Espai grapa

Examen 2017/18-1

Assignatura	Codi	Data	Hora inici
Sistemes operatius	05.566	20/01/2018	09:00

05.566 20 01 18 EX

Enganxeu en aquest espai una etiqueta identificativa amb el vostre codi personal
Examen

Fitxa tècnica de l'examen

- **Comprova que el codi i el nom de l'assignatura corresponen a l'assignatura en la qual estàs matriculat.**
- **Només has d'enganxar una etiqueta d'estudiant a l'espai corresponent d'aquest full.**
- **No es poden adjuntar fulls addicionals.**
- **No es pot realitzar la prova en llapis ni en retolador gruixut.**
- **Temps total:** 2 **h.**
- **En cas que els estudiants puguin consultar algun material durant l'examen, quin o quins materials poden consultar?**
  Un full mida foli/DIN A-4 amb anotacions per les dues cares
- **Valor de cada pregunta:** 2,5 punts
- **En cas que hi hagi preguntes tipus test: Descompten les respostes errònies?** NO **Quant?**

- **Indicacions específiques per a la realització d'aquest examen:**
  Poden portar calculador per realitzar l'examen.

Enunciats

Examen 2017/18-1

Assignatura	Codi	Data	Hora inici
Sistemes operatius	05.566	20/01/2018	09:00

1. Teoria [2.5 punts]

Contesteu **justificadament** les següents preguntes:

a) Donat el següent estat inicial:

```
int x, y;
sem_t semA;
x = 1;
y = 3;
sem_init(semA, 0, 1);  // S'inicialitza el semàfor SemA a 1 (una instància)
```

Enumerar els possibles valors de x per la següent execució, després que els dos fils (Thread1 i Thread2) hagin acabat o es bloquegin. Indicar també si algun dels fils es bloqueja indefinidament i perquè.

```
Thread1:
   sem_wait(SemA);
   y = y*2;
   sem_signal(SemA);
```

```
Thread2:
   sem_wait(SemA);
   x = y+2;
   sem_signal(SemA);
```

En aquest codi tenim dos fils que accedeixen a variables compartides (x, y). Tots dos fils, abans d'accedir, demanen una instància del semàfor A, implementant una secció crítica. D'aquesta manera no es produeixen condicions de carrera. No obstant això, sí que tenim un problema de indeterminisme debut a que depenent de que fil s'executi abans el resultat pot variar.

Els possibles valors que podem obtenir de les variables x,y serien:

- x = 8, y = 6 , si el fil 1 s'executa la seva operació abans que el fil 2.
- x = 5, y = 6, si el fil 2 s'executa la seva operació abans que el fil 1.

Cap fil es bloqueja indefinidament.

b) ¿Indiqueu quants processos, pipes i redireccions necessita l'intèrpret de comandes per executar la següent comanda?

```
$ ls –la | grep nom | wc –l  >> dades.txt
```

Es necessiten 3 processos: un per a executar la comanda ls, un altre per la comanda grep i un tercer per la comanda wc. Es necessiten 2 pipes un per connectar la stdout del procés ls a la stdin del procés grep i un altre per connectar la stdout del procés grep a la stdin del procés wc. També es necessita realitzar 5 redireccions: redirigir la stdout del procés ls a l'entrada del primer pipe; redirigir la stdin del procés grep a la sortida del primer pipe; redirigir la stdout del procés grep a l'entrada del segon pipe; redirigir la stdin del procés wc a la sortida del segon pipe i redirigir el fitxer dades1.txt a la stdout del procés wc.

Assignatura	Codi	Data	Hora inici
Sistemes operatius	05.566	20/01/2018	09:00

c) Es veritat que un dels principals avantatges de la memòria virtual és que podem executar processos que tinguin un espai lògic més gran que $2^{Mida\_Bus\_Adreces}$ paraules (on Mida_Bus_Adreces és la mida del bus d'adreces del processador) independentment del tamany de la memòria física?

Fals, la memòria virtual ens permet executar processos que tinguin un espai lògic més gran que l'espai físic, però en cap cas podrem considerar adreces més grans que que $2^{Mida\_Bus\_Adreces}$ paraules ja que el rang potencial d'adreces lògiques comença a l'adreça lògica 0 i finalitza a l'adreça $2^{Mida\_Bus\_Adreces}-1$.

d) Una aplicació d'usuari pot decidir que no vol rebre una determinada interrupció?

Fals, això implicaria cedir a l'usuari el control sobre el hardware, donant peu a què les aplicacions en fessin mal ús, i causessin falles de protecció o compartició de dispositius, o simplement pèrdues o inconsistències de dades o funcionament. L'accés al sistema d'interrupcions està reservat al mode supervisor.

Examen 2017/18-1

Assignatura	Codi	Data	Hora inici
Sistemes operatius	05.566	20/01/2018	09:00

2. Processos [2.5 punts]

Contesteu les següents preguntes:

a) Indiqueu un motiu que pugui provocar que un procés passi d'estat Wait a estat Ready.

Que es produeixi l'esdeveniment que el procés està esperant (la finalització d'una operació d'entrada/sortida, una temporització,...).

b) Indiqueu quin serà el resultat d'executar el següent programa (jerarquia de processos creada, informació mostrada per la sortida estàndar). Podeu assumir que cap crida al sistema retornarà error.

```
main()
{ int a,b;
  char s[80];

  if (fork() == 0) a=0; else a=1;

  sprintf(s, "a=%d\n", a);
  write(1, s, strlen(s));

  execl("/bin/ls", "ls", NULL);
  write(1, s, strlen(s));
}
```

La primera sentència crea un procés fill; el fill tindrà a=0 i el pare a=1.
Tots dos processos imprimeixen el seu valor de "a" (no podem saber en quin ordre) i després tots dos processos invoquen execl i passen a executar "ls"; per tant, apareix dos cops la llista de fitxers del directori actual.
El write posterior no s'executarà perquè, assumint que l'exec no retornarà error), la crida execl no retorna.

Assignatura	Codi	Data	Hora inici
Sistemes operatius	05.566	20/01/2018	09:00

c) Escriviu un programa que cada cop que l'usuari premi un salt de línia, el programa indiqui quants processos en estat d'execució hi ha al sistema mitjançant la comanda "ps –aux | grep –c R" No és necessari que implementeu el tractament d'errors a les crides al sistema. Cal implementar el programa utilitzant crides al sistema Unix, no és permès utilitzar rutines de biblioteca com ara system().

```
void printPsGrep()
{
  int fd[2];

  if (pipe(fd) < 0) error("pipe");

  switch (fork ())
    {
    case -1:
      error ("fork");

    case 0:
      close(1); dup(fd[1]); close(fd[0]); close(fd[1]);
      execlp ("ps", "ps", "-aux", NULL);
      error ("execl");
    }

  switch (fork ())
    {
    case -1:
      error ("fork");

    case 0:
      close(0); dup(fd[0]); close(fd[0]); close(fd[1]);
      execlp ("grep", "grep", "-c", "R", NULL);
      error ("execl");
    }

  close(fd[0]); close(fd[1]);

  wait(NULL); wait(NULL);
}

int main (int argc, char *argv[])
{
```

Assignatura	Codi	Data	Hora inici
Sistemes operatius	05.566	20/01/2018	09:00

```
  char c;
  int r;

  while ((r = read(0, &c, 1)) > 0)
    {
      if (c=='\n')
        printPsGrep();
    }

  if (r < 0)
    error("read");

  return 0;
}
```

d) Quin esdeveniment provoca que un procés a l'estat zombie/defunct pugui desaparèixer completament del sistema.

Cal que el seu procés pare invoqui la crida al sistema wait.

Assignatura	Codi	Data	Hora inici
Sistemes operatius	05.566	20/01/2018	09:00

3. Memòria [2.5 punts]

Sigui un sistema de gestió de memòria basat en paginació sota demanda on les pàgines tenen una mida de 64KBytes, les adreces lògiques són de 20 bits i l'espai físic és de 512 KBytes.

Sobre aquest sistema es creen dos processos.

- Procés 1: el seu fitxer executable determina que el codi ocuparà dues pàgines, que les dades inicialitzades n'ocupen una, les no inicialitzades dues i la pila també dues.
- Procés 2: el seu fitxer executable determina que les àrees de codi i les dades inicialitzades ocuparan una pàgina cadascuna, que no existeixen dades no inicialitzades i que la pila ocuparà dues pàgines.

Es demana:

a) Estimeu la mida del fitxer executable corresponent al procés 1.

El fitxer executable d'un procés conté el codi i el valor inicial de les dades inicialitzades. En aquest cas, tenim dos pàgines de codi i una de dades inicialitzades per tant, aproximadament, l'executable ocuparà la mida corresponent a tres pàgines, és a dir, uns 192KB.

Per a una estimació més precisa caldria afegir la mida de les capçaleres de l'executable i caldria descomptar la fragmentació interna que pugui existir a la darrera pàgina de codi i de dades inicialitzades.

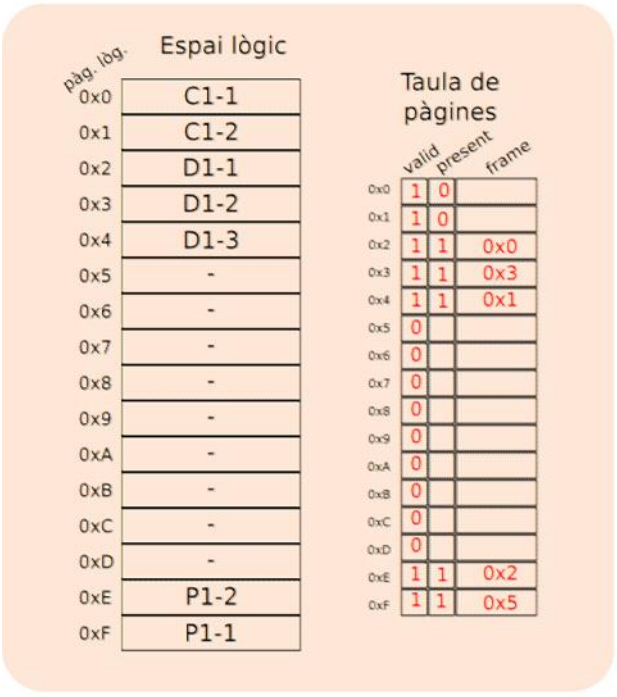
Assignatura	Codi	Data	Hora inici
Sistemes operatius	05.566	20/01/2018	09:00

b) Suposant que les pàgines es carreguen a memòria física tal i com indica el diagrama següent, indiqueu quin serà el contingut de les taules de pàgines de tots dos processos (podeu contestar sobre el diagrama de l'enunciat).

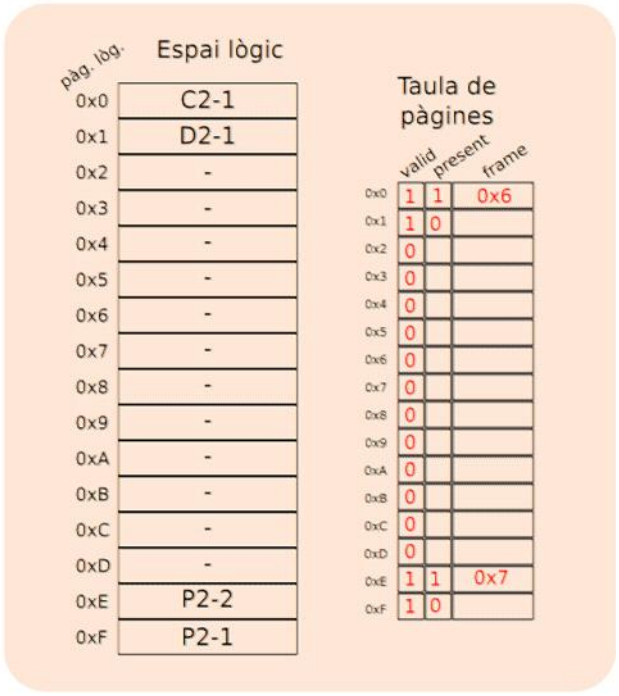
Examen 2017/18-1

Assignatura	Codi	Data	Hora inici
Sistemes operatius	05.566	20/01/2018	09:00

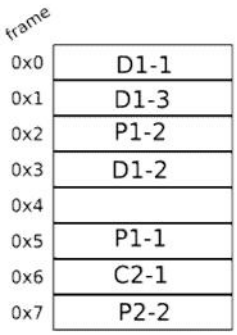
Procés 1

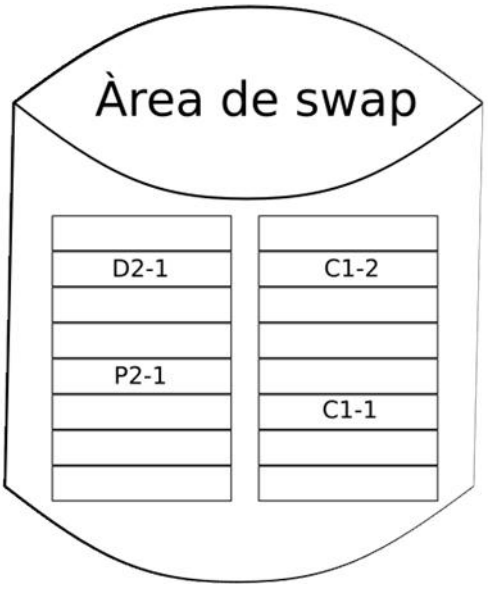


Procés 2



Espai físic





Assignatura	Codi	Data	Hora inici
Sistemes operatius	05.566	20/01/2018	09:00

c) Suposant que el procés en execució és el procés 1, indiqueu quines seran les adreces físiques corresponents a les següents adreces lògiques: 0x0A123 i 0x4B342. Variaria la resposta si el procés en execució fos el procés 2? En cas afirmatiu, indiqueu el motiu i com canviaria.

Primer cal descomposar les adreces lògiques en identificador de pàgina i desplaçament.

Com la mida de pàgina és de 64KB (2^16 bytes), calen 16 bits per a codificar el desplaçament dins de la pàgina, és a dir, quatre dígits hexadecimals. La resta de bits (4) seran l'identificador de pàgina. Per tant, el primer dígit hexadecimal ens indica l'identificador de pàgina lògica i la resta de dígits indiquen el desplaçament dins la pàgina.

Les traduccions serien:

@L	@F Procés 1	@F Procés 2
0x0A123	Fallada de pàgina	0x6A123
0x4B342	0x1B342	Excepció: Adreça invàlida

Són diferents a cada procés perquè cada procés té una taula de pàgines pròpia.

Assignatura	Codi	Data	Hora inici
Sistemes operatius	05.566	20/01/2018	09:00

4. Concurrència [2.5 punts]

La PAC2 de l'assignatura versava sobre el problema de com controlar l'accés a un pont estret per part de conjunt de cotxes, simulats mitjançant fils d'execució, que volien creuar-ho en ambdues direccions. En tot moment, només es permet creuar cotxes en un sentit, havent d'esperar els cotxes de l'altra direcció fins que se'ls concedeixi el torn. L'espera s'implementava mitjançant semàfors amb l'objecte de consumir recursos de CPU (espera passiva).

a) En una de les variants, se'ns demanava que per assegurar la integritat estructural del pont, només es permetés que creuin vehicles en una direcció mentre el seu pes total no excedeixi les 40 tones (4000Kg).

```
Shared int CarsEast=0, CarsWest=0, Weight=0;
sem_t Bridge, MutexEast, MutexWest, MaxLoad;

sem_init(Bridge, 0, 1);
sem_init(MutexEast, 0, 1);
sem_init(MutexWest, 0, 1);
sem_init(MaxLoad, 0, 0);
```

```
CarCrossEast(int car_weight)
{
   sem_wait(MutexEast);
   CarsEast++;
   if (CarsEast==1)
      sem_wait(Bridge);
   Weigth += car_weight;
   while(Weight>40)
   {
      sem_signal(MutexEast);
      sem_wait(MaxLoad);
      sem_wait(MutexEast);
    }
   sem_signal(MutexEast);

   CrossingBridge();

   sem_wait(MutexEast);
   CarsEast--;
   Weigth -= car_weight;
   if (Weight+car_weight>40)
      sem_signal(MaxLoad);
   if (CarsEast==0)
      sem_signal(Bridge);
   sem_signal(MutexEast);
}
```

```
CarCrossWest(int car_weight)
{
   sem_wait(MutexWest);
   CarsWest++;
   if (CarsWest==1)
      sem_wait(Bridge);
   Weigth += car_weight;
   while(Weight>40)
   {
      sem_signal(MutexWest);
      sem_wait(MaxLoad);
      sem_wait(MutexWest);
    }
   sem_signal(MutexWest);

   CrossingBridge();

   sem_wait(MutexWest);
   CarsWest--;
   Weigth -= car_weight;
   if (Weight+car_weight>40)
      sem_signal(MaxLoad);
   if (CarsWest==0)
      sem_signal(Bridge);
   sem_signal(MutexWest);
}
```

Assumiu que tenim 5 cotxes creuant cap a l'Est pel pont, 3 cotxes esperant per excés de pes. Mentre que en l'altre sentit tenim 4 cotxes esperant per creuar cap a l'Oest. Expliqueu en quines instruccions i

Assignatura	Codi	Data	Hora inici
Sistemes operatius	05.566	20/01/2018	09:00

perquè estan esperant cadascun d'aquests cotxes que intenten creuar el pont. Indiqueu el valor que tindran cada un dels semàfors i variables de l'algoritme.

Expliqueu també, com continuaran la seva execució una vegada tots els cotxes de la direcció Est hagin sortit del pont

El primer cotxe que intento creuar cap a l'Est ha demanat la instància del semàfor *Bridge* i se li ha assignat. A continuació els 5 cotxes han creuat el pont cap a l'est. Quan el sisè cotxe fa que el pes total superi les 40 tones, llavors entra al *while* i es queda bloquejat al semàfor MaxLoad. La resta de cotxes (7è i 8è) també es queden bloquejats en aquest semàfor.

Quan estan creuant cotxes cap a l'Est, el primer cotxe que intenti creuar cap a l'Oest es quedarà bloquejat al semàfor Bridge, mentre que la resta de cotxes que volen creuar cap a l'Est es queden bloquejats al semàfor MutexEast.

El valor dels semàfors i variables serà:

- CarsEast = 8.
- CarsWest = 1.
- Weight> 40T;
- Bridge = 0
- MutexEast = 1
- MutexWest = 0
- MaxLoad = 0

A mesura que els cotxes que creuen cap a l'est vagin sortint, reduiran el pes de la variable *Weigh* i si baixa de 40T s'alliberarà una instancia del semàfor *MaxLoad*, permetent que els que cotxes que estan esperant per creuar cap a l'est ho puguin fer.

Una vegada l'últim cotxe de l'Est hagi creuat el pont, alliberarà la instància del semàfor *Bridge*. Això provocarà que el primer cotxe que està esperant per creuar cap a l'Oest, demanant una instància del semàfor *Bridge*, la pugui obtenir i continuar amb l'execució. Una vegada aquest cotxe surti de la secció crítica definida pel semàfor *MutesWest*, alliberarà una instància d'aquest semàfor i la resta de cotxes que estaven esperant en la primera instrucció podran continuar d'un en un (aquests cotxes no demanaran una instància del semàfor *Bridge*, degut a que *CarsWest* és major que 1).

b) Aquesta solució podria generar temps d'espera molt llargs (inanició) en els cotxes d'un dels sentit si hi ha flux continu de cotxes en el sentit contrari.

Modificar la solució proposada perquè els cotxes de cadascun dels sentits es vagi alternant de forma estricta en grups de N. D'aquesta forma, passaran N cotxes en un sentit i a continuació N cotxes del sentit contrari.

```
Shared int CarsEast=0, CarsWest=0, Weight=0;
sem_t Bridge, MutexEast, MutexWest, MaxLoad;

sem_init(Bridge, 0, 1);
sem_init(MutexEast, 0, 1);
sem_init(MutexWest, 0, 1);
sem_init(MaxLoad, 0, 0);
```

Examen 2017/18-1

Assignatura	Codi	Data	Hora inici
Sistemes operatius	05.566	20/01/2018	09:00

```
CarCrossEast(int car_weight)
{
   sem_wait(MutexEast);
   CarsEast++;
   if (CarsEast==1)
      sem_wait(Bridge);
   Weigth += car_weight;
   while(Weight>40)
   {
      sem_signal(MutexEast);
      sem_wait(MaxLoad);
      sem_wait(MutexEast);
    }
   sem_signal(MutexEast);

   CrossingBridge();

   sem_wait(MutexEast);
   CarsEast--;
   Weigth -= car_weight;
   if (Weight+car_weight>40)
      sem_signal(MaxLoad);
   if (CarsEast==0)
      sem_signal(Bridge);
   sem_signal(MutexEast);
}
```

```
CarCrossWest(int car_weight)
{
   sem_wait(MutexWest);
   CarsWest++;
   if (CarsWest==1)
      sem_wait(Bridge);
   Weigth += car_weight;
   while(Weight>40)
   {
      sem_signal(MutexWest);
      sem_wait(MaxLoad);
      sem_wait(MutexWest);
    }
   sem_signal(MutexWest);

   CrossingBridge();

   sem_wait(MutexWest);
   CarsWest--;
   Weigth -= car_weight;
   if (Weight+car_weight>40)
      sem_signal(MaxLoad);
   if (CarsWest==0)
      sem_signal(Bridge);
   sem_signal(MutexWest);
}
```

La solució proposada es basa a afegir dos semàfors de torn (*TurnEast*, *TurnWest*) per bloquejar els cotxes d'aquesta direcció quan no els toqui el torn. Quan li toca el torn als cotxes d'un sentit, la variable Turn val 1, per la qual cosa pot accedir a la resta de codi. A continuació s'incrementa la variable *TurnCars* que compta el nombre de cotxes que han creuat en aquest torn. Si no s'ha arribat a N, s'allibera una instància addicional del semàfor de torn, deixant que creui un altre cotxe en el mateix sentit. Si la variable *TurnCars* ha arribat a N, significa que no pot entrar més cotxes en aquest sentit (no s'allibera la instància del semàfor Turn) i se li concedeix el torn als cotxes de l'altre sentit (s'allibera la instància del semàfor de torn de l'altre sentit).

```
Shared int CarsEast=0, CarsWest=0, TurnCars=0;
sem_t Bridge, MutexEast, MutexWest, TurnEast, TurnWest;

sem_init(Bridge, 0, 1);
sem_init(MutexEast, 0, 1);
sem_init(MutexWest, 0, 1);
sem_init(TurnEast, 0, 1);
sem_init(TurnWest, 0, 0);
```

Examen 2017/18-1

Assignatura	Codi	Data	Hora inici
Sistemes operatius	05.566	20/01/2018	09:00

```
CarCrossEast()
{
   sem_wait(TurnEast);
   sem_wait(MutexEast);
   CarsEast++;
   if (CarsEast==1)
      sem_wait(Bridge);
   TurnCars++;
   if (TurnCars==N) {
      TurnCars=0;
      sem_signal(TurnWest);
   }
   else
      sem_signal(TurnEast);
   sem_signal(MutexEast);

   CrossingBridge();

   sem_wait(MutexEast);
   CarsEast--;
   if (CarsEast==0)
      sem_signal(Bridge);
   sem_signal(MutexEast);
}
```

```
CarCrossWest()
{
   sem_wait(TurnWest);
   sem_wait(MutexWest);
   CarsWest++;
   if (CarsWest==1)
      sem_wait(Bridge);
   TurnCars++;
   if (TurnCars==N) {
      TurnCars=0;
      sem_signal(TurnEast);
   }
   else
      sem_signal(TurnWest);
   sem_signal(MutexWest);

   CrossingBridge();

   sem_wait(MutexWest);
   CarsWest--;
   if (CarsWest==0)
      sem_signal(Bridge);
   sem_signal(MutexWest);
}
```

Examen 2017/18-1

Assignatura	Codi	Data	Hora inici
Sistemes operatius	05.566	20/01/2018	09:00

Examen 2017/18-1

Assignatura	Codi	Data	Hora inici
Sistemes operatius	05.566	20/01/2018	09:00

Examen 2017/18-1

Assignatura	Codi	Data	Hora inici
Sistemes operatius	05.566	20/01/2018	09:00

Examen 2017/18-1

Assignatura	Codi	Data	Hora inici
Sistemes operatius	05.566	20/01/2018	09:00