



PRÀCTICA

1. **(2.5 p) Script Bash.** Heu de fer un script, de nom `binaris.sh`, que mostrarà la conversió a binari d'una seqüència de nombres decimals.

El guió haurà de complir els següents requeriments:

- (a) Nombre i tipus de paràmetres que rep l'script:
 - i. L'script rebrà com a mínim un paràmetre. Es considerarà que el paràmetre introduït és correcte: **un enter entre 1 i 100**.
 - ii. **Si rep menys d'1 paràmetre**, es mostrarà un missatge indicant l'ús correcte de la crida al guió, i se sortirà.
 - iii. **Si rep més d'1 paràmetre**, es mostrarà un avís indicant que només s'usarà el primer paràmetre, i es continuarà amb l'execució del guió.
- (b) El programa mostrarà en binari els nombres de 0 fins al nombre proporcionat com a paràmetre, i a més a més, el nombre de bits necessaris per a la representació en binari.
- (c) Per a calcular la representació en binari d'un nombre decimal i el nombre de bits necessaris, caldrà implementar la funció **dec2bin**:
 - i. Rebrà, per paràmetre d'entrada de la funció, el nombre decimal del qual cal obtenir la representació en binari.
 - ii. Per una part obtindrà la representació en binari.
 - iii. I retornarà, **com a valor de retorn de la funció**, el nombre de bits necessaris per a representar-lo.
 - iv. Per a obtenir la representació en binari cal que useu les operacions de divisió i residu.
- (d) Tot el que no estigui indicat, podeu prendre les vostres decisions.

La crida al l'script és `# ./binaris.sh <nombre1> [<nombre2> ... <nombreN>]`.

Exemples d'execució (en **negreta** es mostra la informació que subministra l'usuari):

```
# ./binaris.sh
```

```
Us:  ./binaris.sh <nombre1>
```

```
# ./binaris.sh 4
```

```
0(d = 0(b - 1 bit/s necessari/s
```

```
1(d = 1(b - 1 bit/s necessari/s
```

```
2(d = 10(b - 2 bit/s necessari/s
```

```
3(d = 11(b - 2 bit/s necessari/s
```

```
4(d = 100(b - 3 bit/s necessari/s
```

```
# ./binaris.sh 3 8
```

```
AVIS: S'han introduït més de 1 parametres; només s'usarà el primer
```

```
0(d = 0(b - 1 bit/s necessari/s
```

```
1(d = 1(b - 1 bit/s necessari/s
```

```
2(d = 10(b - 2 bit/s necessari/s
```

```
3(d = 11(b - 2 bit/s necessari/s
```

2. **(3.25 p) Gestió de Memòria.** Disposem d'un sistema de gestió de la memòria del tipus segmentació paginada. La mida d'una pàgina (cel·la) és de 1.024 paraules. Un segment conté com a molt 3 pàgines. Un procés consta de com a molt 8 segments. La mida d'una paraula és igual a 1 Byte. Totes les taules (de segments i de pàgines) s'implementen en MP. En la següent figura es pot veure el contingut de MP:

0	P1c	10		20		30		40		50		60		70		80		90		100		110		120	SO
1	P1d	11		21	P12c	31		41		51		61		71	P2c	81		91		101		111		121	
2	P2c	12		22		32		42		52		62		72		82	P2d	92		102		112		122	SO
3	P2c	13		23		33		43		53		63		73		83		93	P2p	103		113		123	SO
4	P1d	14		24		34		44		54		64		74		84		94		104		114		124	SO
5		15		25		35		45		55		65		75		85	P2d	95		105		115		125	
6		16	P1c	26		36	P12d	46		56		66		76		86		96		106		116		126	SO
7	P1d	17		27		37		47		57		67		77		87		97		107		117		127	SO
8		18		28		38	P12d	48	P1p	58		68		78		88		98		108		118			
9		19		29		39		49		59		69		79		89		99		109		119			

ACLARACIONS

SO: Sistema Operatiu
P1c: codi del procés P1
P1d: dades del procés P1
P1p: pila del procés P1

nombre
de cel·la

P12c: codi del procés P1 i P2
P12d: dades del procés P1 i P2

- (a) **(2.25 p)** Doneu un exemple de la informació que ha de guardar el sistema operatiu per gestionar la memòria del procés P_1 i P_2 . Supposeu que totes les pàgines del procés P_1 estan plenes. Tota aquesta informació s'ha de posar a la cel·la 9 i ha d'ocupar el mínim espai possible. Digueu què és cada dada que emmagatzemeu, així com la seva adreça física de MP i la seva mida en Bytes.
- (b) **(1 p)** Doneu una possible adreça lògica de l'adreça física 4.097 del procés P_1 . Raoneu la resposta.
3. **(3.25 p) Memòria Virtual.** Disposem d'un sistema de gestió de la memòria del tipus "paginació" i Mem. Virtual del tipus "paginació sota demanda". Assignació local i igualitària de 5 cel·les per procés. La mida d'una pàgina i d'una cel·la és de 16 Bytes. Mida d'una instrucció = Mida d'una paraula = Mida d'un enter = 1 Byte. Donat el programa en Pseudo-assembler següent:

@log	codi
017h	Reg0 \leftarrow 0
018h	Comparar Reg0, 33
019h	Branch_equal to @024h
01Ah	Reg1 \leftarrow Reg0 mod 16
01Bh	Reg2 \leftarrow Reg0 + 1
01Ch	Reg3 \leftarrow Reg0 + 2
01Dh	Load Reg4 \leftarrow Mem(@500h + Reg0)
01Eh	Load Reg5 \leftarrow Mem(@600h + Reg1)
01Fh	Load Reg6 \leftarrow Mem(@700h + Reg2)
020h	Reg7 \leftarrow Reg4 + Reg5 + Reg6
021h	Store Mem(@500h + Reg3) \leftarrow Reg7
022h	Reg0 \leftarrow Reg0 + 1
023h	Branch to @018h

- (a) **(1.75 p)** Quantes fallades de pàgina es produirà en l'execució del programa suposant que l'algorisme de reemplaç és LRU? Feu l'esquema que ens mostri com es van omplint les cel·les a mesura que es van produint les fallades de pàgina.
- (b) **(1.5 p)** Quantes fallades de pàgina es produirà en l'execució del programa suposant ara que l'algorisme de reemplaç és FIFO?
4. **(1 p) Interbloqueig-Evitació.** Tenim 5 processos $\{p_0, p_1, p_2, p_3, p_4\}$ i 3 tipus de recursos $\{r_0 = 10, r_1 = 8, r_2 = 4\}$. Quins processos estan en Interbloqueig donat l'Estat General següent?

	Assignat			Sol·licitud		
	r_0	r_1	r_2	r_0	r_1	r_2
p_0	3	2	1	1	1	3
p_1	2	1	1	7	7	3
p_2	0	0	0	5	0	1
p_3	1	3	0	4	1	1
p_4	3	0	1	0	1	0

Disponible		
r_0	r_1	r_2
1	2	1