

**SISTEMAS OPERATIVOS
PRIMERA PRUEBA
DE
EVALUACIÓN A DISTANCIA
(PED1)**

Curso 2016-2017

Alumno: Julio Alberto Raposo Montaña

Dni: 49041176-R

1. Explique razonadamente si las siguientes afirmaciones son verdaderas o falsas:

I) (1 p) El bloqueo de interrupciones es una solución simple a la exclusión mutua con apoyo del hardware que no presenta serios inconvenientes.

Falsa. Presenta serios inconvenientes, el rendimiento del sistema se puede degradar al no permitir atender las interrupciones más prioritarias en el momento en que llegan. Además, deja en las manos del proceso la decisión de ceder el procesador y si el proceso nunca lo devuelve el sistema se quedará colgado. También en sistemas multiprocesadores no sirve esta forma de conseguir la exclusión mutua.

II) (1 p) El tiempo de conmutación de un cambio de proceso es independiente del hardware.

Falsa. El tiempo de conmutación depende principalmente de factores asociados al hardware como por ejemplo la velocidad de lectura/escritura en memoria principal, el número de registros del procesador que hay que cargar/guardar...

III) (1 p) El grado de multiprogramación viene definido por el número de procesadores existentes en el computador.

Falsa. El grado de multiprogramación viene definido por el número de programas cargados en memoria principal.

IV) (1 p) El algoritmo de planificación basado en prioridades solo puede implementarse como un algoritmo de tipo no expropiativo.

Falsa. El algoritmo de planificación basado en prioridades puede ser expropiativo o no expropiativo.

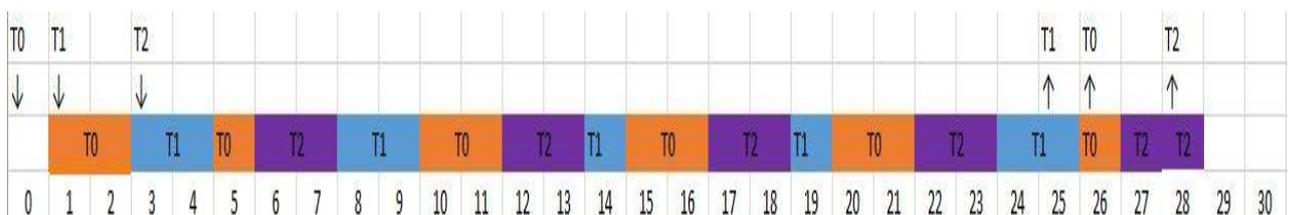
2. Se dispone del siguiente conjunto de trabajos para su planificación:

Trabajo	Instante de llegada (ut)	Duración de las ráfagas de CPU (ut)
T0	0	3, 2, 2, 2, 1
T1	1	2, 3, 1, 2
T2	3	8, 1, 1

Si el sistema operativo utiliza un algoritmo de planificación de tipo round-robin con un quantum $q=2$ ut. Se pide:

- (1 p) Dibujar el diagrama de uso de la CPU.
- (1 p) Calcular los tiempos de espera y finalización de cada trabajo.
- (1 p) Si se considera el tiempo de respuesta medio de un trabajo como el tiempo transcurrido en el sistema hasta completarse cada ráfaga del trabajo dividido por el número de ráfagas del trabajo, calcular los tiempos medios de respuesta para cada trabajo.

A)



B)

	T. llegada	T.espera	T.servicio	T.retorno	T.finalizacion
T0	0	16	10	26	26
T1	1	16	8	24	25
T2	3	15	10	25	28

C)

Tiempos de inicio

Ráfagas	T0	T1	T2
1	0	1	3
2	5	4	23
3	11	14	27
4	16	19	-
5	21	-	-

Tiempos de finalización

Ráfagas	T0	T1	T2
1	5	4	23
2	11	14	27
3	16	19	28
4	21	25	-
5	26	-	-

Tiempos finales – Tiempos iniciales

Ráfagas	T0	T1	T2
1	5	3	20
2	6	10	4
3	5	5	1
4	5	6	-
5	5	-	-

Para T0 tiempo de respuesta medio/número de ráfagas = $(5+6+5+5+5) / 5 = 5.2$

Para T1 tiempo de respuesta medio/número de ráfagas = $(3+10+5+6) / 4 = 6$

Para T2 tiempo de respuesta medio /número de ráfagas = $(20+4+1) / 3 = 8.33$

3)

Usare 3 semáforos y 2 variables globales, un semáforo por cada variable global y el otro para el uso de la bandeja.

```
/*definición de variables y semáforos */
Int piezasA, piezasB =0;
Semáforo_binario A,B,Bandeja;
ProdRobot1() /* producción del robot 1*/
{
While(true){
Wait_sem(A);
If (piezasA<2){
Wait_sem(Bandeja);
Producir();
piezasA= piezasA + 1;
Signal_sem(Bandeja);
}
Signal_sem(A):
}
}
ProdRobot2() /* producción del robot 2*/
{
While(true){
Wait_sem(B);
If (piezasB<2){
Wait_sem(Bandeja);
Producir();
piezasB= piezasB + 1;
Signal_sem(Bandeja);
```

```

}
Signal_sem(B):
}
}
ProdRobot3()/* producción del robot 3*/
{
While (true){
Wait sem(A);
Wait_sem(B):
If(piezasA ==2 && piezasB==2){
    Wait_sem(Bandeja);
    Producir;
    PiezasA= 0;
    PiezasB= 0;
    Wait_sem(Bandeja);
}
Signal_sem(A);
Signal_sem(B);
}
}
Void main (void){
Initialize_sem(A,1);
Initialize_sem(B,1);
Initialize_sem(Bandeja,1);
    Cobegin(ProdRobot1, ProdRobot2 ProdRobot3);
}

```