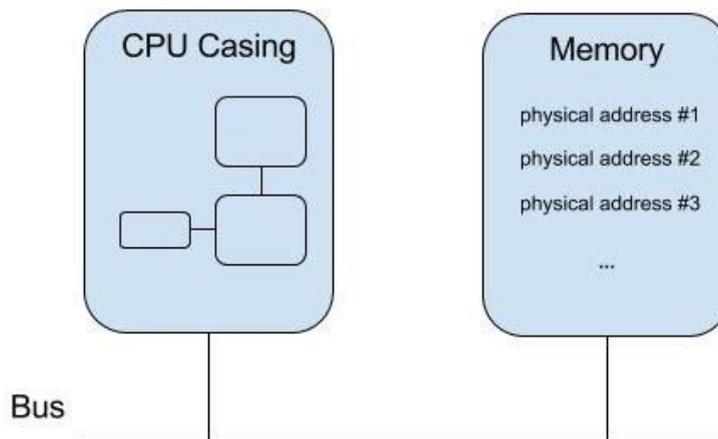




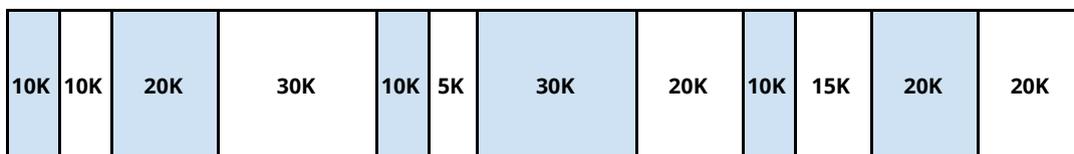
Práctico 6 Gestión de Memoria

6.1. ¿Cuál es la diferencia entre una dirección lógica y una dirección física? ¿Qué tipo de direcciones genera el CPU y la unidad de gestión de memoria (*Memory Management Unit, MMU*)? A partir de lo anterior, identificar la CPU, la MMU, el TLB (*Translation Lookaside Buffer*) y las direcciones en la siguiente imagen:



6.2. Describir el método de *standard swapping*. ¿Es utilizado en los sistemas operativos modernos? ¿Por qué? ¿Cuál es el tiempo de cambio de contexto asociado al *swapping* en un disco duro con una velocidad de transferencia de 2 MB/s si se intercambia un programa de 200 KB de tamaño? Asumir que el tiempo de búsqueda no es considerado y que la latencia promedio es de 15 ms. El resultado debe reflejar sólo la cantidad de tiempo necesario para hacer *swap out* del proceso.

6.3. Considerar el siguiente mapa de memoria, en el cual las áreas sombreadas corresponden a bloques utilizados:



Suponiendo que arriban tres requerimientos de 20 KB, 10 KB y 5 KB respectivamente, determinar cómo serán alocados al utilizar un esquema de asignación dinámica y las siguientes políticas:

- a) *First-fit* (primer-ajuste)
- b) *Best-fit* (mejor-ajuste)
- c) *Worst-fit* (peor-ajuste)
- d) *Next-fit* (siguiente-ajuste)

6.4. Dada la siguiente tabla de segmentos en un sistema que utiliza un esquema de segmentación simple:

Segmento	Base	Longitud (bytes)
0	660	248
1	1752	422
2	222	198
3	996	604

Para cada una de las siguientes direcciones lógicas, determinar su dirección física asociada.

- a) 0,198
- b) 2,156
- c) 1,530
- d) 3,444
- e) 0,222

6.5. Explicar el mecanismo básico de la implementación de paginación. ¿Por qué los tamaños de página son siempre potencia de 2?

6.6. Explicar la diferencia entre fragmentación externa y fragmentación interna.

- a) ¿Qué tipo de fragmentación sufre un esquema segmentado?
- b) ¿Qué tipo de fragmentación sufre un esquema paginado? Considerando un tamaño de página de 4 KB, determinar el nivel de fragmentación producido si un proceso requiere 135748 bytes de memoria. ¿A qué conclusión podemos llegar?

6.7. Considerar un espacio de direcciones lógicas de 64 páginas que consisten de 1.024 *words* cada una, mapeadas a una memoria física de 32 *frames*.

- a) ¿Cuántos bits hay en la dirección lógica?
- b) ¿Cuántos bits hay en la dirección física?

6.8. Describir como el TLB asiste en la traducción de una dirección lógica a una dirección física.

6.9. ¿Podría darse una situación en la cual las tablas de páginas de dos o más procesos referencien al mismo conjunto de *frames*? Justificar adecuadamente.

6.10. Un sistema utiliza una tabla de páginas de dos niveles, con un tamaño de página de 4 KB y un espacio de direccionado lógico de 4 GB. Los primeros 8 bits de la dirección generada por el CPU modelan el índice a la tabla de primer nivel. En este contexto, determinar:

- a) El número de bits que identifican el índice de segundo nivel.
- b) El número de entradas en el primer nivel de la tabla de páginas.
- c) El número de entradas en el segundo nivel de la tabla de páginas.
- d) El número de páginas en el espacio de direccionado virtual.

6.11. Dado un sistema de 32-bit con un tamaño de página de 4 KB:

- a) Considerar que el sistema utiliza una tabla de páginas de dos niveles, donde cada nivel tiene el tamaño de una página y cada entrada es de 4 bytes:
 - i) ¿Cuántas entradas tiene el primer nivel?
 - ii) ¿Cuántas entradas tiene el segundo nivel?
 - iii) Si un proceso fuera a utilizar los 4 GB de memoria virtual disponibles, ¿Cuántas tablas de página del segundo nivel estarían en uso?
 - iv) ¿Qué cantidad de memoria ocupa el primer y segundo nivel de la tabla de páginas?
- b) Considerar el mismo sistema pero con una tabla de páginas de un nivel:
 - i) ¿Cuántas entradas tiene la tabla de páginas?
 - ii) ¿Cuánta memoria ocupa la tabla de páginas?
- c) ¿Por qué un diseñador optaría por la tabla de páginas de dos niveles, incluso si la página de dos niveles completamente llena ocupa más que la de un nivel completamente llena?

6.12. Considerar un sistema paginado en el cual la tabla de páginas se almacena en memoria. Determinar el tiempo de acceso efectivo si:

- a) Cada referencia a memoria demora 150 ns.
- b) Al agregar registros asociativos se obtiene un hit-ratio de un 85 %. Tener en cuenta que la búsqueda asociativa agrega una sobrecarga de 5 ns.
- c) ¿A qué conclusión podemos arribar en función de los resultados obtenidos?

6.13. En un sistema con paginación un proceso no puede acceder a la memoria que no posee; ¿por qué? ¿Cómo podría el sistema operativo permitir el acceso a otra área de memoria?

6.14. ¿Cuál es la diferencia entre paginación simple y paginación de memoria virtual?

6.15. Explicar la diferencia entre un sistema de paginación por demanda y un sistema de paginación con *swapping*.

6.16. Explicar la secuencia de eventos que ocurren cuando se produce un fallo de página.

6.17. Explicar cómo funciona el mecanismo de *copy-on-write*.

6.18. Explicar para qué se utilizan los siguientes bits: *valid*, *modify* y *reference*.

6.19. Supongamos la siguiente tabla de páginas de un proceso que se está ejecutando. Todos los números están expresados en decimales; el tamaño de la página es de 1.024 bytes.

Virtual page number	Valid bit	Reference bit	Modify bit	Page frame number
0	1	1	0	4
1	1	1	1	7
2	0	0	0	-
3	1	0	0	2
4	0	0	0	-
5	1	0	1	0

- a) Describir cómo una dirección virtual generada por la CPU se traduce en una dirección de memoria física.
- b) ¿Qué dirección física se corresponde con las siguientes direcciones virtuales?
 - i) 1.052
 - ii) 2.221
 - iii) 5499

6.20. Un sistema que utiliza un esquema segmentado-paginado posee un espacio de direccionado lógico de 64 KB. A su vez, el sistema utiliza dos segmentos por proceso y un tamaño de página de 4 KB. Dada la siguiente tabla de segmentos y las tablas de páginas:

Segment Table	
Page Table	Length
PTR to Table #0	111
PTR to Table #1	110

Page Table #0	
Valid Bit	FN
1	101011
0	001010
1	001011
1	100110
0	001100
1	110110
0	111010
0	011101

Page Table #1	
Valid Bit	FN
0	010100
1	110101
0	110100
0	011001
1	110011
0	001001
1	000101
1	100010

Para cada una de las siguientes direcciones lógicas, indicar su correspondiente dirección física, o en su defecto, si se produce un *segment-fault* o un *page-fault*.

- a) 00010100 01010111
- b) 11100100 11111111
- c) 11110100 11000111
- d) 00110010 11000111

6.21. Al realizar el *tracing* de un determinado proceso, se deduce que el mismo posee el siguiente *string* de referencia:

0,9,0,1,8,1,8,7,8,7,1,2,8,2,7,8,2,3,8,3

Considerando que el sistema operativo asigna tres *frames* para la ejecución del proceso, determinar el número de *page-faults* bajo los siguientes algoritmos de reemplazo:

- FIFO
- LRU
- Óptimo

6.22. Un proceso posee ocho páginas virtuales en disco, las cuales pueden ser cargadas en un conjunto de cuatro *frames* de memoria principal. Dado el siguiente string de referencia:

1,0,2,2,1,7,6,7,0,1,2,0,3,0,4,5,1,5,2,4,5,6,7,6,7,2,4,2,7,3,3,2,3

- Mostrar la dinámica de asignación bajo un algoritmo de reemplazo LRU, asumiendo que inicialmente todos los *frames* están disponibles, y computar el *hit-ratio*.
- Repetir el inciso anterior utilizando un algoritmo de reemplazo FIFO.
- Comparar los *hit-ratios* obtenidos y argumentar sobre la efectividad de cada algoritmo bajo la traza especificada.

6.23. ¿Cuál es la causa del *thrashing*? ¿Cómo puede ser detectado por el sistema? Una vez que se detecta el *thrashing*, ¿qué puede hacer el sistema para eliminar este problema?

6.24. Suponer un segmento inicial de 1.024 KB donde la memoria se asigna usando el *Buddy system*. Dibujar el árbol que ilustra cómo se asignan las siguientes solicitudes de memoria:

- *request 240 bytes*
- *request 120 bytes*
- *request 60 bytes*
- *request 130 bytes*

A continuación, modificar el árbol a partir de los siguientes *releases* de memoria, realizando la unión de las mismas siempre que sea posible:

- *release 240 bytes*
- *release 60 bytes*
- *release 120 bytes*