



## Final de SO

22 de gener de 2015

Enginyeria de Sistemes TIC

2 HORES

COGNOMS:

NOM:

GRUP de LAB:

**Exercici 1 [4 punts].** Responen breument aquestes preguntes:

1. Descriu la diferència entre un algoritme d'scheduling preemptiu i no preemptiu.
2. Considera quatre processos  $P_1$ ,  $P_2$ ,  $P_3$  i  $P_4$  que arrenquen amb una diferència entre ells d'1 segon. Els seus temps de burst són de 100, 30, 15 i 200 milisegons respectivament. Assumint que l'scheduling és Shortest Job First, quina és la mida mitjana de la cua de ready?
3. Què és una regió crítica i quina relació té amb l'accés a un recurs compartit entre diversos processos?
4. Les comprovacions de seguretat que afecten a les crides del sistema es podrien incorporar en la llibreria estàndard `libc`. Creus que fóra una bona idea? Per què?

**Exercici 2 [2 punts].** Considera el següent programa i assumeix que les crides al sistema sempre funcionen correctament (no hi ha errors). Raona quin és el seu comportament i determina: (a) quants processos es creen i (b) quants caràcters de cada tipus s'escriuen.

```
void main()
{
    ...
    pid1 = fork();
    printf("A");
    if (pid1) pid2 = fork();
    if (!pid2) {
        pid3 = fork();
        printf("B");
    }
    else {
        execvp(...);
        printf("C");
    }
    if (pid1 != pid3) {
        printf("D");
        if (pid1 && pid3) {
            printf("E");
            execvp(...);
        }
        pid4 = fork();
        printf("F");
    }
    else {
        printf("G");
        pid5 = fork();
    }
    if (pid2) {
        pid6 = fork();
        pid7 = fork();
        if (pid6) pid8 = fork();
        printf("H");
    }
    if (!pid1 && !pid2 && !pid3 && !pid4 && !pid5) printf("I");
    execvp(...);
    printf("J");
}
```



**Exercici 3 [2 punts].** Un computador executa 3 processos  $P_1$ ,  $P_2$ , i  $P_3$  que comparteixen els semàfors  $A$ ,  $B$  i  $C$ . Suposant que el valor inicial dels semàfors és  $A = 2$ ,  $B = 1$  i  $C = 1$  i que els processos tenen el següent aspecte:

$P_1$	$P_2$	$P_3$
wait(A)	wait(B)	wait(C)
wait(A)	⋮	wait(C)
wait(A)	⋮	wait(C)
⋮	⋮	⋮
⋮	signal(C)	signal(B)
signal(B)	signal(A)	signal(B)

Determineu el cronograma d'execució dels processos.

**Exercici 4 [2 punts].** Un computador executa 2 processos que accedeixen a un recurs compartit, per exemple una variable, en sengles regions crítiques. Implementeu un mecanisme d'exclusió mútua usant un sistema de comunicació entre processos per missatges.