film, ci arborescentă, semănând cu o carte pe care un utilizator o poate “frunzări” după voie și chiar este ajutat să o facă în mod eficient prin index sau alte metode de trimitere.

Documentele hipertext conțin coduri speciale care sunt interpretate de browser, coduri referitoare la:

- formatarea textului
- includerea în text a unor imagini
- preluarea unor informații de pe Internet,
...., etc.

Trebuie menționat că:

- orice fișier ASCII este afișabil (vizibil) din orice browser; nu este deci neapărat nevoie să știi HTML pentru a afișa un document pe Web;
- totuși, un document formatat HTML arată mai atractiv. În plus, în el se pot include legături (“links”) la alte pagini (a se citi “fișiere”!) plasate pe mașina Dvs. sau pe orice altă mașină accesibilă din lume.

Un document formatat HTML se numește în jargonul curent “pagină Web” (*Web page*, *WWW page*), iar mulțimea paginilor unei instituții se numește *Web site*. Fișierul care conține documentul se numește “fișier sursă” și este un fișier text, având extensia standard `.html` sau `.htm`.

Pagina de “bază” a unei instituții sau a unui utilizator este numită adesea “home page”. Pentru cazul unui utilizator, este de obicei plasată în directorul `~/public_html/` și are numele `Index.html`.

Conceptual, limbajul HTML se aseamănă cu \LaTeX , amândouă satisfăcând prescripțiile *SGML* (*Standard General Markup Language*). Diferă doar sintaxa, dar conceptele care stau la baza celor două limbaje sunt similare. Din acest motiv, traducerea între aceste limbaje este relativ ușoară.

8.8 Coduri de formatare HTML

Codurile de formatare (în limba engleză: “tags”) servesc la marcarea unei “indicații de punere în pagină” de către browser a documentului HTML.

Un **tag** este structurat astfel: semnul `<`, urmat de un cod (numit “directivă”), urmat de semnul `>`:

```
<directivă>
```

Tag-urile sunt de obicei în perechi, de exemplu:

```
<H1> Text </H1>
```

(un titlu de nivel 1, definit prin directiva H1).

Observați similitudinea cu:

```
\begin{document} Text \end{document}
```

din \LaTeX .

Un document HTML are structura minimală:

```
<HTML>

<HEAD>
<TITLE>   Titlul paginii </TITLE>
</HEAD>

<BODY>
. . .
</BODY>

</HTML>
```

8.8.1 Principalele perechi de coduri de formatare HTML

- Heading pentru întregul document

```
<HEAD> ... </HEAD>
```

- Titlu document

```
<TITLE> ... </TITLE>
```

- Titluri secțiuni

```
<H1> ... </H1>
```

```
⋮
```

```
<H6> ... </H6>
```

- Formatare text

`<I> Italic </I>` — *Italic*

` Bold ` — **Bold**

`<U> Subliniat </U>` — Subliniat

- Paragrafe

`<P> ... </P>`

Observație: `</P>` poate lipsi.

- Linie nouă

`
`

- Listă nenumerotată

``

` ... `

` ... `

⋮

``

Observație: `` poate lipsi.

- Listă numerotată

``

` ... `

` ... `

⋮

``

Observație: `` poate lipsi.

- Linie orizontală

`<HR>`

- Includere imagine

``

Exemplu:

``

(Atenție! Numele fișierului `-/graphics/im1.gif-` este doar un exemplu!)

Opțiunile `ALIGN` și `ALT` pot lipsi.

Parametrul *aliniere* poate avea valorile: `top`, `middle`, `bottom` și se referă la alinierea figurii față de textul care urmează.

Opțiunea ALT este folosită de către browser-ele care nu permit afișări grafice. Textul *alternativă* se va afișa pe ecran, în locul unde ar fi trebuit să apară figura.

Exemplu:

```
<IMG SRC = "poza.gif" ALIGN ="middle" ALT = "GIF image"> O imagine
```

- Includerea unei legături la un alt document Internet

```
<A HREF = "URL"> text </A>
```

Perechea de tag-uri este acum <A> și . Textul *text* este afișat diferit față de restul textului; forma cursorului se va modifica atunci când acesta se mută pe *text*.

Exemplu:

```
<A HREF = "alt_inf.html"> Apasati aici pentru alte informatii </A>
```

8.9 Un exemplu de document HTML

Iată un exemplu de utilizare a codurilor prezentate în paragraful anterior:

```
<HTML>
<HEAD>
<TITLE> Pagina de test </TITLE>
</HEAD>
```

```
<BODY>
```

```
<H1> Un titlu mare </H1>
<H3> Un titlu mai mic </H3>
```

Acum urmeaza o lista:

```
<UL>
<LI> mere
<LI> pere
<LI> portocale
</UL>
```

Si inca o lista

```
<OL>
<LI> Primul
<LI> Al doilea
<LI> Al treilea
</OL>
```

O linie orizontala:

```
<HR>
```

Gata!

```
</BODY>
```

... și rezultatul – ceea ce se vede atunci când folosiți browser-ul Netscape pentru a vizualiza documentul:



8.10 Exerciții

EXERCITIUL 1 — “Navigare”

Timp de 15 minute, navigați pe Internet folosind Netscape. Veți constata că minutele trec foarte repede în cursul acestei activități – uitați-vă din când în când la ceas!

Vă recomandăm să atingeți câteva puncte: pagina LMN <http://www.lmn.pub.ro>, pagina UPB <http://www.pub.ro> și pagina Netscape <http://www.netscape.com>.

EXERCITIUL 2 — “Explorare”

Folosind programele de căutare a informațiilor plasate în Internet, (o listă este disponibilă la <http://www.lmn.pub.ro/wwwsearch.html>) realizați o documentare pe Internet privitoare la biblioteci de subrutine matematice.

Începeți să vă construiți propriul “carnet de adrese” Internet: salvați adresele Internet utile folosind butonul **Bookmarks** → **Add bookmark**.

Folosiți editorul dedicat (**Window** → **Bookmarks**) pentru a vă organiza adresele Internet. Creați foldere pe domenii (**Item** → **Insert Folder**), în care mutați (apăsați butonul din stânga și deplasați mouse-ul) adresele corespunzătoare.

Figura 8.3 prezintă aspectul ferestrei Netscape de gestionare a adreselor Internet.

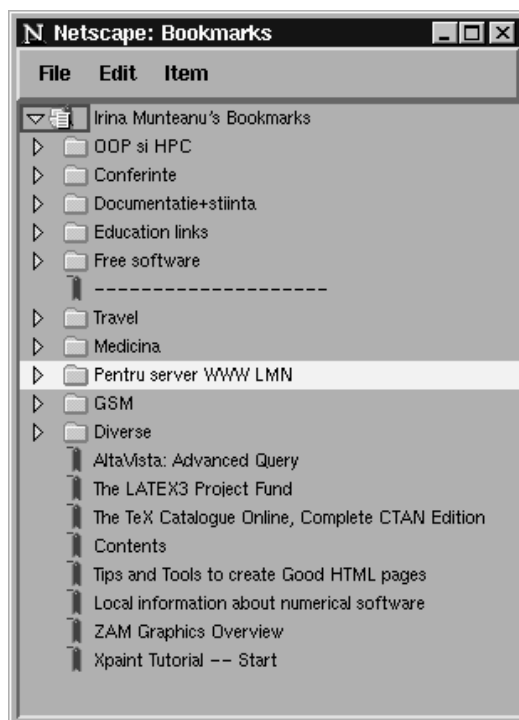


Fig. 8.3: Fereastra “Bookmarks” din Netscape

EXERCITIUL 3 — “Creare de informație electronică”

Scopul acestui exercițiu este de a vă crea propria pagină WWW, pornind de la un model.

1. Creați directorul `public_html` în directorul Dvs. `$HOME`.
2. Accesați cu Netscape adresa: `http://www.lmn.pub.ro/~irina`.
3. Salvați documentul ca “sursă HTML” în fișierul `~/public_html/Index.html`
4. Accesați cu Netscape fișierul nou creat, selectând adresa:
`http://www2.lmn.pub.ro/~nume`
unde *nume* este numele Dvs. de utilizator.
5. Deschideți o nouă fereastră și editați cu `vi` sau cu `emacs` fișierul `Index.html`
6. Identificați principalele părți ale documentului și tag-urile folosite (comparați cu lista de tag-uri din paragrafele următoare).
7. Modificați fișierul astfel încât să conțină propriile Dvs. informații. Salvați fișierul fără să ieșiți din editor.
8. Verificați rezultatul modificărilor folosind comanda “Reload”.
9. Dacă rezultatul este satisfăcător, închideți fișierul `Index.html`.

EXERCITIUL 4

Deschideți pagina Dvs. WWW folosind editorul Netscape (butonul “Edit”).

Familiarizați-vă cu facilitățile acestui editor. Experimentați diferite fonturi, linii de separare, includere de figuri, etc.

Includeți în această pagină ancore la referatele de laborator realizate până acum în format ASCII sau PostScript.

EXERCITIUL 5

Realizați o pagină cu preferințe (**TOP 10**), în care descrieți de exemplu cele mai interesante 10 instrumente Unix, (eventual la fiecare cele mai utile 10 comenzi), cele mai interesante 10 pagini Web pe care le-ați descoperit, cele mai interesante 10 cărți, etc.

În pagina Dvs. home includeți o ancoră la pagina TOP 10. În acest fel, pagina de Web va reflecta personalitatea Dvs.!

EXERCITIUL 6

Folosiți programul `latex2html` pentru a traduce automat în format HTML referatul de laborator referitor la mediul de programare C în Unix. În pagina proprie WWW, adăugați o ancoră la acest document HTML și verificați că el este vizibil din Netscape.

EXERCITIUL 7

Adăugați adresa paginii Dvs. WWW în fișierul `logo` inclus automat în orice mesaj electronic.

Trimiteți un mesaj instructorului, în care îi comunicați finalizarea **ultimului exercițiu** al acestei cărți!

Anexa A

Teste finale de laborator

Testul 1 de verificare a cunoștințelor — Tutorial Unix

1. Creați (temporar, pentru durata acestui test) un director numit `dir1` în directorul Dvs. `$HOME`. În acest director veți lucra pe parcursul testului.
2. Transferați în directorul `dir1` fișierul `test_unix` aflat pe mașina ciuc în directorul `/tmp`.
3. Copiați fișierul `test_unix` în fișierul `test.txt`
4. Editați fișierul `test.txt`:
 - înlocuiți toate aparițiile cuvântului `test` cu cuvântul `text`.
 - înlocuiți orice succesiune de două spații (blancuri) cu un singur blank
 - ștergeți toate liniile care conțin cuvintele `de sters`
 - salvați și ieșiți din editor.
5. Copiați fișierul `test.txt` în fișierul `test1.txt`
6. Listați conținutul directorului `dir1`, redirectând ieșirea (afișarea) în fișierul `dir`
7. Trimiteți prin poștă electronică două mesaje pe adresa `irina`:
 - unul cuprinzând fișierul `test1.txt`
 - unul cuprinzând fișierul `dir`
8. În mod normal, 10 minute vă sunt suficiente pentru a termina acest test. **Aveți la dispoziție 40 minute.**

Succes !

Testul 2 de verificare a cunoștințelor — Tutorial Unix

1. Creați (temporar, pentru durata acestui test) un director numit `dir1` în directorul Dvs. `$HOME`. În acest director veți lucra pe parcursul testului.
2. Transferați în directorul `dir1` fișierul `test_unix` aflat pe mașina ciuc în directorul `/tmp`.
3. Redenumiți fișierul `test_unix` ca `test.tex`
4. Editați fișierul `test.tex` pentru a fi un document \LaTeX :
 - folosiți stilul `article`
 - adăugați titlul `Test de verificare a cunoștințelor`
 - adăugați numele Dvs. (ca autor) și data de astăzi
 - adăugați la sfârșit formula numerotată:

$$y = \sum_{k=1}^n c_k x_k$$

- salvați și ieșiți din editor.
5. Compilați și eliminați eventualele erori.
 6. Trimiteți prin poștă electronică un mesaj pe adresa `irina`, cuprinzând fișierul `test.tex`
 7. În mod normal, 10 minute vă sunt suficiente pentru a termina acest test. **Aveți la dispoziție 40 minute.**

Succes !

Testul 3 de verificare a cunoștințelor — Tutorial Unix

1. Creați (temporar, pentru durata acestui test) un director numit `dir1` în directorul Dvs. `$HOME`. În acest director veți lucra pe parcursul testului.
2. Scrieți un macro Scilab care:

pentru `i = 1, 3`
 preia de la consolă două matrice (sau vectori, sau scalari ...)
 efectuează produsul lor.
3. Rulați programul și verificați corectitudinea
4. Transferați în directorul `dir1` fișierul `test_unix` aflat pe mașina ciuc în directorul `/tmp`.
5. Redenumiți fișierul `test_unix` ca `test.tst`
6. Listați conținutul directorului `dir1`, redirectând ieșirea (afișarea) în fișierul `dir`
7. Trimiteți prin poștă electronică două mesaje pe adresa `irina`:
 - unul cuprinzând fișierul în care ați scris programul Scilab
 - unul cuprinzând fișierul `dir`
8. În mod normal, 10 minute vă sunt suficiente pentru a termina acest test. **Aveți la dispoziție 40 minute.**

Succes !

Testul 4 de verificare a cunoștințelor — Tutorial Unix

1. Creați (temporar, pentru durata acestui test) un director numit `dir1` în directorul Dvs. `$HOME`. În acest director veți lucra pe parcursul testului.
2. Folosind `gnuplot`, reprezentați pe același grafic funcțiile $\sin(x)$ și $\cos(x^2)$, cu următoarele setări:
 - eticheta axei Ox: `x`
 - eticheta axei Oy: `f(x)`
 - graficul pentru funcția sin: puncte și linii, titlu `sinus`
 - graficul pentru funcția cos: puncte titlu `cosinus`
3. Salvați graficul în format Encapsulated PostScript (`eps`) în fișierul `grafic.eps`
4. Transferați în directorul `dir1` fișierul `test_unix` aflat pe mașina ciuc în directorul `/tmp`.
5. Redenumiți fișierul `test_unix` ca `test.nou`
6. Copiați fișierul `test.txt` în fișierul `test1.txt`
7. Listați conținutul directorului `dir1`, redirectând ieșirea (afișarea) în fișierul `dir`
8. Trimiteți prin poștă electronică două mesaje pe adresa `irina`:
 - unul cuprinzând fișierul `grafic.eps`
 - unul cuprinzând fișierul `dir`
9. În mod normal, 10 minute vă sunt suficiente pentru a termina acest test. **Aveți la dispoziție 40 minute.**

Succes !

Testul 5 de verificare a cunoștințelor — Tutorial Unix

1. Pe parcursul acestui test veți lucra în directorul `public_html` (aflat în directorul Dvs. `$HOME`). Dacă directorul `public_html` nu există, creați-l.
2. Transferați în directorul `dir1` fișierul `test_unix` aflat pe mașina ciuc în directorul `/tmp`.
3. Redenumiți fișierul `test_unix` ca `test.html`
4. Editați fișierul `test.html` astfel încât să devină un fișier în limbaj HTML:
 - adăugați marcajele (tag-uri) absolut necesare într-un document HTML
 - adăugați titlul **Test de verificare a cunoștințelor**
 - adăugați numele Dvs. la începutul textului, ca heading de nivel 1.
 - adăugați un marcaj (tag) pentru trecerea la un nou paragraf, astfel încât textul pe ecran să cuprindă doar două paragrafe.
 - adăugați o linie orizontală la sfârșitul textului
 - salvați și ieșiți din editor.
5. Verificați cu NetScape că rezultatul modificărilor este cel așteptat.
6. Listați conținutul directorului curent (`public_html`), redirectând ieșirea (afișarea) în fișierul `dir`
7. Trimiteți prin poștă electronică două mesaje pe adresa `irina`:
 - unul cuprinzând fișierul `test.html`
 - unul cuprinzând fișierul `dir`
8. Ștergeți fișierele `test.html` și `dir`.
9. În mod normal, 10 minute vă sunt suficiente pentru a termina acest test. **Aveți la dispoziție 40 minute.**

Succes !

Anexa B

Regulamentul Laboratorului de Metode Numerice

Instrucțiuni de respectat la folosirea rețelei de calculatoare locale din Laboratorul de Metode Numerice (lmm.pub.ro)

Oricine intră în laboratorul de metode numerice și are deschis un cont activ pe rețeaua locală de calculatoare trebuie să cunoască următoarele reguli.

1. La fiecare intrare in sistem (login time) veți primi diferite mesaje de care trebuie sa țineți cont. Acestea sunt scrise de către administratorul de sistem și au forma:

Incercati sa stergeti din fisierele care nu mai sunt necesare
#####
2. Încercati să vă țineți cât mai actuale datele personale cu ajutorul comenzii chfn și a fisierului .plan
3. Accesul cu diskete este total interzis. Orice problemă care cere introducerea unei diskete din exteriorul laboratorului va fi soluționată numai de administrator.
4. Accesul fizic la calculatoare se va face numai prin intermediul tastaturii și a mouse-ului. Nu acționați nici un alt buton. Nu deconectați/conectați cabluri.
5. Întrucât aveți și acces direct la rețeaua Internet va trebui sa fiți un bun locuitor al acesteia. Este interzis accesul în orice calculator din lume dacă nu aveți un cont personal. Orice încercare de pătrundere frauduloasă în orice sistem sau de producere a oricăror daune asupra datelor din alte sisteme se va pedepsi foarte drastic.

6. Parola este personală și este interzisă folosirea contului de către orice altă persoană în afară de posesorul de drept al acestuia. Nu incredințați parola nimănui altcuiva. Schimbați parola cel puțin lunar cu ajutorul comenzii `yppasswd`. Alegeți o parolă bună, în primul rând foarte diferită de numele de cont. Aceasta trebuie să aibă 6-8 caractere, litere și cifre.
7. Este interzisă înlesnirea accesului altor persoane în contul propriu via `.rhosts`.
8. Încercați să vă rezolvați singuri problemele prin metode soft care nu cer drepturi de root. Când sunteți siguri că nu se poate, scrieți o hârtie cu datele amănunțite ale problemei, cum ați încercat s-o rezolvați și numele d-voastră de cont. Mai sigur prin mail, la adresa:

`help@lmn.pub.ro`
9. Orice resursă costă. De aceea, nu aveți dreptul să folosiți imprimanta decât pentru a imprima lucrările, referatele, etc. realizate de Dvs. în cadrul activităților de laborator sau de elaborare a lucrării de diplomă. Este interzisă imprimarea documentelor de pe Internet, a cursurilor, etc. fără aprobarea șefului laboratorului, prof. Daniel Ioan.

Administratori de rețea:

Bogdan Brătucu (`help@lmn.pub.ro`, `bbg@lmn.pub.ro`)

Victor Stănescu (`help@lmn.pub.ro`, `bruno@lmn.pub.ro`)

Anexa C

Evoluția Laboratorului de Metode Numerice – LMN

Istoria Laboratorului de Metode Numerice - LMN a început în decembrie 1983, când în sala EB-206 a noului local al Politehnicii Bucureștene (pe atunci I.P.B.) s-a instalat primul terminal “la distanță” la sistemul de calcul FELIX C256 al Centrului de Calcul IPB operat cu pachetul de teleprelucrare “SESAM”. Acest terminal era de tip Centronix, un fel de teletype, cu tastatură și imprimare pe hârtie “de calculator”. Chiar dacă acum este un centru de cercetare și învățământ avansat în domeniul utilizării profesionale a calculatoarelor în ingineria electrică, laboratorul și-a păstrat numele tradițional, provenit de la cursul de “Metode Numerice”, oferit studenților din al II-lea an al Facultății de Electrotehnică, începând din 1990.

Dotarea și preocupările laboratorului au urmărit dezvoltarea impresionantă a tehnologiei informației, marcând câteva evenimente, premiere naționale. Trebuie menționate: instalarea primelor calculatoare cu sistem de operare CP/M pentru microprocesoare pe 8 biți –8080 și Z80– (FELIX M118, TPD în 1985), primele calculatoare cu procesoare de 16 biți (8086) ce utilizau sistemul de operare MS-DOS (FELIX PC și, din 1989, compatibile cu IBM PC-AT), trecerea prin toate formatele de dischete flexibile de la 8” și 5” până la 3^{1/2}”, dar și conectarea în 1984, prin linie telefonică dedicată de peste 10 km, a unor terminale inteligente la puternicul –pentru acea vreme– minicalculator WANG VS80, aflat în proprietatea ICPE.

Tehnologiile de imprimare au evoluat și ele, de la imprimanta cu rozetă Robotron la imprimanta cu ace și, din 1990, la imprimanta cu laser. În aceste condiții s-au dezvoltat produse program pentru mai multe aplicații, cum sunt: DISTRIB (analiza circuitelor electrice cu parametri distribuți, dezvoltat în mare parte pe un sistem de calcul foarte puternic de tip CRAY la Electricité de France – DER, începând încă din anul 1978), TEZED (program de editare a tezaurului de termeni ai Comisiei Electrotehnice Internaționale, executabil pe WANG în 1985), LOCAP (program de analiză la nivel de porți a circuitelor logice, combinaționale și secvențiale, dezvoltat sub CP/M în perioada 1985-1988) și FAP (program pentru analiza câmpurilor 2D, dezvoltat inițial sub MS-DOS, începând din 1988).

Un eveniment cu implicații majore în viața laboratorului l-a constituit derularea din 1991 a proiectului TEMPUS JEP 2717, finanțat de Comisia Europeană cu peste 600.000 ECU și dedicat înființării unei școli postuniversitare de inginerie electrică asistată de calculator, cu două direcții: CAD electromagnetic și VLSI. În acest proiect au fost implicate cele mai importante personalități europene din domeniul calculului de câmp electromagnetic și a fost evaluat ca proiect model, continuând în perioada 1995-1996 ca rețea de excelență europeană – TEMPUS JEN 2717. Folosind bugetul acestui proiect, s-a realizat saltul din lumea DOS în universul UNIX, realizându-se în 1992, pentru prima dată în țară, o rețea locală de stații grafice, cu caracter hibrid, bazată pe tehnologia client/server, care încorporează (în sala EB-213) platforme produse de Hewlett Packard (HP 9700), Digital (DEC Station 5000), Sun (SPARC Station) și Silicon Graphics (Iris Indigo). Legătura la Internet, în 1993, prin proiectul Technische Hochschule Darmstadt – THD la care laboratorul a contribuit activ, a reprezentat o altă deschidere majoră. Accesul la rețeaua RoEduNet și, prin intermediul ei, la conectivitate Internet completă, constituie una din cele mai importante resurse disponibile în LMN.

Pachetele de programe achiziționate pentru aceste platforme (sistemele de operare de tip Unix, mediile de dezvoltarea programelor, pachetele de analiză a câmpului electromagnetic FLUX2D, MSC-EMACS și MEGA, precum și pachetele de proiectare VLSI: Cadence, HILO, HSPICE și SOLO) și documentația care le însoțea a avut un impact major atât în activitatea de cercetare, cât și în cea de învățământ avansat (pregătirea doctoranzilor). Cu această ocazie s-au proiectat și realizat practic (prin serviciul EURO-CHIP, din cadrul programului ESPRIT al Uniunii Europene, la care ne-am afiliat din 1992) primele circuite integrate VLSI în tehnologia CMOS de $1\mu\text{m}$. Lista principalelor pachete software instalate în LMN este prezentată în anexa D.2.

Schimbul de profesori și studenți în ambele sensuri a condus la recunoașterea internațională a laboratorului, nu numai în marile universități europene, ci și în SUA și în Japonia, unde avem parteneri în peste 20 laboratoare universitare de mare prestigiu. La aceasta a contribuit, pe lângă proiectele educaționale și de cercetare comune, și organizarea în România a două importante evenimente științifice: Conferința Internațională de Educație Asistată de Calculator a Inginerilor – CAEE'93 (cu 120 participanți din 20 țări) și primul seminar Româno-Japonez în domeniul electromagnetismului și mecanicii aplicate RJJSAEM'96 (cu 30 participanți japonezi).

LMN este printre primele entități din țară care și-au realizat propria pagină de Web (în 1994), în care, în premieră națională au fost postate în 1996 materiale didactice interactive, dezvoltate în Java.

Un alt proiect care a contribuit esențial la dezvoltarea LMN îl constituie proiectul TEMPUS S-JEP 9122 cu titlul *Comunicare Audiovizuală bazată pe Tehnologia Informației*, finanțat de Comisia Europeană și de Fundația Soros pentru o Societate Deschisă. În cadrul acestui proiect, desfășurat în perioada 1995-1998, s-a dezvoltat laboratorul de comunicare profesională (EB-212) bazat pe prima rețea Fast Ethernet din U.P.B., care încorporează calculatoare Multimedia PC-compatibile (cu procesoare Intel Pentium și 32 MB RAM) și două servere, unul Windows NT 4.0 și altul Linux. Tot în cadrul acestui proiect s-a obținut din partea Hogeschool van Utrecht o donație de 17 stații grafice (DEC

VAX), folosite ca Xterminale în laboratorul de inginerie matematică (sala EB-210) precum și 12 calculatoare PC ce completează rețeaua de 12 calculatoare IBM PS/2 (EB-207) finanțată de Ministerul Educației Naționale și folosită în mod curent de studenții din anul II, la disciplina de Metode Numerice. Lista completă a echipamentelor instalate în LMN este prezentată în anexa D.1.

O importantă resursă a LMN pusă la dispoziția studenților și cadrelor didactice este Centrul de documentare și training (EB-206) în care sunt disponibile peste 500 cărți și ultimele numere din 32 de publicații științifice din domeniile: matematici aplicate, calculatoare, câmp electromagnetic și circuite electronice. Lista bibliografică a acestui manual reprezintă doar o parte din documentația pe care o puteți consulta (din păcate fără a o putea împrumuta) în cadrul laboratorului.

Inițiată încă din 1995 de LMN, *Inițiativa Națională pentru Calcule de Înaltă Performanță* urmărește instalarea în rețeaua *RoEduNet* a primului supercalculator din țară (care sperăm să fie menționat cât mai curând în celebra listă TOP500 redactată semestrial de Jack Dongarra și care conține cele mai puternice calculatoare din lume). Realizarea, împreună cu alți parteneri, a proiectului “CoLaborator – Bază de Cercetare cu Utilizatori Multipli în Domeniul Calculelor de Înaltă Performanță”, finanțat începând cu anul 1997 de către Banca Mondială, dă speranța că acest deziderat va fi îndeplinit.

În afara acestui proiect, LMN este implicat în mai multe proiecte europene sau internaționale de cercetare. Acest lucru ar fi imposibil fără permanenta împletire cu activitatea educațională.

LMN organizează anual cicluri de “autoinstruire” în utilizarea calculatoarelor, deschise studenților din anii mici ai tuturor facultăților din UPB. Cei mai talentați studenți sunt ulterior implicați în proiectele de cercetare ale laboratorului.

În afara cursului de Metode Numerice, LMN oferă mai multe cursuri în anii mari ai secției de Inginerie Electrică Asistată de Calculator de la Facultatea de Electrotehnică și organizează anual pregătirea aprofundată (anul VI) pentru 20 studenți ce provin de la diferite facultăți de profil electric din țară (electrotehnică, electronică, automatică, energetică sau calculatoare), direcțiile de specializare fiind “CAD electromagnetic” și microelectronică. În plus, oferă un curs în limba engleză pentru Departamentul de Științe Inginerești din UPB.

În laborator își desfășoară cercetările cca.10 doctoranzi în ingineria electrică (în domeniile: analiza numerică a câmpului electromagnetic, optimizarea dispozitivelor electromagnetice, algoritmi paraleli și distribuiți, calcule de înaltă performanță, efecte de câmp în circuite electronice, defectoscopie electromagnetică nedistructivă, știință și inginerie computaționale, educație asistată de calculator, tehnici numerice în ingineria electrică, proiectarea asistată de calculator în ingineria electrică, proiectare electronică automată, analiza circuitelor cu parametri distribuiți, elemente electromagnetice de circuit, analiza sistemelor cu microunde, probleme fizice cuplate, etc.) și sunt primiți în stagiul studenți și doctoranzi din țară (Iași, Cluj, Brașov, etc.) și din străinătate (Franța, Marea Britanie, Germania, Spania, etc.).

Cercetarea științifică desfășurată în laborator este finanțată de Ministerul Educației Naționale (CNCSU și CNFIS prin proiectele Băncii Mondiale), Ministerul Cercetării

și Tehnologiei, industria din România, instituții și organizații din străinătate (proiecte PECO, COPERNICUS sau proiecte bilaterale, ca RJSAEM).

În ciuda aparenței, LMN este un “laborator deschis” (a nu se confunda cu necesitatea respectării unor reguli stricte de acces, securitate și comportament atunci când se operează într-un mediu tehnologic avansat și complex!), oferind suport și acces la resurse oricărui membru din comunitatea academică (și cu precădere celor talentați, ambițioși și creativi) cu singurele condiții ca acesta să aibă un proiect clar (obiective, termene, metode de abordare și un buget), să respecte regulamentul laboratorului și să-și asume obligația morală de a menționa sprijinul LMN – “acknowledgment” în documentele elaborate în acest proiect. Mediul realizat în LMN (echipamente, software, documentație, servicii, competențe interdisciplinare) este similar cu cel al celor mai importante laboratoare similare din lume și creează condițiile emulației și creativității specifice mediului academic. Fără caracterul său interdisciplinar, deschis intensei cooperări internaționale, acest mediu n-ar putea fi menținut. Dacă accepți exigențele internaționale asupra nivelului necesar cercetării științifice universitare, poți contribui și tu la succesul laboratorului și implicit la succesul tău în carieră.

Credem că prezintă interes câteva din concluziile experienței LMN în utilizarea tehnologiei avansate a informației:

- O importanță deosebită în cercetare o au instrumentele folosite, acestea trebuie să fie dintre cele mai puternice și eficiente, efortul făcut pentru învățarea utilizării lor profesionale fiind din plin răsplătit. Mediul folosit trebuie să fie totuși relativ stabil, deoarece schimbarea prea rapidă a versiunilor sau tehnicilor scade eficiența cercetării.
- Achiziționarea de licențe software este la fel de importantă ca achiziționarea de echipamente (hardware); rezultatele au fost mult mai bune rezervând o mare parte a bugetului pentru software, datorită accesului la documentație și la ultimele versiuni ale pachetelor de programe, lucru ce a modificat mentalitatea utilizatorilor și a crescut productivitatea lor.
- Un impact deosebit în utilizarea tehnicii moderne de calcul îl are conectarea la rețeaua Internet; în acest fel sunt disponibile informații extrem de valoroase care altfel ar fi fost foarte dificil sau imposibil de obținut, dar folosirea Internet-ului nu trebuie făcută numai din poziția de consumator, ci prin producerea de informație generată local se ușurează parteneriatul și integrarea cercetării românești în ansamblul european și global.
- Ingineria electrică are în prezent un puternic caracter interdisciplinar (electrotehnică, electronică, sisteme automate, calculatoare) și se bazează pe solide cunoștințe de matematică. Deciziile strategice privind temele de cercetare și tehnicile ce vor fi folosite trebuie să fie luate astfel încât să se “intuiască” valul domeniului încă înainte ca acesta să ajungă la vârf, când e prea târziu de recuperat decalajul. Păstrarea echilibrului între tradiție și migrarea către temele moderne finanțate generos reprezintă secretul unui bun management științific. Deoarece numărul ideilor

esențialmente noi este mai mic decât s-ar părea la prima vedere, valorificarea creatoare a unei tradiții solide generează suficientă flexibilitate și asigură succesul scontat.

- Componenta umană este mult mai importantă decât orice calculator, software sau documentație; ea este încurajată prin lucrul în echipe interdisciplinare, sprijinirea ideilor tinerilor talentați, crearea unei emulații specifice mediilor academice, respectarea libertății academice, a creativității și inițiativei individuale, contactele personale cu partenerii din străinătate, identificarea și participarea la viața unei comunități științifice de calitate prin participarea la evenimentele periodice (din străinătate, dar și atragerea lor în România!), proiecte comune și valorificarea rezultatelor în publicațiile științifice.
- România are o capacitate limitată de a absorbi resursele financiare disponibile pe plan mondial pentru cercetare și acest lucru se datorește deficitului de deprinderi manageriale moderne și mentalităților deplorabile (credem că respectarea principiului “minimei seriozități” este la fel de importantă sau poate chiar mai importantă decât “*minima moralia*” sau decât aspectele juridice, economice sau politice).

Ne exprimăm încrederea că istoria LMN nu se va opri aici. Ultimele noutăți le puteți afla din pagina noastră de Web: <http://www.lmn.pub.ro>.

Anexa D

Resursele Laboratorului de Metode Numerice

D. 1 Resurse hardware

D. 2 Resurse software

LMN își desfășoară activitatea în sălile:

- EB-206 - Centrul de documentare și training
- EB-207 - Laboratorul de inițiere în metode numerice
- EB-210 - Laboratorul de inginerie matematică
- EB-211 - Secretariat, relații cu publicul
- EB-212 - Laboratorul de comunicare profesională
- EB-213 - Laboratorul de inginerie electrică asistată de calculator

D.1 Resurse hardware

Rețeaua de calculatoare a LMN (subdomeniul `lmn.pub.ro`) are topologia din figurile D.1 – D.4.

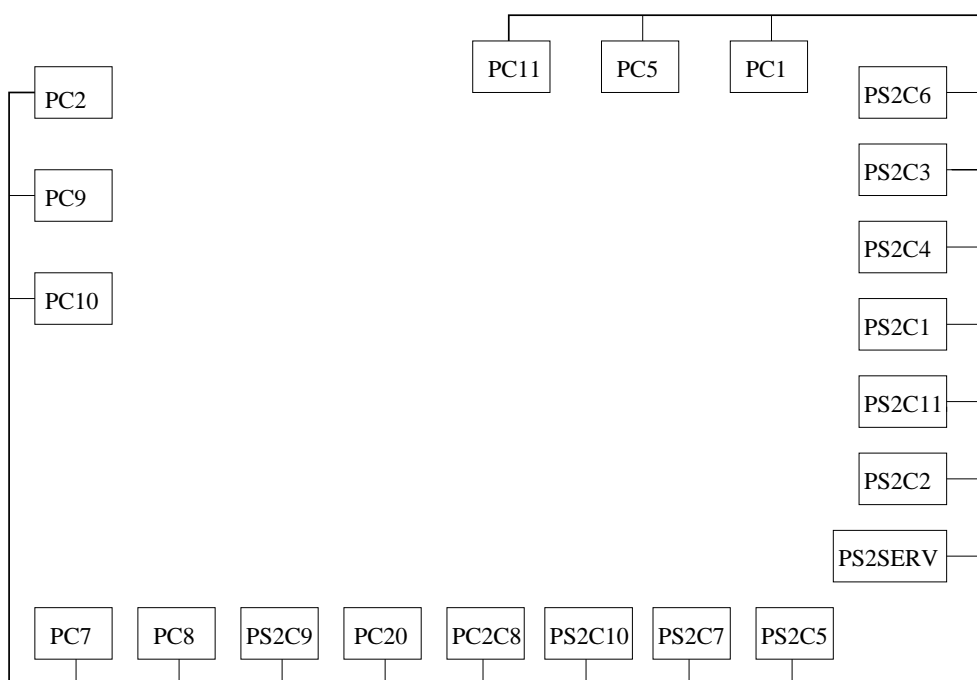


Fig. D.1: Topologia rețelei EB-207

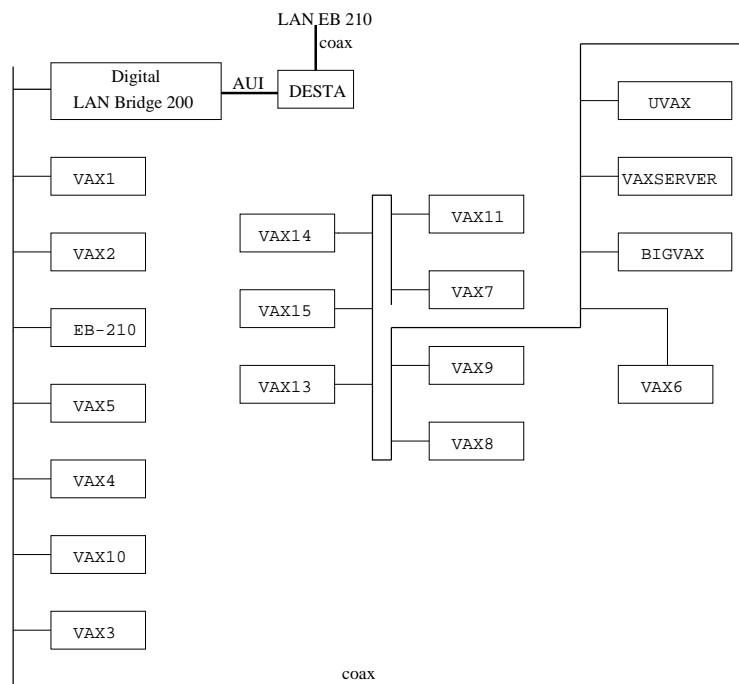


Fig. D.2: Topologia rețelei EB-210

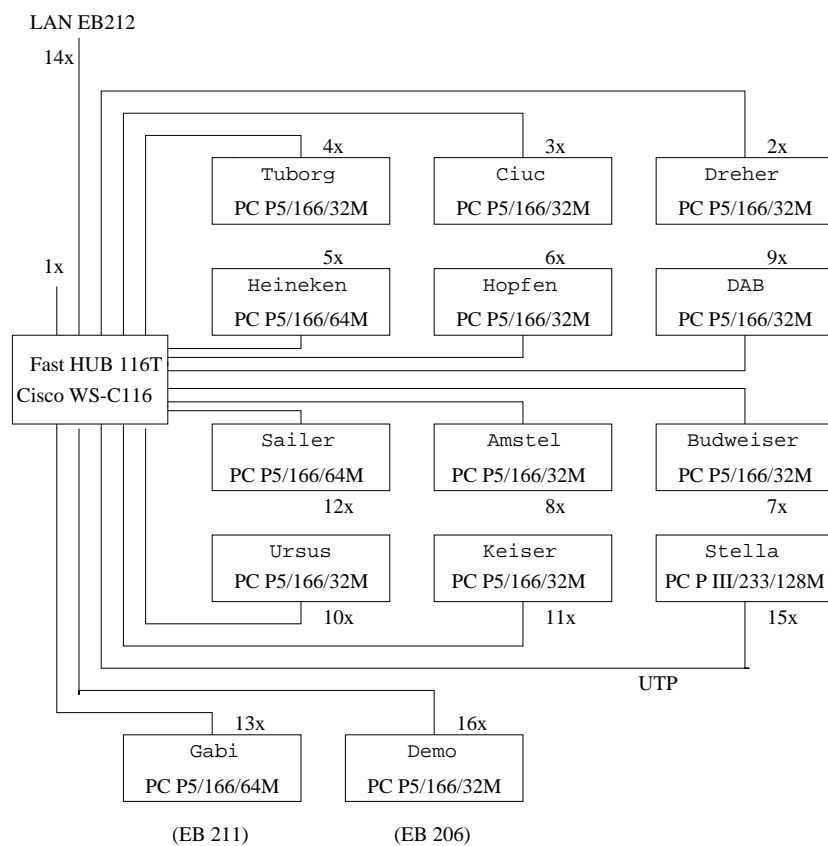


Fig. D.3: Topologia rețelei EB-212

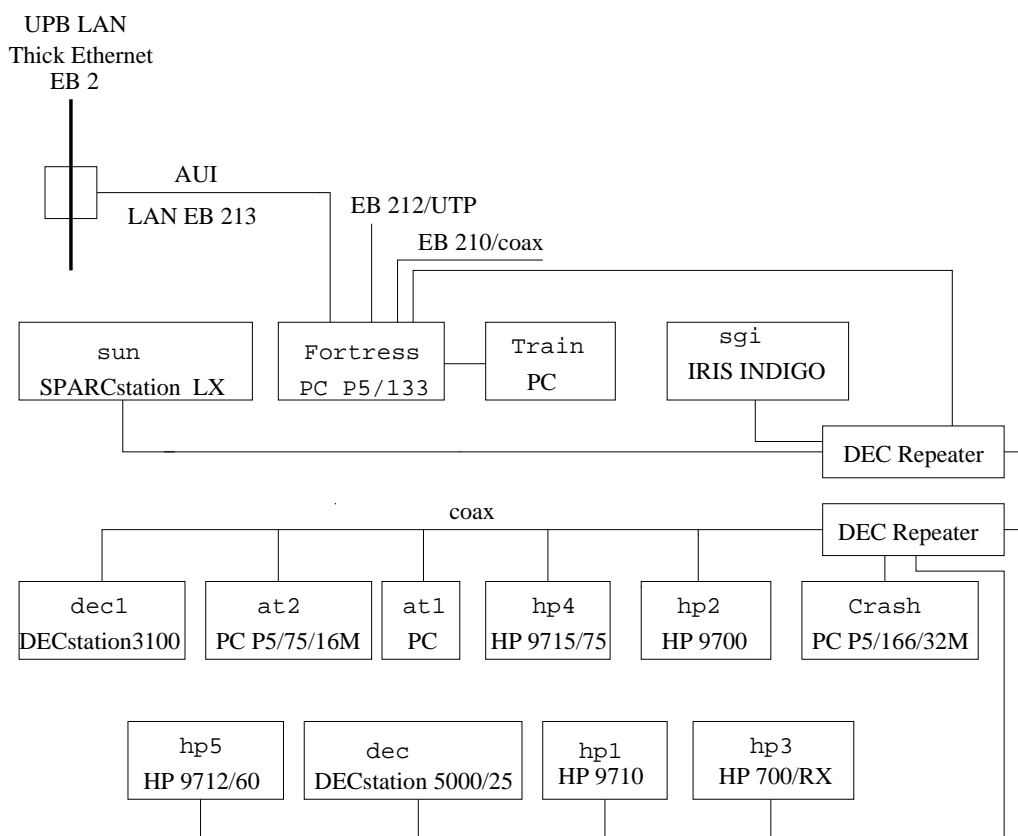


Fig. D.4: Topologia rețelei EB-213

Principalele caracteristici ale sistemelor instalate sunt prezentate în tabelul de pe paginile următoare.

Nume sistem	Model	Sala	Procesor	Monitor	RAM [MB]	HDD [B]	Periferice	Adresa IP 141.85.34.xxx
lmndec	DEC Station 5025	EB-213	RISC	color 21"	40M	820M	95Mb TK tape	4
lmnhp1	HP 9710	EB-213	RISC	color 21"	32M	9 GB	CD-ROM	2
lmnhp2	HP 9705	EB-213	RISC	gray 19"	24M	140M	2 GB DAT tape	1
lmnhp3	HP 700/RX	EB-213	RISC	gray 19"	–	–	–	3
lmnhp4	HP 9715/75	EB-213	RISC	color 21"	64M	540M	HP LaserJet 4	7
lmnhp5	HP 9712/60	EB-213	RISC	color 19"	32M	1G	–	10
lmnsgi	SGI-IRIS INDIGO	EB-213	RISC R4000	color 21"	32M	2.08G	CD-ROM	5
lmnsun	Sun SPARC Station LX	EB-213	RISC SPARC	color 21"	32M	270M	CD- ROM, 150 Mb QIC tape	6
Crash	PC	EB-213	Pentium/166	color 17"	32M	5.8G	CD-ROM	56
at1	PC	EB-213	Pentium/75	color 17"	16M	400M	–	11
at2	PC	EB-213	Pentium/75	color 17"	16M	400M	–	12
Train	PC	EB-213	Pentium/75	color 17"	16M	1.2G	–	8
Fortress	PC	EB-213	80486/100	monocrom 14"	32M	420M	–	129, 9, 193, 65, 34
Sailer	PC	EB-212	Pentium/166	color 17"	64M	4G	Epson Stylus color 3000, MPC	142
Amstel	PC	EB-212	Pentium/166	color 17"	32M	1624M	MPC	138
Dreher	PC	EB-212	Pentium/166	color 17"	32M	1624M	MPC	132
Kaiser	PC	EB-212	Pentium/166	color 17"	32M	1624M	MPC	141
Ursus	PC	EB-212	Pentium/166	color 17"	32M	1624M	MPC	140
Ciuc	PC	EB-212	Pentium/166	color 17"	32M	1624M	MPC	133
DAB	PC	EB-212	Pentium/166	color 17"	32M	1624M	MPC	139
Budweiser	PC	EB-212	Pentium/166	color 17"	32M	1624M	MPC	137
Hopfen	PC	EB-212	Pentium/166	color 17"	32M	1624 M	MPC	135
Tuborg	PC	EB-212	Pentium/166	color 17"	32M	2.1G	MPC	134

Nume sistem	Model	Sala	Procesor	Monitor	RAM [MB]	HDD [B]	Periferice	Adresa IP 141.85.34.xxx
Stella	PC	EB-212	Pentium II/233	color 21"	128M	9G	Scanner Umax Astra 1200, CD-ROM REC, Video	145
Heineken	PC	EB-212	Pentium/166	color 21"	64M	7G	HP LaserJet 5MP, Jaz	131
Gabi	PC	EB-211	Pentium/166	color 17"	64M	1.3 G	HP LaserJetIIP	143
Demo	PC	EB-206	Pentium/166	color 17"	32M	1624 M	MPC	146
EB-210	PC	EB-210	Pentium/166	monocrom 14"	64M	4 G	-	199
VAX1	VAX 3100	Station EB-210		color 21"	16M	120M	-	201
VAX2	VAX 3100	Station EB-210		color 21"	16M	120M	-	202
VAX3	VAX 3100	Station EB-210		color 21"	16M	120M	-	203
VAX4	VAX 3100	Station EB-210		color 21"	16M	120M	-	204
VAX5	VAX 3100	Station EB-210		color 21"	16M	120M	-	205
VAX6	VAX 3100	Station EB-210		color 21"	16M	120M	-	206
VAX7	VAX 3100	Station EB-210		color 21"	16M	120M	-	207
VAX8	VAX 3100	Station EB-210		color 21"	16M	120M	-	208
VAX9	VAX 3100	Station EB-210		color 21"	16M	120M	-	209
VAX10	VAX 3100	Station EB-210		color 21"	16M	120M	-	210

Nume sistem	Model	Sala	Procesor	Monitor	RAM [MB]	HDD [B]	Periferice	Adresa IP 141.85.34.xxx
VAX11	VAX Station 3100	EB-210		color 21"	16M	120M	-	211
VAX13	VAX Station 4000/60	EB-210		color 21"	16M	120M	-	213
VAX14	VAX Station 4000/60	EB-210		color 21"	16M	120M	-	214
VAX15	VAX Station 4000/60	EB-210		color 21"	16M	120M	-	215
VAXSERVER	Microvax 3800	EB-210		monocrom 14"	16M	240M	Scanner	212
UVAX	Microvax II	EB-210		monocrom 14"	-	-	DEC Printer	222
BIGVAX	Microvax 4800	EB-210		monocrom 14"	-	-	CD-ROM	200
PS2 server	IBM PS/2 65SX	EB-207	286	monocrom 12"	2/4M	105 Mb	Printer IBM PS/1	-
PS2C1	IBM PS/2/30	EB-207	286	monocrom 12"	2M	-	-	-
PS2C2	IBM PS/2/30	EB-207	286	monocrom 12"	2M	-	-	-
PS2C3	IBM PS/2/30	EB-207	286	monocrom 12"	2M	-	-	-
PS2C4	IBM PS/2/30	EB-207	286	monocrom 12"	2M	-	-	-
PS2C5	IBM PS/2/30	EB-207	286	monocrom 12"	2M	-	-	-
PS2C6	IBM PS/2/30	EB-207	286	monocrom 12"	2M	-	-	-
PS2C7	IBM PS/2/30	EB-207	286	monocrom 12"	2M	-	-	-
PS2C8	IBM PS/2/30	EB-207	286	monocrom 12"	2M	-	-	-
PS2C9	IBM PS/2	EB-207	286	monocrom 12"	2M	-	-	-
PS2C10	IBM PS/2/30	EB-207	286	monocrom 12"	2M	-	-	-
PS2C11	IBM PS/2/30	EB-207	286	monocrom 12"	2M	-	-	-
PC1	Dec Station 300 digital	EB-207	386SX	color 14"	4M	40 Mb	-	-
PC2	Digital	EB-207	386	color 14"	4M	-	-	-
PC5	Dec Station 320 digital	EB-207	386	color 14"	2M	115 Mb	-	-

Nume sistem	Model	Sala	Procesor	Monitor	RAM [MB]	HDD [B]	Periferice	Adresa IP 141.85.34. <i>xxx</i>
PC7	Digital 320	EB-207	386	color 14"	6M	115 Mb	-	-
PC8	PC 655B3	EB-207	386	color 14"	3M	40 Mb	-	-
PC9	Digital 316	EB-207	386	color 14"	3M	40 Mb	-	-
PC10	Digital 316	EB-207	386	color 14"	5M	-	-	-
PC11	Dec Station 320 digital	EB-207	386	color 14"	4M	-	-	-
PC20	Tulip pc	EB-207	386SX	monocrom 12"	2M	80 Mb	-	-

D.2 Resurse software

Pe sistemele prezentate în figurile D.1 — D.4 sunt instalate mai multe pachete de programe, multe dintre ele accesibile în rețea, dintre care cele mai importante sunt:

1. Software de sistem

- sistem de operare HP-UX pentru stațiile grafice **hp**
- sistem de operare Ultrix pentru stațiile grafice **dec**
- sistem de operare Sun OS pentru **sun**
- sistem de operare Irix pentru **sgi**
- sistem de operare Linux pentru PC din EB-206, EB-210, EB-212 și EB-213 (versiunea 2.0.35 la data publicării)
- interfața grafică (GUI) de tip XWindow/Motif pentru toate sistemele de tip Unix
- sistem de fișiere distribuit NFS pentru toate mașinile
- conexiune Unix-PC prin Digital Pathworks și PC-NFS
- sisteme de operare MS Windows NT (server și workstations) pentru PC (EB-212)
- sisteme de operare MS-DOS, MS Windows 3.1 pentru PS/2 (EB-207)
- NetWare 3.11, sistem pentru rețeaua PS/2 (EB-207)

2. Software de dezvoltare

- Compilator Ultrix C (**dec**)
- Compilator Sun C (**sun**)
- Compilator HP-UX C/C++ (**hp1**)
- Compilator Dec Fortran (**dec**)
- Compilator HP-UX Fortran77 (**hp1**)
- Compilatoarele GNU C, C++, Fortran v2.8.1 (pentru toate sistemele Unix)
- Compilator Turbo Pascal v6.0/Windows
- Compilator Turbo C++ v3.0 /Windows
- Editoare **vi** și **emacs** v2.0.4 pe toate sistemele Unix
- Alte editoare de text specifice fiecărui sistem ca: **Vue-Pad** (**hp**), **XPSView** (**sgi**)
- Medii de depanare și dezvoltare: **gdb** (Linux, **hp4**, **hp5**), **xxgdb**, **ddd** v2.2 (Linux), **dbx** (**dec**, **sgi**, **sun**), **ctrace** (**dec**, **sgi**), **xdb** (**hp1**), **gldebug** (**sgi**), **xwpe** (Linux), **RHide** (Linux, în mod text)
- HP-UX C/C++ Developer's kit (**hp1**)
- Interpretoare scripturi: **Tc/Tkl**, Java

- Biblioteci matematice sursă Fortran (NAG, Linpack, Eispack, acces la Netlib)
- Biblioteci grafice (Xlib, Sun-Direct Xlib, Open GL, Starbase-HP, PIX-HP)
- Medii CASE generatoare ferestre: DEC Vuit, DEC Fuse (`dec`)
- Biblioteci de funcții pentru calcule distribuite: PVM, MPI, PETSc CORBA, LAM, MPICH, NetSolve (EB-212)

3. Software matematic și ingineresc

- Mathematica (2 licențe pe `sgi`)
- Scilab (EB-210, 212, 213)
- MATLAB, SIMULINK, LabVIEW V3.1 Student Ed., MAPLE sub MS-Windows
- CANM - pachet de programe pentru instruirea asistată de calculator în domeniul metodelor numerice dezvoltat în LMN (EB-207)

4. Analiza și proiectarea dispozitivelor electromagnetice (CAD, FEM)

- AutoCAD pentru `hp`, `sgi`, `sun`, MS-Windows
- MSC/EMAS - MacNeil Schwendler - Analiza numerică a câmpului electromagnetic 3D (`hp1`)
- FLUX2D - CEDRAT - Analiza numerică a câmpului electromagnetic 2D (`hp1`)
- MEGA - Donat de Bath University - Analiza câmpului electromagnetic 3D (`hp`)
- FAP - dezvoltat în LMN pentru Analiza câmpurilor 2D, disponibil sub Windows și Linux

5. Proiectarea și analiza microsystemelor (VLSI)

- CADENCE - mediu integrat pentru proiectarea circuitelor VLSI care permite descrierea comportamentală (Hardware Description Language) și structurală, sinteza (inclusiv de nivel înalt), extragerea parametrilor, simularea sistemului și editarea măștilor (o licență Edge pe `dec` și o licență OPUS pe `hp`)
- Licențe pentru tehnologiile European Silincon Structure (ES2): 1.5, 1.0, 0.7 μm
- SOLO 1400 - mediu integrat pentru proiectarea rapidă a circuitelor ASIC în tehnologiile ES2 (10 licențe pe `sun`)
- HILO produs de GENRAD, simulator pentru proiectarea logică (`hp` și `dec`)
- HSPICE produs de META - simulator de circuite la nivel de tranzistori (licențe pe `hp`, `dec`, `sun`)
- CATHEDRAL, mediu pentru proiectarea sistemelor SDP (Digital Signal Processing) (`hp2`)
- ALTERA MAX2PLUS - mediu integrat pentru proiectarea ariilor de porți logice programabile - FPGA (HP și MS-Windows)

- XILINX - mediu integrat pentru proiectarea ariilor de porți programabile FPGA (2 licențe pe MS-Windows)
- Medii de proiectare sub Linux: CACD, CHIPMUNK, MAGIC, OCEAN, UCB-TOOLS, ALLIANCE, SIS, LLL, VERIWELL (EB212)
- SPICE - Berkeley Free Software (3e2) pe dec și Linux
- Pspice - demo sub MS Windows pentru rețeaua PS/2 (EB-207)
- ELCUT - analizor circuite electrice (EB-207)

6. Software de comunicare

- Poșta electronică (elm, pine, MH, mailx)
- Browsere Web și Internet (Mosaic , Netscape)
- Alte instrumente pentru explorare Internet (Xgopher, Xarchie)

7. Software pentru vizualizarea științifică, pregătirea documentelor și producție multimedia

- Pregătire grafice (GNUplot, IRIS Explorer)
- Pregătire figuri (Xfig, XPaint, GIMP 1.0)
- Pregătire documente (TeX, LaTeX, BibTeX, GhostView, LaTeX2html)
- Pregătire prezentări și documente electronice (IRIS Showcase, Acrobat Reader, x11amp, raplayer sub Linux, iar sub MS-Windows: MS-Office97 Professional Ed. Romanian Vers., Word97, Excel97, Powerpoint97, Access97, MS Works, Corell Draw 6.0, 3D Studio MAX, Adobe Photoshop 3.0, Adobe PhotoDeluxe, Adobe Premiere 4.2, Cubasis Audio 1.6, Macromedia Director 4.0, Acrobat Reader 3.0, Visual Cafe, Easy CD-PRO, Miro VIDEO, Miro 3DFX,

O mare parte a comenzilor și instrumentelor din lista de mai sus sunt descrise pe parcursul lucrării.

D.3 Interfața grafică a butoanelor mouse-ului

Apelul pachetelor software descrise în anexa D.2 poate fi realizat utilizând interfața grafică (click cu butoanele mouse-ului pe fundalul ecranului).

În cele ce urmează sunt prezentate meniurile de comenzi disponibile prin intermediul mouse-ului sub Linux și pe stațiile grafice.

D.3.1 Meniuri de comenzi disponibile sub Linux

Meniul “Toolbox” (fig. 1.2) permite accesul la:

- Terminal nou

- NetScape
- Utilitare pentru gestionarea fișierelor
 - TkDesk
 - Midnight Commander (emulează Norton Commander)
 - Xfilemanager
 - Xfm
- Blocarea ecranului
- Calculator de buzunar
- Captură grafică de ferestre
- Vizualizarea încărcării sistemului
- Distrugere fereastră
- Spații de lucru
- Poștă electronică

Meniurile de comenzi (fig. D.5) disponibile sub Linux și accesibile prin click cu butonul stâng pe fundal sunt (EB-210, EB-212):

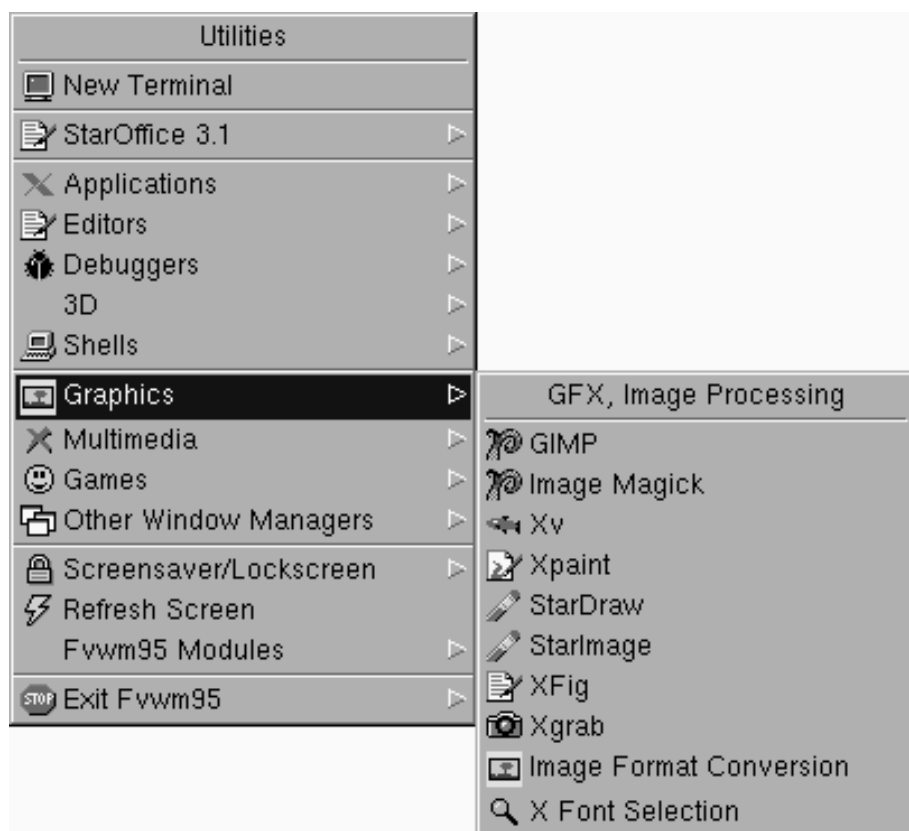


Fig. D.5: Meniul principal Linux

- Terminal nou
- Star Office

- Editor texte - Star Writer
- Editor desene - Star Draw
- Editor figuri - Star image
- Calculator - Star Calc
- Grafice - Star Chart
- Editor ecuații - Star Math
- Aplicații
 - Browser Web
 - * Netscape 3.04
 - * Netscape 4.03
 - * Lynx 2.8 (text)
 - Editoare
 - Depanatoare
 - Cititor PDF
 - Cititor PS
 - Editor bitmap
 - Documentare Man
 - Calendar
- Editoare
 - XEmacs
 - vi
 - xvi
 - joe
 - xedit
- Depanatoare
 - DDD
 - Manual DDD
 - Xxgdb
 - GNU gdb
- 3D
 - Editor Scene VRML (sced)
 - Manual sced
 - Tutorial sced
 - Modelare 3D (nmm)

- Shelluri
 - XTerm
 - Rxwt (VT100)
 - ETerm
- Grafică
 - GIMP
 - Image Magick
 - XV
 - Xpaint
 - XFig
 - Xgrab
 - Conversie format imagine
 - Selecție fonturi X
- Multimedia
 - player
 - CD
 - audio mixer
 - MPEG 3
 - efecte vizuale
- Jocuri
- Managere de ferestre
- Blocare ecran, fundal ecran
- Reinițializare ecran
- Module Fvwm95
- Ieșire din sesiune

D.3.2 Meniuri de comenzi disponibile în rețeaua de stații grafice

Meniurile de comenzi disponibile în rețeaua heterogenă de stații grafice cu procesoare RISC permit accesul de la distanță, în tehnologia client/server. Meniuri distincte sunt mapate pe cele trei butoane ale mouse-ului.

Meniuri de comenzi mapate pe butonul stâng

Prin click cu butonul stâng pe fundalul ecranului sunt disponibile următoarele comenzi (Instrumente generale – fig. D.6):

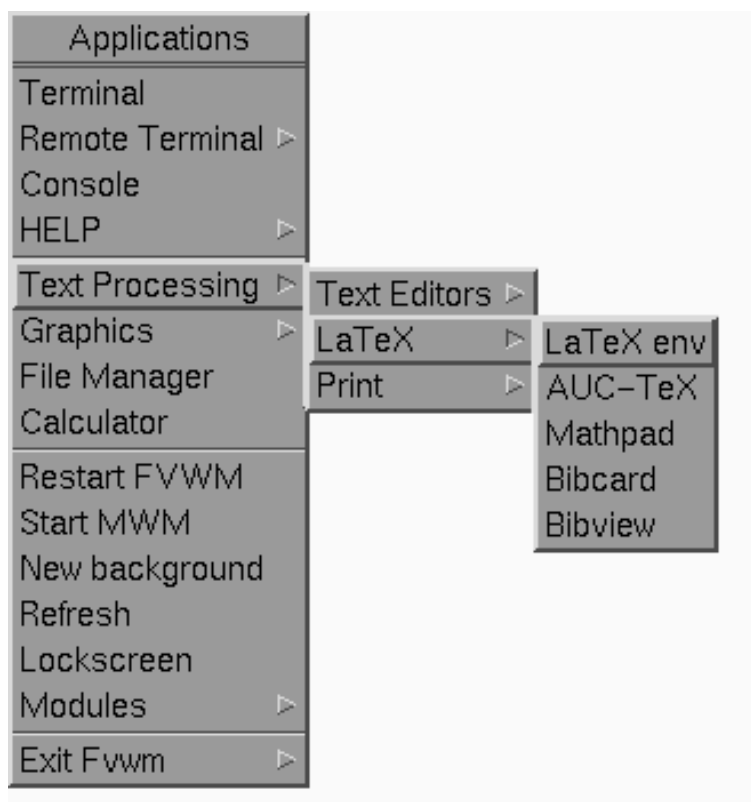


Fig. D.6: Exemplu de meniu asociat butonului stâng

- Terminal nou
- Terminal pe altă mașină
 - lmnhp1
 - lmnhp4
 - lmnhp5
 - lmndec
 - lmnsgi
- Consolă
- Documentație
 - HELP
 - MAN
 - LaserROM
 - Insight
 - XMan
 - OpenBook
- Procesoare texte și documente

- Editoare texte
 - * Emacs
 - * VeEdit
 - * TextEdit
 - * Asedit
- LaTeX
 - * Mediul LaTeX
 - * AUC-TEX
 - * Mathpad
 - * Bibcard
 - * Bibview
- Imprimare
 - * fișier la imprimantă
 - * fereastră la imprimantă
 - * fereastră în fișier
- Grafică
 - XFig
 - Ghostview
 - Adobe PS
 - Acrobat PDF
 - XV
 - XPaint
 - Xmonitory
- Calculator
- Restart sistem ferestre (FVWM)
- Start manager ferestre MWM
- Fundal ecran nou
- Reinițializare ecran
- Module FVWM
- Ieșire

Meniuri de comenzi mapate pe butonul central

Butonul din mijloc permite accesul la următoarele comenzi (Instrumente de proiectare – fig. D.7):

- VLSI
 - Cadence OPUS

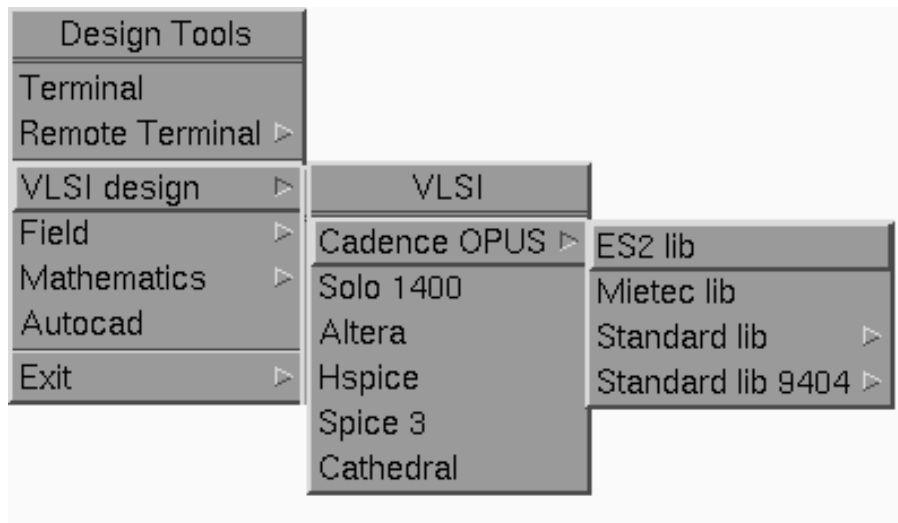


Fig. D.7: Exemplu de meniu asociat butonului central

- Solo 1400
- Altera
- HSpice
- Spice 3
- Cathedral
- Câmp
 - Flux2D
 - Mega
 - Emas
- Matematică
 - Mathematica
 - Scilab
 - GNU Plot
- CAD - AutoCAD

Meniuri de comenzi mapate pe butonul din dreapta

Butonul din dreapta accesează comenzile (Instrumente Internet – fig. D.8):

- e-mail
 - elm
 - HPelm

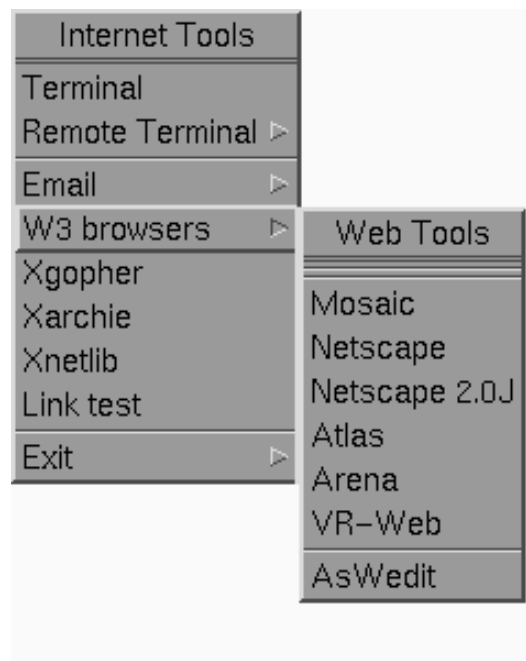


Fig. D.8: Exemplu de meniu asociat butonului din dreapta

- mailx
- mailtool
- Browsere W3
 - Mosaic
 - Netscape
 - Netscape 2.0 (Java)
 - Atlas
 - Arena
 - VR-Web
 - AsWedit
- Xgopher
- Xarchie
- Xnetlib
- Test conexiune pub.net

Bibliografie și Webografie

- [1] ***. Elm Info. <http://www.myxa.com/elm.html>.
- [2] ***. Free XWindow system. <http://www.xfree86.org/>.
- [3] ***. Ghostview. <http://www.cs.wisc.edu/ghost/ghostview/index.html>.
- [4] ***. GIMP – GNU Image Manipulation Program. <http://www.gimp.org/>.
- [5] ***. GNU – Ghostscript.
<http://www.gnu.org/software/ghostscript/ghostscript.html>.
- [6] ***. GNU is Not Unix. <http://www.gnu.org/>.
- [7] ***. The GNU plotting utilities.
<http://www.gnu.org/software/plotutils/plotutils.html>.
- [8] ***. Internet Architecture Board. <http://www.iab.org/iab/>.
- [9] ***. ISO IEC POSIX. <http://osiris.dkung.dk/JTC1/SC22/WG15>.
- [10] ***. Linux. <http://www.linux.org/>.
- [11] ***. Linux applications and utilities.
<http://www.xnet.com/blatura/linapps.html/>.
- [12] ***. The Linux encyclopedia.
<http://www-ocean.tamu.edu/baum/linuxlist.html/>.
- [13] ***. Midnight modeller. <http://SAL.KachinaTech.COM/E/3/MUM.html>.
- [14] ***. The Netlib repository. <http://www.netlib.org/>.
- [15] ***. NetScape home page. <http://home.netscape.com/>.
- [16] ***. The Open Group X/Open. <http://www.opengroup.org/>.
- [17] ***. PETSc - The **P**ortable, **E**xtensible **T**oolkit for **S**cientific computation.
<http://www.mcs.anl.gov/pets>.
- [18] ***. Pine information center. <http://www.washington.edu/pine/>.

-
- [19] ***. Quadpack. <http://www.netlib.org/quadpack/>.
- [20] ***. Sced – scene editor. <http://http.cs.berkeley.edu/schenney/sced.html>.
- [21] ***. Scientific applications on Unix. <http://sal.kachinatech.com/>.
- [22] ***. Scilab: A free CACSD package by INRIA.
<http://www-rocq.inria.fr/scilab/>.
- [23] ***. Stardivision. <http://stardivision.com/>.
- [24] ***. TeX Users Group. <http://www.tug.org/>.
- [25] ***. Transactions author information and LaTeX style files.
<http://www.ieee.org/pubs/authors.html>.
- [26] ***. The UK TeX archive/ The LaTeX3 project. <http://www.tex.ac.uk/>.
- [27] ***. The VIM (Vi IMproved) home page. <http://www.vim.org/>.
- [28] ***. World Wide Web consortium. <http://www.w3.org/>.
- [29] ***. XFig. <ftp://epb1.lbl.gov/xfig>.
- [30] ***. XPaint. <http://www.danbbs.dk/torsten/xpaint/index.html>.
- [31] ***. XV. <http://www.trilon.com/xv>.
- [32] ***. *IRIX User's Reference Manual*. Silicon Graphics Inc., 1990.
- [33] ***. *The Ultimate Guide to the VI and EX Editors*. Hewlett–Packard Company, 1990.
- [34] ***. *Ultrix Reference Pages*. Digital Equipment Co, 1990.
- [35] ***. *A Beginner's Guide to HP-UX*. HP 9000 Computers, Hewlett–Packard Company, 1991.
- [36] ***. *Mail Systems: User's Guide*. HP 9000 Computers, Hewlett–Packard Company, 1991.
- [37] ***. *Text Processing: User's Guide*. HP 9000 Computers, Hewlett–Packard Company, 1991.
- [38] ***. *Sun OS 5.1 Reference Manual*. Sun Soft, 1992.
- [39] ***. *AutoCAD Release 13 User's Guide*. AutoDesk, 1994.
- [40] ***. *Drx. Linux. The Linux Documentation Project*. LSL, 1994.
- [41] ***. *Internet Passport*. Prentice Hall, 1995.

-
- [42] J. D. Foley et al. *Computer Graphics. Principles and Practice*. Addison-Wesley, 1987.
- [43] V. Cristea et al. *Mai multe despre Internet*. Teora, 1995.
- [44] David Baker. Preparing presentation materials using `seminar.sty`.
<http://www.comlab.ox.ac.uk/internal/guides/html/present-27/present.html>, 1995.
- [45] T. F. Coleman. *Handbook for Matrix Computations*. SIAM, 1988.
- [46] Donald A. Curry. *Using C on the UNIX System*. O'Reilly & Associates, Sebastopol, CA, 1989.
- [47] E. Cutler. *SCO Unix in a Nutshell*. Romsym Data, 1995.
- [48] G. F. Forsythe și G. B. Moler. *Computer Solutions of Linear Algebraic Systems*. Prentice Hall, 1967.
- [49] Open Software Foundation. *OSF/Motif Style Guide*. Prentice Hall, 1991.
- [50] D. Frey și R. Addams. *A Directory of Electronic Mail*. O'Reilly & Assoc., 1994.
- [51] Marin Ghinea și Virgiliu Fireșteanu. *MATLAB Calcul numeric – Grafică – Aplicații*. Teora, București, 1995.
- [52] Daniel Gilly. *Unix in a Nutshell*. O'Reilly, 1992.
- [53] Michel Goossens, Frank Mittelbach și Alexander Samarin. *The L^AT_EX Companion*. Addison-Wesley, 1994.
- [54] John Gustafson. Computational verifiability and feasibility of the ASCI program. *Computational Science & Engineering*, vol. 5, nr. 1, pp. 36–45, January–March 1998.
- [55] H. Hahn. *A Student's Guide to UNIX*. McGraw Hill, 1993.
- [56] K. M. Heal, M. L. Hansen și K. M. Rickard. *Maple V Learning Guide*. Springer-Verlag, 1996.
- [57] R. T. Herbst. *Software design using FORTRAN77*. Prentice Hall, 1990.
- [58] Kamran Husain și Tim Parker. *Linux Unleashed*. Sams Publishing, Indianapolis, 1995.
- [59] Daniel Ioan, Irina Munteanu, Bogdan Ionescu, Mihai Popescu, Radu Popa, Mihai Lăzărescu și Gabriela Ciuprina. *Metode numerice în ingineria electrică*. MATRIX ROM, București, România, 1998.
- [60] M. Jalobeanu. *Internet, informare și instruire*. Promedia Plus, 1995.
- [61] B. W. Kernighan și D. M. Ritchie. *The C Programming Language*. Prentice Hall, 1978.

- [62] J. F. Kerrigan. *Migrating to Fortran90*. O'Reilly, 1993.
- [63] Helmut Kopka și Patrick Daly. *A Guide to L^AT_EX*. Addison–Wesley Publishing Co., 1993.
- [64] Ed Krol. *Totul despre Internet*. RomSymData, 1995.
- [65] J. R. Levine și M. L. Young. *Unix pentru toți*. Teora, 1996.
- [66] G. McComb. *Web Programming Languages Sourcebook*. John Wiley, 1997.
- [67] J. D. Murray și W. Van Ryper. *Encyclopedia of Graphics File Formats*. O'Reilly, 1987.
- [68] C. Musciano și B. Kennedy. *HTML, The definitive Guide*. O'Reilly, 1997.
- [69] North West Net. *The Internet Passport*. Prentice Hall, 1995.
- [70] John Purcell, editor. *LINUX MAN – The Essential MAN Pages for Linux. Chap. III (Library Functions)*. Linux System Labs, 1995.
- [71] A. Pusztai și Gh. Ardelean. *L^AT_EX – Ghid de utilizare*. Editura Tehnică, 1994.
- [72] V. Quercia și T. O'Reilly. *XWindow System. User's Guide for X11 Release 5*. O'Reilly, 1993.
- [73] K. Reichard și E. F. Johnson. *Teach Yourself UNIX*. MIS Press, New York, 1992.
- [74] J. R. Rice. *Migrating to Fortran90*. McGraw Hill, New York, 1983.
- [75] M. A. Schoonover și al. *GNU Emacs – UNIX Text Editing and Programming*. Addison–Wesley, 1992.
- [76] A. Simpson. *Netscape Navigator Book*. Netscape Press, 1996.
- [77] Bjarne Stroustrup. *The C++ Programming Language*. Addison-Wesley Publishing Company, Wokingham, England, a doua ed., 1992.
- [78] A. S. Tanenbaum. *Computer Networks*. Prentice Hall, 1996.
- [79] Matt Welsh și Lar Kaufman. *Running Linux*. O'Reilly & Associates, Sebastopol, CA, 1995.
- [80] J. H. Wilkinson și C. Reinsch. *Linear Algebra*. Springer Verlag, 1971.
- [81] Stephen Wolfram. *The Mathematica Book*. Wolfram Media / Cambridge University Press, a treia ed., 1996.
- [82] Timothy Van Zandt. *seminar.sty – A L^AT_EX style for slides and notes. User's guide*. <http://www.tex.ac.uk/tex-archive/macros/latex/contrib/other/seminar/doc/sem-guide.tex>, 1993.

Index

SIMBOLURI

/	27, 28, 30
<	32
< >	208
>	32
>>	32
\$	25, 26
%	26
	32

A

a2ps	32, 63
abbrv, stil bibliografic	99, 103
acm, stil bibliografic	100
Ajutor	<i>Vezi</i> Help
Alias	<i>Vezi</i> elm – alias
alpha, stil bibliografic	99, 102
apropos	34
Arhivare fişiere	204
article, stil L ^A T _E X	71

B

bash	26
bg	33
Bibliografie	96
extensii nume fişiere	100
fişier .bib	97
includere în fişier .tex	100
introducerea referinţelor în L ^A T _E X	99
stil bibliografic	99
abbrv	99, 103
acm	100
alpha	99, 102
ieee	100
ieeetran	100
plain	99, 101
plainrom	99
unsrt	99, 102
unsrtrom	99, 103

utilitare	96
-----------	----

BIB _T E _X	<i>Vezi</i> Bibliografie
bibtex	100
Bitmap	<i>Vezi</i> Figuri – bitmap, <i>Vezi</i> Figuri – bitmap
book, stil L ^A T _E X	71
Browser	198, <i>Vezi</i> Internet – browser
BSD	18

C

Căsuță poștală	<i>Vezi</i> Poștă electronică
Căutări	31
Căutare în Internet	<i>Vezi</i> Netscape
cal	34
calc	35
Calculator de buzunar	<i>Vezi</i> calc, xcalc
Calculule matematice	150
Captură de ecran/fereastră	<i>Vezi</i> xv
cat	29
Cc	57, 61
cd	30
CGI	198
chmod	30
clear	35
Client/server	20, 193, 224
Comenzi Unix	21–35
Comparări de fişiere	31
Compilare	<i>Vezi</i> Programare – compilatoare
compress	199, 204
Computational Science and Engineering	<i>Vezi</i> CSE
Comunicație sub Unix	193
compress, uncompress	204
ftp	203
gzip, gunzip	204
rlogin	204
ssh	204

- talk 201
 tar 204
 telnet 200
 uuencode, uudecode 204
 write 200
 Configurare
 editorul vi 45
 elm 62
 \LaTeX 104
 Scilab 151
 shell 25
 variabile Unix 25
 Conversii de formate grafice ... 128, 135,
 140, 142
 Cota de spațiu de disc 35
 cp 29
 CSE 9, 147
 csh 26
- D**
- date 34
 Depanator (debugger) 176
 Desene 121
 \LaTeX 80
 reprezentare vectorială 122
 XFig 125
 XPaint 138
 diff 31
 Directoare 27
 speciale 27
 curent 27
 home 28
 precedent 28
 rădăcină 28
 Documente 67
 DOS \rightarrow Unix 34
 Drepturi asupra fișierelor 30
 DVI 68
 dvips 70
- E**
- e-mail *Vezi* Poștă electronică
 echo 34
 Editare grafică *Vezi* Figuri
 Editarea documentelor 39
 Editarea textelor 39
 Editoare de texte
 emacs 47
 vi 40
 electronic mail ... *Vezi* Poștă electronică
 elm 59, 199
 comenzi
 citire mesaje 59
 creare alias 61
 ștergere/mutare mesaje 62
 trimitere mesaje 60
 configurare 62
 emacs 47
 comenzi
 ștergere 49
 încărcare fișier 50
 înlocuire 50
 acces la help 51
 căutare 50
 deplasare cursor 49
 dezvoltare programe 51
 ferestre 50
 formatare 49
 indentare 49
 macrocomenzi 50
 manipulare blocuri 50
 mod dezvoltare programe 51
 salvare fișier 50
 Unix 51
 ieșire 47
 lansare 47
 mod LaTeX 51
 spelling 51
 structură ecran 47
 tutorial 51
 Ethernet 195
 Exemple de utilizare
 apel FORTRAN din C 179
 apel rutine externe din Scilab ... 164
 configurare \LaTeX 105
 configurare vi 45
 gnuplot 136
 HTML 210
 index 117
 \LaTeX 83–95
 make 109, 175

-
- planșe de prezentare 107
 - Scilab 160
 - tar 204
 - traducere Scilab → FORTRAN . 166
 - exit 24
 - Extensii nume fișiere
 - .adi 122
 - .aux 83
 - .bbl 100
 - .bib 97, 99, 100, 104
 - .bin 157
 - .blg 100
 - .bmp 123
 - .bst 100, 104
 - .c, .cc, .f, .h, .o 173
 - .dvi 83
 - .eps 123
 - .fig 122, 128
 - .gif 122
 - .glo 83
 - .gnu 136
 - .hpg 122
 - .html 207
 - .idx 83
 - .ist 118
 - .jpg 123
 - .lof 83
 - .log 83
 - .lot 83
 - .pbm 123
 - .pcx 123
 - .pdf 123
 - .pgm 123
 - .pk 104
 - .pov 123
 - .ppm 123
 - .ps 123
 - .sci 157, 158
 - .sty 104
 - .tex 83, 104
 - .tif 123
 - .toc 83
 - .vrm 123
 - .wmf 123
 - .xbm 123
 - .xpm 123
 - fișiere grafice 122–123
 - F**
 - f2c 170
 - Fast Ethernet 195
 - Fereastră *Vezi* XWindow
 - fg 33
 - Fișiere 27
 - afișare listă 29
 - comparare 31
 - copiere 29
 - imprimare 32
 - listare conținut 29
 - mutare 29
 - nume 27
 - redenumire 29
 - schimbarea drepturilor 30
 - ștergere 29
 - Fișiere de comenzi
 - gnuplot 136
 - Scilab 158
 - Unix *Vezi* Script
 - Figuri 121
 - apel din meniuri . 239, 241, 242, 244, 245
 - bitmap 122
 - incluse în HTML 209
 - LaTeX 72, 73, 80, 83
 - metafile 122
 - PostScript .. 123, 125, 128, 135, 140, 142, 159, 160
 - XFig 125, 159, 160
 - XPaint 138
 - XV 142
 - file 205
 - find 31
 - finger 33, 199
 - Fire torsadate *Vezi* UTP
 - Format fișier . *Vezi* Extensii nume fișiere
 - Formatarea documentelor ... *Vezi* LaTeX
 - Formule matematice *Vezi* LaTeX
 - FORTRAN 77 169, 172
 - Fortran 90 169
 - forward 57
 - fromdos 34

ftp.....199, 203, 205

G

g++ 172
 g77 172
 gcc 172
 gdb 176
 ghostview.....32, 70
 GNU 19, 172
 gnuplot 129
 gopher 205
 GPL 19
 Grafice.....121
 Gnuplot 129
 Scilab 159
 grep 31
 GUI 17
 gunzip.....204
 gzip 199, 204

H

HDCL 195
 Help .. 34, 45, 51, 59, 129, 134, 135, 139,
 140, 142, 158
 Hipermedia 198
 hostname 21
 HP-UX 18
 HTML 205
 coduri de formatare 208–210
 structură 207
 tags 208–210

I

Ieșire în sistem 24
 ieee, stil bibliografic 100
 ieeetran, stil bibliografic 100
 Imprimare.....32, *Vezi* lpr
 \index.....117
 Indirectări.....32
 Internet 193, 195
 browser 205
 domenii naționale.....56
 domenii SUA 56
 domeniul .ro.....198
 file 205
 ftp 205

gopher 205
 HTML 205
 istoric 194
 URL 205

Interoperabilitate 170
 Intrare în sistem..... 21
 IP 195
 IPv6 195
 IRIX 18
 Istoricul LMN 223

J

Java 198
 JavaScript 198
 Joe 40

K

kill 33
 ksh 26

L

LAN..... *Vezi* Local Area Network
 Lansare/oprire comenzi Unix 33
 L^AT_EX

 accente 81
 cadru 79
 caractere speciale 81
 comenzi 74
 configurare 104
 definire de noi comenzi..... 80
 despărțire în silabe 82
 diacritice pentru limba română... 81
 exemple de utilizare..... 83
 extensii nume fișiere 83
 formule matematice 86
 index 80
 macrodefiniții..... 80
 makefile 109
 modul LR 78
 modul matematic 77
 modul paragraf 75
 moduri de lucru 75
 realizarea fișierului imprimabil... 70
 stiluri
 article 71
 book 71

letter 71
 report 71
 structură fișier 70
 structura paginii 83
 tipuri de comenzi 74
 latex 70
 latex2html 214
 letter, stil L^AT_EX 71
 Linux 19
 LMN
 adresă WWW 227
 istoric 223
 meniuri de comenzi 239
 regulament 221
 resurse hardware 230, 232
 resurse software 237
 Local Area Network 193
 locate 31
 lock 25
 login 21
 logout 24
 lpq 33
 lpr 32, 70
 ls 29
 Lucru cu directoare 30
 Lucru cu ferestre X 21, 35, 130
 Lucru cu fișiere 29
 Lucrul cu ferestre X
 *Vezi și* XWindow
 Lynx 205

M
 Macro *Vezi* Scilab, L^AT_EX, emacs
 Make
 pentru compilare 174
 pentru procesarea fișierelor .tex 109
 makeidx 117
 \makeindex 117
 makeindex 117
 MAN.. *Vezi* Metropolitan Area Network
 man 34
 MAPLE 149
 Mathematica 147
 MATLAB 148
 Medii de programare 169
 Meniu *Vezi* XWindow

Mesaje ... *Vezi* Poștă electronică, write
 msg 199
 Metafile *Vezi* Figuri – metafile
 Metropolitan Area Network 193
 mkdir 30
 Modelare
 fizică 147
 matematică 147
 numerică 147
 more 29
 mv 29

N

Navigație în Internet *Vezi* Netscape
 NetLib 170
 Netscape 205
 Nucleul sistemului de operare 17

O

OOP 170
 Open System Interconnection... *Vezi* OSI
 OSI 195

P

Parolă *Vezi* password
 passwd 25
 password 21
 Pearl 198
 ping 199
 plain, stil bibliografic 99, 101
 plainrom, stil bibliografic 99
 Planșe de prezentare 105
 Poștă electronică 55
 adresă 56
 elm 59
 reguli de stil 57
 semnătură electronică 58
 smileys 58
 POSIX 19
 PostScript 148, *Vezi* Figuri – PostScript
 PPP 195
 \printindex 117
 Procesarea textelor 39
 Programare
 apel rutine FORTRAN
 din C 179

din Scilab 163
 compilatoare 172
 depanare 172, 176
 editare programe în **emacs** 51
make 174
 medii 169
 orientată pe obiecte 170
 în Scilab 156
 stilul profesional 173
 structurată 170
 traducere automată
 FORTRAN → C 170
 Scilab → FORTRAN 166
ps 33
 Pseudonim *Vezi alias*
pwd 30

Q

quota 35

R

Rețea de calculatoare 193
 Referat de laborator 111, 112
 Regulament LMN 221
report, stil **L^AT_EX** 71
 Reprezentare metafișier ... *Vezi Figuri – metafile*
 Reprezentare pixel *Vezi Figuri – bitmap*
 Reprezentare vectorială .. *Vezi Desene – reprezentare vectorială*
 Request For Comments *Vezi RFC*
 Resurse hardware LMN 230, 232
 Resurse software LMN 237
RFC 196
rlogin 204
rm 29
rmdir 30
 RoEduNet 194, 198, 225

S

Salvare mesaje... *Vezi Poștă electronică*
SCICOS 149
Scilab 147
 apel de rutine externe 163
 componente 150
 demo 152

fișiere de comenzi 158
 formatul comenzilor 151
 funcții predefinite 157
 grafice 159
 instrucțiuni 157
 programare 156
 structuri de date 152
 traducere automată
 în FORTRAN 166
scilab 150
Script 26, 237
sdiff 31
seminar 105
SGML 207
sh 26
 Shell Unix 17
SLIP 195
 Smileys 58
SMTP 195
SNMP 195
Solaris 18
sort 35
ssh 24, 26, 204
 Standard General
 Markup Language .. *Vezi SGML*
 Structura de directoare 26
Subject 57
Sun OS 18
SVR4 18

T

Tabele 72, 73, 80
talk 199, 201
tar 199, 204
TCP 195
Telnet 195
telnet 199, 200
 Teste de verificare 216–220
T_EX 68, 148
Text 39
todos 34
Toolbox 11, 23
 Traducătoare
 ASCII → PostScript 32, 63
 FORTRAN → C 170
 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ → HTML 214

U

Ultrix	18
umask	31
uname	21
uncompress	204
Uniform Resource Locator ... <i>Vezi</i> URL	
UNIX	11, 18
Unix	11, 18–20
Unshielded Twisted Pair	<i>Vezi</i> UTP
unsrt, stil bibliografic	99, 102
unsrtrom, stil bibliografic	99, 103
URL	205
Usenet	195
username	21
UTP	194
uudecode	199, 204
uuencode	199, 204

V

Vectorial	<i>Vezi</i> Desene
vi	40
comenzi	
înlocuiri	44
căutări	44
deplasare cursor	42
editare	43
ieșire	44
salvare	44
configurare	45
intrare în mod text	42
moduri de lucru	41
comandă	41
text	41
Vizualizare	<i>Vezi</i> ghostview, xdvi

W

w	33
WAN	<i>Vezi</i> Wide Area Network
wc	35
Web	<i>Vezi</i> WWW
Web page	207
Web site	207
whatis	34
who	33
whoami	33
Wide Area Network	193

World Wide Web	<i>Vezi</i> WWW
write	199, 200
WWW	198, 205
WYSIWYG	40, 67, 198

X

X11	20
xcalc	35
xclock	35
xdvi	70
Xedit	40
Xenix	18
xfig	125
xlock	25
xpaint	138
xterm	35
xv	142
XWindow	20, 21, 130
xxgdb	176