

Sistemas Operativos – OSTEP

Ejercicio 1.

- (a) Indique que valores puede tomar x para el programa de la Figura 1, donde la atomicidad es línea-a-línea. Explique. Notar que el multiprograma no termina.
- (b) Exprese todos los valores posibles que puede tomar el programa de la Figura 1 si las asignaciones no son atómicas, o sea $a_i = x; a_i = a_i + 1; x = a_i$, con i el número de proceso.
- (c) ¿Qué se puede asegurar sobre el valor de x entre los incrementos y decrementos, si la atomicidad es línea a línea? Explique.

Pre: $x = 0$		
1 P0: while (1) { 2 $x = x + 1;$ 3 $x = x - 1;$ }	a P1: while (1) { b $x = x + 1;$ c $x = x - 1;$ }	x P2: while (1) { y $x = x + 1;$ z $x = x - 1;$ }

Figura 1:

Ejercicio 2. Utilice semáforos para sincronizar los procesos como lo indica el grafo de sincronización de la Fig. 2.

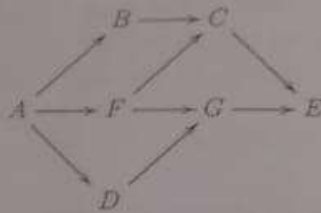


Figura 2:

Init	
P0: while (true) { NCS0 enterCS() CS0 leaveCS() }	P1: while (true) { NCS1 enterCS() CS1 leaveCS() }

Figura 3:

Ejercicio 3.

- (a) Mostrar el código de entrada `enterCS()`, de salida `leaveCS()` y de inicialización `Init` del problema de la región crítica implementada con Spinlocks. (Ver Figura 3.)
- (b) Explicar porque vale la condición de seguridad: "Hay a lo más un proceso dentro de la CS0, CS1".
- (c) Dar dos (2) ventajas y dos (2) desventajas de esta implementación de problema de la región crítica con respecto a la implementación con el algoritmo de Peterson.

Ejercicio 4. Para la tabla de procesos de la Figura 4 realice el diagrama de planificación para RR(Q=2).

Proceso	Arribo	Uso CPU	Demora E/S	Uso CPU
A	0	2	8	2
B	0	7	1	7
C	0	3	4	3

Figura 4:

Ejercicio 5. Determine si las siguientes proposiciones son verdaderas o falsas justificando su respuesta:

- (a) En un sistema operativo donde periódicamente se mata algún proceso al azar no se pueden producir deadlocks.
- (b) Existe una ejecución del sistema de la Fig. 5 lleva a deadlock.

P0: acquire(A);	P1: acquire(A);	P2: acquire(B);
acquire(B);	acquire(C);	acquire(C);
release(A);	release(A);	release(C);
release(B)	acquire(B);	release(B)
	release(B);	
	release(C)	

Figura 5:

Ejercicio 6. Dado el sistema de paginado de dos niveles del i386: direcciones virtuales de 32 bits, direcciones físicas de 32 bits, 10 bits de índice de *page directory*, 10 bits de índice de *table directory*, y 12 bits de *offset* dentro de la página. Mostrar el directorio y las tablas de página para el siguiente mapeo no inyectivo de virtual a física que mapea en total 32 KiB de memoria. Dar los 20 bits o los 5 dígitos hexa con el número de marco físico.

Virtual	Física
[4092 KiB, 4108 KiB)	→ [0 KiB, 16 KiB)
[8188 KiB, 8204 KiB)	→ [16 KiB, 32 KiB)

Ejercicio 7. Se define un *page directory* donde la última entrada, la 1023, apunta a la base del mismo, Figura 6.

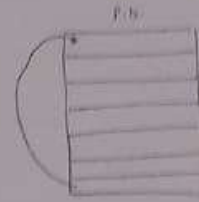


Figura 6: Page directory autoreferenciado en su última entrada

- (a) ¿A dónde apunta la dirección virtual 0xFFC00000?
- (b) ¿Y la dirección virtual 0xFFFE0000?
- (c) Indique a donde apunta la dirección virtual 0xFFFF0000.
- (d) Finalmente, describa para que sirve este esquema de memoria virtual.

Ejercicio 8. En un sistema de archivos de tipo UNIX, tenemos los bloques de disco dispuestos dentro del *i-nodo* con 8 bloques directos, 1 bloque indirecto y 1 bloque doble indirecto. Cada bloque es de 1 KiB y los números de bloque ocupan 16 bits.

- (a) Calcule la capacidad máxima de un archivo.
- (b) En este FS se está descargando un archivo MP3 por protocolo HTTP, y hasta el momento se tienen 530999 bytes de datos. Un segundo después llega un paquete TCP/IP de 1536 bytes para agregarlo al final del archivo. Explique cual era la situación de disposición del archivo y como cambia, en cuanto a todos los nuevos bloques que deben ser agregados.
- (c) ¿Es posible llegar a esta longitud máxima de archivo calculada anteriormente? Explique.

Ejercicio 9. Usted forma parte de un equipo al que le fue encargado el diseño de un sistema de archivos UNIX-like y uno de sus compañeros propuso la idea de aprovechar los 75 bytes que sobran en los *i-nodos* (Cada *i-nodo* ocupa 181 bytes y se decidió "alinearlo" a 256 bytes para que entraran 2 por bloque de disco) almacenando allí los archivos que no superen esta capacidad¹.

- (a) Discuta porque sí o porque no esta decisión de diseño es acertada, en cuanto a eficiencia, velocidad y simpleza. Soporte sus argumentos con las mediciones de la Figura 7².
- (b) ¿Tiene sentido que estos 75 bytes siempre se utilicen sin importar la longitud del archivo? Discuta.

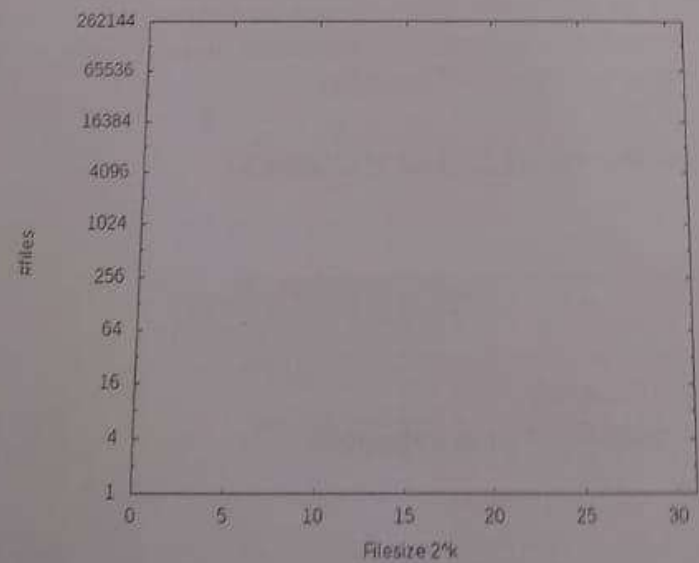


Figura 7: Distribución de tamaños de archivos.

Ejercicio 10. La persistencia o filesystem se divide en tres (3) capas. Identifíquelas y describa brevemente las funcionalidades de cada una.

¹Esto ya existe y se conoce como *immediate files*, tanto en MinixFS como en NTFS.
²La figura muestra la cantidad de archivos que poseen determinado tamaño en todo el disco duro el día 20141214. Los estándares