

Grado en Ingeniería Informática

NOTAS:

- * La fecha de publicación de las notas, así como de revisión se notificarán por Aula Global.
 - * Para la realización del presente examen se dispondrá de 1:50 horas.
 - * No se pueden utilizar libros ni apuntes
 - * Será necesario presentar el DNI o carnet universitario para realizar la entrega del examen
-

Ejercicio 1 (3 puntos). Autotest.

Responda a las preguntas del autotest en los cuadros adjuntos indicando la letra de la respuesta válida. Recuerde que por cada 3 fallos se quita un punto. No contestadas no penalizan.

NOMBRE:

GRUPO:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

PREGUNTAS

1 - ¿Qué sucede cuando un proceso recibe una señal?

- A) Si la tiene capturada, se ejecuta el manejador y muere
- B) Si no la tiene capturada, se restaura la acción por defecto
- C) Si la tiene ignorada se restaura la acción por defecto.
- D) Si tiene la acción por defecto el proceso muere.

2 - El cambio de contexto es realizado por:

- A) El planificador.
- B) El dispatcher.
- C) La aplicación del usuario.
- D) El proceso init.

3 - En un sistema operativo con threads de kernel sobre un monoprocesador, que emplea una política pura FIFO para planificar los threads pertenecientes a un proceso, se ejecuta un proceso con 3 threads. ¿Cuál de las siguientes afirmaciones es falsa?

- A) Las llamadas al sistema no bloquean necesariamente al proceso.
- B) Un thread no puede escribir en la pila de otro thread.
- C) Si los threads incluyen llamadas al sistema bloqueantes, entonces ejecutan los tres.

Grado en Ingeniería Informática

D)Un proceso ligero puede escribir en un fichero creado por otro proceso ligero.

4 - ¿Cuál de las siguientes instrucciones máquina no debería ejecutarse en modo protegido?

A)Inhibir interrupciones.

B)Instrucción trap.

C)Modificar el reloj del sistema.

D)Cambiar el mapa de memoria.

5 - ¿Cuál de las siguientes técnicas hardware tiene mayor influencia en la construcción de un sistema operativo?

A)DMA

B)Microprogramación del procesador.

C)Cache de la memoria principal

D)RISC

6 - Se tienen 3 procesos: P1, P2 y P3, con tiempos de ejecución: 65, 45 y 120 ms, respectivamente. Si actúa el planificador a corto plazo según el algoritmo SJF (Largest Job First) se obtiene que:

A)Los procesos se encuentran en la lista de preparados en el orden: P2, P1 y P3.

B)Los procesos se ejecutan en el orden: P3, P2 y P1.

C)Los procesos se ejecutan en el orden de llegada al sistema: P1, P2 y P3.

D)Los procesos se ejecutan en el orden: P3, P1 y P2.

7 - ¿Cuál de las siguientes afirmaciones es falsa?

A)Después de la ejecución de cada instrucción, se comprueba la existencia o no de alguna señal de interrupción.

B)Las interrupciones o ticks de reloj hacen que el sistema operativo entre en funcionamiento.

C)Cualquier instrucción que se ejecute en un procesador debe estar almacenada en la memoria principal.

D)Ninguna de las respuestas anteriores es correcta.

Grado en Ingeniería Informática

8 - ¿Cuál de las siguientes comunicaciones no está permitida en un micronúcleo?

- A) Proceso de usuario con tareas del núcleo.
- B) Gestor de Memoria con Sistema de Ficheros.
- C) Tareas del núcleo entre sí.
- D) Tarea del núcleo con Gestor de Memoria.

9 - ¿Cuál de las siguientes políticas de planificación es más adecuada para un sistema de tiempo compartido?

- A) Shortest Job First.
- B) Round Robin.
- C) Prioridades.
- D) FIFO.

10 - ¿Cuál de los siguientes criterios de prioridad asegura que no se produzca inanición en la planificación de procesos?

- A) Más prioridad cuanto más larga haya sido la última racha de CPU del proceso.
- B) Más prioridad cuantas menos rachas de CPU lleve ejecutadas el proceso.
- C) Más prioridad cuantas más veces se haya quedado bloqueado el proceso.
- D) Más prioridad cuanto más reciente haya sido la última racha de CPU del proceso.

Grado en Ingeniería Informática

11 - ¿En cuál de los siguientes algoritmos de planificación de procesos el tratamiento de una interrupción que indica el final de una operación de E/S bloqueante puede producir cambio de proceso?

- A)FIFO
- B)Menor tiempo restante (Shortest-Remaining-Time First)
- C)Turno rotatorio (Round-Robin)
- D)Primero el proceso más corto (Shortest-Job First o Shortest-Process Next)

12 - Indique qué es falso en relación con la planificación de procesos:

- A)El método round-robin es un sistema de planificación equitativo.
- B)El método shortest-job-first puede producir inanición.
- C)Para evitar la inanición se pueden utilizar técnicas de envejecimiento de procesos.
- D)Los sistemas de planificación con expulsión son adecuados para planificación por lotes.

13 - Las ventajas del control de threads a nivel de núcleo frente al control a nivel de usuario son:

- A) El cambio de contexto es más rápido.
- B) Puede aprovechar el paralelismo hardware del sistema.
- C) Los threads se pueden planificar de forma independiente evitando que el thread esté en estado suspendido si está esperando por una E/S.
- D) Todas las anteriores.

14 - ¿Qué es falso respecto a la llamada al sistema EXEC?

- A)Cambia la imagen de memoria de un proceso.
- B)Permite pasar argumentos al programa invocado.
- C)Cambia el identificador de fichero del proceso.
- D)Mantiene los descriptores abiertos.

15 - ¿Qué algoritmo de planificación será más conveniente para optimizar el rendimiento de la CPU en un sistema que sólo tiene procesos en los cuales no hay entrada/salida?

- A)FIFO

Grado en Ingeniería Informática

B)Prioridad con expulsión.

C)Round Robin

D)Prioridad con expulsión y envejecimiento.

Grado en Ingeniería Informática

Ejercicio 2 (3,5 puntos).

Dado el siguiente conjunto de procesos, con el tiempo de ráfaga de CPU dada en milisegundos:

PROCESO	SERVICIO	PRIORIDAD
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

Se supone que los procesos llegaron en orden P1, P2, P3, P4 y P5, todos ellos en el momento 0.

(Ejemplo de representación gráfica

P1
P2
P3
P4
 0-----3-----7-----12-----24)

1. Represente gráficamente la ejecución de estos procesos empleando la planificación

a. FCFS (First Come, First Serve)

P1 P2 P3 P4 P5
 0---10---11---13---14---19

b. SJF (Shortest Job First)

P2 P4 P3 P5 P1
 0---1---2---4---9---19

c. Round Robin con quantum=1

P1 P2 P3 P4 P5 P1 P3 P5 P1 P5 P1 P5 P1 P5 P1 P1 P1 P1 P1
 0---1---2---3---4---5---6---7---8---9---10---11---12---13---14---15---16---17---18---19

Grado en Ingeniería Informática

d. Planificación expulsiva por prioridades en la cual un número de prioridad menor implica una mayor prioridad.

P2 P5 P1 P3 P4

0---1---6---16---18---19

2. Calcule el tiempo de retorno, espera y retorno normalizado para cada una de las planificaciones descritas en el apartado anterior.

SOL: TR= FIN LLEGADA; TE= RETORNO - SERVICIO

a. FCFS (First Come, First Serve)

PROCESO	TIEMPO DE RETORNO	TIEMPO DE ESPERA	RETORNO NORMALIZADO
P1	10	0	1
P2	11	10	11
P3	13	11	6,5
P4	14	13	14
P5	19	14	3,8

b. SJF (Shortest Job First)

PROCESO	TIEMPO DE RETORNO	TIEMPO DE ESPERA	RETORNO NORMALIZADO
P1	19	9	1,9
P2	1	0	1
P3	4	2	2
P4	2	1	2
P5	9	4	1,8

Grado en Ingeniería Informática

c. Round Robin con quatum=1

PROCESO	TIEMPO DE RETORNO	TIEMPO DE ESPERA	RETORNO NORMALIZADO
P1	19	9	1,9
P2	2	1	2
P3	7	5	3,5
P4	4	3	4
P5	14	9	2,8

d. Planificación expulsiva por prioridades en la cual un número de prioridad menor implica una mayor prioridad.

PROCESO	TIEMPO DE RETORNO	TIEMPO DE ESPERA	RETORNO NORMALIZADO
P1	16	6	1,6
P2	1	0	1
P3	18	16	9
P4	19	18	19
P5	6	1	1,2

no es necesario realizar el programa en C, solo poner las llamadas al sistema dentro de la tabla que se facilita y de acuerdo a este ejemplo.

Proceso Padre	Proceso hijo1	Proceso hijo2	Proceso hijo3
fork()	close(1); fork(); exec(ls);	exec(ps.)	
SOLUCIÓN: Proceso Padre	Proceso hijo1	Proceso hijo2	Proceso hijo3

Grado en Ingeniería Informática

SOLUCIÓN:

Proceso Padre

Proceso hijo1

Proceso hijo2

Proceso hijo3

```

pipe(fd1)
fork()
f=open(fichero)
close(1)
dup(fd1[1])
close(fd1[0])
close(fd1[1])
exec(cat,fichero)
exit()

```

```

pipe(fd2)
fork()
close(0)
dup(fd1[0])
close(fd1[0])
close(fd1[1])
close(1)
dup(fd2[1 ])
close(fd2[0])
close(fd2[1])
exec(sort)
exit()

```

```

close(0)
dup(fd2[0])
close(fd2[0])
close(fd2[1])
close(fd1[0])
close(fd1[1])
exec(wc)
exit()

```