



Presentació

Aquesta PAC planteja un seguit d'activitats amb l'objectiu que l'estudiant es familiaritzi amb la temàtica dels primers mòduls de l'assignatura.

Cada pregunta indica una possible temporització per poder acabar la PAC abans de la data límit i el pes de la pregunta a l'avaluació final de la PAC.

Competències

Transversals:

- Capacitat per a la comunicació escrita en l'àmbit acadèmic i professional

Específiques:

- Capacitat per a analitzar un problema en el nivell d'abstracció adequat a cada situació i aplicar les habilitats i coneixements adquirits per a abordar-lo i resoldre'l

Enunciat

1. Mòdul 2 [Del 30 d'octubre a 2 de novembre] (Pes 30% = 5% + 5% + 5% + 5% + 10%)

Responen **justificadament** les següents preguntes relacionades amb el mòdul 2 de l'assignatura:

- 1.1 En què es diferencien les interrupcions hardware de les interrupcions software (crides al sistema)? Poseu un exemple de cada tipus.

La diferència radica en el motiu que les causa. Les interrupcions hardware són causades pels dispositius físics per notificar algun canvi d'estat (el teclat avisa per indicar que una tecla ha estat premuda, el disc avisa per indicar que la darrera operació de lectura ha finalitzat,...). Les interrupcions software són provocades explícitament pel codi en execució mitjançant una instrucció del llenguatge màquina (per exemple, quan el programa invoca la crida al sistema `write`).

- 1.2 Indiqueu un esdeveniment que pugui causar la transició des de l'estat Run (Execució) a l'estat Ready (Preparat, Llest). I des de Ready a Run? I des de Run a Wait (Bloqueig, Espera, Blocked)?

- Run a Ready: la finalització del quantum provoca que el planificador de la CPU passi a Ready el procés que actualment està a Run
- Ready a Run: quan el planificador de la CPU assigna la CPU a un procés que està a Ready



- Run a Wait: quan el procés invoca una crida al sistema bloquejant (per exemple, la crida `read` sobre el teclat).

1.3 En un mateix moment, pot haver varis processos a l'estat Wait? I a l'estat Ready? I a l'estat Run?

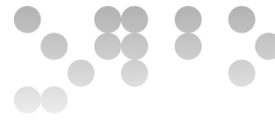
Depèn del nombre de *cores* del processador i de si té capacitats multithreading. A un sistema amb un processador amb un únic core i sense multithreading, a un moment donat únicament pot haver un procés a l'estat de Run. Si el processador disposa de varis cores o de multithreading, diversos processos poden estar a Run simultàniament.

En canvi, l'únic límit a la quantitat de processos que poden estar en Wait o en Ready és la quantitat de processos vius que pugui gestionar el SO.

1.4 Indiqueu semblances i diferències entre l'estat de Wait i l'estat Ready.

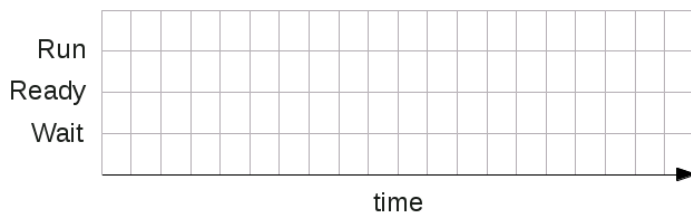
La semblança principal és que tant un procés a Ready com un procés a Wait no estan fent ús del processador. Una altra semblança seria que mentre el nombre de processos a Run està limitat pel nombre de processadors al sistema, el nombre de processos a Wait o a Ready només està limitat pel nombre màxim de processos que poden existir al sistema. La diferència principal és que un procés a Ready està en condicions de continuar la seva execució (només fa falta que el SO li assigni un processador), però un procés a Wait està esperant algun esdeveniment (per exemple, finalització d'una operació d'E/S). Una altra diferència és que l'entrada a l'estat Wait acostuma a ser voluntària (per exemple, quan el procés invoca una crida al sistema bloquejant) però l'entrada a l'estat Ready acostuma a ser involuntària (per exemple, quan el planificador realitza un canvi de context per l'expiració del quantum).

1.5 Feu un cronograma on es mostri com podria evolucionar l'estat d'un procés que executi els següents bucles a un SO multiprocés, tot indicant els motius que provoquen els canvis d'estat.



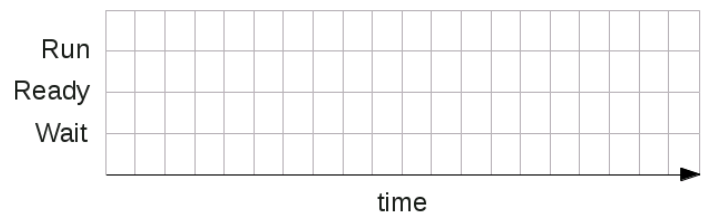
i)

```
while(1);
```



ii)

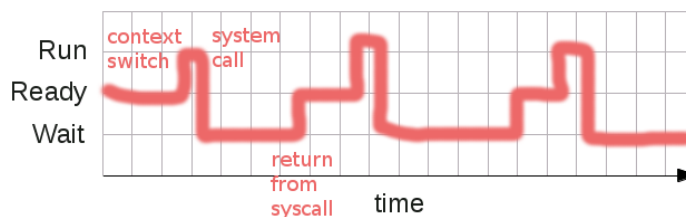
```
while(1) {
    /* system call that reads from standard input */
    read(0, &c, 1);
}
```



i)



ii)



2. Mòdul 3 [Del 3 al 10 de novembre] (Pes 60% = 5% + 5% + 10% + 5% + 35%)

Responen **justificadament** les següents preguntes relacionades amb el mòdul 3 de l'assignatura:

- 2.1 L'operador & del llenguatge C retorna l'adreça de memòria on està emmagatzemada una variable. Sobre una màquina amb un SO tipus Unix, aquesta adreça és lògica o física?

Des del punt de vista del programa, totes les adreces són lògiques. El sistema de gestió de memòria ens amaga les adreces físiques on s'emmagatzemen les variables.

- 2.2 És possible que l'espai lògic d'un procés sigui més gran que la quan-



titat de memòria física instal·lada a la màquina?

Sí. Gràcies a mecanismes com la paginació sota demanda, l'espai lògic d'un procés pot estar emmagatzemat tant a memòria física com a l'àrea de swap. Per tant, la mida de l'espai lògic d'un procés pot arribar a ser més gran que la mida de la memòria física instal·lada a la màquina.

2.3 Donat el següent codi, indiqueu a quina zona de l'espai lògic (codi, dades estàtiques, dades dinàmiques/heap, pila) s'emmagatzemen els objectes **a**, **b**, **c**, **d**, **e**, **e[3]**.

```
int a;

void c(int d)
{
}

main()
{
    int b;
    int *e;

    c(b);
    e = (int*) malloc(10*sizeof(int));
}
```

- **a**: és una variable global. S'emmagatzema a la regió de dades estàtiques.
- **b**: és una variable local. S'emmagatzema a la regió de pila.
- **c**: és una rutina. S'emmagatzema a la regió de codi.
- **d**: és un paràmetre d'una rutina. S'emmagatzema a la regió de pila.
- **e**: és una variable local. S'emmagatzema a la regió de pila.
- **e[3]**: és un objecte creat dinàmicament. S'emmagatzema a la regió de dades dinàmiques (heap).

2.4 A un sistema de gestió de memòria basat en paginació sota demanda, què vol dir que una pàgina lògica sigui invàlida?

Indica que la pàgina en qüestió no forma part de l'espai lògic del procés. Intentar accedir a aquesta pàgina provocarà una excepció d'adreça invàlida i, molt probablement, el SO avortarà el procés.

2.5 Sigui un sistema de gestió de memòria basat en paginació sota demanda on les pàgines tenen una mida de 4KBytes, les adreces lògiques són de 16 bits i l'espai físic és de 32 Kbytes. Sobre aquest sistema es creen dos processos.

- Procés 1: el seu fitxer executable determina que el codi ocuparà



dues pàgines, que no té dades inicialitzades, té dues pàgines de dades sense inicialitzar i una pàgina de pila.

- Procés 2: el seu fitxer executable determina que el codi ocuparà una pàgina, que les dades no inicialitzades ocuparan una pàgina, que no existeixen dades inicialitzades i que la pila ocuparà dues pàgines.

Es demana:

- 2.5.1 Estimeu la mida del fitxer executable corresponent al procés 1.
- 2.5.2 Suposant que les pàgines es carreguen a memòria física tal i com indica el diagrama de la pàgina 9 d'aquest enunciat, indiqueu quin serà el contingut de les taules de pàgines de tots dos processos (considereu com a invàlides les pàgines marcades amb el símbol -). Indiqueu per quin motiu totes dues taules de pàgines tenen 16 entrades.
- 2.5.3 Suposant que el procés en execució és el procés 1, indiqueu quines seran les adreces físiques corresponents a les següents adreces lògiques: 0x3124 i 0xFABC. Variaria la resposta si el procés en execució fos el procés 2? En cas afirmatiu, indiqueu el motiu i com canviaria.

- 2.5.1 Al fitxer executable cal guardar el contingut de la zona de codi i de les dades inicialitzades. Per tant, com no tenim dades inicialitzades, només caldria emmagatzemar les dues pàgines de codi (la fragmentació interna a aquestes pàgines pot causar que la mida real sigui inferior a 8 Kbytes). A més, cal sumar la mida de les capçaleres del fitxer executable.
- 2.5.2 Les taules de pàgines són de 16 entrades perquè és el resultat de dividir la mida de l'espai lògic (2^{16} bytes) entre la mida de la pàgina (4 Kbytes, és a dir, 2^{12} bytes). El resultat és 2^4 , és a dir, 16 pàgines. El contingut de les taules de pàgines és el que es mostra a la figura (el contingut dels camp en blanc és irrellevant al tractar-se de pàgines invàlides o de pàgines no presents a memòria física).



Procés 1

Espai lògic

0x0	C1-1
0x1	C1-2
0x2	D1-1
0x3	D1-2
0x4	-
0x5	-
0x6	-
0x7	-
0x8	-
0x9	-
0xA	-
0xB	-
0xC	-
0xD	-
0xE	-
0xF	P1-1

Taula de pàgines

	valid	present	frame
0x0	1	1	0x1
0x1	1	1	0x7
0x2	1	0	
0x3	1	0	
0x4	0		
0x5	0		
0x6	0		
0x7	0		
0x8	0		
0x9	0		
0xA	0		
0xB	0		
0xC	0		
0xD	0		
0xE	0		
0xF	1	1	0x2

Procés 2

Espai lògic

0x0	C2-1
0x1	D2-1
0x2	-
0x3	-
0x4	-
0x5	-
0x6	-
0x7	-
0x8	-
0x9	-
0xA	-
0xB	-
0xC	-
0xD	-
0xE	P2-2
0xF	P2-1

Taula de pàgines

	valid	present	frame
0x0	1	1	0x3
0x1	1	0	
0x2	0		
0x3	0		
0x4	0		
0x5	0		
0x6	0		
0x7	0		
0x8	0		
0x9	0		
0xA	0		
0xB	0		
0xC	0		
0xD	0		
0xE	1	0	
0xF	1	1	0x6

2.5.3 La resposta depèn del procés en execució perquè cada procés utilitza una taula de pàgines diferent. Com les pàgines són de 2^{12} bytes, els 12 bits de menys pes (3 dígits hexadecimal) constitueixen el desplaçament dins de la pàgina. La resta de bits (els 4 bits alts, el primer dígit hexadecimal) constitueixen l'índex a la taula de pàgines. El resultat de les traduccions seria el següent:

Adreça Lògica	Adreça física Procés 1	Adreça física Procés 2
0x3124	Excepció: fallada de pàgina	Excepció: adreça invàlida
0xFABC	0x2ABC	0x6ABC

Es recorda que realitzar divisions on el denominador és una potència de 2 és trivial si treballem en base 2 (o en base 16). Dividir entre 2^n no és més que descartar els n bits de menys pes del numerador (expressat en base 2); el residu de la divisió entre 2^n són els n bits baixos del numerador.

3. Mòdul 4 [Del 11 al 13 de novembre] (Pes 10% = 5% +5%)
 Responen **justificadament** les següents preguntes relacionades amb el mòdul 4 de l'assignatura:

3.1 Indiqueu si els següents dispositius són físics, lògics o virtuals:

3.1.1 el canal de lectura sobre una pipe ordinària

És un dispositiu virtual, com qualsevol altre canal. És l'abstracció que ens ofereix el SO per a que un procés pugui accedir a un dispositiu d'entrada/sortida.

3.1.2 un llapis de memòria USB



Un llapis de memòria és un dispositiu físic.

3.1.3 un fitxer emmagatzemat a un llapis de memòria USB

Tot fitxer, independentment d'on s'emmagatzemi, és un dispositiu lògic. És una abstracció oferta pel SO per treballar més còmodament sobre el dispositius d'emmagatzemament.

3.1.4 el canal 0 d'un procés

El canal 0 és un dispositiu virtual, com qualsevol altre canal.

3.1.5 el fitxer `/dev/null`

`/dev/null` és un dispositiu lògic perquè és un fitxer ofert pel SO. En aquest cas és un pseudofitxer (el seu contingut no està reflectit a cap dispositiu físic) però la seva interfície és com la de qualsevol altre fitxer.

3.2 En què consisteix la independència de dispositius? Poseu un exemple de la seva utilitat.

Consisteix en que, des del punt de vista de l'usuari, tots els dispositius s'accedeixen de la mateixa forma. Els processos treballen sobre dispositius virtuals i és el SO qui s'encarrega de fer l'accés sobre el dispositiu físic. Des del punt de vista del procés, és el mateix escriure sobre una pantalla que sobre un fitxer. Això simplifica la programació de les aplicacions i facilita la reutilització del codi.

Recursos

- Mòduls 1, 2, 3 i 4 de l'assignatura.
- L'aula "Laboratori de Sistemes Operatius" (podeu plantejar els vostres dubtes relatius a l'entorn Unix, programació,...).

Criteris d'avaluació

Es valorarà la justificació de les respostes presentades.

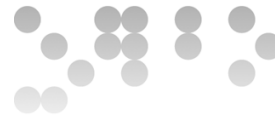
El pes de cada pregunta està indicat a l'enunciat.



Format i data de lliurament

Es lliurarà un fitxer **zip** que contingui un fitxer **pdf** amb la resposta a les preguntes i, si s'escau, els fitxers addicionals que també vulgueu lliurar.

Data límit de lliurament: 24:00 del 13 de novembre de 2019.



Procés 1

pàg. lòg.	Espai lògic		Taula de pàgines		
0x0	C1-1		valid	present	frame
0x1	C1-2				
0x2	D1-1				
0x3	D1-2				
0x4	-	0x0			
0x5	-	0x1			
0x6	-	0x2			
0x7	-	0x3			
0x8	-	0x4			
0x9	-	0x5			
0xA	-	0x6			
0xB	-	0x7			
0xC	-	0x8			
0xD	-	0x9			
0xE	-	0xA			
0xF	P1-1	0xB			
		0xC			
		0xD			
		0xE			
		0xF			

Procés 2

pàg. lòg.	Espai lògic		Taula de pàgines		
0x0	C2-1		valid	present	frame
0x1	D2-1				
0x2	-	0x0			
0x3	-	0x1			
0x4	-	0x2			
0x5	-	0x3			
0x6	-	0x4			
0x7	-	0x5			
0x8	-	0x6			
0x9	-	0x7			
0xA	-	0x8			
0xB	-	0x9			
0xC	-	0xA			
0xD	-	0xB			
0xE	P2-2	0xC			
0xF	P2-1	0xD			
		0xE			
		0xF			

Espai físic

frame	
0x0	
0x1	C1-1
0x2	P1-1
0x3	C2-1
0x4	
0x5	
0x6	P2-1
0x7	C1-2

