



PRÀCTIQUES

1. **Shell Script (2.5 p.).** Heu de fer un script, de nom “numbers.sh”, que realitzarà la tasca matemàtica que triï l’usuari a través d’un argument. Les opcions que pot triar l’usuari seran: la taula de multiplicar d’un nombre (-t), i una progressió aritmètica (-p).

La crida a l’script serà:

```
# ./numbers.sh {-t|-p}
```

Depenent de l’opció triada, es demanarà més informació:

- -t) es demanarà el nombre sobre el que s’haurà de mostrar per pantalla la taula de multiplicar.
- -p) es demanaran dos nombres, el 1^{er} element de la progressió i l’increment de la progressió. S’hauran de mostrar 5 elements de la progressió.

* Si la taula de multiplicar i la progressió es realitzen en 2 funcions separades, es sumarà **0.5 punts** addicionals.

Exemples d’execució:

```
# ./numbers.sh
```

Ús : ./numbers.sh {-t|-p}

```
# ./numbers.sh -n
```

Ús : ./numbers.sh {-t|-p}

```
# ./numbers.sh -t
```

Introduceix un nombre: 3

Taula del 3:

3 * 1 = 3

3 * 2 = 6

3 * 3 = 9

3 * 4 = 12

3 * 5 = 15

3 * 6 = 18

3 * 7 = 21

3 * 8 = 24

3 * 9 = 27

3 * 10 = 30

```
# ./numbers.sh -p
```

Introduceix el primer terme de la progressió: 1

Introduceix l’increment: 3

Progressió aritmètica : 1, 4, 7, 10, 13

TEORIA

2. **Gestió de Memòria (3.5 p).** Disposem d'un sistema de gestió de Memòria del tipus segmentació paginada. La mida d'una paraula és igual a 1 Byte. Mida pàgina = Mida cel·la = 1.024 paraules. Un segment conté com a molt 4 pàgines. Un procés consta de com a molt 8 segments. En la següent figura es pot veure el contingut de MP:

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

ACLARIMENTS:

P1c: codi del procés P1; P1d: dades del procés P1; P1p: pila del procés P1; P12c: codi dels processos P1 i P2.

- (a) **(1 p)** Suposant que totes les taules (de segments i de pàgines) s'implementen en Memòria Principal, doneu l'esquema de traducció d'adreces lògiques a físiques, amb totes les mides de les estructures (o registres que hi apareixin). Poseu-hi també el(s) multiplicador(s) que cregueu convenient(s).
- (b) **(1.5 p)** Doneu un exemple de la informació que ha de guardar el sistema operatiu per gestionar la memòria del procés P1. Supposeu per fer-ho que totes les cel·les de P1 estan plenes. Supposeu també que tota aquesta informació s'ha de posar a la cel·la 6.
- (c) Un cop fet l'apartat (b), respongueu:
- (0.5 p)** Doneu el rang d'adreces lògiques i físiques del procés P1.
 - (0.5 p)** També pel procés P1, quina és l'adreça lògica corresponent a l'adreça física 10.250_(d)?
3. **Memòria Virtual (3.5 p).** Disposem d'un sistema de gestió de la memòria del tipus segmentació paginada i Mem. Virtual del tipus paginació sota demanda. Assignació local de cel·les (6 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 sencer = 1 Byte. Donat el programa següent:

@log	codi
002Bh	Reg0 ← 1
002Ch	Comparar Reg0, 40
002Dh	Branch_equal to @0035h
002Eh	Load Reg1 ← Mem(@6000h + Reg0 - 1)
002Fh	Load Reg2 ← Mem(@7000h + Reg0 + 1)
0030h	Load Reg3 ← Mem(@8000h + (Reg0 mod 16))
0031h	Reg4 ← Reg1 + Reg2 + Reg3
0032h	Store Mem(@5000h+Reg0) ← Reg4
0033h	Reg0 ← Reg0 + 1
0034h	Branch to @002Ch

- (a) **(2.5 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 p)** Feu el mateix que a l'apartat (a) però utilitzant l'algorisme de reemplaç FIFO i canviant la línia “002Ch” per “Comparar Reg0, **25**”.