Sistemas Operativos

Práctica 4 – Administración de memoria

Notas preliminares

- Los ejercicios marcados con el símbolo ★ constituyen un subconjunto mínimo de ejercitación. Sin embargo, aconsejamos fuertemente hacer todos los ejercicios.

Ejercicio 1

Indicar las diferencias entre dirección de memoria lógica, lineal y física.

Ejercicio 2

Explique la diferencia entre los conceptos de fragmentación interna y externa.

Ejercicio 3 ★

Se tiene un sistema con 16 MB de RAM que utiliza particiones fijas para ubicar a los programas en memoria. Cuenta con particiones de 8 MB, 1 MB, 4 MB, 512 KB, 512 KB y 2 MB, en ese orden. Se desean ejecutar 5 programas de los siguientes tamaños: 500 KB, 6 MB, 3 MB, 20 KB, 4 MB, en ese orden.

1. Indique cómo asignaría las particiones utilizando *best-fit*. ¿Cuál es la cantidad de bytes de memoria desperdiciados?
2. ¿Alguna de las estrategias de asignación vistas en clase (peor-ajuste, primer-ajuste) produce como resultado la imposibilidad de ejecutar los 5 programas a la vez?
3. ¿Cuál algoritmo hace el uso más eficiente de la memoria?

Ejercicio 4

¿Por qué las tablas de páginas son de tamaño potencia de 2?

Ejercicio 5 ★

Considere un sistema con paginación por demanda donde los procesos están haciendo acceso secuencial a los datos de acuerdo a los siguientes patrones de uso:

- Uso de CPU: 20 %
- El sistema hace thrashing.
- Uso del resto de los dispositivos de E/S: 10 %

Como se ve, la CPU está siendo ampliamente desaprovechada.

Para cada uno de los siguientes cambios en el sistema indique si es probable o no que mejore la utilización de la CPU.

a) Instalar una CPU más rápida.
b) Instalar un disco de paginado más grande.
c) Incrementar el grado de multiprogramación.
d) Disminuir el grado de multiprogramación.
e) Instalar más memoria principal.
f) Instalar un disco más rápido.
g) Incorporar *prepaging* a los algoritmos de reemplazo de páginas.
h) Incrementar el tamaño de página.
i) Incrementar la velocidad del bus de E/S.

Ejercicio 6 ★

Se tiene un espacio de direcciones lógicas de 8 páginas de 1024 palabras cada una, mapeado en una memoria que tiene 32 *frames* de capacidad.

a) ¿Cuántos bits tiene una dirección lógica?
b) ¿Cuántos bits tiene una dirección física?

Ejercicio 7

Un sistema asigna espacios de direccionamiento de 65536 bytes, divididos en páginas de 4096 bytes. Un programa particular tiene 32768 bytes de texto, 16836 bytes de datos y requiere de 15870 bytes para la pila (stack). ¿Se puede ejecutar dicho programa en el espacio de direccionamiento disponible? ¿Cambia la situación si el tamaño de página es de 512 bytes?

Ejercicio 8 ★

Considere un sistema de paginación en el que la tabla de páginas esté almancenada en memoria.

a) Si una referencia a memoria tarda en realizarse 200 nanosegundos, ¿cuánto tiempo tardará una referencia a memoria paginada?
b) Si añadimos una TLB y el 75 % de todas las referencias a las tablas de paginación se encuentran presentes en la TLB, ¿cuál es el tiempo que se espera que tarde una referencia a memoria en promedio? (suponer que el acceso a la TLB tarda tiempo 0).

Ejercicio 9 ★

¿Bajo qué circunstancias se produce un *page-fault*? ¿Cuáles son las acciones que realiza el sistema operativo para resolver la situación?

Ejercicio 10 ★

Considere la siguiente secuencia de referencias a páginas:

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

¿Cuántos fallos de página se producirán con los siguientes algoritmos de reemplazo, suponiendo que se tienen 1, 2, 3, 4, 5, 6 ó 7 frames? Al comenzar todos los frames se encuentran vacíos, por lo que la primer referencia a una página siempre genera fallo de página.

a) Con reemplazo LRU.
b) Con reemplazo FIFO.
c) Con reemplazo óptimo.

Ejercicio 11

Se tiene la siguiente matriz:

```
int A[][] = new int[100][100];
```

donde `A[0][0]` está cargado en la posición 200, en un sistema de memoria paginada con páginas de tamaño 200. Un proceso de manipulación de matrices se encuentra cargado en la primer página, de la posición 0 a 199, por lo que todo *fetch* de instrucciones es a la misma página.

Si se tienen sólo 3 *frames* de páginas, ¿Cuántos fallos de página serán generados por los siguientes ciclos, utilizando el algoritmo de reemplazo LRU? Suponer que el programa se encuentra en el primer *frame*, y los otros dos están vacíos.

a)
```
for (int j = 0; j < 100; j++)
  for (int i = 0; i < 100; i++)
    A[i][j] = 0;
```

b) Notar el orden de los índices en los ciclos...

```
for (int i = 0; i < 100; i++)
  for (int j = 0; j < 100; j++)
    A[i][j] = 0;
```

Ejercicio 12 ★

Dado un sistema que no realiza *copy-on-write*, ¿cómo le agregaría esa funcionalidad? Considere:

- Llamadas al sistema a modificar.
- Cambios de HW.
- Cambios en el manejo de segmentos y páginas.

Ejercicio 13 ★

Se tiene un sistema operativo que debe controlar un celular, cuya función principal es atender llamadas.

a) ¿Tiene sentido implementar segmentación? ¿Y si el usuario puede descargarse programas de Internet?
b) ¿Tiene sentido que haya páginas que nunca swappen? En caso afirmativo, ¿Tiene sentido que estas p'aginas est'en en marcos de p'agina prefijados?

Ejercicio 14

Se tienen dos sistemas embebidos:

A) Hace procesamiento secuencial de archivos. Los bloques se leen, se procesan y se escriben.

B) Medidor de clima. Hay un proceso principal que detecta fenómenos meteorológicos (lluvia, vientos, granizo, sol intenso) y lanza programas específicos para hacer mediciones apropiadas. El clima puede cambiar abruptamente y cuando aparece el fenómeno nuevo se lo debe medir de inmediato.

1. ¿Implementaría segmentación? Justifique.
2. Indicar cuál de las siguientes políticas de reemplazo de páginas es más apropiada para cada uno. Justifique.
   a) Bajar la página más recientemente usada.
   b) LRU
   c) Segunda oportunidad + páginas estáticas.

Ejercicio 15

Suponer que se tiene un sistema con 2 MB de RAM y se desea ejecutar un programa de 4 MB ubicado en un disco de 200GB.

1. Explicar cómo funciona el mecanismo de paginación que permite ejecutar un programa más grande que la memoria física disponible.

2. Si el tamaño de *frame* es de 4 KB y suponiendo que el programa tarde o temprano ejecuta todo su código. ¿Cuántos fallos de página se producirán como mínimo?

3. ¿Bajo qué contexto tiene sentido que varios procesos compartan páginas? Indique por lo menos 2 situaciones y justifique.

Ejercicio 16

Se tiene un sistema operativo para una arquitectura multiprocesador con un modelo de memoria plano, es decir, donde las direcciones virtuales son las direcciones físicas. Interesa modificarlo para poder cargar los programas en cualquier lugar de la memoria.

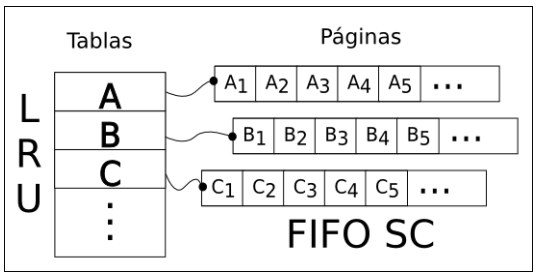
1. Explique por qué en un sistema como el descripto (en la versión actual, sin modificaciones) no resultaría posible cargar un programa en una posición arbitraria de la memoria física.

2. ¿Puede la **segmentación** ayudar a lograr el objetivo? En caso negativo, explique por qué no. En caso afirmativo, explique cómo, sin omitir una descripción de las acciones que debe llevar a cabo el SO al cargar un nuevo proceso en memoria. Indique también si será o no necesario modificar los programas existentes antes de poder utilizarlos con la nueva versión.

3. ¿Puede la **paginación** ayudar a lograr el objetivo? En caso negativo, explique por qué no. En caso afirmativo, explique cómo, sin omitir una descripción de las acciones que debe llevar a cabo el SO al cargar un nuevo proceso en memoria. Indique también si será o no necesario modificar los programas existentes antes de poder utilizarlos con la nueva versión.

Ejercicio 17 ★

Un motor de base de datos está encargado de cargar y descargar de memoria sus tablas de disco a medida las va usando. A la hora de construir un motor es necesario contemplar la política de sustitución de páginas de las tablas que se irán leyendo para así evitar la mayor cantidad de fallos de página. Recientemente los ingenieros de SOracle idearon una nueva política de sustitución de páginas de bases de datos distinta a la tradicionalmente usada (FIFO). Esta nueva política es una composición de políticas ya vistas en la materia (LRU + FIFO SC).



En el momento de hacer un desalojo el motor toma la tabla que hace más tiempo haya sido utilizada (LRU) y entre sus respectivas páginas cargadas toma la que se haya cargado hace más tiempo, excepto que haya sido utilizado nuevamente luego de haber sido cargada en cuyo caso se la reencolará (FIFO SC) tomandosé así la siguiente. En términos implementativos, ésta política mantiene un listado con las tablas que actualmente tienen alguna página en memoria, cada elemento de esta lista hace referencia al listado de páginas siendo utilizado por esa tabla. Ambos listados no están ordenados de cualquier manera, el primero (listado de tablas) los elementos están ordenado por que tan recientemente fueron accedidos (siendo el primero el más recientemente accedido), mientras que el segundo listado (listado de páginas usados por una tabla) está ordenado por orden en que se cargó cada página. Cabe destacar que el último mencionado los elementos tienen un bit de "uso" el cual indica si la página fue usada desde que se cargó.

Una tabla de bases de datos se puede leer de manera completa secuencial (notación: A^*) ó de manera aleatoria sobre uno o varios de sus ítems al azar (notación: A^k). Por ejemplo la secuencia

$Q^2R^*Q^1$ realiza la lectura de dos items de Q, luego una lectura completa de R seguido de la lectura de otro item de Q . Los items pueden o no ser el mismo además cada ítem requiere la carga de una página entera.
Dada la siguiente secuencia:

$A^*B^1A^1C^4A^*B^*C^*$

Sabiendo que:

- A ocupa 20 páginas
- B ocupa 4 páginas
- C 15 páginas
- Tenemos 24 frames disponibles en memoria.

Indicar tanto para la política dada como para FIFO a secas:

- El estado de la memoria al finalizar la secuencia.
- El *hit rate* de cada política.