

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
FACULTAD DE CIENCIAS E INGENIERÍA**

**SISTEMAS OPERATIVOS**

**3ra práctica (tipo a)  
(Segundo semestre de 2019)**

Horario 0781: prof. V. Khlebnikov  
Horario 0782: prof. F. Solari A.

Duración: 1 h. 50 min.

Nota: No se puede usar ningún material de consulta.

**La presentación, la ortografía y la gramática influirán en la calificación.**

**La práctica debe ser desarrollada en el cuadernillo usando lapicero.**

**Lo escrito con lápiz NO será evaluado.**

Puntaje total: 20 puntos

---

**Pregunta 1 (2 puntos – 10 min.)**

**a) (1 punto – 5 min.)** ¿Por qué es deseable contar con soporte de reubicación en el *hardware* del CPU? Dé un ejemplo que considere algo de código y acceso a datos.

**b) (1 punto – 5 min.)** Explique en qué consisten la fragmentación externa y la fragmentación interna.

**Pregunta 2 (4 puntos – 20 min.)** Considere un computador con una memoria real de 1 MiByte, y un sistema que administra dicha memoria dinámicamente con listas de espacios libres (huecos). La unidad de asignación es de 1 KiByte. El sistema base ocupa los primeros 128 KiBytes, y luego son cargados programas que ocupan 68 KiB, 130 KiB, 280 KiB, 62 KiB, 45 KiB, y 180 KiB en este orden. Luego se quiere cargar otro programa de 300 KiB.

**a) (1 punto – 5 min.)** Represente el estado de la memoria luego de la carga de los primeros programas mediante una lista de espacio libre y, si lo necesita, un esquema de las posiciones de memoria en sí.

Cada proceso tiene límites de los recursos del sistema. En la línea 21 se consulta el límite de la pila que se asigna a cada proceso. Se puede ver la pila es de 8 MB. Entonces, crear un arreglo grande dentro de la pila no es una buena idea. Mejor sería usar el segmento de datos o el segmento *heap* que puede crecer. Pero, en nuestro caso, usaremos la pila para el arreglo. Si declaramos el arreglo de 8 MB, obtendremos el error *Segmentation fault* durante la ejecución del programa porque otras variables también necesitan un espacio en la pila. Por eso el tamaño del arreglo está establecido en (8MB – 16KB).

**b) (1 punto – 5 min.)** Si termina el programa de 280K y el de 45K, ¿ya se contaría con suficiente memoria para cargar el nuevo programa de 300 KiB? Explique su respuesta, representando el estado de la memoria en la lista.

**c) (2 puntos – 10 min.)** Al liberarse el espacio del programa de 62 KiB, luego de lo ocurrido en b), ¿ya se contaría con suficiente memoria para cargar el nuevo programa de 300 KiB? Explique su respuesta, representando en secuencia el estado de la memoria en la lista durante: liberación inicial, comprobación de contigüidad, condensación, y nueva asignación. Haga los cálculos respectivos.

**Pregunta 3 (6 puntos – 30 min.)**

a) (1 punto – 5 min.) Complete la figura.

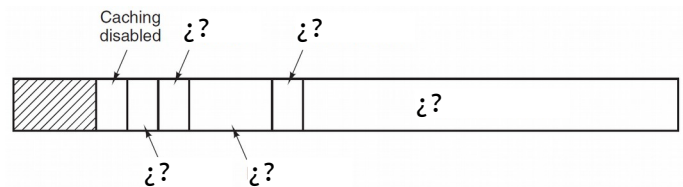


Figure 3-11. A typical page table entry.

b) (1 punto – 5 min.) Complete la siguiente frase: "... More detail about possible implementations. In any paging system, two major issues must be faced: ..."

c) (2 puntos – 10 min.) Suppose that a machine has 48-bit virtual addresses and 32-bit physical addresses. If pages are 4 KB, how many entries are in the page table if it has only a single level? Explain. Presente los valores finales con el prefijo adecuado: K, M, ...

d) (2 puntos – 10 min.) Suppose that a machine has 38-bit virtual addresses and 32-bit physical addresses. What is the main advantage of a multilevel page table over a single-level one? (1 punto) With a two-level page table, 16-KB pages, and 4-byte entries, how many bits should be allocated for the top-level page table field and how many for the nextlevel page table field? Explain. (1 punto más)

**Pregunta 4 (6 puntos – 30 min.)** Below is an execution trace of a program fragment for a computer with 512-byte pages. The program is located at address 0x5fc, and its stack pointer at 0x2200 (the stack grows toward 0 before every push operation).

a) (4 puntos – 20 min.) Give the page reference string generated by this program. Each instruction occupies 4 bytes (1 word) including immediate (instruction inside) constants. Both instruction and data references count in the reference string.

Load word 0x1a00 into register 0  
Push register 0 onto the stack  
Call a procedure at 0x1600, stacking the return address  
Subtract the immediate constant 0x10 from the stack pointer  
Compare the actual parameter (in the stack) to the immediate constant 4  
Jump if equal to 0x1620

b) (1 punto – 5 min.) Si ninguna página de este programa está cargada en la memoria, ¿cuántos fallos de página sucederían para 3 marcos con el algoritmo óptimo?

c) (1 punto – 5 min.) Si ninguna página de este programa está cargada en la memoria, ¿cuántos fallos de página sucederían para 3 marcos con el algoritmo LRU?

**Pregunta 5 (2 puntos – 10 min.)** Un sistema de archivos tipo Unix, que utilizan índices, o nodos-i, o también llamados *i-nodes*, tiene una estructura de *i-node* de 256 bytes, de los cuales se utilizan 40 bytes para índices directos de 32 bits, 4 bytes para el índice simplemente indirecto y 4 bytes para el índice doble indirecto, reservando algunos bytes para lo que podría usarse como índice triple indirecto.

Si cada bloque es de 1 kibi-Byte, muestre como se asignarían bloques a un archivo de entre 9 y 10 kibi-bytes (1 punto), y a uno de poco más de 10 kibi-bytes y menos de 11 kibi-bytes (1 punto más). Considere bloques libres a partir del bloque #8192d en cada caso.



La práctica ha sido preparada por FS(1,2,5) y VK(3,4)  
en Linux Mint 19.2 "Tina" con LibreOffice Writer

Profesores del curso: (0781) V. Khlebnikov  
(0782) F. Solari A.

Pando, 15 de noviembre de 2019