

Alumne 1:	Nota:
Alumne 2:	

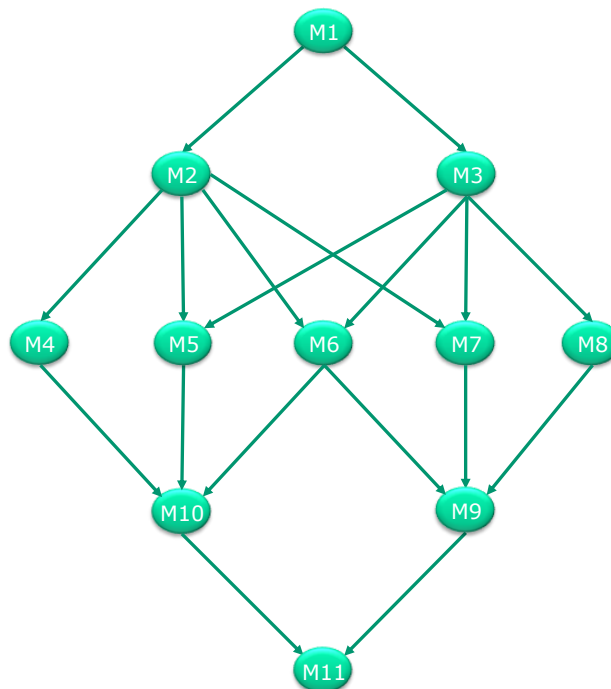
### Data de Lliurament

La data límit de lliurament del problema és **el dimarts 22 de desembre 2020 a les 9:00h.**

### Enunciat Problema

#### PROBLEMA 1 [3.0 punts]

A partir del següent graf de precedència obtingut en el problema 1 d'avaluació contínua:



Es demana:

- Implementeu el graf de precedència mitjançant els constructors ParBegin/ParEnd i assumint que tots els processos s'arrenquen només quan són necessaris. Minimitzar el nombre de semàfors utilitzats. [1.5 punts].
- Implementeu el graf de precedència mitjançant els constructors ParBegin/ParEnd, assumint que tots els processos s'arrenquen simultàniament a l'inici de l'aplicació. Minimitzar el nombre de semàfors utilitzats. [1.5 punts].

### PROBLEMA 2 [2.0 punts]

Justifiqueu si el següent algorisme (sol es dona el pseudo-codi del procés  $P_i$ , per l'altre procés ( $P_j$ ), ja l'heu de saber) compleix totes les condicions de la secció crítica per a dos processos.

#### Procés $P_i$

*Inicialment:  $Bandera[i] = Bandera[j] = Falso$ ;*

**Do** {

Bandera [i] = Cierto;

**While** (Bandera [j]==Cierto) {

Bandera [i] = Falso;

**While** (Bandera [j]==Cierto);

Bandera [i] = Cierto;

}

**REGION CRÍTICA**

Bandera[j] = Falso;

.....  
} **while**(1);

### PROBLEMA 3 [5.0 punts]

La companyia d'informació telefònica *Timofonica* vol fer un estudi per determinar l'eficiència d'implantar un nou sistema de gestió de les trucades rebudes. Per fer-ho ens ha encarregat que fem una simulació, mitjançant processos, del procés de recepció i atenció de les trucades.

Cada cop que es rep una nova trucada aquesta es posa a la cua de recepció, que té un capacitat limitada. Quan un operador queda disponible, captura una trucada de la cua de recepció i respon la informació demanada. Un cop s'ha resolt la informació demanada per l'usuari, l'operador que ha atès la trucada torna a estar lliure.

Per fer la simulació considerarem un procés Trucada per cada trucada entrant; un procés Receptor que s'encarrega d'emmagatzemar les trucades entrants; un procés Assignador que assigna les trucades als operadors; i tants processos Tramitador com operadors, que es responsabilitzen d'atendre les peticions d'informació.

Cada procés Trucada simplement ha d'indicar la seva presència al sistema.

El procés Receptor ha de comprovar que hi ha espai suficient a la cua de recepció per a guardar-hi la trucada; i en cas afirmatiu, guardar-la. La cua de recepció pot ser accedida simultàniament pel procés Receptor i els processos Operadors.

El procés Assignador ha de comprovar que a la cua de recepció hi ha alguna trucada pendent de ser atesa, i que hi ha algun operador disponible. Un cop comprovat això, treu la trucada de la cua d'espera (i n'allibera l'espai corresponent), i disminueix el nombre d'operadors lliures. Nota: com que els operadors són indistingibles no es manté una marca de lliure/ocupat per cada operador, sinó simplement un control de quants operadors tenim lliures o ocupats.

Cada procés Tramitador espera a detectar una trucada assignada per l'Assignador; l'atén; i un cop resolta la petició d'informació, allibera l'operador.

Considerem que cua de recepció té una capacitat de  $T$ , i que la companyia disposa de  $O$  operadors.

Es demana:

- Us proposen la següent solució, on hi manquen totes les instruccions del control de la concurrència i la sincronització de tasques. Afegiu-les en els recuadres les operacions amb semàfors necessaris (**en cada recuadre poden anar 0,1 o més instruccions**). Tingueu especial cura en l'accés a recursos compartits. [4 punts]
- Per cada semàfor utilitzat doneu-ne una descripció del seu funcionament, el tipus i el seu valor inicial. [1 punt]

### TRUCADA

/\* Indiquem que arriba UNA TRUCADA \*/

### RECEPTOR

mentre(cert)

{

/\* Està atent a l'arribada de noves trucades. Si a la cua de recepció hi ha prou espai, l'encua \*/

guarda la trucada a la cua de recepció

}

### ASSIGNADOR

mentre(cert)

{

/\* Si hi ha alguna trucada en espera a la cua de recepció, i un operador lliure, redirecciona la trucada cap a un operador \*/

desencua la trucada i la redirecciona a un operador

}

### TRAMITADOR

mentre(cert)

{

/\* Atén la trucada \*/

atén la trucada

}