

Tema 3. Els nombres de l'ordinador

Què aprendràs

- Per què l'ordinador només fa servir zeros i uns.
- Llegir un nombre binari i dir quant val.
- Escriure qualsevol nombre en binari.
- Fer servir l'hexadecimal, que és la drecera dels informàtics.
- Sumar dos nombres en binari.
- Entendre com es guarden les lletres: ASCII, Unicode i UTF-8.
- Fer servir l'octal, que va de tres en tres.
- Saber per què el teu disc d'1 TB diu que en té menys.
- Calcular quant triga a baixar un fitxer, sense caure a la trampa dels bits.

Curriculum del mòdul 0222 (RD 1691/2007): RA1 b).

La teva xuleta per a tot el tema

Abans de començar, queda't amb aquesta fila de nombres. **La faràs servir a tots els exercicis.**

128	64	32	16	8	4	2	1
-----	----	----	----	---	---	---	---

Fixa-t'hi: comença per l'1, a la dreta, i **cada nombre és el doble de l'anterior**. 1, 2, 4, 8, 16, 32, 64, 128. Si algun dia se te'n va del cap, comença per l'1 i ves doblant.

Consell. Cada vegada que comencis un exercici, copia aquesta fila a dalt del full. T'estalviarà la meitat de les errades. No és fer trampa: els professionals també la tenen a mà.

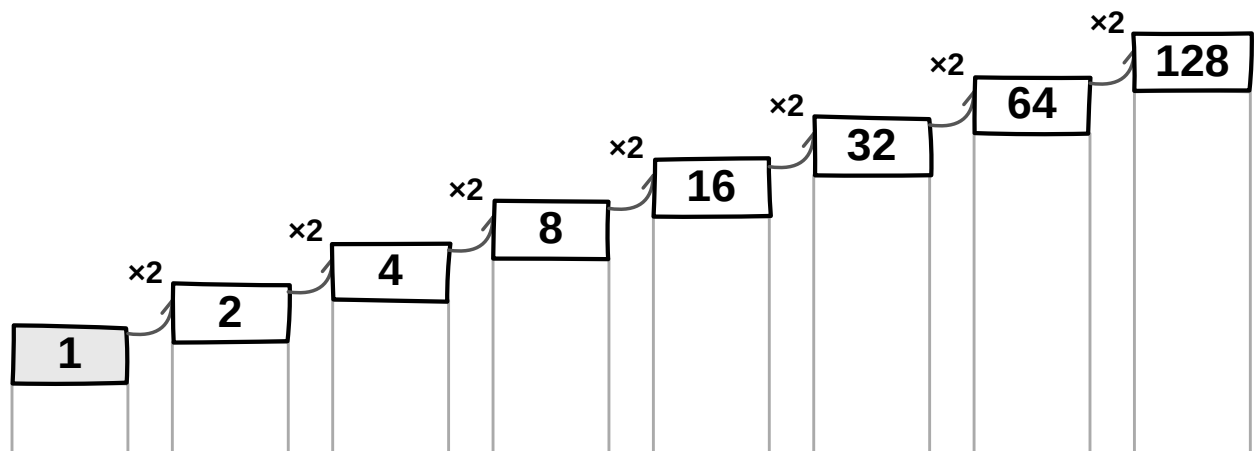


FIG 3.2 La fila de pesos: cada nombre és el doble del que té a la dreta.

Escalfa motors

Escriu la fila de memòria, sense mirar:

Ara segueix doblant quatre nombres més enllà del 128:

1. L'ordinador només sap dir sí o no

Pensa en l'interruptor del llum de la teva habitació. Només pot estar de dues maneres: encès o apagat. No hi ha terme mitjà.

Dins de l'ordinador hi ha milions d'interruptors diminuts. Cada un està encès o apagat. Res més.

Per escriure-ho fem servir dos símbols:

- **0** = apagat
- **1** = encès

Cada un d'aquests interruptors es diu **bit**.

Nosaltres comptem del 0 al 9 perquè tenim deu dits. L'ordinador compta amb 0 i 1 perquè només té dues posicions. És l'única diferència.

El que importa. Canviar de sistema no canvia la quantitat. Dotze galetes són dotze galetes. Les escriguis **12** o les escriguis **1100**, te les menges igual. Canvia l'etiqueta, no les galetes.

2. Llegir un nombre binari

Aquí és on fas servir la xuleta. El mètode té dos passos:

1. Posa el nombre binari sota la fila, **quadrant per la dreta**.
2. Suma els nombres de dalt on a sota hi hagi un **1**. On hi hagi un 0, passes de llarg.

Ja està. No hi ha més.

Exemple resultat 1

Quant val **1011**?

8	4	2	1
1	0	1	1

Hi ha un 1 sota el 8, el 2 i l'1. Els sumo:

$$8 + 2 + 1 = 11$$

1011 val **11**.

Exemple resultat 2

Quant val **1011 0110**?

128	64	32	16	8	4	2	1
1	0	1	1	0	1	1	0

Sumo els que tenen un 1 a sota:

$$128 + 32 + 16 + 4 + 2 = 182$$

Si et costa de veure, imagina't els vuit interruptors d'abans. Els que estan encesos són els que sumen:

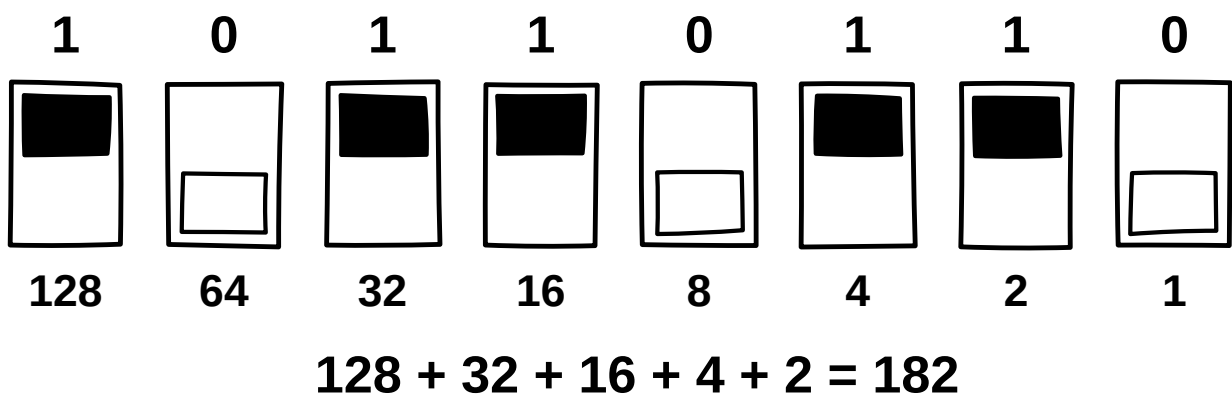


FIG 3.1 El nombre **1011 0110** vist com a vuit interruptors. Només sumen els encesos.

Truc. Els nombres binaris llargs s'escriuen en grups de quatre, com **1011 0110**. Així no et perds comptant. Fes-ho sempre.

Guiat 3.1

Quant val **0110 1101**? Col·loca els bits a la taula i suma.

128	64	32	16	8	4	2	1

Suma:

Practica 3.2 — Comença pels curts

Digues quant val cada un.

a) **10** = b) **11** = c) **100** =

d) **101** = e) **110** = f) **1000** =

g) **1001** = h) **1100** = i) **1111** =

(Comprovació del primer: **10** val 2.)

Practica 3.3 — Ara de vuit bits

Dibuixa tu la taula al costat.

a) **0000 1010** =

b) **0011 1010** =

c) **1000 0001** =

d) **1111 0000** =

e) **1111 1111** =

(Comprovació del primer: **0000 1010** val 10.)

Comprova-ho tu sol. Si el binari acaba en **0**, el resultat és **parell**. Si acaba en **1**, és **senar**. Si no et quadra, has sumat malament. Busca-ho abans de seguir.

3. Escriure un nombre en binari

Ara al revés. Et donen un nombre normal i l'has d'escriure amb zeros i uns.

Fem servir la mateixa xuleta i una sola pregunta, repetida: **hi cap?**

Vas d'esquerra a dreta. A cada nombre de la fila et preguntes si hi cap en el que et queda:

- Si **hi cap**, poses un **1** i el restes.
- Si **no hi cap**, poses un **0** i no restes res.

Exemple resultat 3

Escriure **45** en binari.

Xuleta	Hi cap en el que em queda?	Hi poso	Em queda
128	128 en 45, no hi cap	0	45
64	64 en 45, no hi cap	0	45
32	32 en 45, sí que hi cap	1	$45 - 32 = 13$
16	16 en 13, no hi cap	0	13
8	8 en 13, sí que hi cap	1	$13 - 8 = 5$
4	4 en 5, sí que hi cap	1	$5 - 4 = 1$
2	2 en 1, no hi cap	0	1
1	1 en 1, sí que hi cap	1	$1 - 1 = 0$

Llegeixo la columna "Hi poso" de dalt a baix: **0010 1101**

Quan arribes a 0 a l'última columna, has acabat. Si no arribes a 0, alguna cosa ha fallat.

Comprova-ho sempre: $32 + 8 + 4 + 1 = 45$. Correcte.

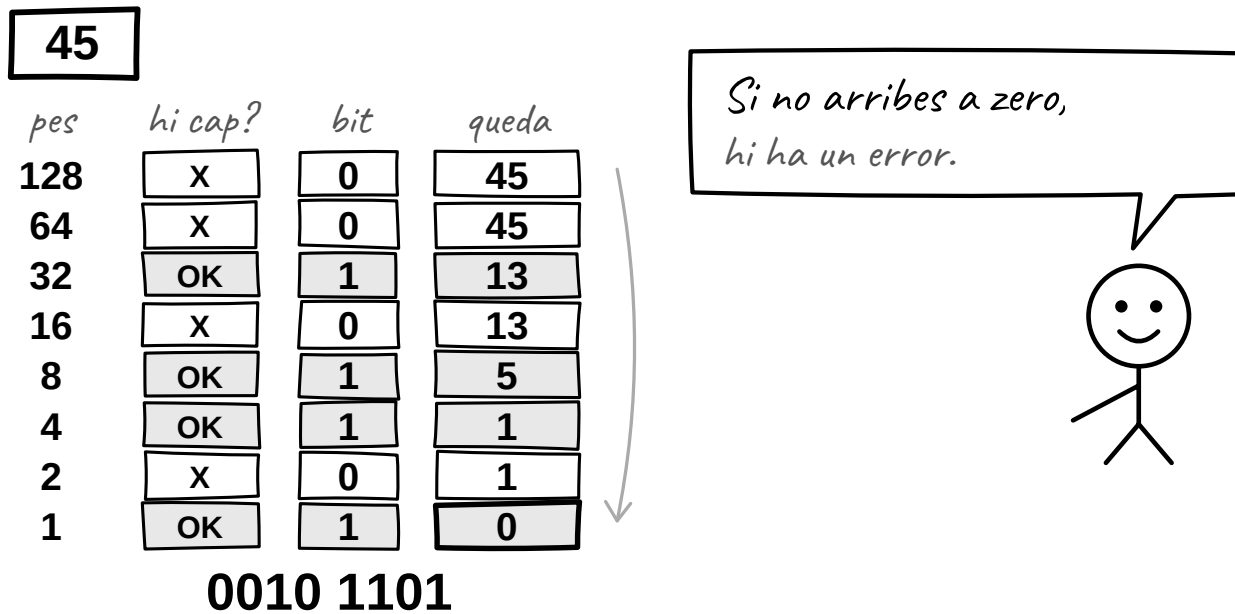


FIG 3.3 L'escala del «hi cap?»: el 45 es va buidant fins a arribar a 0.

Guiat 3.4

Escriure **83** en binari. Completa la taula.

Xuleta	Hi cap?	Hi poso	Em queda
128	no hi cap	0	83
64	sí que hi cap	1	$83 - 64 = 19$
32			
16			
8			
4			
2			
1			

Resultat:

Comprovació (suma els pesos on has posat 1):

pes	hi cap?	bit	queda
128			
64			
32			
16			
8			
4			
2			
1			

--	--	--	--	--	--	--	--

FIG 3.3m Completa l'escala: escriu el bit i el que et queda a cada pas.

Practica 3.5

Escriu en binari. Fes la taula cada vegada, encara que et sembli lent. Al cap de cinc et sortirà de memòria.

a) 7 =

b) 20 =

c) $100 =$ d) $176 =$ e) $255 =$ *(Comprovació del primer: 7 és 111.)***3.1. Una altra manera: dividir entre 2**

Hi ha un altre mètode. Alguns professors el demanen i a alguna gent li agrada més.

Divideixes entre 2 una vegada i una altra, apuntes la resta, i al final **llegeixes les restes de baix cap a dalt**.

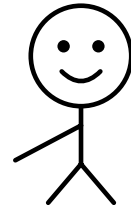
45 en binari:

Divisió	Quocient	Resta
$45 \div 2$	22	1
$22 \div 2$	11	0
$11 \div 2$	5	1
$5 \div 2$	2	1
$2 \div 2$	1	0
$1 \div 2$	0	1

De baix cap a dalt: **101101**. El mateix resultat que abans.

45	÷	2	=	22	1
22	÷	2	=	11	0
11	÷	2	=	5	1
5	÷	2	=	2	1
2	÷	2	=	1	0
1	÷	2	=	0	1

*Les restes es llegeixen
al revés.*



101101

FIG 3.4 Les divisions entre 2 i les restes, que es llegeixen de baix cap a dalt.

Atenció. L'errada típica és llegir les restes de dalt a baix. Es llegeixen **al revés**, començant per l'última. Fes servir el mètode que vulguis, però fes servir sempre el mateix.

4. Hexadecimal: la drecera

El binari funciona, però escriure'l cansa. Una adreça de memòria són 32 uns i zeros seguits. Ningú copia això sense equivocar-se.

Per això els informàtics fan servir l'**hexadecimal**. Té 16 símbols. Com que amb deu xifres no arriba, se segueix amb lletres:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Després del 9 ve la A, no el 10. La F és el 15, el més gran d'una sola xifra.

DEC	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
BIN	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

FIG 3.5 Els setze símbols de l'hexadecimal, del 0 a la F.

L'has vist sense saber-ho: els colors de les pàgines web (**#FF6600**), l'adreça MAC de la teva targeta de xarxa, els codis d'error de Windows.

4.1. De binari a hexadecimal és regalat

Aquí està la gràcia de l'assumpte:

Cada 4 bits són exactament un símbol hexadecimal.

No cal dividir ni restar. Només partir en grups de quatre **començant per la dreta** i traduir cada grup.

Exemple resultat 4

Passar **1011 0110** a hexadecimal.

Ja ve partit en dos grups de quatre:

- **1011** → $8 + 2 + 1 = 11$ → a la taula, l'11 és la **B**
- **0110** → $4 + 2 = 6$ → el 6 és **6**

Resultat: **B6**

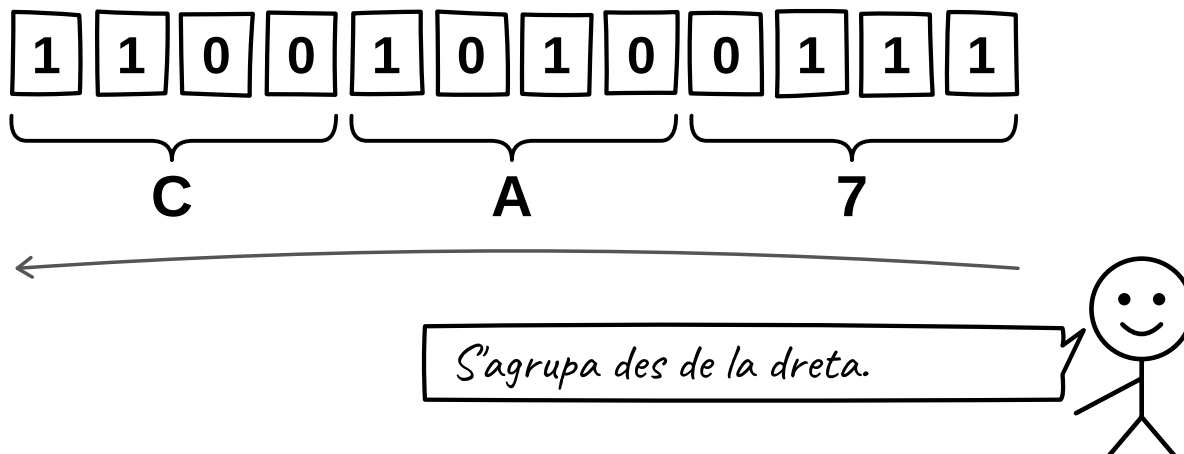


FIG 3.6 Cada grup de quatre bits es converteix en un sol símbol hexadecimal.

I al revés igual de fàcil: cada lletra o xifra es converteix en els seus quatre bits.

Atenció. S'agrupa **des de la dreta**. Si sobren bits a l'esquerra, omple amb zeros. **1 0110** s'agrupa com **0001 0110**. Mai com **1011 0000**.

Guiat 3.6

Passar **1100 1010 0111** a hexadecimal.

Grup	1100	1010	0111
Quant val			
Símbol hex			

Resultat:

Practica 3.7

a) **1111 0001** a hexadecimal =

b) **0010 1101** a hexadecimal =

c) **3F** a binari =

d) **A0** a binari i després a decimal =

(Comprovació del primer: **1111 0001** és **F1**.)

4.3. L'octal: el mateix però de tres en tres

L'**octal** té 8 símbols: del **0** al **7**. No hi ha ni 8 ni 9.

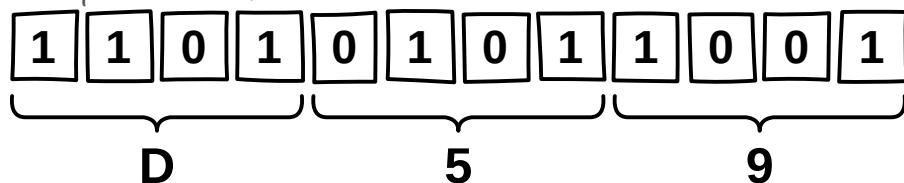
Funciona igual que l'hexadecimal, però amb un canvi:

Cada 3 bits són exactament un símbol octal.

Tres bits, perquè amb tres bits caben vuit combinacions: de **000** a **111**, que és del 0 al 7.

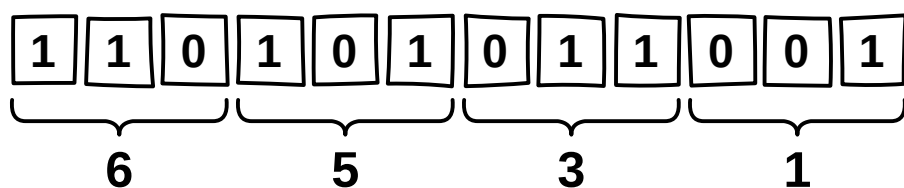
Octal	0	1	2	3	4	5	6	7
Binari	000	001	010	011	100	101	110	111

de quatre en quatre



HEX
D59

de tres en tres



OCT
6531

FIG 3.13 Hexadecimal de quatre en quatre, octal de tres en tres.

Exemple resultat 5

Passar **110 101 011 001** a octal.

Ja ve partit en grups de tres. Tradueixo cada grup:

- **110** → $4 + 2 = 6$
- **101** → $4 + 1 = 5$
- **011** → $2 + 1 = 3$
- **001** → **1**

Resultat: **6531**

I al revés: **542** en octal és **101 100 010** en binari.

Atenció. També s'agrupa **des de la dreta**. I si sobren bits a l'esquerra, s'omple amb zeros fins a tenir grups de tres, no de quatre.

Practica 3.8

a) Passa **111 010** a octal.

b) Passa **354** en decimal a octal. (Pista: passa'l primer a binari.)

c) Quin és el nombre més gran que es pot escriure amb dues xifres octals? Digues-lo en octal i en decimal.

d) Omple la taula.

Decimal	Binari	Octal	Hexadecimal
75			
	1 0110 1011		
		113	

(Comprovació de la primera fila: 75 en binari és **1001011**.)

Per què existeix l'octal? Perquè els ordinadors antics treballaven amb paraules de 12, 24 o 36 bits, tots múltiples de 3. Avui és més rar, però encara el trobaràs als **permisos de fitxers de Linux**: quan algú escriu **chmod 755**, aquell 755 és octal.

5. El byte i el que hi cap a dins

Vuit bits junts formen un **byte**. És la mesura de tota la vida en informàtica.

Quant és el màxim que hi cap en un byte? Posa un 1 als vuit llocs i suma la xuleta sencera:

$$128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$$

Així que **un byte va del 0 al 255**. En total, 256 valors diferents, comptant el zero.

Aquest 0-255 et perseguirà: és el rang de les màscares de xarxa, el de cada color d'un píxel, i el d'un munt de coses més.

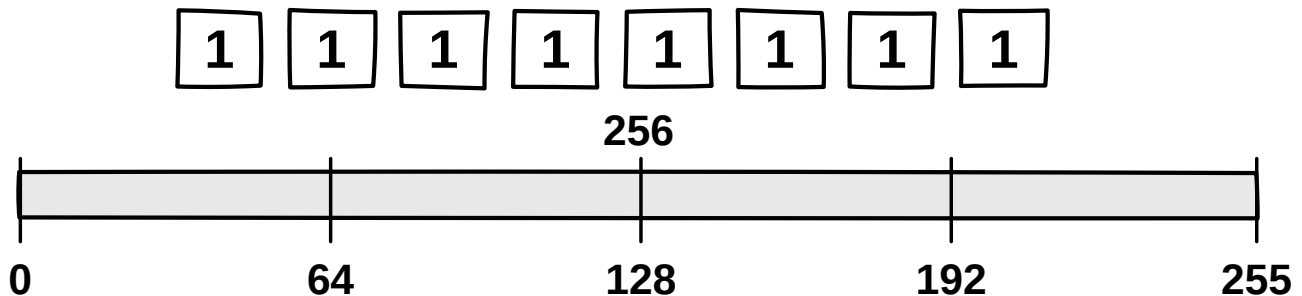


FIG 3.8 Els 256 valors d'un byte, del 0 al 255.

Cada bit que hi afegeixes, dobla el que hi cap:

Bits	Quants valors hi caben
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8 (un byte)	256

És la mateixa fila de doblar de sempre. Amb 10 bits hi caben 1.024 valors. Aquest nombre et sonarà d'aquí a un moment.

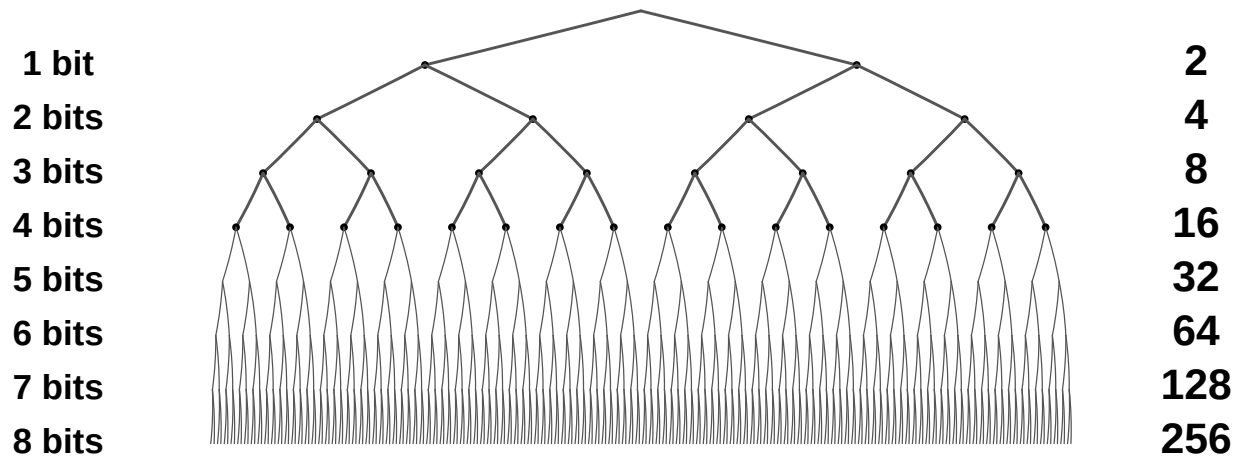


FIG 3.7 Cada bit nou parteix en dos el que hi cabia abans.

Practica 3.9

a) Quants valors hi caben en 6 bits?

b) Quin és el nombre més gran que cap en 4 bits? Escriu-lo en binari i en decimal.

c) Vull numerar els 780 alumnes del centre. En tinc prou amb 8 bits? I amb 10?

6. Sumar en binari

Sumar en binari és com sumar de tota la vida, però amb un sol canvi: **quan arribes a 2, te'n dus una**. En decimal te'n dus una en arribar a 10; aquí, en arribar a 2.

Només hi ha quatre casos. Aprèn-te'ls i ja saps sumar:

Suma	Resultat	Em duc
0 + 0	0	—
0 + 1	1	—
1 + 0	1	—
1 + 1	0	1

L'únic estrany és l'últim. 1 + 1 són 2, però el 2 no existeix en binari. S'escriu **10**. Així que poses **0** i et duus **1**, igual que quan en decimal sumes 7 + 5 i poses 2 duent-te'n 1.

Exemple resultat 5

Sumar **1011 + 0110**.

Es comença per la dreta, com sempre:

		1	1		← em duc
	1	0	1	1	
+	0	1	1	0	
=	1	0	0	1	

Anem columna per columna, de dreta a esquerra:

1. $1 + 0 = 1$
2. $1 + 1 = 0$ i **em duc 1**
3. $0 + 1 + 1$ (la que em duia) = 0 i **em duc 1**
4. $1 + 0 + 1$ (la que em duia) = 0 i **em duc 1**

5. Queda la que em duia: 1

Resultat: 1 0001

$$\begin{array}{r}
 \\
 \\
 + \\
 \hline
 1 \\
 11 + 6 = 17
 \end{array}$$

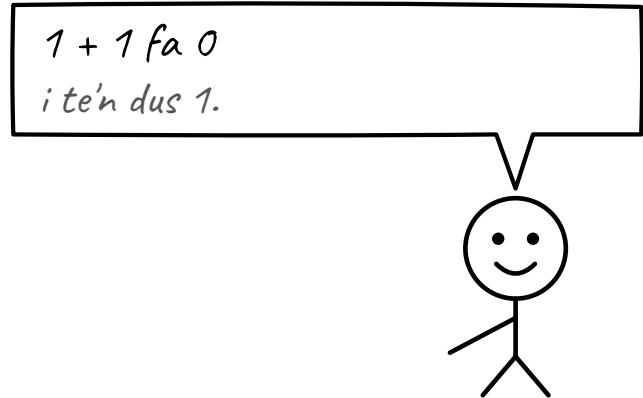


FIG 3.9 La suma columna a columna i la que es duu cap a l'esquerra.

Comprova: 1011 és 11, 0110 és 6, i $11 + 6 = 17$. I 1 0001 és $16 + 1 = 17$. Correcte.

Comprova sempre així. Passa els dos nombres a decimal, suma'ls de cap, i mira si coincideix amb el teu resultat en binari. És la manera més ràpida de caçar l'error.

Practica 3.10

a) $101 + 011 =$

b) $1100 + 0011 =$

c) $1111 + 0001 =$

d) **1010 1010 + 0101 0101 =**

(Comprovació del primer: **101 + 011 = 1000**, o sigui $5 + 3 = 8$.)

7. Les lletres també són nombres

Fins ara hem guardat nombres. I les lletres?

L'ordinador no sap guardar una «A». Només guarda nombres. Així que es van posar d'acord en una taula: **a cada caràcter li toca un nombre**. Això és un codi de caràcters.

7.1. ASCII

El primer va ser l'**ASCII**. Fa servir 7 bits, així que hi caben 128 caràcters. Suficient per a l'anglès: lletres, xifres, signes de puntuació i poca cosa més.

Alguns que convé conèixer:

Caràcter	Nombre
espai	32
O	48
A	65
Z	90
a	97
z	122

Fixa't en dues coses útils:

- Les xifres van seguides des del 48. El **7** és $48 + 7 = 55$.
- Les majúscules comencen al 65 i les minúscules al 97. **Hi ha 32 de diferència** entre una lletra i la seva minúscula. La **B** és 66 i la **b** és 98.

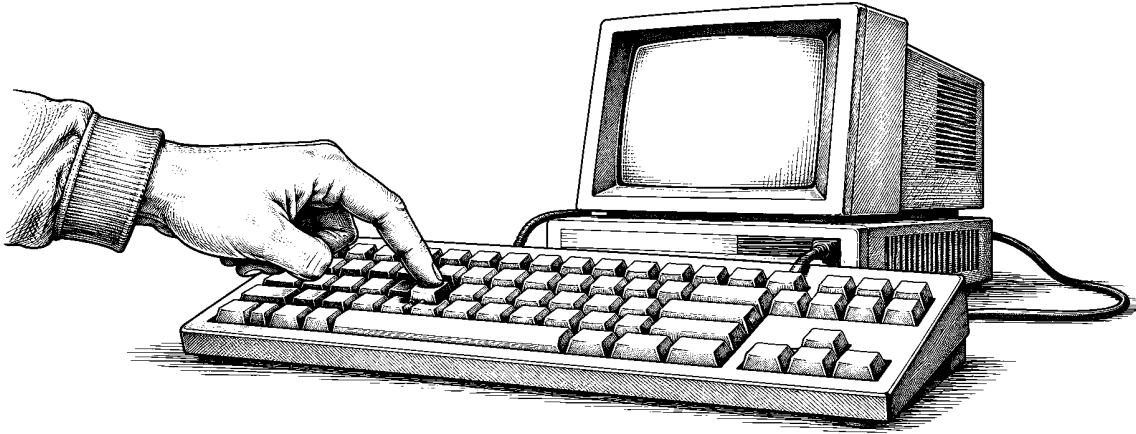


FIG 3.10 De la tecla al nombre i del nombre als bits.

Compte. En ASCII el caràcter **7** **no val 7**. Val 55. El nombre set i la lletra set són coses diferents per a l'ordinador. D'aquí venen moltes errades de programació.

7.2. ISO-8859-1

L'ASCII no té ni la **ñ**, ni la **ç**, ni les vocals accentuades. Amb 7 bits no hi caben.

S'hi va afegir un bit més, fins a 8, i amb això 256 caràcters. La segona meitat de la taula es va omplir amb les lletres que faltaven per a les llengües d'Europa occidental. Aquest és l'**ISO-8859-1**, també anomenat Latin-1.

Problema: cada zona del món va fer la seva pròpia taula. El mateix nombre significava una lletra a Europa i una altra de diferent a Rússia o a Grècia. Obries un fitxer d'un altre país i sortia tot malament.

7.3. Unicode i UTF-8

Unicode és la solució: una sola taula per a **tots** els caràcters de tots els idiomes del món, més símbols i emojis. Avui n'hi ha més de 140.000.

UTF-8 és la forma més usada de guardar Unicode. És llest: fa servir **l'espai que cal i no més**.

Quin caràcter	Quants bytes fa servir
Els d'ASCII de tota la vida (A, 7, espai)	1
Lletres amb accent, ñ, ç, grec, ciríl·lic	2
Xinès, japonès, coreà	3
Emojis	4

La gràcia de l'UTF-8 és que **un text en anglès ocupa el mateix que en ASCII**. Per això va guanyar i avui és el que fan servir gairebé totes les pàgines web.

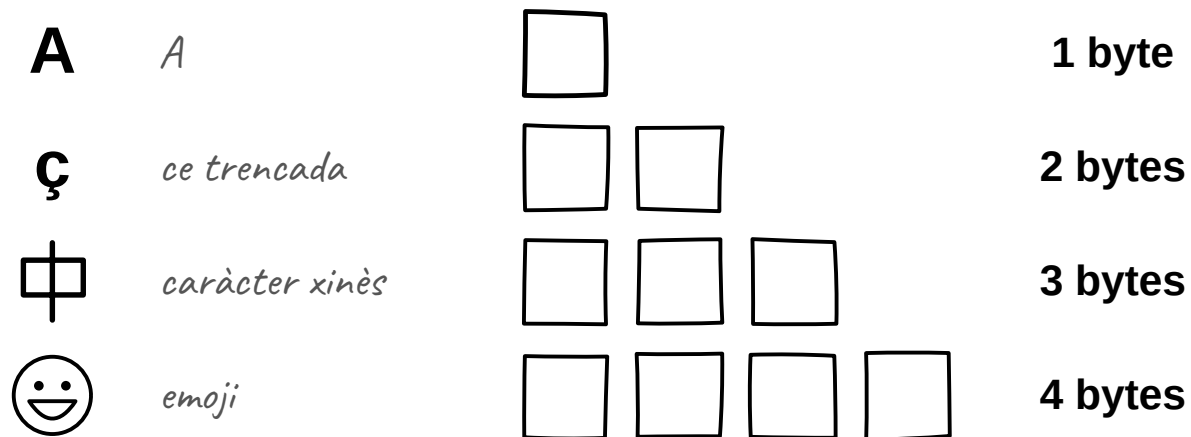


FIG 3.11 Quants bytes gasta l'UTF-8 segons quin caràcter sigui.

Quan vegis «Ã±» on hi hauria d'haver «ñ», ja saps què ha passat: el fitxer està en UTF-8 i el programa el llegeix com si fos ISO-8859-1. És l'error de codificació més comú que et trobaràs.

Practica 3.11

a) Si la **A** és 65, quin nombre li toca a la **D**?

b) Si la **a** és 97, quina lletra és el nombre 101?

c) La **M** és 77. Quin nombre té la **m**?

d) Escribe la paraula **Sol** en nombres ASCII. (Pista: la **S** és 83.)

e) Quants bytes ocupa en UTF-8 la paraula **any**? I la paraula **açò**? Compta lletra per lletra.

(Comprovació del primer: la **D** és 68.)

8. Per què el teu disc d'1 TB en té menys

Compres un disc d'1 TB. L'endolles. Windows diu 931 GB. No t'han enganyat.

El que passa és que hi ha **dues maneres de comptar**:

- El **fabricant** compta de mil en mil: 1 kB = 1.000 bytes.
- El **sistema operatiu** compta de 1.024 en 1.024: 1 KiB = 1.024 bytes.

I per què 1.024? Perquè és el que cap en 10 bits, el nombre d'abans. A l'ordinador li surt rodó encara que a nosaltres no ens ho sembli.

Del fabricant	Val	Del sistema	Val
1 kB	1.000 bytes	1 KiB	1.024 bytes
1 MB	1.000 kB	1 MiB	1.024 KiB
1 GB	1.000 MB	1 GiB	1.024 MiB
1 TB	1.000 GB	1 TiB	1.024 GiB

La diferència és petita al principi, però es va acumulant. En 1 TB ja són gairebé 70 GB de diferència.

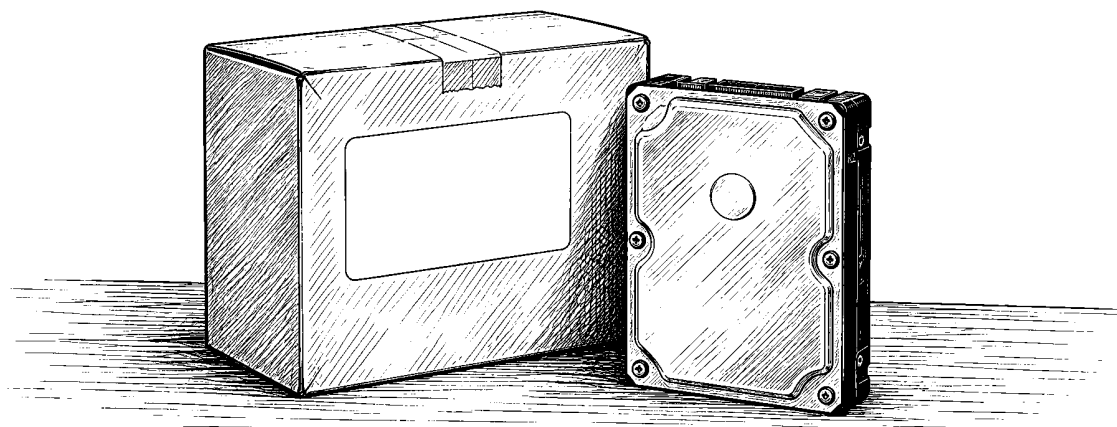


FIG 3.12 El mateix disc mesurat amb dues unitats diferents.

Que quedi clar. El disc té l'espai que diu la caixa. El que canvia és la unitat amb què el compta cada un.

Practica 3.12

a) Quants bytes hi ha en 4 KiB?

b) Quants bytes hi ha en 1 MiB? (Pista: 1.024×1.024)

c) Un pendrive posa 32 GB. Quants GiB veuràs a l'ordinador, més o menys?

9. Quant trigarà a baixar?

Aquí hi ha la trampa que enganya tothom, i cau a l'examen.

La velocitat de connexió es mesura en bits. La mida dels fitxers es mesura en bytes.

Mira't les unitats amb lupa:

S'escriu	Es llegeix	És de
Mbps	megabits per segon	bits
MB/s	megabytes per segon	bytes

La **b** minúscula és bit. La **B** majúscula és byte. I ja saps que **1 byte són 8 bits**.

Per això una connexió de 100 Mbps no baixa 100 MB cada segon, sinó **vuit vegades menys**: $100 \div 8 = 12,5 \text{ MB/s}$.

Per què ho fan així? Les operadores anuncien en megabits perquè el número surt vuit vegades més gran. «300 megues» impressiona més que «37,5 MB/s», que és el mateix.

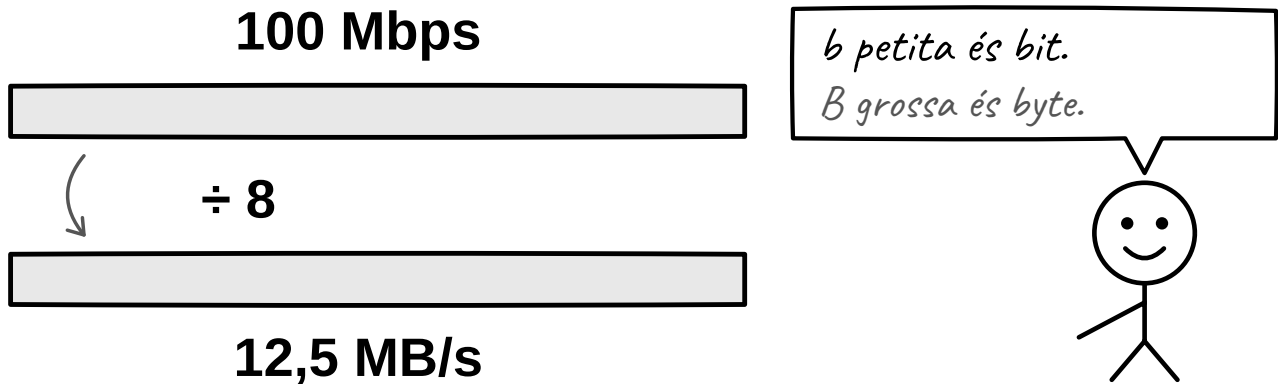


FIG 3.14 La mateixa velocitat dita de dues maneres.

9.1. El mètode, en tres passes

1. **Passa la velocitat de bits a bytes:** divideix entre 8.
2. **Posa la mida del fitxer en la mateixa unitat.**
3. **Divideix:** mida entre velocitat. El resultat són segons.

Exemple resolt 8

Vols baixar un fitxer de **100 GB** amb una connexió de **200 Mbps**. Quant triges?

Passa 1. La velocitat en bytes: $200 \div 8 = 25 \text{ MB/s}$

Passa 2. El fitxer en MB: $100 \text{ GB} = 100.000 \text{ MB}$

Passa 3. Divideix: $100.000 \div 25 = 4.000 \text{ segons}$

I ara en unitats humanes: $4.000 \div 60 = 66,7 \text{ minuts}$, una mica més d'una hora.

Comprova-ho sempre. Si el resultat et surt vuit vegades massa gran o massa petit, t'has oblidat de dividir entre 8. És l'error número u.

Practica 3.13

a) Tens 50 Mbps. Quants MB/s són?

b) Una pel·lícula en alta definició ocupa 3 GB. Amb una connexió de 100 Mbps, quant triges a baixar-la?

c) Una operadora anuncia «1 giga de fibra». Quants MB/s són de veritat?

d) Has trigat 2 minuts a baixar un fitxer de 900 MB. A quina velocitat anaves, en MB/s i en Mbps?

(Comprovació de la primera: 50 Mbps són 6,25 MB/s.)

El que has de saber sí o sí

- La xuleta: **128, 64, 32, 16, 8, 4, 2, 1**. Cada un el doble de l'anterior.
- **Llegir binari**: suma els de la xuleta on hi hagi un 1.
- **Escriure binari**: pregunta "hi cap?" d'esquerra a dreta, resta i apunta 1 o 0.
- Comprova sempre tornant enrere. Si no quadra, hi ha una errada.
- **4 bits = 1 símbol hexadecimal**. S'agrupa des de la dreta.
- Després del 9 en hexadecimal ve la A. La F val 15.
- **Un byte són 8 bits** i va del 0 al 255.
- **Sumar en binari**: 1 + 1 fa 0 i et duus 1.
- Les lletres es guarden com a nombres. La **A** és 65, la **a** és 97.
- **UTF-8** val per a tots els idiomes i gasta només els bytes que calen.
- **Cada 3 bits són un símbol octal**, i cada 4 bits un d'hexadecimal.
- **kB són 1.000 bytes**. **KiB són 1.024**. Per això el disc sembla més petit.
- **Mbps és de bits** i **MB/s és de bytes**. Per passar de l'un a l'altre, divideix entre 8.

Paraules noves

Paraula	Què és	Les meves notes
Bit	Un interruptor. Val 0 o 1.	
Byte	Vuit bits junts. Del 0 al 255.	
Binari	Sistema amb dos símbols: 0 i 1.	
Hexadecimal	Sistema amb 16 símbols: 0 a 9 i A a F.	
kB	1.000 bytes. El que fa servir el fabricant.	
KiB	1.024 bytes. El que fa servir el sistema.	
ASCII	Taula de 128 caràcters. Val per a l'anglès.	
UTF-8	Forma de guardar Unicode. Serveix per a tots els idiomes.	

Repàs del tema

A. Omple la taula

Falta una casella de cada fila.

Decimal	Binari	Hexadecimal
12	1100	
	0001 1111	1F
42		2A
64	0100 0000	
255		FF
	1100 1000	C8
7	0111	

B. Vertader o fals?

Frase	V o F	Per què
1010 val 10		
En un byte hi caben 256 valors		
FF val 255		
1 kB i 1 KiB són el mateix		
Un binari que acaba en 1 és senar		

C. Problemes

Problema 1. Cada color d'un píxel fa servir un byte: un per al vermell, un per al verd i un per al blau. Quants valors diferents pot prendre cada color?

Problema 2. El color **#FF0000** és vermell del tot. Escriu els tres bytes en binari i digues quin valor té cada color.

Problema 3. Un error de Windows mostra l'adreça **0x00A4**. Passa-la a binari.

Problema 4. Tens un pendrive de 8 GB. Vols copiar-hi 2.000 fotos de 4 MB cada una. Hi caben? Fes els comptes.

Problema 5. El teu mòbil té 128 GB i ja en té 96 ocupats. Vas de viatge i fas 40 fotos cada dia durant 12 dies. Cada foto ocupa 5 MB i cada vespre graves un vídeo de 800 MB. T'hi cabrà tot?

Problema 6. Vols baixar una sèrie sencera: 10 capítols de 1,5 GB cada un. Tens fibra de 600 Mbps. Quant triges? Dona el resultat en minuts.

Problema 7. Has de pujar un vídeo de 2 GB a internet. La teva connexió té 600 Mbps de baixada però només 50 Mbps de pujada. Quant trigaràs? Per què no serveix el número gran?
