

SISTEMAS OPERATIVOS

PRIMERA PRUEBA

DE

EVALUACIÓN A DISTANCIA

(PED1)

Curso 2012-2013

José Manuel Sáez Vicioso

DNI: 75765391 – X

Centro asociado de Cádiz



SISTEMAS OPERATIVOS

Primera Prueba de Evaluación a Distancia (PED1)

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

- I) (1 p) La estrategia de *prevención de interbloqueos* consiste en conceder a un proceso solamente aquellas peticiones de recursos que tengan garantizado que no conducirán a un estado de interbloqueo.
- II) (1 p) La planificación expropiativa produce un menor sobrecarga al sistema que una planificación no expropiativa.
- III) (1 p) Una de las principales ventajas que tiene implementar una determinada aplicación como uno o varios procesos multihilos es que permite aumentar el rendimiento del sistema.
- IV) (1 p) Un sistema operativo multiacceso debe ser capaz de soportar necesariamente multiprogramación.

Solución

- I) Falso. Esa es la estrategia de evitación de interbloqueos.

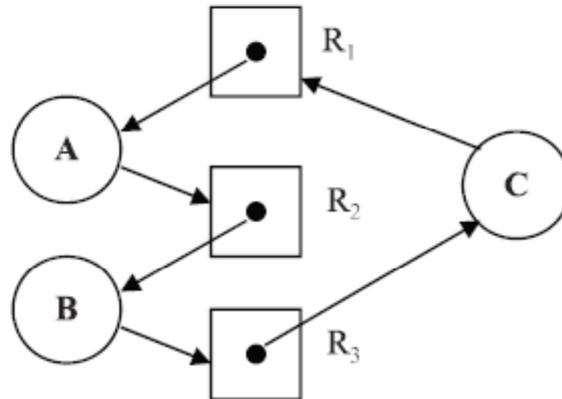
Lo correcto sería: “La estrategia de *evitación de interbloqueos* consiste en conceder a un proceso solamente aquellas peticiones de recursos que tengan garantizado que no conducirán a un estado de interbloqueo”.

- II) Falso. La planificación expropiativa produce una **mayor** sobrecarga al sistema que una planificación no expropiativa, ya que se ejecuta con mayor frecuencia y se realizan más cambios de proceso.

- III) Verdadero. La creación de un hilo nuevo dentro de un proceso ya existente requiere menos tiempo que la creación de un nuevo proceso. Además, se ahorran recursos, la comunicación es más eficiente, se consigue un mayor aprovechamiento de las arquitecturas multiprocesador y la simplificación de la estructura de las aplicaciones.

- IV) Falso. El multiacceso **no** requiere necesariamente la existencia de multiprogramación. Algunos sistemas informáticos dedicados al procesamiento de transacciones, como por ejemplo los sistemas de reserva de billetes, soportan centenares de terminales activos que son atendidos por un único programa.

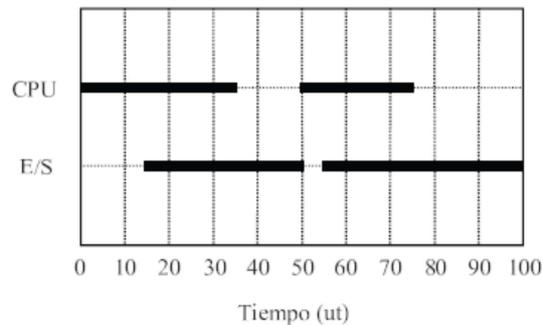
2. (2 p) Supóngase un sistema que permite retención y espera, exclusión mutua y expropiación en sus recursos. En un determinado instante de tiempo el grafo de asignación de los recursos R1, R2 y R3 del sistema a los procesos A, B y C es el que se muestra en la figura. Explicar **razonadamente** si se cumplen las condiciones para que se produzca una situación de *interbloqueo*.



Solución

En la figura tenemos el ciclo $R_1 \rightarrow A \rightarrow R_2 \rightarrow B \rightarrow R_3 \rightarrow C \rightarrow R_1$, pero no es condición suficiente para que exista interbloqueo. Por otra parte, los recursos tienen una sola instancia. Por tanto, se produce interbloqueo. El proceso A espera por el recurso R_2 que está asignado al proceso B, el cual espera por el recurso R_3 que está asignado al proceso C. Y por su parte, C tiene asignado R_3 y está bloqueado en espera de que le asignen R_1 .

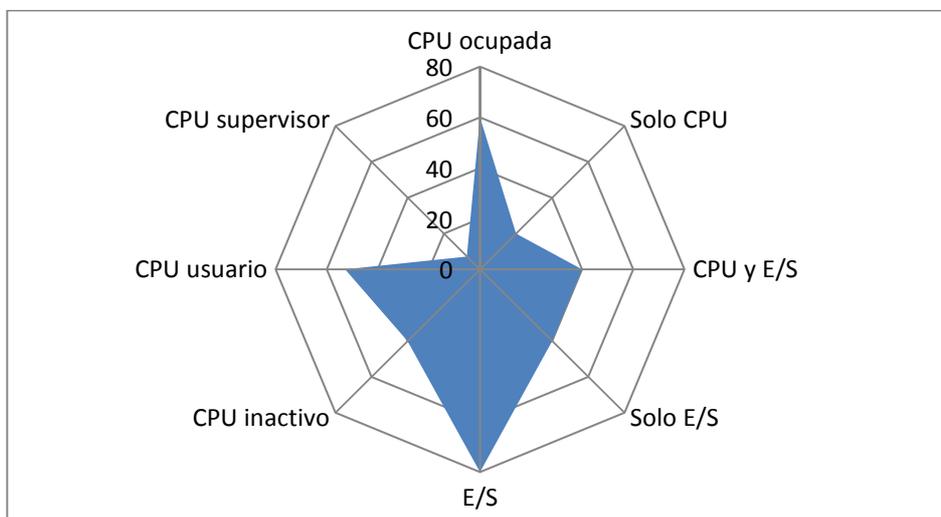
3. (2 p) Durante un cierto intervalo de observación de 100 ut, la CPU y los dispositivos de E/S de un computador han sido utilizados por el sistema operativo y varios procesos de usuarios de acuerdo con el diagrama de uso que se muestra en la figura. Dibujar y comentar el diagrama de Kiviatt - Kent asociado a este sistema informático. Suponer que el tiempo de ejecución en modo supervisor ha sido el 12% del tiempo total de uso de la CPU.



Solución

- La CPU ha sido utilizada durante 60ut. Luego $CPU\ ocupada = 60\%$.
- La CPU ha estado en modo supervisor durante 12% del tiempo total de uso de la CPU (60%). Luego $CPU\ supervisor = 7.2\%$.
- El tiempo que la CPU ha estado en modo usuario se obtiene restando el tiempo que ha estado en uso la CPU y el tiempo que ha estado ejecutándose en modo supervisor: $60 - 7.2 = 52.8$. Luego $CPU\ usuario = 52.8\%$.
- La CPU ha estado inactiva 40ut, luego $CPU\ inactiva = 40\%$.
- Los dispositivos de E/S han estado en uso durante 80ut, luego $E/S = 80\%$.
- Los dispositivos de E/S han estado en uso sin estarlo la CPU durante 40ut, luego $Solo\ E/S = 40\%$.
- La CPU ha estado en uso sin estarlo los dispositivos de E/S durante 20ut, luego $Solo\ CPU = 20\%$.
- La CPU y los dispositivos de E/S han estado en uso simultáneamente durante 40ut. Por lo tanto se tiene que $CPU\ y\ E/S = 40\%$.

Este diagrama se corresponde con la denominada flecha de entrada-salida, que quiere decir que el sistema está limitado por el subsistema de E/S aunque presenta una buena utilización de la CPU (en modo usuario y en modo supervisor).



4. (2 p) Dos procesos A y B se ejecutan concurrentemente en un determinado sistema. El proceso A ejecuta unas tareas ("Tareas 1") y alcanza un punto de encuentro. Posteriormente realiza otras tareas ("Tareas 2") y finaliza. Por su parte el proceso B ejecuta unas tareas ("Tareas 3") y llega al punto de encuentro. Posteriormente realiza otras tareas ("Tareas 4") y finaliza. El primer proceso que llega al punto de encuentro no puede continuar su ejecución hasta que no llegue el otro proceso. No se sabe qué proceso comienza a ejecutarse primero o cuál es el primero que termina. Escribir en pseudocódigo un programa de nombre *coordinación* que usando **semáforos binarios** coordine la actividad de los procesos A y B. Dicho programa debe tener cuatro partes: declaración de variables y semáforos, código del proceso A, código del proceso B y código para inicializar los semáforos y lanzar la ejecución concurrente de ambos procesos.

Solución

```
semáforo_binario S1,S2; /* Definición semáforos binarios */

void proceso_A() /* Proceso A */
{
    tareas_1();
    signal_sem(S2);
    wait_sem(S1);
    tareas_2();
}

void proceso_B() /* Proceso B */
{
    tareas_3();
    signal_sem(S1);
    wait_sem(S2);
    tareas_4();
}

main() /* Inicialización de semáforos y ejecución concurrente */
{
    Init_sem(S1,0);
    Init_sem(S2,0);
    ejecución_concurrente(proceso_A, proceso_B);
}
```