# Exercicis Tema 4 - Sistemes Operatius

Computadors – Grau en Ciència i Enginyeria de Dades – 2019-2020 Q2

Facultat d'Informàtica de Barcelona

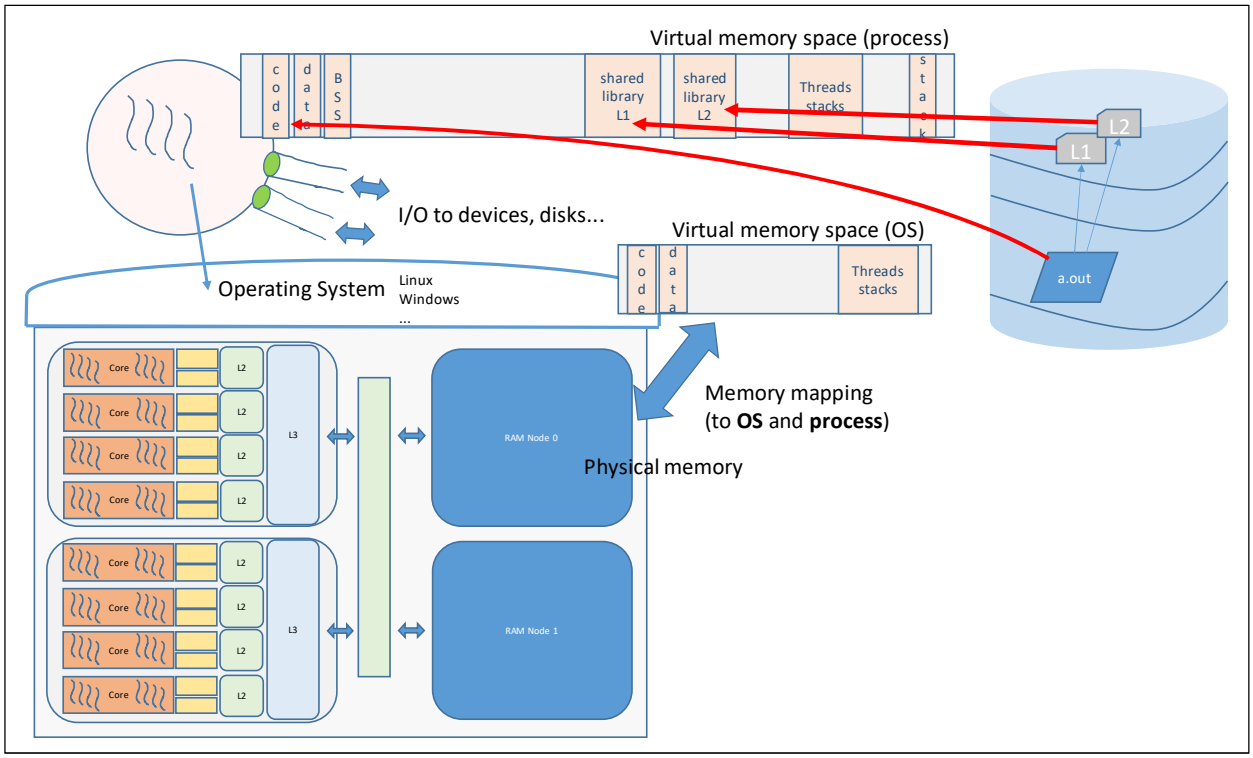



Figura 1: l'entorn d'execució

## Exercici 1

Expliqueu en quins llocs d'aquest programa hi ha una entrada al sistema per qualsevol dels 3 mecanismes existents (interrupció, excepció i crida a sistema):

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

char buffer[2];
char command[256];

int main (int argc, char * argv[])
{
   int res;
   res = read (0, buffer, 2);
   unsigned long long i;
   for (i=0; i < 10*1024*1024*1024L; i++) {
          ;
   }
   if (res > 0) {
      write (1, buffer, res);
   }
   sprintf (command, "cat /proc/%d/status", getpid());
   system(command);

   res = res - 2;

   printf ("result %d\n", 8/res);

   return 0;
}
```

## Exercici 2

En el programa anterior, indiqueu, per a cada sentència del codi, quins elements de la figura 1 – entorn d'execució fa servir.

## Exercici 3

a) Tenim el següent codi:

```
int main(int argc, char **argv)
{
    int i;
    for (i=0; i < 4; i++)
       fork();
    return 0;
}
```

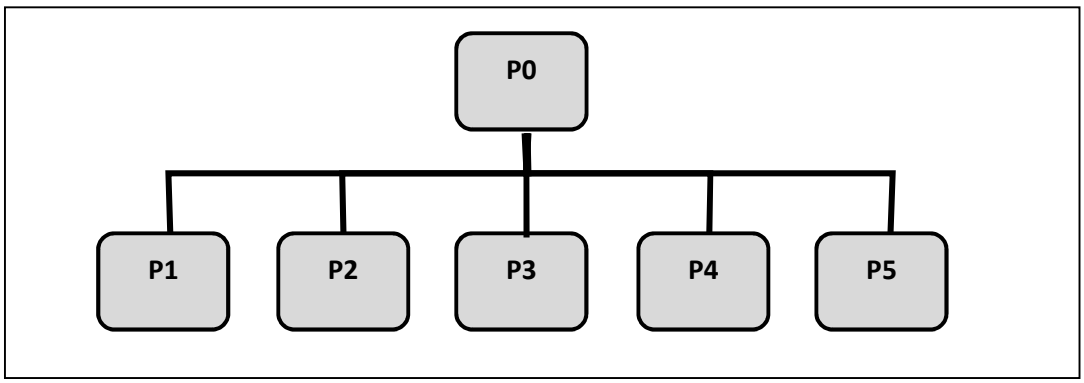
Dibuixa la jerarquia de procesos que es crea.

b) Dibuixa la jerarquia corresponent a aquesta altra variant del codi:

```
int main(int argc, char **argv){
    int res;
    int i;
    for (i=0; i < 4; i++) {
       res = fork();
       if (res == 0) {
          do_work (...);
          exit(0);
       }
       if (res < 0) {
          perror("fork");
          break;
       }
    }
    for (i=0; i < 4; i++) {
       res = waitpid(-1, NULL, 0);
       if (res < 0) {
          perror ("waitpid");
       }
    }
    return 0;
}
```

## Exercici 4

En un moment donat, tenim la següent jerarquia de procesos:

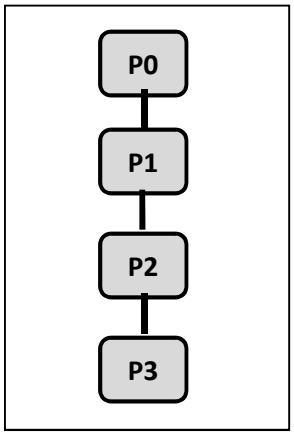



Amb aquesta informació, contesta raonadament a les següents preguntes:

a) Els processos fills s'executen seqüencialment o concurrentment?
b) Si "P0" finalitza abans que els altres, qui s'encarrega d'alliberar els PCBs dels processos "P1", "P2", "P3", "P4", i "P5" quan passin al estat zombie?

## Exercici 5

En un moment donat, tenim la següent jerarquia de procesos:




Amb aquesta informació, contesta raonadament a les següents preguntes:

a) Els processos fills s'executen seqüencialment o concurrentment?

b) Si "P0" finalitza abans que els altres, qui s'encarrega d'alliberar els PCBs dels processos "P1", "P2" i "P3" quan passin a l'estat *zombie*?

## Exercise 6

a) Feu el dibuix de l'entorn d'execució complet que resulta a l'executar el següent tros de codi. Mostreu el sistema operatiu, els dos processos que intervenen i el terminal amb la sortida dels *printf*'s:

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/wait.h>

int main ()
{
   int res;
   int pid;

   printf ("Hola, fent fork\n");
   pid = fork();
   if (pid == 0) {
       printf ("Soc el fill\n");
       exit(0);
   }
   else if (pid < 0) {
      perror ("fork");
      exit(1);
   }

   printf ("Soc el pare\n");

   res = waitpid(pid, NULL, 0);
   if (res < 0) {
      perror ("waitpid");
      exit(1);   // opcionalment... exit(2);
   }

   return 0;
}
```

b) Expliqueu perquè es podria fer un exit(2) en cas que la crida a sistema *waitpid* acabi amb error.

## Exercise 7

a) Dibuixeu la jerarquia de processos que resulta de la següent execució, indicant quin programa està executant cadascun:

```
 /* master.cpp */

#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/wait.h>

int main ()
{
   int res;
   int pid, pid2;

   pid = fork();
   if (pid == 0) {
       execlp ("./producer", "./producer", "file.transfer", NULL);
       perror("execlp (producer)");
       exit(1);
   }
   else if (pid < 0) {
       perror ("fork");
       exit(1);
   }

   pid2 = fork();
   if (pid2 == 0) {
       execlp ("./consumer", "./consumer", "file.transfer", NULL);
       perror("execlp (consumer)");
   }

   res = waitpid(pid, NULL, 0);
   if (res < 0) {
       perror ("waitpid (pid)");
   }

   res = waitpid(pid2, NULL, 0);
   if (res < 0) {
       perror ("waitpid (pid)");
       exit(1);
   }

   return 0;
}
```

b) Indica si els processos es creen de forma seqüencial o concurrent

c) Critica positivament o negativa l'ús de les crides exit(X)

## Exercici 8

Indica si els següents serveis són crides de llibreria del llenguatge o són crides a sistema. Per a cadascuna, descriu breument què fa:

| Servei | Llenguatge/sistema | Breu descripció |
|---|---|---|
| fork | | |
| printf | | |
| sprintf | | |
| open | | |
| execvp | | |
| malloc | | |
| fopen | | |
| exit | | |
| waitpid | | |