Convocatoria Oficial – Sistemas Operativos

Ordinaria del 30 enero de 2021

Departamento de Ingeniería de Computadores

Escuela Politécnica Superior de Elche

Para los resultados del test y el ejercicio de planificación utilizar la plantilla recibida o una similar. Hay tres rejillas por si necesitáis rectificar o usar como borrador. **Marca claramente cuál es la definitiva.**

Test

(1.5 puntos, 0.15 cada una)

1.-En un sistema con planificación expulsiva SRTF llegan los procesos en los momentos indicados en la tabla para ejecutarse, y cada uno necesita la cantidad de tiempo indicada para ejecutarse.

Proceso Momento de llegada Necesidad de CPU
llegada/cpu
P1 0/7
P2 4/4
P3 9/3

El tiempo de retorno de cada proceso es:
a) T1=7 T2=7 T3=7
b) T1=8 T2=4 T3=7
c) T1=7 T2=7 T3=4
d) T1=7 T2=7 T3=5

2.- El repartidor (dispatcher) se encarga de:
a) Planificar los distintos trabajos
b) Definir las políticas de planificación
c) Realizar el cambio de contexto.
d) Desbloquear los procesos que están esperando una E/S cuando ésta finaliza

3.- El planificador (scheduler) es la parte del sistema operativo encargada de:
a) Realizar el cambio de contexto entre los procesos.
b) Determinar el orden de ejecución entre los procesos
c) Mantener la tabla de procesos
d) Descargar los procesos a disco cuando la memoria del sistema escasea.

4.- El cambio de contexto:
a) Lo realiza el scheduler.
b) Modifica la entrada en la tabla de procesos del proceso desalojado
c) Siempre se origina por una interrupción
d) Se produce siempre que el proceso abandona la cola de procesos en espera y pasa a la cola de procesos preparados

5.- ¿Cuándo puede un proceso entrar en un monitor?
a) Sólo cuando tiene definidas variables de condición.
b) Sólo es necesario que el monitor se encuentre "vacío".
c) Hace falta que el monitor esté vacío y no haya procesos esperando en ninguna cola de condición.
d) Ninguna del resto de las contestaciones es correcta.

6.- El semaforo elimina la espera activa porque:
a) Se inicializa al número máximo de recursos que se comparten
b) Las operaciones de wait y signal se implementan como acciones indivisibles
c) El semaforo no elimina la espera activa
d) Se implementa con una cola de tareas a la cual se añaden los procesos que estan en espera del recurso.

7.- Los monitores proporcionan exclusion mutua porque:
a) Sólo un proceso puede estar activo cada vez para ejecutar un procedimiento del monitor
b) Para ello utilizan variables de condición
c) No proporcionan exclusión mútua sólo garantizan la sincronización
d) Se diseñan como procedimientos encapsulados en un módulo

8.- El algoritmo de Peterson corresponde a:
a) Una estrategia de sincronización de procesos
b) Un método para mantener el grado de multiprogamación a un valor dado
c) Un algoritmo que evita la espera activa
d) Una solución al problema de la exclusión mutua

9.- Si se usa un semáforo para lograr la sincronización de procesos
a) Éste debe inicilizarse al número de procesos que se desean sincronizar
b) Se deben incluir variables de condición, pues el semaforo sólo proporciona exclusión mutua
c) Las operaciones wait y signal se utilizan dentro del mismo proceso
d) Las operaciones wait y signal se utilizan en procesos separados

10.- La operacion wait de un semáforo y la cwait (delay) de una varialbe de condición de un monitor se diferencian en:
a) que en el caso de la variable de condición siempre se suspende el proceso que la invoca
b) que en el caso de la variable de condición no se elimina la espera activa
c) No existe diferencia pues en ambos casos siempre sirve para lograr la exclusión mutua de la sección crítica
d) No existe diferencia pues en ambos casos siempre sirve como mecanismo de sincronización

Contestar utilizando esta plantilla del pdf que habéis recibido vía anuncio de la asignatura.

1	2	3	4	5	6	7	8	9	10

Preguntas

1 – (1 Punto)

- Explica las diferencias entre threads ULT y KLT
- Enumera las ventajas e inconvenientes de usar un esquema u otro.

Ejercicios

2 – 5 puntos

En un parking público caben 100 coches. El parking tiene un número determinado de entradas y de salidas. Por la vía publica hay circulando 200 coches, tras un periodo de circulación un coche decide entrar al parking y estacionar. Para ello comprobará primero si hay sitio en el parking. Si hay sitio entrará al parking. Si no hay sitio volverá a circular otro tiempo determinado. Cuando el coche está en el interior del parking, está estacionado durante un tiempo determinado hasta que decida salir del parking.

SE PIDE:
Gestionar mediante **semáforos y mediante paso de mensajes** el comportamiento del sistema, de tal forma que no puedan entrar más coches al parking de los permitidos y que se controle el aforo del parking correctamente.
Semáforos: utilizar sólo un semáforo binario sbMutex
Paso de mensajes: receive bloqueante, send no bloqueante
Para cada solución se dispone de un esqueleto.
Entregar los distintos bloques TODO indicando la numeración de cada uno.
Solución con semáforos: 2.5 puntos
Solución con paso de mensajes: 2.5 puntos

ESQUELETO SOLUCION SEMAFOROS

```
int plazasLibres=100;
semaphore sbMutex;
function circulando(){
pause(random);
}
function estacionado(){
pause(random);
}
function entrar(){
//BEGIN TODO 1 --------------------
//END TODO 1----------------------
}
function salir(){
//BEGIN TODO 2 --------------------
//END TODO 2 ----------------------
}
function Coche(){
while true(){
//BEGIN TODO 3 ----------------------
//END TODO 3
}
}
begin
sCreate(sbMutex);
sInit(sbMutex,1);
//Se lanzan 200 procesos coche en paralelo
parbegin
Coche[200]();
parend
end
```

ESQUELETO SOLUCION PASO DE MENSAJES.

```
int MAXPLAZAS=100;
mailbox bPlazas;
function circulando(){
pause(random);
}
function estacionado(){
pause(random);
}
function Coche(){
tMessage msg=null;
while true(){
//BEGIN TODO 2 ----------------------
//END TODO 2
}
}
begin
int i;
tMessage msg=null;
//BEGIN TODO 1 ----------------------------
//Inicializamos el buzón
//END TODO 1
//Se lanzan 200 procesos coche en paralelo
parbegin
Coche[200]();
parend
end
```

3 – 2.5 puntos

PRIORIZAMOS AL QUE LLEGA FRENTE AL QUE SALE DE CPU

ORDEN FIFO SI COINCIDE LA SALIDA DE BLOQUEO

Proceso	Llegada	Carga	Carga total
P0	0	3 (1) 3 (1)	8
P1	1	3 (1) 3	7
P2	2	2 (2) 3	7
P3	3	3 (3) 2	8

N	Ráfaga de CPU
(N)	Ráfaga de E/S

Rellenar correctamente el diagrama de planificación de procesos para los datos mostrados.

- (1.25 puntos) Utilizando un algoritmo Round Robin con quantum=2.
- (0.75 puntos) Rellenar una tabla de datos como la adjunta para proporcionar los tiempos medios de: Espera, Retorno y Respuesta. Calcular la Tasa media de CPU de los procesos.
- (0.5 puntos) Ante la ejecución con un quantum=2 y con la carga del sistema, explica que consecuencias tendría incrementar el quantum a 3 u.t.

	Carga			Tiempo de			Tasa CPU
	E/S	CPU	Total	Retorno	Espera	Respuesta	
P0							
P1							
P2							
P3							
			Medias