

## 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.

*FALSA, ya que la prevención de interbloqueos consiste en impedir que se produzca alguna de las cuatro condiciones necesarias para que se produzca el interbloqueo.*

- II) (1 p) La planificación expropiativa produce un menor sobrecarga al sistema que una planificación no expropiativa.

*FALSA, ya que la planificación expropiativa produce una mayor sobrecarga al sistema que la planificación no expropiativa, ya que el planificador se ejecuta más veces y se realizan más cambios de procesos. Además, la planificación expropiativa requiere que el núcleo del sistema operativo tenga mecanismos de sincronización para evitar la posible corrupción de las estructuras de datos.*

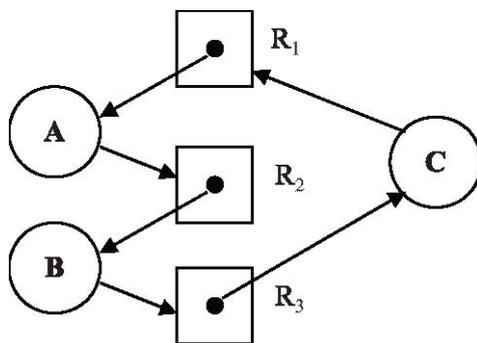
- III) (1 p) Una de las principales ventajas que tiene implementar una determinada aplicación como uno o varios procesos multihilos es que permite aumentar el rendimiento del sistema.

*VERDADERA, ya que crear un hilo dentro de un proceso ya existente requiere menos tiempo que crear un proceso nuevo. También se tarda menos en finalizar un hilo que un proceso. A su vez, un cambio de proceso requiere más tiempo que un cambio de hilo dentro de un mismo proceso.*

- IV) (1 p) Un sistema operativo multiacceso debe ser capaz de soportar necesariamente multiprogramación.

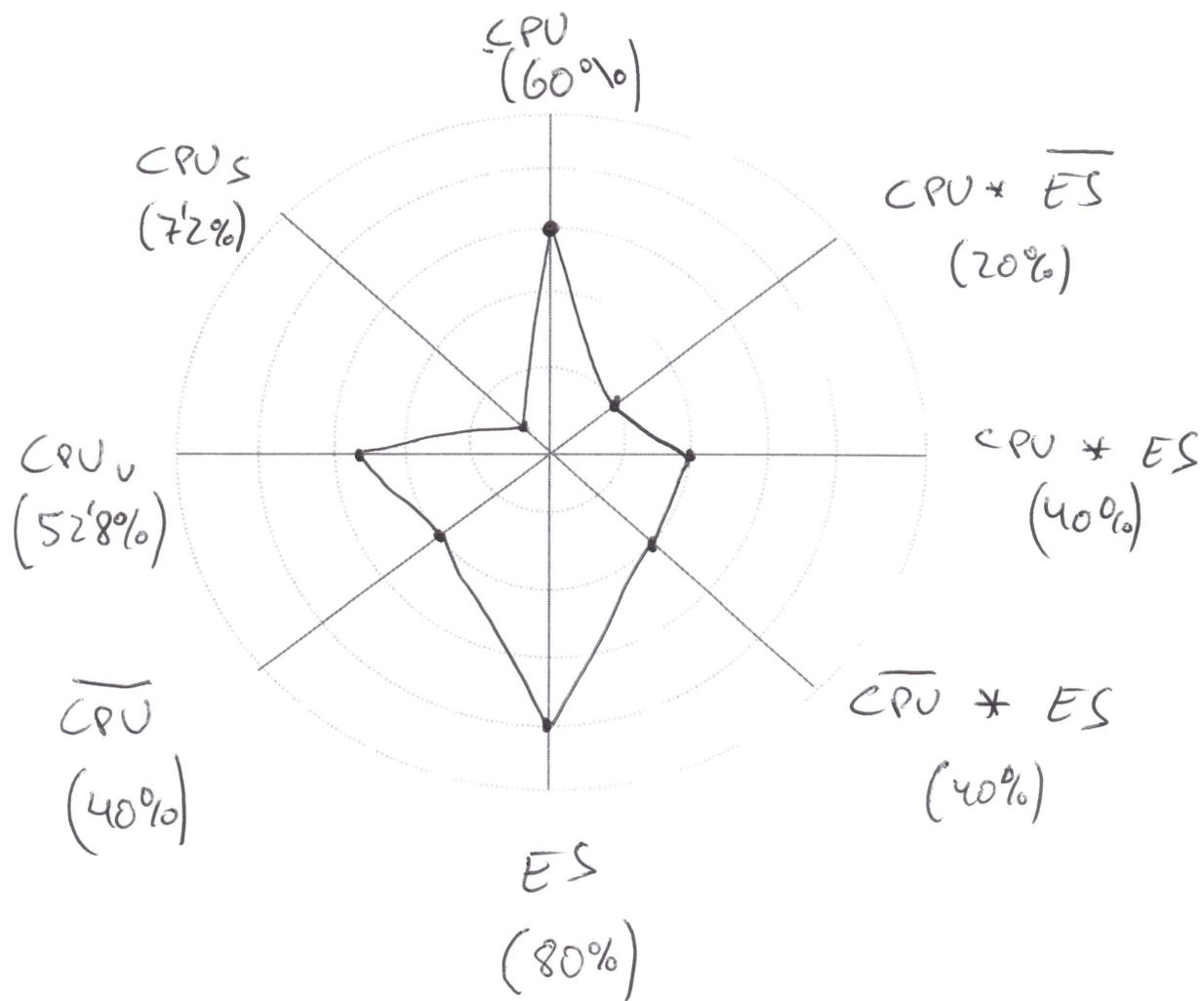
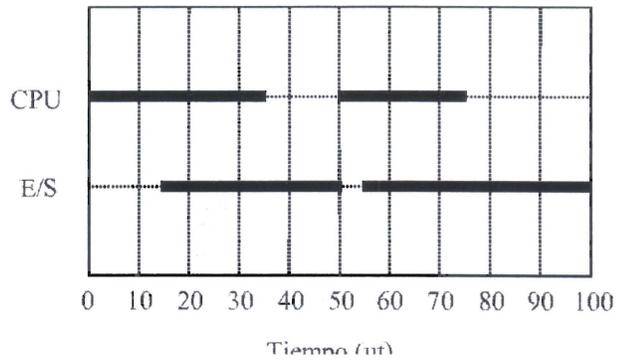
*FALSA. Un sistema operativo multiacceso es aquel que permite el acceso al sistema informático a través de dos o más terminales. No requiere necesariamente multiprogramación.*

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*.



No se puede producir el interbloqueo dado que la condición necesaria de la **no existencia de expropiación** no se cumple.

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.



- La CPU se ha sido utilizado durante 60 ut (CPU = 60%).
- La CPU en modo supervisor el 12% del uso de la CPU, por lo que es 7,2 ut (CPU s = 7,2%)
- La CPU en modo usuario  $60 - 7,2 = 52,8$  ut (CPU u = 52,8%)
- La CPU ha estado inactiva 40 ut (CPU = 40%)
- La CPU y los dispositivos E/S han estado en uso simultáneamente 40 ut (CPU\*E/S = 40%)
- La CPU ha estado en uso sin estarlo los dispositivos E/S 20 ut (CPU\*E/S = 20%)
- Los dispositivos de E/S han estado en uso sin hacer lo la CPU 40 ut (CPU\*E/S = 40%)
- Los dispositivos de E/S han estado en uso 80 ut (E/S = 80%)

Este es el diagrama de Kiviatt-Kent de este sistema informático. Para analizarlo lo comparamos con la “Estrella de Kiviatt” que es el diagrama de Kiviatt-Kent que debería representar un sistema informático ideal. En esta ejercicio se observa que el uso de la CPU y los dispositivos E/S es correcta, pero para mejorar el rendimiento del sistema habría que aumentar el uso de la CPU simultáneamente con los dispositivos de E/S, así como aumentar el uso de la CPU en modo usuario.

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.

PROGRAMA *coordinacion*;

semaforo\_binario SA,SB;

void procesoA()

```
{
    tareas 1;
    signal_sem(SA);
    wait_sem(SB);
    tareas 2;
}
```

void procesoB()

```
{
    tareas 3;
    signal_sem(SB);
    wait_sem(SA);
    tareas 4;
}
```

void main()

```
{
    init_sem(SA,0);
    init_sem(SB,0);
    ejecucion_concurrente(procesoA,procesoB);
}
```