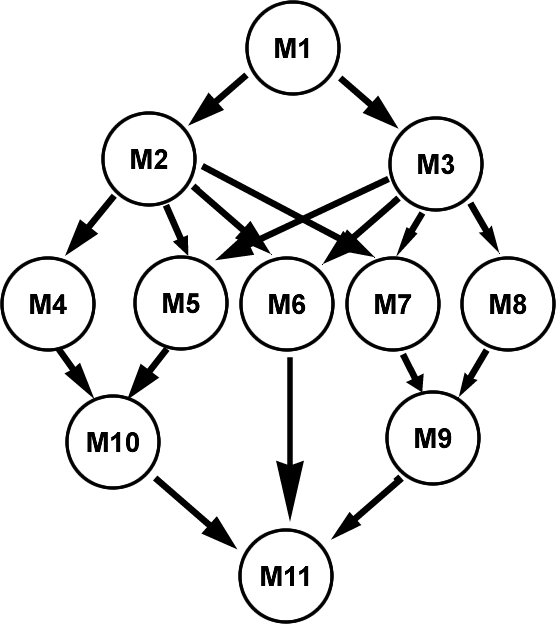
**Problema d’Introducció a la Concurrència**

Joel Aumedes Serrano i Joel Farré Cortés

# Apartat a)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 | M10 | M11 |
| M1 |  | No con. | No con. | Con. | Con. | Con. | Con. | Con. | Con. | Con. | Con. |
| M2 |  |  | Con. | No con. | No con. | No con. | No con. | Con. | Con. | Con. | Con. |
| M3 |  |  |  | Con. | No con. | No con. | No con. | No con. | Con. | Con. | Con. |
| M4 |  |  |  |  | Con. | Con. | Con. | Con. | Con. | No con. | Con. |
| M5 |  |  |  |  |  | Con. | Con. | Con. | Con. | No con. | Con. |
| M6 |  |  |  |  |  |  | Con. | Con. | Con. | Con. | No con. |
| M7 |  |  |  |  |  |  |  | Con. | No con. | Con. | Con. |
| M8 |  |  |  |  |  |  |  |  | No con. | Con. | Con. |
| M9 |  |  |  |  |  |  |  |  |  | Con. | No con. |
| M10 |  |  |  |  |  |  |  |  |  |  | No con. |
| M11 |  |  |  |  |  |  |  |  |  |  |  |

****

# Apartat b)

Per a fer aquest apartat hem definit els següents blocs dins del diagrama:

Begin:

M1;

Cobegin:

M2;

M3;

Cobegin:

M4;

M5;

M6;

M7;

M8;

Coend;

Cobegin:

M9;

M10;

Coend;

Coend;

M11;

End;

# Apartat c)

Per a fer aquest apartat hem definit els següents camins concurrents:

int num = 3, num\_2 = 2, num\_3 = 2, num\_dummy = 7;

M1;

fork Rama1;

M2;

DummySinc1:

join num\_dummy;

fork Rama2;

fork Rama3;

fork Rama4;

M4;

Sinc2:

join num\_2;

M10;

Sinc1:

join num;

M11;

exit;

Rama1:

M3;

fork Dummy1;

M8;

Sinc3:

join num\_3;

M9;

goto Sinc1;

Rama2:

M5;

goto Sinc2;

Rama3:

M6;

goto Sinc1;

Rama4:

M7;

goto Sinc3;

Dummy1:

goto DummySinc1;

# Apartat d)

Per a fer aquest apartat hem fet servir els mateixos camins que hem definit a l’apartat anterior.

M1;

signal(SIGUSR1, SIG\_IGN);

if((int pid1 = fork()) == 0){

M3;

kill(getppid(),SIGUSR1);

M8;

exit(0);

}

M2;

if((int pid5 = fork()) == 0){

M4;

}

pause();

if((int pid2 = fork()) == 0){

M5;

}

if((int pid3 = fork()) == 0){

M6;

}

if((int pid4 = fork()) == 0){

M7;

}

if((int pid6 = fork()) == 0){

wait(pid1);

wait(pid4);

M9;

}

wait(pid2);

wait(pid5);

M10;

wait(pid6);

wait(pid3);

M11;

# Apartat e)

a)

...

... ...

... ...

... 

En l’aplicació seqüencial s’executen totes les tasques una darrere l’altra, i en total es tarden 83 unitats de temps.

b)

...

... ...

... ...

... 

L’aplicació concurrent amb un sol processador no aprofita al màxim la concurrència bloquejant-se en alguns moments, tardant un total de 65 unitats de temps.

c)

L’aplicació seqüencial, per molts processadors que tingui, seguirà executant les tasques una a una, tornant a tardar 83 unitats de temps amb el mateix diagrama que a l’apartat a.

d)

...

... ...

... 

L’aplicació concurrent, amb 2 processadors, té suficient per a aprofitar la concurrència i tardar 50 unitats de temps.

e)

...

... ...

... 

L’aplicació concurrent pot utilitzar els 4 processadors per acabar l’execució en 40 unitats de temps.