

Ejercicio Evaluación II Sistemas Operativos

Concurrencia

Problema 1

Se define un programa con un vector y dos hilos de trabajo que acceden concurrentemente al vector. Teniendo en cuenta que el acceso al vector debe estar protegido con semáforos, se pide escribir el código que implementa las funciones suma y reset.

```
int vector[10];
semafor s;
main() {
    int i;
    sem_init(&s, 1);
    crear_flux(suma);
    crear_flux(reset);
    esperaFinalFluxos();
}

// suma el valor dels elements del vector
int suma() {
    int valor;

    return valor;

}

// inicialitza el contingut del vector a zeros
void reset() {

}
```

Problema 2

En una tienda de pájaros están teniendo problemas para tener a todos sus canarios felices. Los canarios comparten una jaula en la que hay un plato con alpiste y un columpio para hacer ejercicio. Todos los canarios quieren inicialmente comer del plato y después, columpiarse. Pero se encuentran con el inconveniente de que solo tres de ellos pueden comer del plato al mismo tiempo y solo uno puede columpiarse. Defina un proceso que ejecuten los canarios concurrentemente de forma que sincronicen sus actividades empleando semáforos.

Gestión Memoria

Problema 3

Se tiene un sistema de gestión de memoria contigua que actualmente tiene esta lista de huecos libres, cuyos tamaños son 1.800 Kb, 1.000 Kb, 900 Kb, 1.200 Kb. La lista está ordenada tal y como se ha escrito aquí.

Llegan al sistema varias peticiones de memoria, en este orden: 1.200 Kb, 1.000 Kb, 900 Kb.

- a) ¿Cómo atenderán estas peticiones las políticas de "primer ajuste" (*first-fit*) y "mejor ajuste" (*best-fit*)?
- b) ¿Cuál de las dos políticas aprovecha mejor la memoria disponible?

**Problema 4

Supongamos un sistema paginado de dos niveles, que utiliza paginación por demanda. Se conocen los siguientes datos del sistema:

- tiempo de acceso a memoria principal: 80 nseg.
- tiempo de acceso a la **TLB** (registros asociativos): 10 nseg.
- porcentaje de aciertos de la **TLB**: 95 %
- tiempo para resolver un fallo de página: 20 mseg.

Se quiere saber:

- a) Si no hay fallos de página, ¿cuál es, en promedio, el tiempo efectivo de acceso a memoria?
- b) Si la tasa de fallos de página es de uno entre un millón de accesos, ¿cuál es el tiempo efectivo de acceso a memoria?
- c) ¿Qué tasa de fallos de página ha de darse para que el tiempo efectivo de acceso a memoria sea inferior a 100 nseg?