

CC41B Sistemas Operativos
Examen – Semestre Primavera 2007
Prof.: Luis Mateu

Pregunta 1

En un parque de diversiones la atracción principal es el tagada. Los turistas son representados por múltiples threads que ejecuta el código de más abajo a la izquierda. Cuando un turista desea divertirse en el tagada invoca **subirAlTagada** con su nombre. El tagada funciona llamando a **lanzarTagada** que recibe un arreglo de turistas y el número de turistas. Este procedimiento recibe un máximo de 10 turistas y tarda unos 3 minutos, tiempo durante el cual los turistas mencionados se divierten en el tagada.

La implementación actual de **subirAlTagada** aparece abajo a la derecha y sólo atiende a un turista a la vez, lo cual es ineficiente.

```

void turista(char *nombre) {
    ...
    subirAlTagada(nombre);
    ...
}

monitor ctrl;
void subirAlTagada(char *nombre) {
    char *grupo[10];
    grupo[0]= nombre;
    nEnter(ctrl);
    lanzarTagada(grupo, 1);
    nExit(ctrl);
}
    
```

Re programe el procedimiento **subirAlTagada** de modo que el tagada se use eficientemente de acuerdo a la siguiente estrategia. El tagada está detenido si no hay turistas. Cuando llega un turista, Ud. lanza el tagada de inmediato con el turista que llegó. Al terminar puede ocurrir que (i) no hay turistas, en cuyo caso el tagada permanece detenido, o bien (ii) hay turistas esperando subirse al tagada. Para (ii) Ud. debe subir tantos turistas como pueda hasta un máximo de 10 y relanzar el tagada. Los demás turistas si los hay permanecen en espera. La atención de los turistas debe realizarse por orden de llegada. La implementación debe hacerse con los monitores de nSystem. El procedimiento **lanzarTagada** es dado.

Hint: Use un thread adicional que se encargue de operar el tagada. Además utilice una cola de tipo `FifoQueue` para almacenar requerimientos de uso del tagada.

Pregunta 2

Parte a.- Una máquina Unix corre varios procesos intensivos en CPU y memoria, pero *no en E/S*. Durante la última media hora Ud. observa que el LED de acceso al disco se enciende permanentemente, el comando `top` se demora mucho e indica persistentemente que el porcentaje de uso de la CPU es 1%. Ninguno de los procesos parece avanzar. Conteste (i) ¿qué está sucediendo? (ii) indique qué estrategia de reemplazo de páginas utiliza el sistema operativo, (iii) señale qué haría Ud. como administrador de la

máquina, (iv) ¿en qué estrategia de reemplazo de páginas esto no puede ocurrir y por qué?

Parte b.- El siguiente diagrama muestra (con una x) los accesos a la memoria para un proceso que se ejecuta en un sistema Unix que utiliza la estrategia del *working set*.

página	6	x x x	x	x	x		x	
5		x x			x			
4	x x	x x				x	x x	x x
3		x	x		x x x	x x x	x	
2	x x			x x		x	x	
1						x x x		
0								
		A	B	C	D	E	F	G

Tiempo

La letras A, B, C, etc. denotan los intervalos para los que se calcula el *working set*. Los números 0, 1, 2, etc. denotan las páginas del proceso. Cada x representa una lectura o escritura en la página. Conteste:

- i. Indique para los períodos C a F qué accesos pueden producir page-faults. Utilice coordenadas del estilo (A, 4, 1er. acceso).
- ii. Indique el valor del atributo *Referencia* para todas las páginas al inicio del intervalo E y al final de ese intervalo.
- iii. Considere que el computador en donde se ejecuta este proceso posee solo 5 páginas reales. Explique en qué momento la estrategia del *working set* no funcionará adecuadamente.

Pregunta 3

Parte a.- Suponga que en un instante dado Ud. necesita ejecutar 4 procesos independientes entre sí en una máquina *quad-core*. Discuta si lanzaría los procesos simultáneamente, secuencialmente o en algún orden específico considerando que (i) todos los procesos son intensivos en CPU, pero no en E/S, (ii) todos los procesos son intensivos en E/S, pero no en CPU, hay un solo disco, (iii) 2 procesos son solo intensivos en E/S y los otros 2 procesos son solo intensivos en CPU, hay un solo disco.

Parte b.- En un sistema Unix un archivo tiene un tamaño de $(12+256)*512+1$ bytes. Haga 2 diagramas que muestren inodo, bloques de datos y bloques de indirección suponiendo que el tamaño de los bloques de la partición es (i) 512 bytes y (ii) 4096 bytes. Supongo que un programa usa `lseek` para posicionarse en el último byte del archivo y lo lee. Explique para los casos (i) y (ii) cuantos bloques será necesario leer. Supongo que el cache de disco está completamente vacío.

Parte c.- Suponga que un sistema operativo tiene pendiente lecturas de los bloques 10, 45, 90. El scheduler de disco elige leer el bloque 45. Mientras lee este bloque 2 procesos solicitan al scheduler de disco escribir los bloques 40 y 55. En qué orden se leerán todos los bloques pendientes suponiendo que la estrategia de scheduling de disco es (i) *shortest seek first* y (ii) método del ascensor. Explique.