

INTRODUCCIÓ A LA PROGRAMACIÓ EN FORTRAN 77

Juan Carlos Paniagua
Albert Solé

Departament de Química Física



UNIVERSITAT DE BARCELONA



TEXTOS DOCENTS

328

INTRODUCCIÓ A LA PROGRAMACIÓ EN FORTRAN 77

Juan Carlos Paniagua
Albert Solé

Departament de Química Física

Publicacions i Edicions



UNIVERSITAT DE BARCELONA



Índex

PRÒLEG	i
1 Nocions d'informàtica	1
1.1 L'ordinador	1
1.2 El maquinari	1
1.3 Codificació de la informació	4
1.4 El programari	7
1.5 Sistemes operatius	8
1.6 Estructuració dels discos	8
1.7 El sistema operatiu GNU/Linux	10
1.7.1 La GUI	11
1.7.2 Instruccions en mode text	12
2 Programació en llenguatge Fortran 77	17
2.1 Llenguatges de programació	17
2.2 Intèrprets i compiladors	18
2.3 Primeres nocions del llenguatge Fortran	19
2.3.1 Un exemple preliminar	19
2.3.2 Tipus d'instruccions	20
2.3.3 Sintaxi	21
2.3.4 Variables i constants	23
2.3.5 Tipus de dades	24
2.4 Instruccions d'entrada i de sortida	27
2.5 Instruccions d'assignació. Operadors aritmètics	29
2.5.1 Ordre d'avaluació d'operacions aritmètiques	31
2.6 Funcions incorporades	33
2.7 Disseny descendent d'algorismes	34
2.7.1 Estructures algorísmiques bàsiques	35
2.7.2 Estructura repetitiva <i>per a</i>	36
2.7.3 Algorismes complexos	36
2.8 Estructura alternativa	37
2.8.1 Formes simplifiades	42
2.9 Instruccions de transferència de control	42
2.10 Estructures repetitives	43
2.10.1 <i>Do</i> amb índex	44
2.10.2 Acumuladors	46
2.10.3 <i>Do while</i>	49
2.10.4 Comptadors	51
2.11 Variables indexades	53
2.11.1 Vectors	53

2.11.2	<i>Do</i> implícit	57
2.11.3	Matrius	58
2.12	Entrada i sortida amb fitxers	64
2.13	Subprogrames	67
2.13.1	Subrutines	67
2.13.2	Funcions externes	74
2.13.3	Avantatges dels subprogrames	77
3	Informació addicional i altres exercicis	79
3.1	Taules	79
3.2	Integració numèrica	81
3.3	Variables indexades	83
3.3.1	Índexs amb valor mínim diferent d'1	83
3.3.2	Component més gran o més petit d'un vector	85
3.3.3	Ordenació dels components d'un vector	85
3.3.4	Producte de matrius	87
3.3.5	Diagonalització de matrius	88
3.3.6	Variables indexades en subprogrames	92
3.4	Escriptura i lectura amb format	93
3.4.1	Codis per a especificacions de format	96
3.5	Instrucció <i>data</i>	97
3.6	Instrucció <i>common</i>	98
3.7	Instrucció <i>go to</i>	98
3.8	<i>Do</i> amb etiqueta	99
3.9	Altres exercicis	100
	APÈNDIXS	107
A	Evolució dels sistemes operatius	109
B	Programari lliure	113
C	Xifres significatives, taules i gràfiques	115
C.1	Xifres significatives	115
C.2	Criteris per construir taules	116
C.3	Criteris per construir gràfiques	117
D	Bibliografia	119
D.1	Enciclopèdies i diccionaris generals	119
D.1.1	Català	119
D.1.2	Castellà	119
D.1.3	Anglès	119
D.1.4	Recopilació de diccionaris	119
D.1.5	Diccionari anglès-català	119
D.2	Diccionaris de termes informàtics	120
D.2.1	Català	120
D.2.2	Castellà	120
D.2.3	Anglès	120
D.3	Distribucions de GNU/Linux	120
D.4	Fortran 77	120
D.5	Aplicacions	120

1 Nocions d'informàtica

En aquest capítol s'introduiran els termes més emprats en l'àmbit informàtic amb un llenguatge molt planer, la qual cosa ens farà sacrificar, de vegades, el rigor en favor de la simplicitat. Aquesta introducció serà força incompleta, atesos els objectius d'aquest text i l'amplitud del camp; a més es tracta d'una matèria que evoluciona molt ràpidament, per la qual cosa remetem el lector interessat a ampliar o actualitzar la informació proporcionada a les referències recollides a la bibliografia. Com que les obres impreses necessiten més temps per posar-se al dia que les publicades a través de la xarxa Internet, hem optat per incloure més referències del segon tipus. D'altra banda, una opció pràctica per consultar el significat d'un terme determinat és fer una cerca estesa a tota la xarxa mitjançant un buscador de paraules (*www.google.com*, *search.yahoo.com*, *www.live.com*, etc.).

1.1 L'ordinador

Un *ordinador* és una màquina capaç d'entendre instruccions externes per realitzar una tasca, fer-la i mostrar-ne el resultat. La informació subministrada pot consistir en dades numèriques, paraules (teclejades o seleccionades dins d'un menú o d'una finestra), imatges, so, etc., i la tasca pot implicar operacions algebraïques, comparacions entre dades, còpia d'informació, etc. De fet, hi ha molts aparells capaços d'entendre certes instruccions per realitzar determinades tasques (un equip de música, una rentadora, etc.), però el que caracteritza els ordinadors és la seva versatilitat; és a dir, la varietat d'instruccions i tasques que admeten. La *informàtica* estudia el disseny d'ordinadors i les seves aplicacions.

En qualsevol ordinador podem distingir dos tipus de components: el maquinari i el programari. El *maquinari* o *hardware* és el conjunt de components físics de l'ordinador; és a dir, tot allò que es pot tocar: teclat, monitor, circuits electrònics,¹ etc. Anomenem *programari* o *software* les instruccions (components lògics) que permeten manipular el maquinari en la forma desitjada (llegir el contingut d'un disc magnètic o enregistrar-hi informació, efectuar operacions matemàtiques amb dades introduïdes a través del teclat, imprimir-ne el resultat en una impressora, retocar una imatge, etc.). De vegades s'inclou també sota la denominació *software* la informació emmagatzemada a l'ordinador (textos, dades numèriques, imatges, so, etc.).

Per clarificar els conceptes de maquinari i programari podem emprar el símil d'un receptari de cuina: les receptes (intangibles) contingudes en el llibre equivaldrien al programari, i el paper i la tinta (tangibles) constituïrien el maquinari.

1.2 El maquinari

Els components de maquinari que conté un ordinador poden ser molt diversos, però sempre hi ha un processador, una memòria ROM, una memòria RAM i uns perifèrics (figura 1.1).

¹Compte amb les enramades: certs components interns dels ordinadors poden produir descàrregues elèctriques fins i tot amb l'aparell desendollat.

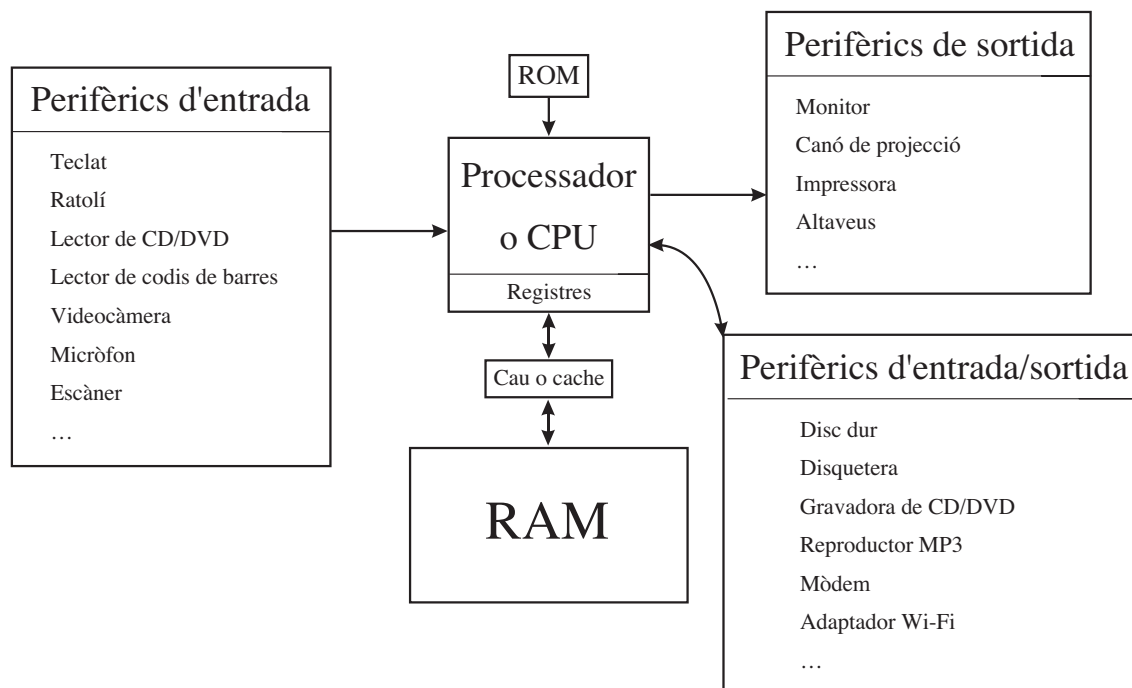


Figura 1.1: Esquema del maquinari d'un ordinador.

El *processador* o *CPU* (Central Processing Unit: unitat central de procés) és l'òrgan director del funcionament de l'ordinador (el “cervell” de la màquina). En endavant utilitzarem preferentment la paraula *processador*, atès que, en el llenguatge del carrer, se sol anomenar CPU o *torre* la capsa que conté la major part dels components d'un ordinador no portàtil (exclosos el teclat, el ratolí, el monitor, la impressora, etc.). Els processadors que porten actualment la majoria d'ordinadors —tant els d'ús domèstic o *personals* (PC: Personal Computer) com els d'ús professional— són *escalars*, en el sentit que fan les operacions d'una en una (Intel Core, AMD Athlon...). Alguns ordinadors, però, tenen processadors *vectorials*, els quals poden fer una mateixa operació sobre diferents dades alhora, processadors amb més d'un *core* (Dual Core, Quad Core...), que funcionen com si fossin conjunts de processadors, o diversos processadors que treballen de manera coordinada (*SMP*: Symmetric Multi-Processing). El ritme de treball d'un processador ve determinat per un rellotge amb una freqüència fixa que, actualment, se sol mesurar en gigahertz ($1\text{ GHz} = 10^9\text{ Hz}$). Per a una mateixa família de processadors —per exemple, els Pentium 4— la velocitat de procés augmenta amb la freqüència del rellotge, però aquesta dada no permet comparar rendiments de famílies de processadors diferents —per exemple, un Pentium 4 amb un Pentium M o amb un Athlon. Aquesta comparació s'ha de fer a partir dels temps que triguen a fer una mateixa tasca. En l'àmbit científic s'utilitza molt el nombre de *flops* per comparar el rendiment dels processadors. Un *flops* (Floating-Point Operations per Second) és una operació amb nombres reals —també anomenats de *coma flotant*— per segon i, com que els processadors moderns són molt ràpids, s'utilitzen múltiples d'aquella quantitat: megaflops ($1\text{ Mflops} = 10^6\text{ flops}$), gigaflops ($1\text{ Gflops} = 10^9\text{ flops}$), etc.

La informació que arriba al processador i la que aquest produeix es guarda en els *registres*, que són petites zones de memòria temporal d'accés molt ràpid. Així, si el processador ha de sumar dos nombres, cada un d'aquests es guardarà en un registre i, una vegada calculada la seva suma, el resultat queda guardat en un altre registre (que pot ser el mateix que guardava un dels sumands). Com que la capacitat dels registres és molt petita, el processador necessita poder accedir a zones de memòria més grans, com ara la memòria RAM. La comunicació entre el processador i altres parts de l'ordinador es fa mitjançant un *bus*, que és un conjunt de conduccions elèctriques (16, 32, 64..., depenent de la màquina) que transmeten informació de manera simultània (*en paral·lel*).

Anomenem *memòria ROM* (Read Only Memory) una zona de memòria que no es pot modificar i que no s'esborra en apagar l'ordinador. Aquesta memòria conté instruccions molt bàsiques per al

funcionament de l'ordinador. Així, quan l'engeguem, el processador llegeix les instruccions emmagatzemades en la ROM, les quals poden fer que aquell busqui instruccions per acabar d'arrencar en algun disquet o CD que haguem introduït prèviament o en el disc dur, etc.

Anomenem *memòria RAM* (Random Access Memory)² o *principal* una zona de memòria que es pot modificar i que s'esborra quan apaguem l'ordinador. Es diu, per això, que la informació que guarda aquesta memòria és *volàtil*. A la memòria RAM —que és molt més gran que la ROM— s'emmagatzemen les instruccions que indiquen al processador les tasques que volem que faci, així com les dades necessàries per realitzar-les i els resultats que puguin produir aquestes tasques.

En la major part de les calculadores, les memòries de treball són de tipus RAM (podem guardar-hi dades, esborrar-les, etc.); en canvi, les instruccions per calcular funcions trigonomètriques, logaritmes, etc. estan enregistrades en memòria ROM, ja que no s'esborren en apagar la calculadora. La informació impresa en paper proporciona un símil no electrònic per a aquests dos tipus de memòria: un llibre de la biblioteca es pot considerar un suport de memòria ROM, ja que la informació que conté pot ser llegida però no podem alterar-la; una llibreta, en canvi, representaria una memòria RAM, en la qual podem anotar dades, esborrar-les, etc. (tot i que no té el caràcter volàtil de la RAM de l'ordinador).

Quan comprem un ordinador no solament hem de fixar-nos en la capacitat de la seva memòria RAM; és important que aquesta es pugui ampliar afegint *mòduls* o *xips*³ de memòria o substituint els que té per uns altres de més capacitat, ja que les noves versions del programari suposen, sovint, un augment en les exigències de memòria.

El ritme de treball del processador és més ràpid que l'accés a la memòria RAM, per la qual cosa se sol intercalar entre aquells dos components una petita memòria d'accés molt ràpid (i més cara) que s'anomena *cau* o *cache*. Aquesta acumula les dades transferides al processador o des del processador en blocs per tal d'augmentar la velocitat de transferència.

Els *perifèrics* o *dispositius* permeten que el processador es comuniqui amb l'exterior. N'hi ha de molts tipus, però es poden classificar en tres grups:

- els *d'entrada* permeten transferir informació al processador: el teclat, el ratolí, el lector de discos òptics —CD-ROM (Compact Disc Read Only Memory) i DVD (Digital Video Disc o Digital Versatile Disc)—, un micròfon, un escàner, un lector de codis de barres, etc.;
- els *de sortida* mostren informació produïda pel processador: el monitor o pantalla, un canó de projecció, una impressora, uns altaveus, etc., i
- els *d'entrada/sortida* permeten transferir informació en ambdós sentits: *discos magnètics* durs, interns (fixos) o externs, discos magnètics flexibles (*floppy disks*) o *disquets*, *gravadores* de discos òptics (CD-R, CD-RW, DVD-R, DVD-RW, DVD+R, DVD+RW, etc.), *dispositius USB de memòria flaix* (anomenats també *pen drive*, *USB flash drive*, *memòria USB* o *reproductor MP3* si permeten escoltar música comprimida amb aquest format), *mòdems* o *routers* per connectar l'ordinador a una línia telefònica, etc.

El principal mitjà d'emmagatzematge no volàtil d'informació solen ser els discos magnètics i, per això, una característica important d'un ordinador és la capacitat del seu disc dur. Aquesta capacitat es pot ampliar substituint el disc o afegint-hi discos addicionals. L'accés a disc (o a altres perifèrics) és molt més lent que l'accés a la memòria RAM, per la qual cosa se sol intercalar una petita memòria ràpida —un *buffer* o *cache de disc*— que acumula blocs d'informació que s'ha d'escriure o llegir abans de transferir-los entre el disc i la memòria RAM.

Els perifèrics s'acoblen a l'ordinador a través de *connexions* que també s'anomenen *ports*, *sortides* o *entrades*, com ara *USB*, *FireWire* (anomenat també *IEEE 1394* o *iLink* en les videocàmeres), *ATA* o *IDE*, *Serial ATA*, *SCSI*, *paral·lel*, *sèrie*, etc. Les *connexions a xarxa* permeten que l'ordinador es comuniqui amb altres ordinadors. La més utilitzada avui dia s'anomena *Ethernet*. Si no disposem de línia Ethernet, necessitem un mòdem per transformar la informació que ha de sortir de l'ordinador o entrar-hi en senyals elèctrics transmissibles a través d'una línia telefònica (mòdems *RTB* o *ADSL*),

²Aquesta denominació és poc afortunada: de fet, tant la memòria RAM com la ROM són d'*accés aleatori* (Random Access), en el sentit que podem accedir a qualsevol dada emmagatzemada de manera directa (sense haver de llegir les dades anteriors).

³La paraula *xip* se sol aplicar a qualsevol component microelectrònic amb circuits integrats i moltes connexions elèctriques.

o de la xarxa elèctrica (mòdem *PLC*). La comunicació amb altres ordinadors o amb perifèrics es pot fer també sense fils (mitjançant ones electromagnètiques), fent servir les tecnologies *Bluetooth*, *Wi-Fi* o *Airport*, etc. El nombre de connexions d'un ordinador es pot ampliar si aquest disposa de *ranures d'expansió* o *slots* (*PCI*, *PCMCIA*, etc.), que són espais preparats per introduir-hi *targetes* (plaques amb components electrònics) que poden incorporar noves connexions. Aquestes targetes poden realitzar certes operacions, com ara la conversió d'un senyal *analògic* (que varia de forma contínua) produït per un aparell en un senyal *digital* (expressat en forma numèrica) intel·ligible per l'ordinador.

La torre conté, normalment, una placa amb un circuit imprès —la *placa base*— en què estan endollats el processador, les memòries ROM i RAM, el disc dur, la disquetera, el lector i/o la gravadora de CD/DVD, diversos ports i ranures d'expansió, targetes gràfica⁴ i de so, una font d'alimentació, etc. En els ordinadors compactes tot això va integrat amb el monitor i els altaveus, i en els portàtils s'hi afegeix també el teclat i una mena de ratolí.

Exercici 1.1

Classifiqueu els perifèrics següents com d'entrada, de sortida o d'entrada/sortida: a) disc magnètic; b) impressora; c) teclat; d) monitor; e) mòdem.

Solució: a) E/S; b) S; c) E; d) S (si no és tàtil); e) E/S.

1.3 Codificació de la informació

Els ordinadors treballen amb informació codificada en forma *digital* —és a dir, numèrica— tant si es tracta de xifres com si són lletres, imatges, sons, etc. Així, una imatge es pot codificar en format digital (*digitalitzar*) fent servir un escàner, que la divideix en una quadrícula molt fina i emmagatzema el color de cada quadratet (*píxel*) mitjançant tres nombres naturals que representen les intensitats de tres colors primaris; per exemple el vermell, el verd i el blau (*RGB*: Red, Green and Blue).

Per a la major part de les aplicacions, l'usuari d'un ordinador no necessita saber com es codifica la informació que introdueix o que extreu de l'ordinador, ja que els perifèrics de sortida (monitor, impressora...) li presenten descodificada i els d'entrada (teclat, ratolí...) la codifiquen de manera automàtica. Això no obstant, és interessant tenir una idea de com es fa aquesta codificació.

La codificació que fan servir actualment els ordinadors és de tipus *binari*, la qual cosa vol dir que utilitzen només dos dígit: el 0 i l'1. D'una manera molt simplista, podríem imaginar la memòria RAM com un taulell amb moltes bombetes, cada una de les quals representa un 0 o un 1 segons estigui apagada o encesa. D'acord amb aquest símil, la CPU aniria encenent i apagant bombetes quan ha de guardar informació en la memòria, i per llegir aquesta informació haurà de comprovar l'estat —apagat o encès— de les bombetes. És evident que un sistema com aquest només permet emmagatzemar informació codificada amb un sistema binari, i que la informació es perdrà si s'interromp el subministrament elèctric i no ha estat prèviament copiada en un suport no volàtil; per exemple, un disc magnètic. En aquest, un 0 o un 1 es podrien enregistrar magnetitzant una petita superfície (un "punt") del disc en una direcció o en l'oposada, respectivament. Aquesta magnetització no es perd quan s'apaga l'ordinador, de manera que les dades enregistrades es conserven. Per llegir després la informació caldrà anar detectant el sentit de la magnetització a cada punt del disc.

Veiem com s'expressa un nombre natural qualsevol en codificació o notació binària. Per expressar un nombre natural en notació decimal —la qual s'identificarà afegint-hi la indicació ₁₀— escrivim els dígit decimal (0, 1, 2, 3, 4, 5, 6, 7, 8 o 9) que, multiplicats per successives potències de 10 i sumats, donen aquest nombre:

$$3504_{10} = 3 \times 10^3 + 5 \times 10^2 + 0 \times 10^1 + 4 \times 10^0 = 3000 + 500 + 0 + 4$$

Anàlogament, la notació binària d'un nombre natural consisteix en una successió de dígit binaris (0 o 1) que, multiplicats per successives potències de 2 i sumats, donen el nombre representat. Per exemple, la notació binària 1101 representa el nombre decimal 13:

$$1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 0 + 1 = 13_{10}$$

⁴La presentació d'imatges al monitor sol estar gestionada per un processador dedicat exclusivament a aquesta tasca (*GPU*: Graphics Processing Unit), el qual pot estar a la placa base o a la targeta gràfica.

Exercici 1.2
Indiqueu els nombres naturals en codificació decimal que corresponen a les notacions binàries $1010)_2$, $1000)_2$, $1111101000)_2$ i $10000000000)_2$.
Solució: $10)_{10}$, $8)_{10}$, $1000)_{10}$ i $1024)_{10}$.

Per expressar un nombre decimal en notació binària es divideix successivament per 2 fins a obtenir un quocient més petit que 2. Aquest quocient, seguit de tots els residus de les divisions anteriors (cada un dels quals pot ser 0 o 1) començant per l'últim, constitueixen la notació binària d'aquell nombre. Per exemple, per expressar el $13)_{10}$ en notació binària, farem les divisions:

13

2

1

6

2

0

3

2

1

1

que condueixen al resultat $1101)_2$.

Exercici 1.3
Escriviu en notació binària els nombres $63)_{10}$, $64)_{10}$ i $65)_{10}$.
Solució: $111111)_2$, $1000000)_2$, $1000001)_2$.

La unitat d'informació més petita que pot manipular un ordinador que treballa amb codificació binària és un dígit binari o *bit* (B-inary dig-IT). Qualsevol memòria de l'ordinador està estructurada com una successió de bits, cadascun dels quals podrà emmagatzemar un 0 o un 1. Com que el bit és una unitat de mesura molt petita, s'ha definit l'*octet* o *byte*, que equival a 8 bits, i se simbolitza mitjançant una 'B'. Els múltiples decimals del bit o del byte es poden identificar amb els mateixos prefixos que s'utilitzen per a qualsevol altra unitat (kilo, mega, giga, tera, etc.). No obstant això, el fet que els ordinadors utilitzin codificació binària fa que sigui convenient utilitzar múltiples en què el factor sigui una potència de 2 en lloc de 10. Això s'explica perquè les potències de 2 són nombres rodons en notació binària, de la mateixa manera que les de 10 ho són en la notació decimal; per exemple, $2^6 = 64)_{10} = 1000000)_2$ (exercici 1.3). Com que la potència de 2 més propera a 1000 és $2^{10} = 1024)_{10} = 10000000000)_2$ (exercici 1.2), se sol utilitzar aquest nombre en lloc de 1000 per definir els múltiples kilo, mega, etc., de la mateixa manera que s'ha utilitzat un nombre rodó en binari, $1000)_2 = 8)_{10}$, en comptes del nombre rodó en decimal $10)_{10} = 1010)_2$ per definir el byte. Això ha creat una ambigüitat en el significat d'aquests prefixos que ha estat resolta per la Comissió Electrotècnica Internacional el desembre de 1998 mitjançant la introducció de prefixos específics per als múltiples binaris. La taula següent recull aquests noms juntament amb els factors corresponents, els símbols que els representen i els prefixos decimals dels quals deriven:

Factor binari	Nom	Símbol	Factor decimal	Nom	Símbol
1024^1	kibi	Ki	1000^1	kilo	k
1024^2	mebi	Mi	1000^2	mega	M
1024^3	gibi	Gi	1000^3	giga	G
1024^4	tebi	Ti	1000^4	tera	T
1024^5	pebi	Pi	1000^5	peta	P
1024^6	exbi	Ei	1000^6	exa	E

Els prefixos binaris encara s'utilitzen poc i, pel que fa a la memòria RAM, la major part de la gent fa servir les unitats kB, MB, GB, etc. per referir-se als KiB, MiB, GiB, etc. Ara bé, en el cas particular dels discs durs, la informació comercial sol expressar la seva capacitat en GB o TB decimals. Per exemple, si la capsa d'un disc dur indica que aquest és de 100 GB, en consultar la seva capacitat a través de l'ordinador podem trobar-nos que aquest indica un valor d'uns 93,1 GB, quan hauria de dir

93,1 GiB (gibibytes) (vegeu l'exercici 1.4).

Exercici 1.4
Quina capacitat en GiB té un disc de 100 GB?
Solució: uns 93,132 GiB.

Exercici 1.5
Quanta memòria, expressada en KiB, ocupa una imatge de 640×480 píxels si es fan servir 8 bits per codificar la intensitat de cada un dels tres colors primaris (imatge de 24 bits)?
Solució: 900 KiB.

En contextos tècnics també s'utilitza, de vegades, la unitat *paraula* per mesurar quantitats d'informació. La paraula ve a ser la unitat d'informació amb què treballa el processador, i pot ser de 32, 64... bits, depenent del tipus de processador. Els múltiples de la paraula es defineixen de la mateixa manera que els dels bytes o els bits; així, una mebiparaula —en anglès *mebiword* (MiW)— són 1024^2 paraules. La posició de cada paraula s'identifica mitjançant un nombre que es coneix com l'*adreça* de la posició de memòria. Per exemple, una representació esquemàtica de la memòria RAM d'un ordinador amb paraules de 32 bits podria ser:

Adreça	Contingut de la paraula de memòria
0	01000101 01001101 00010010 01001010
1	10100001 01010011 01000101 01001101
2	00010010 01001010 00010010 01001010
...	...

La velocitat de transmissió d'informació a través d'una xarxa (Ethernet, línia telefònica, etc.) sol expressar-se en múltiples de bit per segon; així, en el moment d'escriure aquest text, la xarxa telefònica admet una velocitat de transmissió d'informació de 56 Kibit/s (kibibits per segon) —és a dir, 56×1024 bit/s— si s'utilitza un mòdem convencional i de 20 Mibit/s (mebibits per segon) amb el sistema ADSL (Asymmetric Digital Subscriber Line); les targetes d'Ethernet arriben fins a 10 Gibit/s.⁵ En la pràctica, les velocitats efectives solen ser més petites que aquests valors màxims (almenys un 20%), entre d'altres coses, perquè l'enorme flux d'informació (incloent-hi els protocols de comunicació entre ordinadors) satura tant les línies per les quals es transmet com els ordinadors que la distribueixen (els *servidors*).

Exercici 1.6
Indiqueu el temps que trigarà (com a mínim) a transmetre's 700 MiB d'informació a través de la xarxa telefònica si fem servir un mòdem de 56 Kibit/s. ¿I si fem servir una línia ADSL que permet *baixar* informació a 3 Mibit/s i *pujar-la* a 320 Kibit/s (el lector haurà endevinat per què se'n diu "asimètric" aquest sistema de comunicació)?
Solució aproximada: 28 h; 31 min; 5 h.

Per codificar nombres enters (inclosos els negatius) se sol reservar un bit per al signe (una manera de fer-ho és posar-hi un 0 si el nombre és positiu i un 1, si és negatiu). Per codificar nombres amb decimals es guarden les seves xifres significatives per una banda i la posició de la coma per l'altra. El signe s'indica en un altre bit. Per codificar lletres, xifres (com a caràcters, no com a nombres) i altres símbols ortogràfics, matemàtics, etc. es fa servir normalment la codificació ASCII (American Standard Code for Information Interchange) —també anomenada US-ASCII—, que atribueix una

⁵ Aquestes xifres canvien molt ràpidament i és probable que estiguin obsoletes quan el lector llegeixi aquest text. D'altra banda, no sempre és clar si les dades comercials estan expressades amb múltiples decimals, binaris o, fins i tot, barreges dels dos tipus.

successió de 7 bits a cada caràcter. A continuació mostrem alguns exemples d'aquesta codificació:

Caràcter	Codificació US-ASCII
A	$1000001_2 = 65_{10}$
a	$1100001_2 = 97_{10}$
2	$0110010_2 = 50_{10}$
+	$0101011_2 = 43_{10}$

D'aquesta manera es poden codificar 128 caràcters (vegeu l'exercici 1.7).⁶ Com succeeix amb molts "estàndards", hi ha diverses variants i extensions de la codificació US-ASCII. Entre les més utilitzades hi ha les extensions ISO/IEC 8859-1 (o Latin-1) i la ISO/IEC 8859-15, que afegeixen un bit a la US-ASCII per tal de duplicar el nombre de codis (vegeu l'exercici 1.7) i poder incloure lletres accentuades i certs caràcters que no existeixen en anglès.⁷

Exercici 1.7

Quants caràcters es poden codificar mitjançant la codificació ASCII original (de 7 bits per caràcter)? I si fem servir una codificació ASCII estesa (de 8 bits per caràcter)?

Solució: $2^7 = 128$; $2^8 = 256$.

Exercici 1.8

Quants nombres enters podem guardar (en format binari) en 4 bytes?

Suggeriment: Tingueu en compte que, en codificació decimal, tres dígitos permeten codificar des del 0 fins al 999; és a dir, 10^3 nombres.

Solució: $2^{32} = 4\,294\,967\,296$.

Exercici 1.9

Quantes pàgines de text es poden emmagatzemar (en codificació ASCII de 8 bits per caràcter) en un CD-ROM de 700 MiB? (suposeu que hi ha, per terme mitjà, 32 línies per pàgina i 64 caràcters per línia).

Solució: 358 400 pàgines.

Exercici 1.10

Indiqueu si és certa o falsa cadascuna de las afirmacions següents: a) Tota la informació que s'emmagatzema en un ordinador està codificada amb els dígitos 0 i 1. b) La informació continguda en la memòria ROM es perd quan es tanca l'ordinador. c) La informació que es troba en la memòria RAM es pot esborrar i modificar. d) El monitor d'un ordinador forma part del seu software.

Solució: a) Certa; b) falsa; c) certa; d) falsa.

1.4 El programari

D'una manera molt general, podem classificar el *software* o programari en diferents categories: el sistema operatiu, les aplicacions i, si adoptem l'accepció més ampla del terme *software*, els documents.

El *sistema operatiu* és un conjunt d'instruccions que permeten començar a treballar amb l'ordinador i fer les tasques més generals, com llegir el contingut d'un CD, copiar-ne una part en el disc fix,

⁶De fet, una part de les 128 combinacions de bits (les que corresponen als nombres 0 a 31_{10} i 127_{10}) estan reservades per a caràcters de control no imprimibles, els quals s'utilitzen per controlar certs dispositius; per exemple, impressores. Així, el caràcter 10_{10} —*line feed*— significa canvi de línia. El lector interessat en els diferents estàndards de codificació de caràcters pot consultar la web: <http://es.wikipedia.org/wiki/ASCII>

⁷Tot i així, la gran varietat de caràcters que s'utilitzen en les diferents llengües del món fa que les 256 possibilitats que ofereixen 8 bits es quedin molt curtes, i s'estan desenvolupant codificacions que pretenen ser pràcticament exhaustives. La més estesa és la UNICODE, que admet $1\,114\,112 (= 2^{20} + 2^{16})$ caràcters, dels quals a hores d'ara estan assignats uns 96 000. Els primers 256 coincideixen amb la codificació ISO/IEC 8859-1, els 128 primers de la qual són els mateixos que els de la US-ASCII.

imprimir una imatge en una impressora, mostrar el contingut d'un dispositiu de memòria flaix, etc. El caràcter general d'aquestes instruccions fa que sigui un programari necessari independentment de la utilització concreta que es vulgui fer de l'ordinador. Hi ha sistemes operatius per a cada tipus de processador, i pot haver-n'hi més d'un que funcioni en un ordinador determinat. Per exemple, un ordinador amb un processador Intel Pentium pot funcionar amb diferents versions dels sistemes operatius Windows, GNU/Linux i MacOS X, entre d'altres.

Una *aplicació* o *programa* és un conjunt d'instruccions que permet realitzar tasques lligades a un ús concret de l'ordinador. Per exemple, un *editor* o *processador de textos* és una aplicació que permet escriure textos, guardar-los en discos, llegir-los i modificar-los, imprimir-los, etc.; un *full de càlcul* és una aplicació que permet fer diverses operacions amb col·leccions de dades disposades en forma de taula; un *gestor de bases de dades* és una aplicació que permet manipular la informació d'un banc de dades (ordenar-la, classificar-la, seleccionar-ne un part, etc.); un *navegador* és una aplicació que permet *navegar* per la xarxa Internet (cercar informació, visualitzar-la, transferir-la al disc fix, etc.); i els *compiladors* i *intèrprets* són aplicacions que tradueixen un conjunt d'instruccions escrit en un llenguatge de programació estàndard a un llenguatge propi de l'ordinador, tal com veurem en el capítol 2.

Determinades aplicacions permeten crear *documents* que contenen informació. És el cas dels editors de textos, que creen i modifiquen documents textuais (cartes, informes, etc.); els fulls de càlcul, que creen i modifiquen taules de valors que poden estar interrelacionats (documents que s'anomenen també *fulls de càlcul*); els gestors de bases de dades, que creen o modifiquen bases de dades; etc.

La classificació anterior no és estricta. Així, les aplicacions que s'instal·len en l'ordinador juntament amb el sistema operatiu —com ara una aplicació per navegar per Internet, un visualitzador de vídeo o un reproductor de so— i els documents d'informació i ajuda poden considerar-se part del sistema.

1.5 Sistemes operatius

Quasi tots els sistemes operatius actuals permeten manipular l'ordinador de dues maneres:

- En *mode gràfic* o *amb interfície gràfica* (GUI: Graphical User Interface): el monitor presenta una sèrie de *menús* desplegable on les instruccions del sistema operatiu apareixen escrites i se seleccionen amb l'ajuda d'un ratolí (*clicant* sobre un menú i *arrossegant* el ratolí fins que *s'activi* la instrucció desitjada), *finestres* (rectangles dibuixats sobre la pantalla) que poden contenir gràfics i *botons clicables* per escollir certes opcions, dibuixets anomenats *icones* per representar certs components de maquinari o de programari, etc.
- En *mode text*: cada instrucció s'ha de teclejar i s'executa en prémer la tecla de retorn () , anomenada també *Entrar* o *Intro*. El monitor —o una finestra d'aquest on apareix el text teclejat— pot mostrar caràcters però no gràfics. Se'n sol dir una *consola* o *terminal*.

El mode text presenta certs inconvenients: dificultat de memoritzar les instruccions menys usuals, lentitud quan s'han d'escriure instruccions llargues, facilitat de cometre errors en teclejar-les, etc. El mode gràfic resulta més intuïtiu i fàcil d'aprendre, cosa que redueix considerablement les necessitats de memorització i les possibilitats d'errors. Els sistemes més utilitzats en l'actualitat són el Windows i les diverses derivacions de l'Unix: GNU/Linux, Mac OS X, Solaris, etc. El mode gràfic és, amb diferència, la manera d'operar més estesa, i el treball en mode text està cada dia més restringit als desenvolupadors de programari i als encarregats del manteniment de sistemes.

Als apèndixs A i B podeu trobar una breu ressenya sobre el desenvolupament històric dels sistemes operatius i, en particular, de les interfícies gràfiques.

1.6 Estructuració dels discos

De la mateixa manera que es necessita un cert grau d'ordre en els documents que tenim al nostre despatx per poder localitzar-los amb facilitat, és convenient estructurar la informació guardada en el disc fix —o en qualsevol altre mitjà d'emmagatzematge d'informació— d'un ordinador. Una bona

part de les instruccions dels sistemes operatius serveixen per crear aquesta estructuració i treure'n profit.

Un *fitxer* o *arxiu* és un bloc d'informació que s'identifica mitjançant un nom. La informació continguda en un fitxer pot ser tant una part del sistema operatiu, com una aplicació (o una part d'aquesta) o un document. Així, quan escrivim un document amb un editor de textos i el guardem en el disc dur donant-li un nom, estem creant al disc un fitxer amb aquest nom.⁸ El mateix editor de textos estarà també emmagatzemat en un fitxer del disc (o en més d'un). Els fitxers poden contenir també dades per efectuar un càlcul matemàtic, els resultats d'aquest, l'aplicació que l'efectua, imatges o so digitalitzats, etc.

En el sistema operatiu DOS, antecessor del Windows, així com en les primeres versions d'aquest, els noms de fitxer poden tenir, com a màxim, vuit caràcters i, opcionalment, un punt i tres caràcters més. Aquests últims s'anomenen l'extensió del nom, i se solen utilitzar per indicar el tipus d'informació continguda al fitxer. Els caràcters poden ser lletres, xifres i caràcters especials excepte / \ [] : ; < > + = . , . Per exemple, un fitxer que conté una aplicació per calcular el producte de dues matrius es podria anomenar **prod2m.exe**, on l'extensió **exe** indicaria que el fitxer conté una aplicació; és a dir, que es tracta d'un fitxer *executable*. Les dades que necessita l'aplicació, és a dir, les matrius a multiplicar es poden introduir amb el teclat, però també es poden llegir d'un fitxer on estiguin guardades, que es podria anomenar **prod2m.dad**. El resultat del càlcul —la matriu producte— es pot mostrar al monitor, escriure en una impressora o guardar en un fitxer del disc amb el nom de, per exemple, **prod2m.res**.

En la major part dels sistemes operatius actuals (Windows, GNU/Linux, Mac OS X...) hi ha poques restriccions en els noms que es poden donar als fitxers. Així, el fitxer de dades considerat abans es podria dir, en qualsevol d'aquests sistemes, **Matrius a multiplicar**. No obstant això, és convenient no utilitzar caràcters que tinguin un significat especial per al sistema, com /, \, - o un punt com a primer caràcter, ni espais en blanc.⁹ D'altra banda, encara hi ha aplicacions per al Windows que només funcionen correctament si els noms dels documents de treball que empen s'ajusten al sistema de 8+3 caràcters abans esmentat.

Un disc pot contenir una sèrie de fitxers sense cap altra estructuració i, de fet, aquesta és la manera habitual de disposar els fitxers en un disquet. No obstant això, quan el nombre de fitxers és considerable —com succeeix en el disc fix— és convenient agrupar-los per temàtiques en diferents *carpetes* o *directoris*, com ho faríem amb els papers del despatx. Així, una carpeta pot contenir els fitxers que constitueixen el sistema operatiu, una altra les aplicacions, una altra els documents creats per l'usuari, etc. Els noms de les carpetes han d'ajustar-se a les mateixes especificacions que els dels fitxers. Per identificar un fitxer que es troba en una carpeta determinada s'escriu el nom d'aquesta i, a continuació, el del fitxer separat pel caràcter \ (*barra inversa*) en el sistema Windows i per / (*barra inclinada*) en els altres sistemes esmentats. Per exemple, en Windows, la identificació completa d'un fitxer amb nom **apunts_mates** que es troba en la carpeta **documents** del disc dur —el qual s'identifica, normalment, mitjançant la notació **C:**—, serà **C:\documents\apunts_mates**. Als altres sistemes la identificació seria **/documents/apunts_mates**. En endavant ens referirem a sistemes de la família Unix, inclosos el Linux i el Mac OS X, sempre que no indiquem el contrari. Per aplicar la discussió al sistema Windows s'haurien de substituir les barres inclinades (/) per barres inverses (\) i afegir a l'esquerra la identificació del disc: **A:** per a la disquetera i **C:**, **D:**... per als discs durs, CD, etc.

Cada carpeta o directori es pot estructurar en noves *carpetes* o *subdirectoris*. Per exemple, podríem definir, en el directori **documents**, dos subdirectoris: **dibuixos** i **apunts** i ens referirem a un document d'aquest últim com **/documents/apunts/apunts_mates**. Les paraules carpeta, directori i subdirectori s'utilitzen indistintament (tot directori és subdirectori d'algun altre o del disc sencer). La indicació completa del directori on es troba un fitxer s'anomena el *camí* (*path*) o *ubicació* del fitxer. Així, en l'exemple anterior **/documents/apunts/** és el camí que condueix al fitxer **apunts_mates**. Els directoris confereixen al disc una estructura d'arbre, en què les branques serien els directoris i les fulles els fitxers (figura 1.2). El nivell més baix d'aquesta estructura se sol anomenar l'*arrel* (/)

⁸La denominació *fitxer* o *arxiu* es remunta als orígens de la informàtica: un fitxer es podia emmagatzemar com un conjunt de fitxes perforades, que eren fitxes de cartolina sobre les quals s'enregistrava la informació mitjançant una perforadora. Cada fitxa contenia una línia del fitxer, amb un màxim de 80 caràcters. La perforadora s'assemblava a una màquina d'escriure, i escrivia els caràcters de la línia a la part superior de la fitxa al temps que feia uns forats a sota cada caràcter; aquests permetien a la lectora de fitxes —un perifèric d'entrada— identificar els caràcters i transmetre la informació al processador.

⁹En certs casos s'han d'escriure entre cometes els noms de fitxer que contenen caràcters d'aquests.

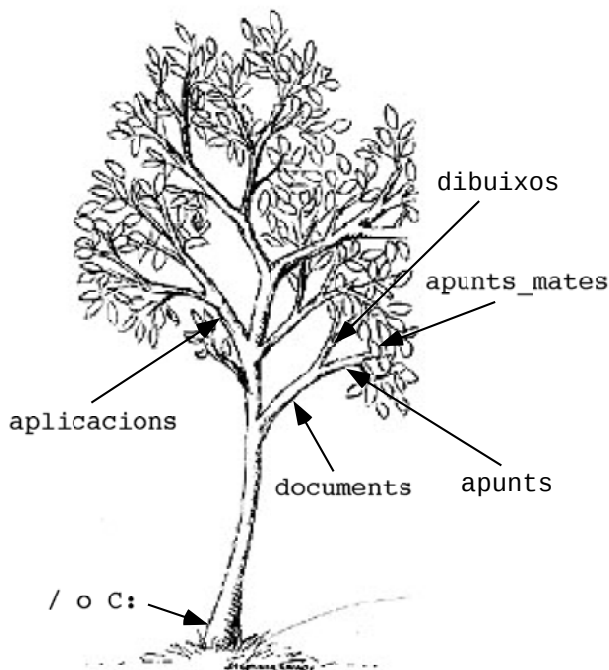


Figura 1.2: Estructuració d'un disc.

(s'hauria de parlar d'un arbust, més que d'un arbre, ja que no hi ha tronc). Es diu que els fitxers i subdirectoris *pengen* del directori on s'ubiquen.

1.7 El sistema operatiu GNU/Linux

El GNU/Linux és un sistema operatiu *lliure* (vegeu l'apèndix B) que es pot obtenir gratuïtament a la xarxa Internet; per exemple, de l'organisme sense ànim de lucre Debian, o de l'empresa Canonical Ltd. (distribució Ubuntu, basada en Debian). També es pot comprar —a un preu força moderat— a empreses que recopilen el que consideren més important i el distribueixen (RedHat, SuSe, Mandriva ...). A més, aquestes proporcionen suport tècnic durant un cert període a partir de la data de compra. Les distribucions de GNU/Linux inclouen diferents GUI (KDE, Gnome, etc.) que faciliten les operacions més usuals; no obstant això, és interessant saber com es poden escriure i executar les instruccions del sistema, ja que s'ha de treballar en mode text per poder treure-li tot el suc. Les diferents GUI funcionen de manera bastant semblant. No obstant això, s'ha de tenir en compte que, fins i tot per a una mateixa distribució i GUI, poden haver-hi petites diferències depenent de la versió i de com s'hagi configurat el sistema.

Quan s'engega un ordinador, aquest busca un sistema operatiu en els discos accessibles i, si el troba, comença a llegir les seves instruccions d'arrencada i copiar-les o *carregar-les* en la memòria RAM. Si tenim més d'un sistema operatiu instal·lat al disc dur, per exemple, Windows i GNU/Linux, trobarem de seguida una pantalla que ens oferirà la possibilitat d'escollir el sistema que vulguem utilitzar. Seleccionarem l'opció Linux amb les *tecles del cursor* (que estan a la part inferior, entre el teclat alfanumèric i el numèric) i, en prémer la tecla de retorn () es carregarà aquest sistema.

El sistema GNU/Linux també es pot carregar des d'un disc compacte d'arrencada o *autònom* (un *live CD*), la qual cosa permet utilitzar-lo sense haver d'instal·lar res al disc dur. Només caldrà engegar el PC amb el CD autònom ficat en el lector de CD i, si la BIOS (Basic Input/Output System) està configurada de la manera adient,¹⁰ arrencarà el sistema del CD. En el moment d'escriure aquest text, els CD autònoms més utilitzats deriven d'un que s'anomena *Knoppix* (vegeu la bibliografia).

¹⁰Si l'ordinador busca el sistema operatiu en el disc dur abans que en el CD s'haurà de modificar la BIOS per invertir l'ordre de cerca. Per modificar la BIOS s'ha d'interrompre l'arrencada (quan s'inicia) prement una tecla que pot ser Supr, F2, F1, Esc, etc., depenent del fabricant de la placa base. En començar l'arrencada sol aparèixer (de manera força fugaç) una indicació de la manera d'entrar en la BIOS.

El GNU/Linux és un sistema *multiusuari*, és a dir, permet definir diferents usuaris, cada un amb el seu nom o *codi* i la seva *paraula de pas* (*password*), que poden utilitzar el mateix ordinador —en moments diferents o, fins i tot, simultàniament si hi ha altres ordinadors que hi estan connectats via xarxa— sense perill d'interferències: cada usuari té assignat un *directori propi* (el *directori* o *carpeta de l'usuari*), pot protegir els seus fitxers perquè no puguin ser esborrats, modificats o, fins i tot, vists per un altre, pot configurar l'entorn de treball al seu gust, etc. Sempre hi ha un usuari definit —el *root*— que té accés a tota la informació i pot executar qualsevol instrucció. Els altres usuaris tenen algunes opcions vetades per tal de garantir la integritat del sistema envers errors accidentals o accions malintencionades.

1.7.1 La GUI

La resta d'aquest capítol està concebuda per a un lector que disposi d'un ordinador arrencat amb el sistema operatiu GNU/Linux per tal de posar en pràctica, a mesura que llegeix el text, les instruccions que s'aniran introduint.¹¹ Per iniciar una sessió de treball s'ha d'introduir, normalment, un codi i una paraula de pas prèviament donats d'alta per l'usuari *root*. En endavant suposarem que el codi de l'usuari és *usuari1* i que el seu directori és */home/usuari1/*. Una vegada iniciada la sessió, el monitor mostra algunes icones distribuïdes per la pantalla —l'*escriptori*— amb el nom corresponent a sota: pot haver-hi una paperera, un disquet o *floppy*, un disc dur etiquetat com a *hda1*, etc. A la part inferior (de vegades a la superior) hi ha una franja amb més icones (la *barra de menús*). Una d'aquestes sol ser una caseta que representa la carpeta de l'usuari (la seva *home*). Clicant-hi a sobre una sola vegada amb el botó esquerre del ratolí s'obre una finestra que mostra el contingut de la carpeta; és a dir, els noms dels fitxers que conté i els dels subdirectoris (o carpetes) que hi pengen.

Una de les aplicacions incloses amb el sistema és un editor de textos senzill que ens serà molt útil per crear i modificar fitxers amb programes. Pot ser que hi hagi una icona en l'escriptori o en la barra de menús que representi aquest editor (un llapis, una llibreta, un paper amb una ploma, etc.); en aquest cas, *obrirem* l'editor clicant sobre la icona. Si no, haurem de desplegar el *menú d'aplicacions* (sol estar a la part esquerra de la pantalla), buscar un editor i seleccionar-lo. En obrir-se o *arrencar-se*, l'editor es llegeix del disc i es copia o *carrega* en la memòria RAM perquè puguem treballar-hi. Un cop carregat, apareix una finestra en blanc que representa un fitxer nou. Si teclegem un text, aquest apareixerà en la finestra just a continuació del *cursor* (la ratlleta vertical que parpelleja). El text que escrivim es va guardant a la memòria RAM i, en conseqüència, es perdrà si, per exemple, es produeix un tall de corrent. Per guardar-lo en una memòria permanent (disc dur, disquet, memòria USB, etc.) hem de seleccionar l'opció *Desar* o *Guardar* del menú *Fitxer* o *Arxiu*, a la part superior de la finestra de l'editor. En fer-ho, s'obre una finestra perquè hi escrivim el nom que vulguem donar al fitxer. A la part superior d'aquesta finestra s'indica la ubicació que tindrà el fitxer, la qual es pot modificar clicant-hi i escrivint la nova ubicació. En clicar sobre el botó *Acceptar* o *Desar*, el fitxer es gravarà al disc, com podrem comprovar clicant sobre la finestra de la carpeta de l'usuari i buscant-lo. Si tornem a l'editor (clicant sobre la seva finestra) i seguim escrivint o modifiquem el text que hi havia, les modificacions afectaran només la versió del fitxer que resideix a la memòria RAM, la qual no es guardarà al disc fins que tornem a seleccionar l'opció *Desar*. Ara ja no caldrà introduir cap nom, ja que es mantindrà el que vam assignar la primera vegada. Si volem conservar la versió del fitxer que hi havia al disc i crear-ne un de nou amb la versió modificada que resideix a la memòria RAM, haurem de seleccionar l'opció *Desar com. . .*, que ens demanarà el nom que volem donar al nou fitxer.¹²

Convé dedicar un cert temps a escriure sobre el fitxer i fer diferents modificacions del text escrit per familiaritzar-se amb el funcionament de l'editor: clicar sobre el text escrit per insertar-hi una paraula, clicar sobre el text i arrossegar el ratolí per seleccionar un fragment que vulguem esborrar (tecla ) , copiar (opció *Copiar* del menú *Editar* i, després de clicar en un altre punt del text, opció *Enganxar*), moure (opcions *Tallar* i *Enganxar*), insertar línies en blanc amb la tecla de retorn () , moure's pel text fent servir les tecles del cursor, etc.

¹¹S'ha de tenir present que poden haver-hi moltes variacions en els detalls de funcionament i aspecte de la GUI —dependent, sobretot, de la distribució de GNU/Linux emprada— respecte del que descrivim en aquest apartat.

¹²Si s'està treballant amb el sistema operatiu arrencat des d'un *live CD* el directori de l'usuari està en un *disc virtual* o *ramdisk*, que és una zona de la memòria RAM que es presenta com si fos un disc. Per tant, qualsevol fitxer guardat en aquest directori que es vulgui conservar després de tancar la sessió de treball s'haurà de copiar prèviament en una memòria permanent.

Si tornem a la carpeta de l'usuari podrem eliminar un dels fitxers que hem creat arrossegant-lo amb el ratolí fins a la icona de la paperera. En realitat, aquesta operació no elimina definitivament el fitxer, que encara es pot recuperar clicant sobre la paperera i arrossegant-lo de nou sobre la carpeta de l'usuari (o qualsevol altre punt de l'escriptori). Per esborrar-lo definitivament hauríem de buidar la paperera, després de llençar-hi el fitxer, clicant-hi amb el botó dret del ratolí i seleccionar l'opció *Buidar la paperera*. També es pot eliminar definitivament un fitxer seleccionant l'opció *Esborrar* en el menú que apareix en clicar amb el botó dret del ratolí sobre la icona del fitxer.

Si, al menú *Editar* de la carpeta de l'usuari, seleccionem l'opció *Crear nou* seguida de *Carpeta...* i introduïm un nom en la finestra que apareix —per exemple, **exercicis**— haurem creat un subdirector del directori de l'usuari que es representarà mitjançant una nova icona amb forma de carpeta. Clicant sobre aquesta icona s'obrirà una finestra en blanc, ja que el nou subdirector no conté cap fitxer ni carpeta (està buit). La fletxa que apunta cap amunt a la part superior d'aquesta finestra ens tornarà a la carpeta de l'usuari. Des d'aquesta carpeta podem arrossegar un dels fitxers que conté a la nova carpeta **exercicis**, la qual cosa farà aparèixer un petit menú: si escollim l'opció *Moure* el fitxer canviarà d'ubicació (desapareixerà d'**usuari1** i passarà a **exercicis**) i si escollim *Copiar* es mantindrà a la ubicació original i n'aparèixerà un duplicat a la carpeta **exercicis**.

Per copiar fitxers a un disquet o des d'un disquet hem d'introduir-lo en la disquetera —amb el foradet de la cantonada superior esquerra tancat si volem escriure-hi—, clicar sobre la icona amb forma de disquet i esperar que s'obri una finestra que en mostrarà el contingut.¹³ Llavors, el disquet estarà disposat per treballar-hi (*muntat*), la qual cosa s'indica mitjançant un petit triangle al costat de la icona. Si volem copiar fitxers al disquet o des d'aquest al disc fix els haurem d'arrossegar de la mateixa manera que quan els copiem entre diferents carpetes d'un mateix disc. A diferència del que succeeix al sistema operatiu Windows, no hem de prémer directament el botó d'expulsió del disquet quan vulguem treure'l després de guardar-hi informació; abans hem de *desmuntar-lo*, per la qual cosa tancarem la finestra del disquet, clicarem sobre la seva icona amb el botó dret del ratolí i seleccionarem l'opció *Desmuntar*. A més, s'ha d'esperar que s'apagui el llumet que hi ha al costat del botó d'expulsió, indicatiu d'activitat a la disquetera (això també s'aplica al sistema Windows).

Quan endollem un dispositiu USB o FireWire (per exemple, un dispositiu de memòria flaix o un disc dur extern) apareix una icona en l'escriptori que el representa. Amb aquests dispositius es treballa de manera similar que amb els disquets: en clicar sobre la icona del dispositiu aquest es munta per poder treballar-hi, i abans de desendollar-lo s'ha de desmuntar clicant sobre la icona amb el botó dret del ratolí i seleccionant l'opció *Desmuntar* en el menú que hi apareix (el muntatge s'efectua automàticament si està activada l'opció corresponent del sistema).

Encara que aquesta breu introducció a una GUI és suficient per al que farem en endavant, el lector interessat pot explorar les diferents opcions de cada menú per descobrir per si mateix altres possibilitats que li ofereix el sistema operatiu.

1.7.2 Instruccions en mode text

La GUI proporciona una imatge gràfica de l'estructuració del disc fix en directoris i fitxers molt útil per comprendre la filosofia general del sistema operatiu. És convenient, no obstant això, tenir una idea de com es treballa en mode text, la qual cosa s'entendrà de seguida executant instruccions que reproduïxin algunes de les tasques que es poden realitzar de manera gràfica.

Per obrir una consola o terminal (en Unix/Linux s'anomena també una *shell*) clicarem sobre la icona que normalment representa un monitor o una petxina (si aquesta no està en la barra de menús s'haurà de buscar en el menú d'aplicacions). A la part superior esquerra de la consola apareixerà una línia de text —el *prompt* o *indicador*— que, depenent de com estigui configurat el sistema, pot indicar el nom de l'usuari, una identificació de l'ordinador, el nom del directori on s'està treballant, etc. A continuació del *prompt* hi ha el *cursor*, que és un quadradet (o rectangle) que assenyalava la posició de la finestra on començarà a aparèixer el text que escriurem des del teclat. El *prompt* que trobareu, juntament amb el cursor, pot ser d'aquest estil:

```
[/home/usuari1] $ █
```

L'aparició del *prompt* indica que l'ordinador està en disposició de rebre instruccions del sistema operatiu. Per executar-ne una hem d'escriure-la i prémer la tecla de retorn (). Així, si escrivim:

¹³Si aquesta icona no està en l'escriptori s'haurà d'executar la instrucció **mount** en mode text.

pwd

(abreviació de *print working directory*) i premem , ens apareixerà una indicació de quin és el *directori de treball*, és a dir, el directori on s'està treballant (també se'n diu directori *actiu* o directori en què s'està). Si aquest és el directori de l'usuari apareixerà:

```
/home/usuari1
```

Per treure una llista dels fitxers ubicats en aquest directori i els subdirectoris que hi pengen podem executar la instrucció **ls** (abreviació de *list*). Els ítems d'aquesta llista es corresponen amb les icones que apareixen en clicar sobre la icona de la carpeta de l'usuari. Si volem que aquesta llista inclogui més informació de cada fitxer o subdirectori podem afegir l'opció **-l** a la instrucció **ls**:

```
ls -l
```

Si, per exemple, hi ha dos fitxers anomenats **fitxer1** i **fitxer2** i dos subdirectoris anomenats **cartes** i **informes** apareixerà una cosa semblant a això:

```
total 4
drwx-----  3 usuari1  usuari1      1360 nov 27 20:04 cartes
-rw-r--r--   1 usuari1  usuari1        48 nov 27 20:04 fitxer1
-rw-r--r--   1 usuari1  usuari1        62 nov 27 20:04 fitxer2
drwxr-xr-x   2 usuari1  usuari1      1360 nov 27 20:04 informes
```

és a dir, se'ns indica quin és el nombre total d'ítems —fitxers o subdirectoris— i s'escriu una línia per a cadascun d'aquests ítems. El primer caràcter d'aquesta línia és una **d** si l'ítem és un subdirectori i un guió quan es tracta d'un fitxer normal. A continuació s'indica el grau de protecció que té cada ítem (les lletres **r**, **w** i **x** indiquen permisos de lectura, escriptura i execució per part de diferents categories d'usuari), l'*inode* (un nombre que té un significat massa tècnic per ser descrit aquí) seguit del nom del propietari de l'ítem i el del grup d'usuaris al qual pertany, la memòria que ocupa en el disc mesurada en bytes, la data i l'hora de la seva última modificació i el seu nom.

Per crear un subdirectori del directori actiu s'utilitza la instrucció **mkdir** (*make directory*), equivalent a l'opció *Crear nova carpeta* del menú *Editar* de la GUI:

```
mkdir nomsubdir
```

on **nomsubdir** és el nom que vulguem donar al nou subdirectori. Per *passar* a aquest, és a dir, per fer que el nou subdirectori sigui l'actiu, usarem la instrucció **cd** (*change directory*):

```
cd nomsubdir
```

En executar-la el *prompt* canviarà per indicar que ens trobem en el subdirectori **nomsubdir** del directori de l'usuari; per exemple:

```
[/home/usuari1/nomsubdir] $ █
```

Aquest subdirectori encara no tindrà cap ítem, com es pot comprovar executant la instrucció **ls**. El directori del qual penja, és a dir, l'anterior directori de treball, es pot representar abreujadament mitjançant dos punts seguits. Així, per tornar a aquell directori des del **nomsubdir** podem fer:

```
cd ..
```

La instrucció **cd** permet passar directament a qualsevol directori identificat mitjançant la seva designació completa; per exemple, la instrucció

```
cd /home/usuari2
```

ens passarà al directori de l'usuari2 (si no està protegit). Després podem executar la instrucció

```
cd /home/usuari1
```

per tornar al nostre directori.

Per referir-nos a un subdirectori o fitxer del directori actiu no cal incloure la indicació d'aquest últim. Així, tot i que es podria passar des del directori de l'usuari1 al subdirectori **nomsubdir** amb la instrucció:

```
cd /home/usuari1/nomsubdir
```

és més pràctic fer-ho de la manera que hem vist abans (`cd nomsubdir`).

Per duplicar un fitxer, és a dir, per crear-ne un de nou que contingui la mateixa informació que l'original s'utilitza la instrucció `cp` (*copy*). Per exemple, si el fitxer original s'anomena `fitxer1` i volem que el nou s'anomeni `fitxer2`, ambdós ubicats al directori actiu, farem:

```
cp fitxer1 fitxer2
```

Si la ubicació d'un d'aquests fitxers no és el directori actiu haurem d'indicar-ho; per exemple:

```
cp fitxer1 nomsubdir/fitxer2
```

copiarà el fitxer `fitxer1` del directori actiu en un nou fitxer del subdirectori `nomsubdir` anomenat `fitxer2`. Si s'omet el nom del nou fitxer s'agafarà el mateix nom que tenia l'original; així, la instrucció

```
cp fitxer1 nomsubdir
```

farà una còpia del fitxer `fitxer1` al subdirectori `nomsubdir` amb el mateix nom. Per fer aquesta mateixa operació des del directori `nomsubdir` (després d'haver fet `cd nomsubdir`) podem usar la instrucció:

```
cp ../fitxer1 .
```

on el punt que apareix al final identifica el directori actiu.

La instrucció `rm` (*remove*) permet esborrar fitxers, la qual cosa equival a llençar-los a la paperera des de la GUI i buidar-la (o seleccionar l'opció *Esborrar* després de clicar-hi amb el botó dret). Així,

```
rm informe
```

esborra del directori actiu un fitxer anomenat `informe`.

Per moure un fitxer d'un directori a un altre, la qual cosa equival a copiar-lo i esborrar després l'original, usarem la instrucció `mv` (*move*). Per exemple, executar la instrucció

```
mv document1 nomsubdir
```

equival a executar les dues instruccions

```
cp document1 nomsubdir
rm document1
```

La instrucció `mv` també permet canviar el nom d'un fitxer; per exemple, perquè el fitxer `fitxer1` passi a anomenar-se `fitxer2` farem:

```
mv fitxer1 fitxer2
```

En moltes instruccions, com ara `cp`, `mv` i `rm`, es pot utilitzar un asterisc per representar qualsevol seqüència de caràcters. Per exemple,

```
cp fitxer* nomsubdir
```

copia tots els fitxers el nom dels quals comenci per `fitxer` al subdirectori `nomsubdir`, i

```
rm *
```

esborra tots els fitxers del directori actiu.

Per eliminar un subdirectori s'han d'esborrar primer tots els fitxers que conté i, després, executar la instrucció `rmdir` (*remove directory*):

```
rmdir nomsubdir
```

Si volem veure el contingut d'un fitxer anomenat `informe` i no necessitem modificar-lo podem utilitzar la instrucció `more`:

```
more informe
```

Si el contingut del fitxer no cap a la consola, l'escriptura s'atura quan la consola està plena, i s'ha de prémer la tecla d'espaiat per prosseguir la visualització. Per tornar a la pàgina anterior s'ha de prémer la tecla `b`. Es pot abandonar la visualització abans d'arribar al final del fitxer prement la tecla `q`.

Una instrucció molt útil és **man** (de *manual*), que ens mostra la part del manual de referència del sistema que explica el funcionament d'una instrucció i les opcions que admet. Així, la instrucció:

```
man cp
```

presenta informació sobre la instrucció **cp**. Es pot controlar la visualització de la manera descrita per a la instrucció **more**.

Per als que coneguin el sistema operatiu DOS pot ser interessant saber que algunes de les seves instruccions funcionen en GNU/Linux afegint davant la lletra **m**. Per exemple, la instrucció:

```
mformat a:
```

formata o *inicialitza* un disquet, prèviament introduït en la disquetera, la qual cosa esborra tot el que pogués contenir i el prepara per gravar-hi informació. La instrucció

```
mcopy *.f a:
```

copia tots els fitxers el nom dels quals acabi en **.f** al disquet. Aquesta instrucció munta i desmunta el disquet cada vegada que s'executa; en conseqüència, aquest no haurà d'estar muntat i es podrà expulsar prement el botó d'expulsió directament, com es fa al sistema Windows. Sempre s'ha d'esperar, però, que s'apagui el llumet que hi ha al costat del botó.

Per acabar, un truc que estalvia temps i errades: el sistema memoritza les instruccions que executem en mode text, i podem recuperar-les amb les tecles `↑` i `↓` per no haver de tornar a teclejar-les. Una instrucció recuperada es pot modificar abans de tornar-la a executar (es poden emprar les tecles `→` i `←` per desplaçar el cursor sobre el text).

Exercici 1.11

Escriu les instruccions de GNU/Linux necessàries per crear un subdirectori de l'actiu amb nom **programes**, passar al nou subdirectori, moure-hi els fitxers del directori anterior el nom dels quals acabi en **.f** i veure la relació d'ítems que conté el nou subdirectori.

Solució:

```
mkdir programmes
cd programmes
mv ../*.f .
ls
```