



*Sistemas Operativos, Pauta Certamen #3,
Segundo Semestre de 2023.*

1. Paginación y Uso de TLB

Se ha instalado la última versión de Windows XP en una arquitectura de hardware que cuenta con una TLB. Se tiene 1GB de memoria principal distribuido en 524.288 marcos gestionados a través de paginación. Los arreglos se guardan por fila en la memoria. El código, datos y stack del proceso utilizan 1 página cada uno.

```
int anuel[512][512];
for (i=0; i<512; i++)
    for (j=0; j<512; j++)
        anuel[j][i] = i + 1;
```

Cuadro 1: Extracto de Código

- (a) Determine la cantidad mínima de entradas de la TLB para generar la menor cantidad de fallos al direccionar el código del cuadro 1.

Respuesta: El mínimo número de fallos se produce cuando se genera solo uno por cada nueva página cargada. Se requieren 3 fallos para código, datos y stack. El tamaño del arreglo es de $512 \times 512 \times 4B = 1MB$. Como el tamaño de la página es 2KB, se requieren 512 páginas para guardar el arreglo. El mínimo número de entradas es de 515.

- (b) Determine la tasa de fallos de TLB que se producen al ejecutar el código del cuadro 1 con 10 entradas.

Respuesta: En cada ciclo solo hay un acceso a memoria principal, dado como se recorre el arreglo, cada referencia a una página genera un fallo, por lo que la tasa de fallos es de 100%.

- (c) Si cada acceso a memoria principal toma 1 [μs] determine el tiempo total requerido en el punto (b).

Respuesta: Cada fallo de TLB produce 2 accesos a memoria, dado que la tasa es de 100% el tiempo total es de $(3 + (512 \times 512)) \times 2 \times 1 \approx 0,5[s]$

2. Paginación

Se instaló Windows 11 en una arquitectura que gestiona 1GB de memoria principal utilizando paginación. El tamaño del espacio virtual es de 4GB.

- (a) Si la memoria se divide en 131.072 marcos y se sabe que el tamaño máximo de la tabla de páginas es de 2MB determine el tamaño de la página y el tamaño de cada entrada en la tabla de páginas.

Respuesta: Con la información del enunciado:

$$T_{pag} = \frac{1GB}{131.072} \rightarrow \frac{2^{30}}{2^{17}} = 2^{13}B = 8KB$$

$$TTp_{Max} = 2^{21}B = 2^{19} \times T_{entrada} \rightarrow T_{entrada} = 4B$$

- (b) Determine la fragmentación interna producida por un conjunto de procesos al llenar la memoria principal. Considere un tamaño medio de 4MB para cada proceso y que se pierde un 25% promedio.

Respuesta: Cada proceso requiere $4MB / 8KB = 512$ páginas más lo necesario para la tabla de páginas. Como cada entrada a la tabla tiene 4B, el tamaño de ésta es de $4B \times 512$, es decir, 2KB. Dado el tamaño de la memoria principal y de las páginas podríamos llevar a ésta $131.072 / 512 = 256$ procesos, pero nos faltaría espacio para las tablas de páginas, por lo que llevaremos 255 procesos, ocupando 130.560 marcos, dejando 512 disponibles para asignar las tablas de páginas. Con tan pocos marcos, el único escenario posible es juntar las tablas de páginas. Se requieren $2KB \times 255 / 8KB = 63,75$ marcos para las tablas de páginas, lo cual genera 0,25KB de fragmentación interna.



En la paginación la fragmentación interna siempre se produce en la última página, por lo que tendremos 2KB por proceso. El cálculo total es $2\text{KB} * 255 + 0,25\text{KB}$.

3. Tiempos y Segmentación Paginada

Un sistema operativo gestiona la memoria principal a través de segmentación paginada. Cada proceso puede tener a lo más 8 segmentos. La tabla de segmentos se implementa únicamente con registros y el tiempo de acceso a ellos es igual a 2[ns]. El tiempo de acceso a la memoria principal es de 20[ns]. Para mejorar el rendimiento del sistema se utiliza una TLB en donde se guarda tanto el segmento como la página. El tiempo de acceso y búsqueda en promedio es de 10[ns].

- (a) Explique brevemente los pasos que se seguirían para realizar la conversión de una dirección.

Respuesta: Para traducir una dirección lógica a física se debe conocer el segmento, consultar en la tabla de segmentos para ver si la dirección es correcta y luego acceder al nivel de la paginación:

- Se accede a los registros para consultar la tabla de segmentos $\rightarrow T_{reg}$
- Se accede a la TLB para buscar el segmento y la página $\rightarrow T_{TLB}$
 - Si se encuentra, se realiza la conversión y se accede a memoria. $\rightarrow T_{AM}$
 - Si no se encuentra, hay que consultar la tabla de páginas que se encuentra en la memoria y luego acceder a la memoria principal. $\rightarrow 2*T_{AM}$

- (b) Si la tasa de aciertos en la TLB es del 90%, determine el tiempo de acceso efectivo que genera el sistema.

Respuesta: Del punto (a) podemos desprender cual es el comportamiento y los tiempos asociados:

$$T_{AE} = (T_{reg} + T_{AM} + T_{TLB}) * 0,9 + (T_{reg} + 2 * T_{AM} + T_{TLB}) * 0,1$$

$$T_{AE} = (2 + 20 + 10) * 0,9 + (2 + 2 * 20 + 10) * 0,1$$

$$T_{AE} = 36,7[\text{ns}]$$

4. Memoria Virtual

La versión Jammy Jellyfish de Ubuntu se ha instalado en una arquitectura de hardware que gestiona la memoria principal a través de **paginación bajo demanda**. Las direcciones lógicas y físicas tienen un tamaño de 20 bits, se dispone de un total de 1024 marcos y el tamaño de la memoria es de 64MB.

La ejecución de un proceso genera la siguiente secuencia de direcciones:

0xAFFFF, 0xBAFBF, 0xAA000, 0xFA888, 0xCB123, 0xFB004, 0xAA440, 0xCB008, 0xBC004, 0xCA44F,
0xFB008, 0xAB440, 0xAA001, 0xFB008, 0xCAA00, 0xFA001

Respuesta: El tamaño de página es $64\text{MB} / 1024 = 2^{16}\text{B}$, por lo que se utilizan 16 bits para el offset, por lo que la cadena de referencias a estudiar es:

A	B	A	F	C	F	A	C	B	C	F	A	A	F	C	F
10	11	10	15	12	15	10	12	11	12	15	10	10	15	12	15

- (a) La asignación directa de marcos utiliza la formula $p \bmod m$ para asignar, en donde p es el número de página y m el número de marcos. Utilizando asignación directa y sabiendo que hay 3 marcos disponibles, determine el total de fallos de página.

Respuesta: 9 fallos de página.



A	B	A	F	C	F	A	C	B	C	F	A	A	F	C	F
			F	C	F		C			F				C	F
A	A		A	A	A		A			A				A	A
	B		B	B	B		B			B				B	B
1	2		3	4	5		6			7				8	9

- (b) Con la misma cantidad de marcos de (a) y utilizando LRU, determine el total de fallos de página.

Respuesta: 7 fallos de página.

A	B	A	F	C	F	A	C	B	C	F	A	A	F	C	F
A	A		A	A				A		F	F				
	B		B	C				C		C	C				
			F	F				B		B	A				
1	2		3	4				5		6	7				

- (c) Para la preguntas (a) y (b) exprese una fórmula que permita calcular el tiempo de acceso a memoria principal. Compare y concluya.

Respuesta: Con mapeo directo un 56% de las referencias producen fallo de página. El cálculo:

$$T_{AE} = 0,4375 * MA + 0,5625 * W$$

Con asignación asociativa y ocupando LRU un 44% de las referencias a memoria genera un fallo de página.

$$T_{AE} = 0,5625 * MA + 0,4375 * W$$

En este ejercicio en particular es una coincidencia de que las fórmula estén al revés, lo que sí se puede concluir es que mapeo directo genera más fallo de página, impactando de peor forma el tiempo de ejecución de un programa.