

Sistemas Operativos

I. T. Informática de Sistemas

Córdoba, 17 de diciembre de 2003

Cuestiones

- (1) Proponga una implementación de semáforos en pseudocódigo en cada uno de los siguientes casos:
 - Implementación de semáforos generales utilizando deshabilitación de interrupciones.
 - Implementación de semáforos con espera activa utilizando la instrucción TSL.
- (1) Sea la solución al problema del productor/consumidor con búfer limitado siguiente:

```
void
Productor ()
{
    extern SEMAFORO pmutex, full, empty;
    T dato;

    while (TRUE)
    {
        ProducirDato (dato);
        wait (empty);
        wait (pmutex);
        EntrarDato (dato);
        signal (pmutex);
        signal (full);
    }
}
```

```
void
Consumidor ()
{
    extern SEMAFORO cmutex, full, empty;
    T dato;

    while (TRUE)
    {
        wait (full);
        wait (cmutex);
        SacarDato (dato);
        signal (cmutex);
        signal (empty);
        ConsumirDato (dato);
    }
}
```

- a) ¿Qué ventaja tiene utilizar dos semáforos diferentes, **cmutex** y **pmutex**, para la exclusión mutua de productores y consumidores?
- b) Indique cómo habría que modificar esta solución si el tamaño del búfer se considera infinito.
3. (1) Supóngase que el *principio de localidad* no existiera, y los accesos a memoria de un proceso se distribuyeran de forma aleatoria en todo su espacio de memoria, ¿cómo afectaría este hecho a los algoritmos de reemplazo de páginas en un sistema de memoria virtual?
4. (1) Un algoritmo de planificación por prioridades asigna a cada proceso i una prioridad, p_i dada por la fórmula:

$$p_i = \alpha \frac{1}{t_w + 1} + (1 - \alpha)t_e, \quad (1)$$

donde t_w es el tiempo de espera del proceso hasta el instante actual, t_e es la duración estimada de la próxima ráfaga y α , $0 \leq \alpha \leq 1$, es un parámetro que fija el administrador del sistema. La prioridad mayor es la numéricamente menor. Explique el funcionamiento del algoritmo dependiendo del valor de α .

5. (1) Indique las diferencias entre las variables condición y los semáforos. Ponga un ejemplo breve de su diferente uso en alguno de los problemas clásicos de programación concurrente.

Problemas

1. (1.5) Supóngase en un sistema de cómputo la siguiente secuencia de procesos con los instantes de llegada y tiempos de espera que se indican:

Proceso	Instante de llegada	Tiempo de ráfaga
1	0	10 ms
2	4	13 ms
3	9	20 ms
4	12	5 ms
5	18	16 ms

Considere ahora el siguiente algoritmo de planificación. Los procesos en el sistema se ejecutan según un algoritmo por prioridades apropiativo. La prioridad de todos los procesos cuando llegan al sistema es 0, los procesos que están en espera aumentan su prioridad a razón de $\alpha = 2$ cada unidad de tiempo, el proceso que está en ejecución aumenta su prioridad a razón de $\beta = 1$ cada unidad de tiempo. La prioridad mayor es la numéricamente mayor. Cuando un proceso en espera iguala o supera la prioridad del proceso en ejecución este es apropiado y comienza su ejecución el proceso más prioritario.

- a) Realice la planificación de la tabla de procesos anterior con este algoritmo. Indique el diagrama de Gant y el tiempo medio de espera.
- b) Explique como se comportaría el algoritmo, sin obtener de nuevo el diagrama de Gant, para los casos:
- 1) $\beta > \alpha > 0$.
 - 2) $\alpha > 0, \beta = 0$.
 - 3) $\beta = \alpha$.

2. (2) Supóngase el siguiente problema de sincronización:

Parque Jurásico consiste en un museo de dinosaurios y un parque para un paseo safari. Hay m visitantes al parque que desean hacer el paseo y n coches para realizarlo, cada uno de ellos con capacidad para 2 pasajeros. Los pasajeros deambulan durante un rato por el museo y luego hacen cola en la parada de los coches. Cuando un coche está disponible carga a sus 2 pasajeros y circula por el parque una cantidad de tiempo aleatoria. Si todos los coches están en uso, los pasajeros que desean usarlos esperan en la cola hasta que haya un coche disponible. Si el coche está disponible pero no hay pasajeros en espera el coche espera hasta que alguien desee subir. Si un coche está disponible y sólo hay un pasajero dispuesto a subir ambos deben esperar hasta que haya un segundo pasajero.

- a) Identifique los problemas de programación concurrente que pueden aparecer en el sistema.
 - b) Describa los principios generales de la solución que propondría a este problema (sin indicar el pseudocódigo de la solución).
 - c) Implemente en pseudocódigo una posible solución al problema.
3. (1.5) Dada la siguiente secuencia de referencias a memoria, de lectura y de escritura, y una memoria real de 4 marcos de página y una memoria virtual de 8 páginas:

2 1 3 2 4 5 2 3 4 7 6 1 1 6 5 4 0 1 2 7 1 1 3 4 0 2 4
E E L L L E L E E L L E L L E E L E L E L L E E L E L

Supóngase que modificamos el algoritmo de reemplazo FIFO con segunda oportunidad, o de reloj, añadiéndole un bit de modificación, que se pone a 1 cuando la página es accedida para escritura, además del bit de referencia usual. A continuación el algoritmo funciona de la siguiente forma:

- a) Comenzando en la cabeza actual de la cola FIFO se recorre toda la cola sin modificar ningún bit, ni el de modificado ni el de reemplazo. El primero marco que tenga ambos bits a 0 es reemplazado.
- b) Si en el paso anterior ninguna página cumple el criterio se realiza una segunda pasada hasta encontrar una página con el bit de referencia a 0. En esta segunda pasada los bits de referencia se van poniendo a 0 a medida que se pasan las páginas.
- c) Si el paso anterior también falla todas las páginas tendrán su bit de referencia a 0. Se repite entonces el paso 1, y el dos si fuera necesario, que en esta segunda ocasión obtendrán una página a reemplazar con seguridad.

Explique las ventajas de este algoritmo respecto al FIFO con segunda oportunidad estándar.

Aplice este algoritmo a la cadena de referencias anterior obteniendo el número de fallos de página.

Tiempo de realización: **4 horas**. Valoración aproximada de cada pregunta entre paréntesis.