Sistemas Operativos 2/2018
Laboratorio 3

Profesores:
Cristóbal Acosta (cristobal.acosta@usach.cl)
Miguel Cárcamo (miguel.carcamo@usach.cl)
Fernando Rannou (fernando.rannou@usach.cl)
Ayudantes:
Esteban Alarcón (esteban.alarcon.v@usach.cl
Marcela Rivera (marcela.rivera.c@usach.cl))

I. Objetivos Generales

Este laboratorio tiene como objetivo aplicar los conceptos de *memoria virtual*, tales como tabla de página, tabla multinivel, caché, entre otros.

II. Objetivos Específicos

1. Conocer y usar las funcionalidades de `getopt()` como método de recepción de parámetros de entradas.
2. Implementar paginación.
3. Comprender e implementar tabla multinivel.
4. Implementar caché (TLB) y políticas de reemplazos.
5. Conocer y practicar uso de makefile para compilación de programas.

III. Conceptos

III.A. Memoria virtual

Todas las referencias a memoria son direcciones lógicas y son mapeadas a direcciones físicas en tiempo de ejecución. Por otro lado, los procesos pueden ser particionados en páginas o segmentos.

III.B. Memoria virtual con paginación

Los procesos se dividen en bloques pequeños de igual tamaño, llamados páginas, a su vez la memoria también se divide en bloques pequeños llamados marcos, los cuales almacenan páginas. De lo anterior se desprende que los marcos y las páginas son del mismo tamaño.

Debido a la memoria virtual, no todas las páginas o segmentos de un proceso tienen que residir en memoria principal durante su ejecución, lo cual permite que se puedan almacenar en memoria una mayor cantidad de procesos.

Por otro lado, el formato de dirección de memoria consiste en un número de página y offset.

Número de Página	Offset

Table 1. Formato de direccionamiento lógico

III.C. Tabla de página

Para conocer la ubicación de las páginas, cada proceso tendrá una tabla de página la cual contiene el marco donde se encuentra almacenada la página. Su formato de dirección es:

P	M	Número de página	offset

Table 2. Formato de direccionamiento tabla de página.

III.D. Dirección física

Como se menciona inicialmente, las direcciones lógicas son traducidas a dirección física en tiempo de ejecución. El formato de direccionamiento es:

Marco	Offset

Table 3. Formato de direccionamiento físico

III.E. Tabla multinivel

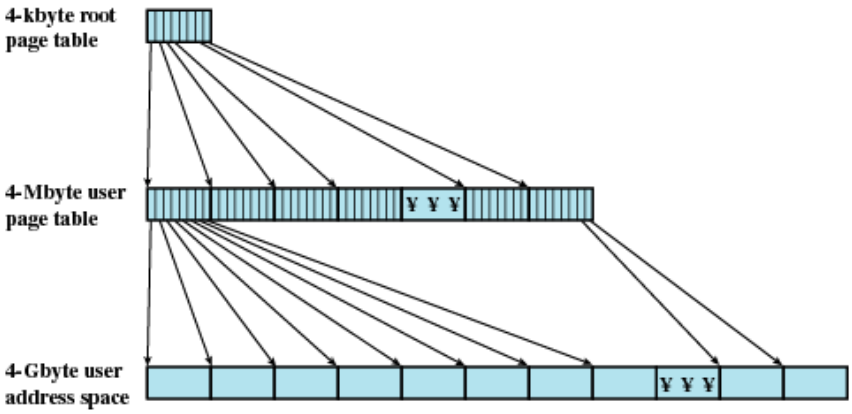
Debido a que el tamaño de la tabla de página depende del espacio de direccionamiento, el sistema de administración debe paginar la tabla de página. Por lo tanto, el problema radica en el tamaño de la página, ya que si se tiene una TP de 4MB con páginas de 4KB, nos da 1024 páginas.

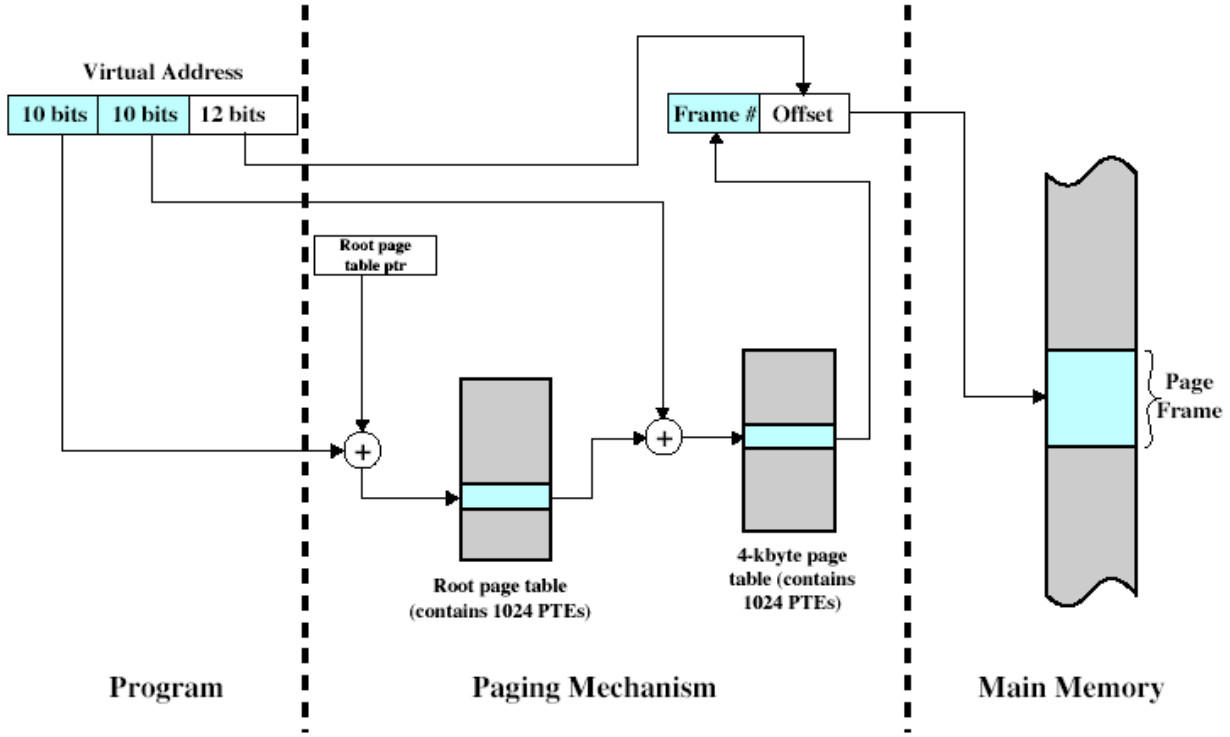
Lo anterior hace referencia a tabla multinivel, ya que se pagina la tabla de página. Para el caso de dos niveles, se tiene una tabla raíz (primer nivel), la cual contiene la tabla de página en la cual se encuentra la tabla que se está buscando. El formato de dirección es:

Nivel 1	Nivel 2	Offset

Table 4. Formato de direccionamiento físico tabla multinivel

En la figura 2 se presenta como apunta la tabla raíz al resto de niveles. Por otro lado, gráficamente, el camino lógico de la tabla multinivel se encuentra en la figura 1.





III.F. TLB

Para optimizar la traducción de dirección lógica a física, se mantiene una caché de alta velocidad encargada de almacenar una pequeña parte de la tabla de páginas. Esto se hace debido a que la memoria virtual puede provocar dos accesos a memoria, el primero para acceder a la tabla de páginas y el segundo para acceder al dato mismo. La idea es que esta cache llamada TLB, almacene el mapeo de páginas− >marco de las páginas recientemente usadas.

El procedimiento para utilizar la TLB, consiste en: a partir de una dirección virtual el procesador examina la TLB buscando el mapeo de la página buscada, en caso de encontrarse hay un HIT, por lo que a continuación se recupera el número de marco y se genera la dirección física (concatenando marco con offset), en caso contrario, hay miss.

En caso de miss, se va a buscar a la tabla de páginas. Luego, se chequea que la página esté en memoria principal, si no, el SO emite una interrupción con page-fault. Finalmente, se actualiza la TLB para reflejar la nueva página referenciada.

III.G. Algoritmos de reemplazo

Permiten determinar qué página es reemplazada, cuando no hay marcos libres. Para efectos del laboratorio, se utilizarán los siguientes algoritmos con el fin de determinar que páginas reemplazar en la TLB.

III.G.1. FIFO

Se reemplaza la página que lleva más tiempo en memoria, es uno de los algoritmos más simples de implementar. Sin embargo no tiene el mejor rendimiento debido a que se puede reemplazar una página que podría necesitarse nuevamente (principio de localidad).

III.G.2. LRU

Algoritmo que reemplaza la página que no ha sido referenciada durante más tiempo. Por el principio de localidad, la página seleccionada por LRU debiera ser aquella que es menos probable de ser referenciada en

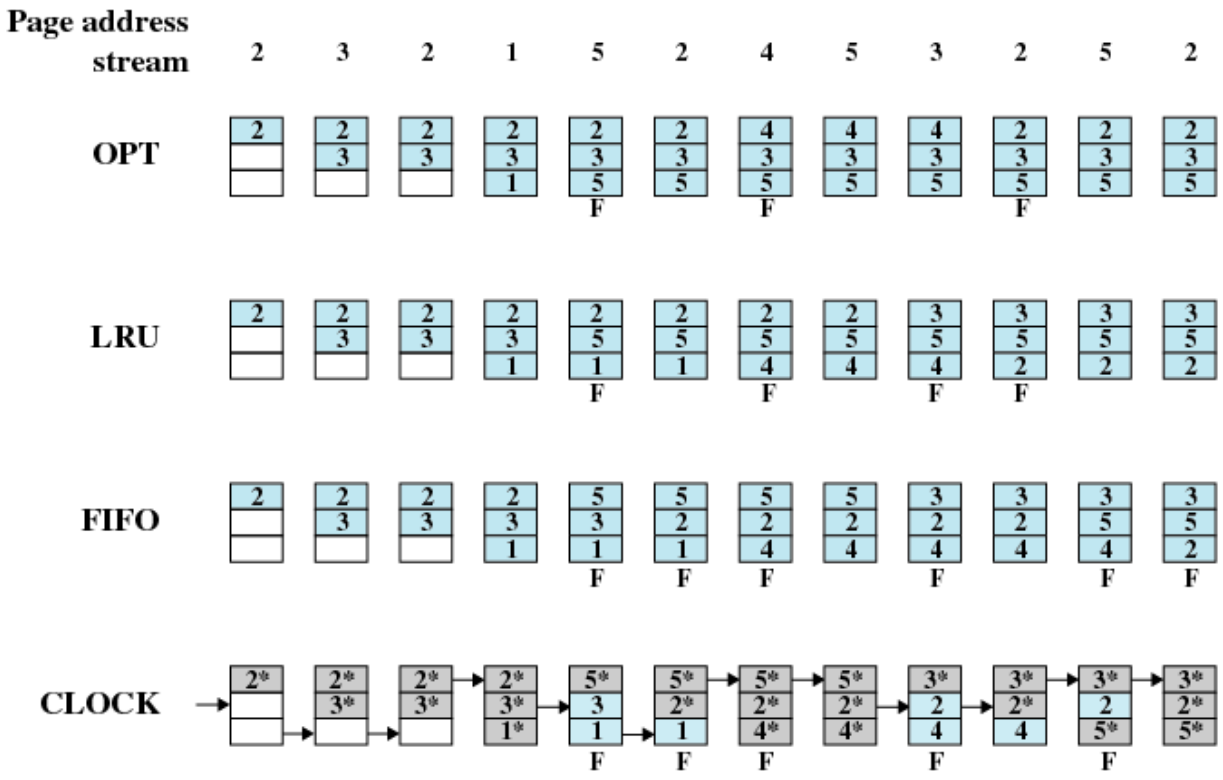
el futuro cercano.

III.G.3. Reloj

Intenta ser similar a LRU pero con mayor eficiencia. Para conseguir esto, se usa un bit de uso para cada marco (1 o 0). Los pasos son:

1. Cuando el marco se carga con una página, se setea el bit en 1.
2. Cuando la página es referenciada, se setea el bit en 1.
3. Cuando es necesario reemplazar una página, se busca en orden secuencial el primer marco con bit de uso en 0; dicha página es reemplazada.
4. Durante la búsqueda los bits de uso en 1 son cambiados a 0
5. ¿Qué pasa si todos los bits están en 1?, se setean todos a cero y se escoge según FIFO.

Para un mejor entendimiento de los algoritmos previamente descritos, se presenta un ejemplo obtenido del libro Stallings.



IV. Actividad

Se debe construir un sistema de paginación de dos niveles para traducción de direcciones lógicas a físicas. Este programa sólo traduce direcciones de 16 bits.

La dirección lógica incluye los bits para direccionar la tabla raíz, bits para la tabla secundaria y el restante determina el offset del marco (depende del tamaño de página). Como parámetros de entrada se ingresan los

bits para tabla raíz y tabla secundaria, por lo tanto, el tamaño de página se infiere:

$$TamañoPágina = 16 - bitsRaiz - bitsSecundaria. \quad (1)$$

Notar que $bitsRaiz + bitsSecundaria < 16$, es decir, offset mínimo debe ser 1.

El sistema debe incluir una TLB, el cual su número de entradas se entrega como parámetro. Este debe ocupar uno de tres algoritmos de reemplazo propuestos: FIFO, CLOCK, LRU. El programa debe calcular la cantidad de hits y miss dentro de la TLB.

Finalmente, debe almacenar el resultado de las traducciones de direcciones lógicas a físicas.

Existen dos archivos de entrada; el primero contiene las direcciones lógicas a traducir y el segundo, contiene los marcos a leer e ingresar a las tablas de páginas secundarias. Por ejemplo, si los bits de tabla raíz es 5 y bits de tabla secundaria son 6, entonces deben leerse e ingresarse un total de $2^5 \cdot 2^6 = 2^{11}$ marcos,repartidos secuencialmente en las tablas secundarias (Notar que no necesariamente deben leerse todos los marcos del archivo).

IV.A. Ejemplo archivo entrada 1

```
0xA06C
0xF43D
.
.
.
0x7B34
```

IV.B. Ejemplo archivo entrada 2

```
0x34FV
0x4CA1
.
.
.
0x58D7
```

V. Parámetros de entrada

Los parámetros que deben ser entregados por la terminal son:

- -r: Cantidad de bits para tabla raíz.
- -s: Cantidad de bits para tablas secundiaria.
- -t: Número de entradas para TLB.
- -b: Permite ver los hits en TLB para cada dirección lógica traducida.

VI. Salida del programa

El programa debe generar dos archivos de salida: traduccion.txt y tasas.txt.

El archivo traduccion.txt contiene cada dirección física obtenida en la traducción de las direcciones lógicas ingresadas incialmente; el formato debe ser igual al del archivo de entrada de direcciones lógicas.

El archivo tasas.txt contiene cantidad de hits y miss total obtenidos en TLB para los 3 algoritmos de reemplazo.

VII. Entregables

Debe entregarse un archivo comprimido que contenga al menos los siguientes archivos:

1. *Makefile*: archivo para make que compile los programas.
2. dos o mas archivos con el proyecto (*.c, *.h)

El archivo comprimido debe llamarse: RUTESTUDIANTE1_RUTESTUDIANTE2.
Ejemplo: 19689333k_189225326.zip

VIII. Fecha de Entrega

Jueves 17 de Enero, hasta las 23:55 hrs.