



Nota:

Apellido y Nombre	Profesor	Tomé conocimiento de la nota: (Sólo aplazos)

Preguntas teóricas					Ejercicios	
1	2	3	4	5	1	2

Contexto

En un Sistema Operativo que corre sobre una PC con un procesador de un core, se realizan diferentes tipos de pruebas. Para esto, se utiliza un proceso de **Benchmark con 3 KLTs (A, B y C)**, que corren variando algunos parámetros, como por ejemplo el algoritmo de planificación de procesos, la cantidad de frames asignados, etc. La PC en cuestión, utiliza un **disco de estado sólido ultra rápido para swap (que permite realizar entrada-salida en paralelo)**, y un disco rígido tradicional con **dos particiones**, una para el FileSystem Unix donde se aloja el Sistema Operativo y otra FAT32, para archivos y programas.

Al ejecutar el Benchmark, con un determinado algoritmo, se produce la ejecución que se muestra en el siguiente gantt:

A											E/S	E/S	E/S	E/S	E/S					
B					E/S	E/S	E/S	E/S	E/S											
C													E/S	E/S	E/S	E/S	E/S			
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	

Cada KLT utiliza **una y sólo una** determinada página durante el tiempo indicado posteriormente entre paréntesis. Cuando la página necesaria no se encuentre en memoria principal, ocurrirá un fallo de página, el cual siempre tomará 5 ut en ser realizado, siendo estas las únicas entrada-salidas realizadas. Los tiempos de cada KLT son:

- KLTA -> P1 (5 ut) -> P2 (2ut)
- KLTB -> P5 (1 ut) -> P8 (3ut) -> P5 (2ut)
- KLTC -> P3 (4 ut) -> P1 (1ut)

	T=0	T=9	T=14	T=17
Frame 1	P1	P1	P2	P2
Frame 2	P5	P5	P5	P5
Frame 3	P3	P3	P3	P1
Frame 4	-	P8	P8	P8

Nota: Se resaltan los fallos de página

Se utiliza además una TLB de 3 entradas.

A) Teoría: Explícitamente defina como VERDADERA o FALSA cada una de estas afirmaciones justificando brevemente.

- 1) Al ejecutar por consola “./benchmark”, desde el directorio raíz del FS Unix, se corre correctamente el Benchmark. Esto no sería posible si el programa estuviese en la partición FAT32.
- 2) El proceso “Benchmark” podría generar el mismo gantt si los hilos fuesen ULTs, utilizando una biblioteca de hilos que tenga algún algoritmo apropiado.
- 3) Con una TLB de 4 entradas se generarían menos fallos de página
- 4) Dado que esta PC tiene un único procesador, no es recomendable que los semáforos del Sistema Operativo realicen espera activa
- 5) Cada operación de E/S asociada a un fallo de página se realiza utilizando una entrada-salida bloqueante

B) Práctica: Resuelva los ejercicios justificando las respuestas

1) Sabiendo que los KLTs no utilizan prioridades entre sí, y que llegan a la cola de listos en el orden A, B, C

- Indique qué algoritmo/s de planificación podrían haber sido utilizado/s. Justifique indicando **al menos 2 instantes** donde se aplica/n. No justifique realizando nuevamente el gantt.
- Indique todos los cambios de modo provocados por **interrupciones de hardware**.

2) El proceso Benchmark utiliza 4 marcos, con asignación fija y reemplazo local. Además, el Sistema Operativo utiliza la técnica de buffering de páginas, y esto permite que en este sistema **se corra el algoritmo de reemplazo una vez finalizada la E/S del fallo de página**:

- Indique el/los algoritmo/s de reemplazo que podrían haber sido utilizado/s. Justifique explicando que es lo que ocurre para que esto sea así
- Indique cual es la menor cantidad de fallos de página que podría causar esta ejecución.



RESPUESTAS

TEORÍA

- 1) Falso. Si “./benchmark” fuera un symbolic link apuntando al path absoluto del ejecutable en la otra particion, entonces sería posible ejecutar el programa
- 2) Falso. Los ULTs deberían bloquearse ante cada entrada-salida
- 3) Falso. La TLB no disminuye la cantidad de fallos de página. Disminuye los accesos a memoria a la hora de traducir páginas.
- 4) Verdadero. La espera activa generaría demasiado overhead (a diferencia de un sistema multiprocesador donde otros procesos podrían ejecutar mientras en una cpu se ocupa una región crítica). Para un caso así, se podrían deshabilitar las interrupciones, dado que suele ser menos costoso porque solo hay un procesador..
- 5) Verdadero. Es necesario que el KLT en cuestión no continúe hasta que la página esté cargada en memoria principal, dado que en su ejecución da por sentado que todas las páginas que necesita estarán en memoria

1. En los instantes 0 a 3 y 4 a 7 se ve claramente la presencia de un quantum = 3. En 9, se da prioridad a el KLT B, que vuelve de entrada-salida, pero solamente se asignan dos unidades de tiempo. Estamos en presencia de un Virtual Round Robin.

2. El algoritmo correcto podría ser LRU o bien Clock.

- Para LRU
 - En T=14 -> P1 fue utilizada hasta T=9, P5 esta en uso, P3 fue utilizada hasta T=12 y P8 hasta T=13
 - En T=17 -> El mismo criterio del instante 14.
- Para Clock
 - En T=14 -> Todos los bits de uso se bajan a cero. Se reemplaza P1. Para que esto sea posible, el puntero debio estar apuntando a P1. El puntero ahora apunta a P5.
 - En T=17 -> P5 fue utilizada en T=14, el bit de uso queda en 1. P2 fue utilizada en T=15, el bit de uso queda en uno. Al correr el algoritmo, se baja el bit de uso de P5 y se reemplaza P3 (con el bit de uso en cero).

Los otros algoritmos pueden ser descartados:

- FIFO
 - En T=17 debería reemplazar a P5
- Clock modificado
 - Se desconocen las escrituras realizadas.