1. (1 punto) ¿Se puede afirmar cuál es el factor que tiene más influencia en el rendimiento de un sistema informático (el planificador de corto plazo, el administrador de memoria virtual, los módulos de entrada/salida, etc…)? ¿Por qué?
2. (1 punto ) ¿Por qué razones puede finalizar la ejecución de un proceso? Enumerar al menos tres razones posibles.
3. (0,5 puntos) Respecto a los posibles estados de un proceso, ¿existe alguna ventaja en la implementación del estado bloqueado con varias colas de bloqueados? ¿Por qué?
4. (0,5 puntos) ¿En un sistema monoprocesador pueden ejecutarse simultáneamente varios procesos? ¿Por qué?
5. (1 punto) ¿Cuál es el planificador que decide si se debe desalojar un proceso de la memoria principal debido a que el sistema está sobrecargado?
6. (1 punto) En sistemas multiprocesador fuertemente aplicados, ¿es conveniente que los procesadores compartan memoria principal? ¿Por qué?
7. (1 punto) Se dispone de los siguientes datos sobre los tiempos de llegada y ejecución de un conjunto de cuatro procesos por lotes (conocidos de antemano por el planificador):

|  |  |  |
| --- | --- | --- |
| **Proceso** | **Llegada** | **Servicio** |
| **A** | **0** | **5** |
| **B** | **1** | **3** |
| **C** | **2** | **4** |
| **D** | **4** | **2** |

Suponiendo la política de planificación a corto plazo SRT para un sistema monoprocesador:

1. Hacer diagrama.
2. ¿Cuántos cambios de contexto tendrán lugar? ¿Por qué?
3. (1 punto) En el caso de la pregunta anterior, si hubiera sido FIFO…
4. ¿Se hubiese producido el mismo número de cambios de contexto? ¿Por qué?
5. ¿Siempre sería así (mismo o diferente número de cambios de contextos) para cualquier conjunto de procesos? ¿Por qué?
6. (1 punto) En el momento de creación de un proceso hijo, ¿cuál, o cuáles, son las diferencias entre padre e hijo? ¿Y entre procesos hijos del mismo padre?
7. (2 puntos) Suponga que el siguiente programa nunca diera error:

main(){

int i=0, retorno;

for (i=0; i<2; i++) {

retorno=fork();

printf(“a”);

if (retorno>0) {

printf(“b”);

fork();

}

}

}

1. Explique razonadamente cuántos procesos son creados en total, SIN contar el proceso que inicia la ejecución del programa.
2. Dibuje la jerarquía de procesos que resulta de la ejecución del código anterior.
3. Muestre y explique cuál sería una posible salida del programa anterior.
4. ¿La salida anterior será siempre la misma? ¿Por qué?