

Sistemas Operativos

Cursada 2022

Comisión S21 y S22

Memoria Virtual (Algoritmos)

Algoritmos de selección de Pagina Victima

FIFO

Optimo

LRU

LRU por aproximación

- Bits de referencias adicionales
- Segunda chance o reloj

Memoria Virtual (Algoritmos)

Algoritmo FIFO (Ya conocido)

Supongamos que tenemos las siguientes direcciones lógicas

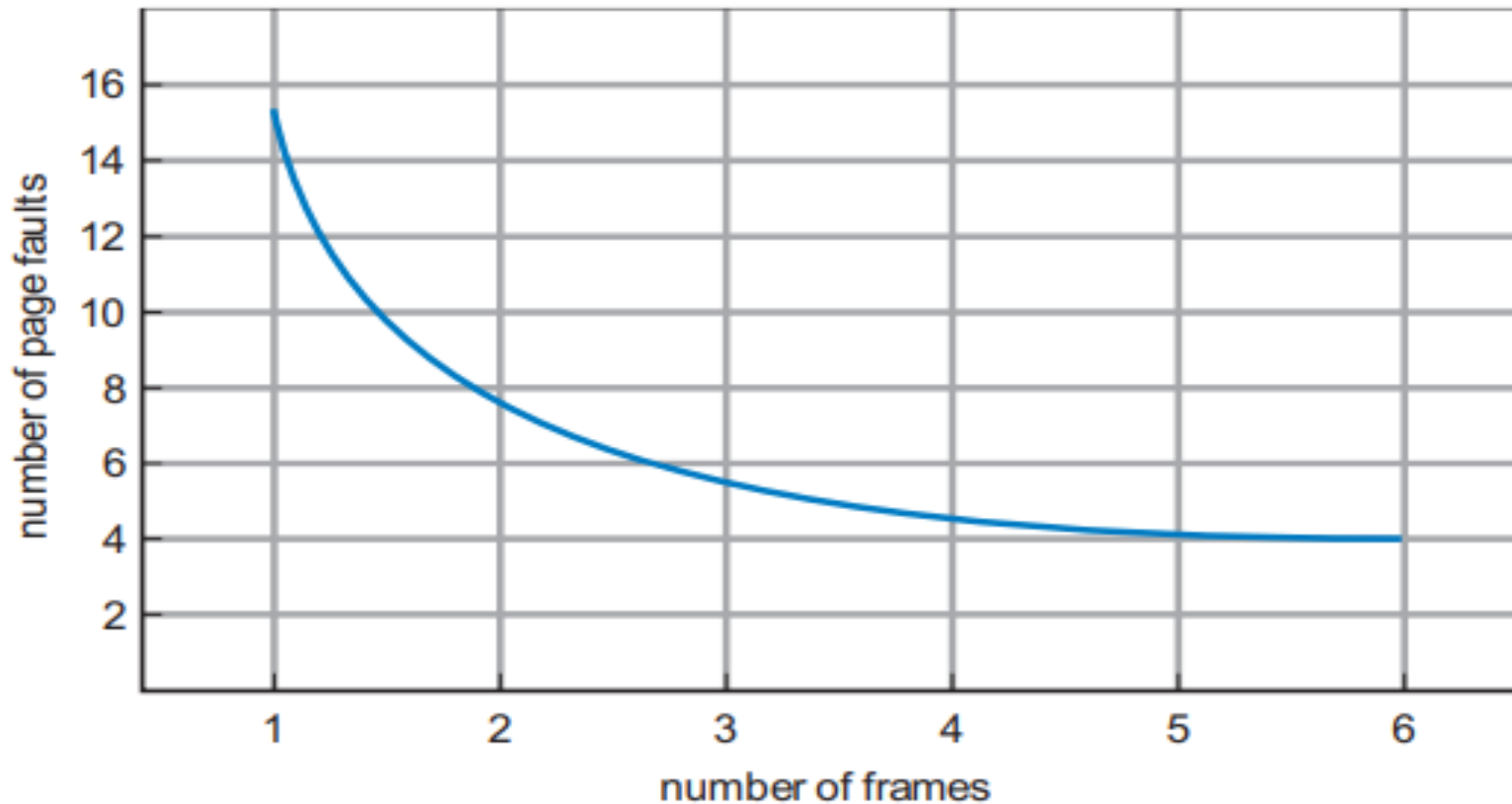
El tamaño de páginas es de 1K

512 7420 2500 7421 6000 8300 9750 2500 4098

Marcos	0	7	2	7	5	8	9	2	4
0	0	0	0	0	0	8	8	8	8
1		7	7	7	7	7	9	9	9
2			2	2	2	2	2	2	4
3					5	5	5	5	5
	FP	FP	FP		FP	FP	FP		FP

Memoria Virtual (Algoritmos)

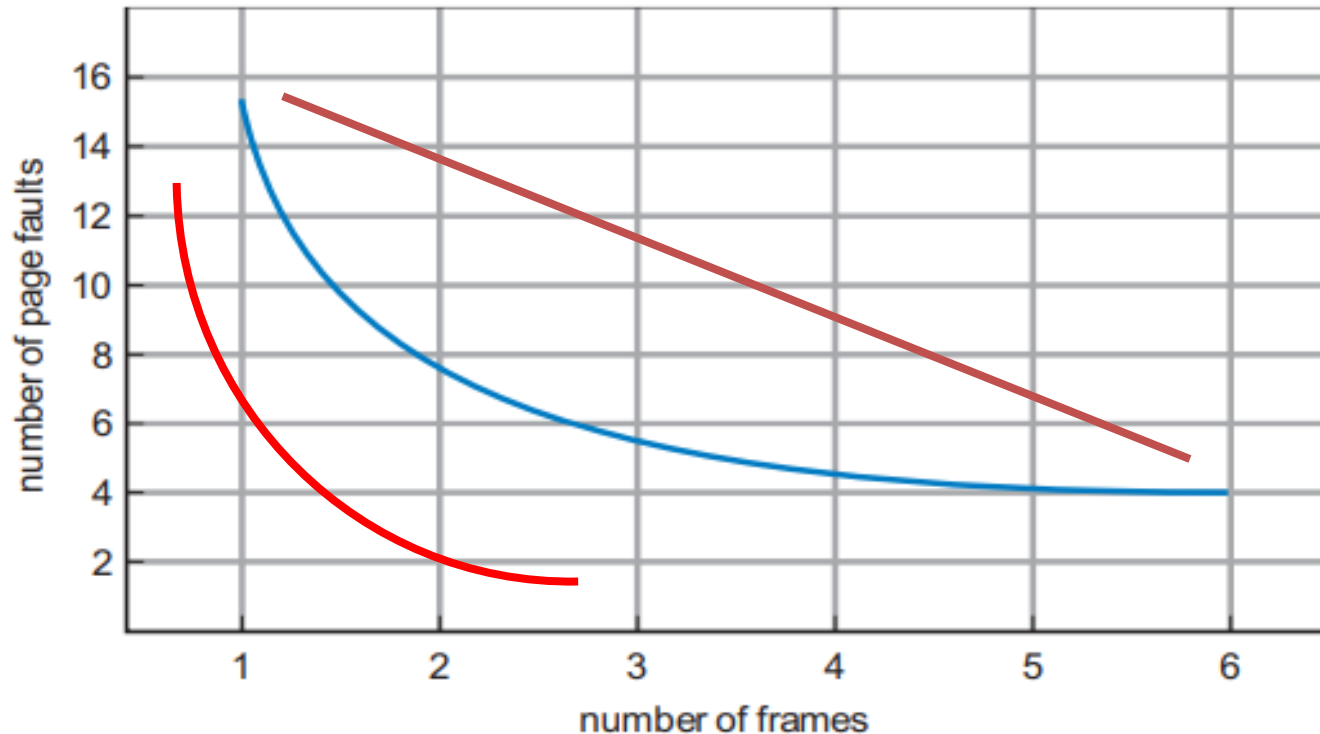
Para FIFO deberíamos tener una expectativa de aciertos que se refleja en esta curva



Porque no llega a 0 (siempre habrá 1 FP)

Memoria Virtual (Algoritmos)

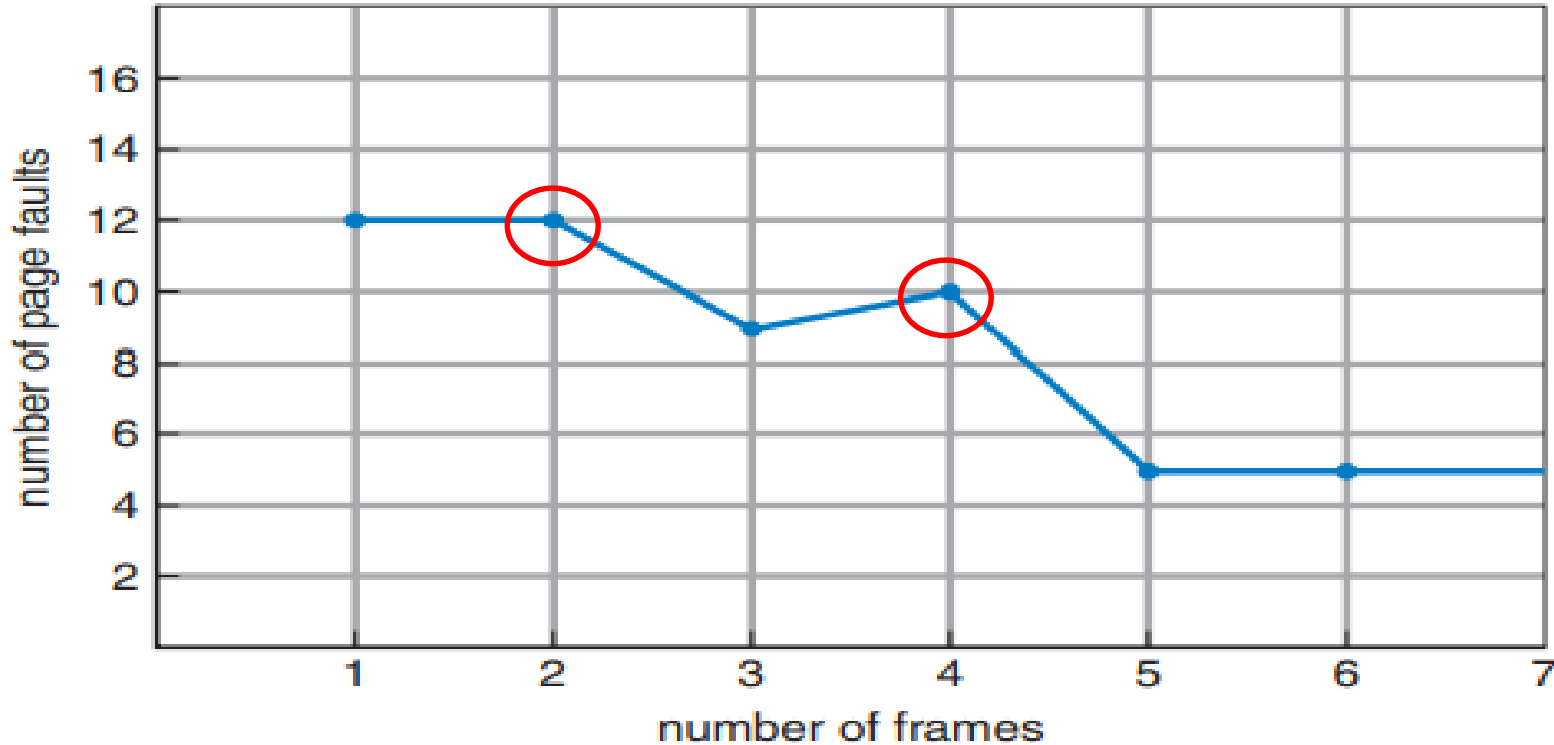
Se estudio el comportamiento de la curva.
Si el algoritmo era realmente eficiente
Belady tomo la posta y estudio este grafico



Memoria Virtual (Algoritmos)

Después de estudiar el comportamiento

En base a muchos Datos Estadísticos



Cuanto mas marcos asigno anda peor??

Esto se llama ***“Anomalía de Belady”***

Memoria Virtual (Algoritmos)

Algoritmo Optimo

Supongamos que tenemos las siguientes referencias a paginas

Marcos	7	0	1	2	0	3	0	4	2	3	0	3	2	1
0	7	7	7	2	2	2	2	2	2	2	2	2	2	1
1		0	0	0	0	0	0	4	4	4	0	0	0	0
2			1	1	1	3	3	3	3	3	3	3	3	3
	FP	FP	FP	FP		FP		FP			FP			FP

Fallos de pagina = 8

Como no podemos conocer el futuro, el algoritmo no es aplicable, nos sirve para comparar el resto de los algoritmos

Memoria Virtual (Algoritmos)

Algoritmo LRU (least recently used)

- En el optimo, para ver el futuro, analizamos los datos históricos, en muchos casos funciona
- Vamos analizar un algoritmo que mire el pasado

Supongamos que tenemos las siguientes referencias a paginas

Marcos	7	0	1	2	0	3	0	4	2	3	0	3	2	1
0	7	7	7	2	2	2	2	4	4	4	0	0	0	1
1		0	0	0	0	0	0	0	0	3	3	3	3	3
2			1	1	1	3	3	3	2	2	2	2	2	2
	FP	FP	FP	FP		FP		FP	FP	FP	FP			FP

Memoria Virtual (Algoritmos)

Algoritmo LRU (least recently used)

En principio como se implemento

- **Contador**
- **Pila**
- **Lista doblemente enlazada**

Ninguna de estas opciones fueron muy eficientes

Memoria Virtual (Algoritmos)

Algoritmo LRU por Aproximación

- **Algoritmo de bits de referencias adicionales**

Se usan registros de desplazamientos de 8bits

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Nunca fue usada

0	0	0	0	0	0	1	1
---	---	---	---	---	---	---	---

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

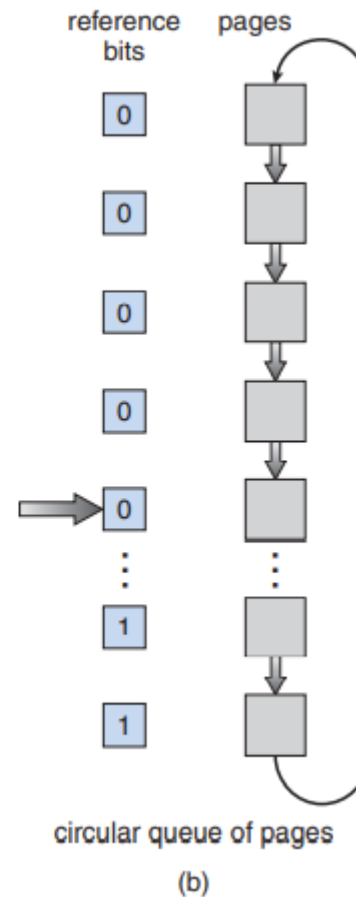
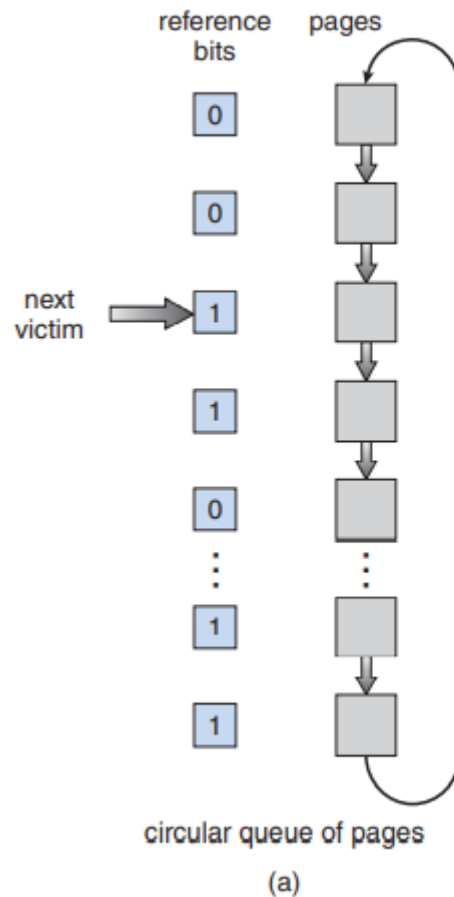
Se usó al menos 1 vez x periodo

1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Memoria Virtual (Algoritmos)

Algoritmo LRU por Aproximación

- Algoritmo segunda chance o reloj



Memoria Virtual (Algoritmos)

Algoritmo LRU por Aproximación

- Algoritmo segunda chance o reloj

Supongamos que tenemos las siguientes referencias a

Marcos	7	0	1	2	0	3	0	4	2	3	0	3	2	1
0	7	7	7	2	2	2	2							
1		0	0	0	0	0	0							
2			1	1	1	3	3							
	FP	FP	FP	FP		FP								

Cola: 7⁰ 0⁰ 1⁰ 2⁰

Cola: 7⁰ 0¹ 1⁰ 2⁰

Cola: 7⁰ 0¹ 1⁰ 2⁰ 3⁰

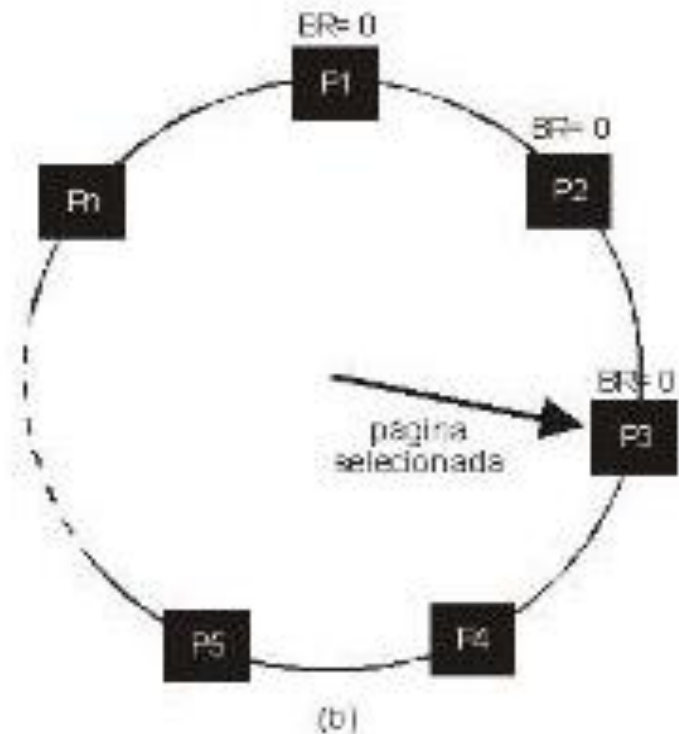
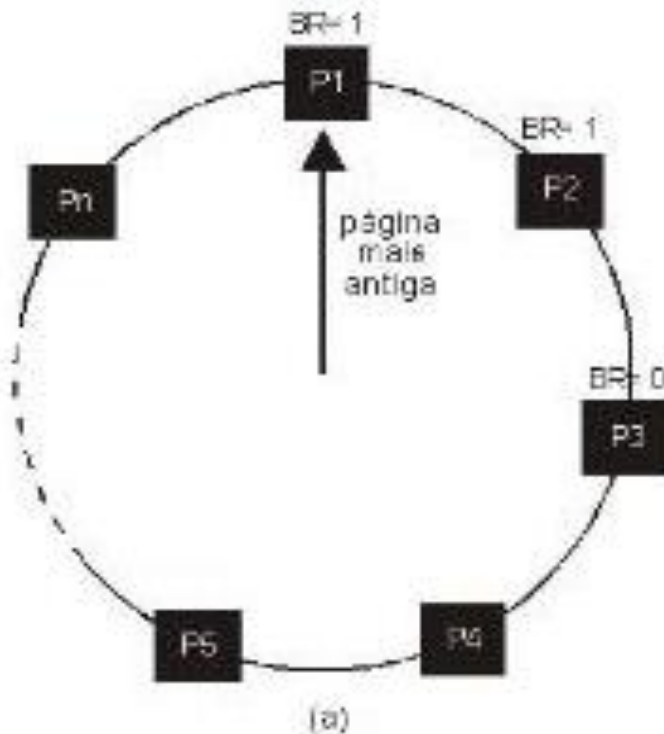
Cola: 7⁰ 1⁰ 2⁰ 3⁰ 0⁰

Cola: 7⁰ 1⁰ 2⁰ 3⁰ 0¹

Memoria Virtual (Algoritmos)

Algoritmo LRU por Aproximación

- Algoritmo segunda chance o reloj
 - FIFO Circular (Clock)



Memoria Virtual (Algoritmos)

Algoritmo LRU por Aproximación

- **Algoritmo segunda chance o reloj mejorado**

Este algoritmo usa 2 bits en lugar de uno

(0,0) Ni usada – Ni modificada

(0,1) No Usada – Si modificada

(1,0) Si usada – No modificada

(1,1) Si Usada –Si modificada

Memoria Virtual (Algoritmos)

Algoritmos con base de conteo

Tienen un contador que indica la cantidad de veces que se uso

LFU Menos frecuentemente Usada

MFU Mas frecuentemente Usada

Implementaciones costosas, por lo tanto no fueron implementadas

Memoria Virtual (Algoritmos)

Algoritmos con base de conteo

Tienen un contador que indica la cantidad de veces que se uso

LFU Menos frecuentemente Usada

MFU Mas frecuentemente Usada

Son costosas, por lo tanto no fueron implementadas

Los S.O. modernos en realidad tienen en cuenta de no quedarse con la pila de marcos vacíos, por lo tanto hacen pre sacrificio de marcos

Memoria Virtual (Algoritmos)

Otra problemática en la administración de memoria

- La paginación bajo demanda o autosatisfacción no funcionan muy bien.
- Debemos tener en cuenta los tipos de procesamiento

Asignación inicial de marcos

- **Equitativa** (Interactivo)
- **Proporcional** (batch)

Sustitución de marcos

- **Local** (Interactivo)
- **Global** (batch)

Memoria Virtual

Cantidad de marcos mínimos asignado a los procesos

Si hablamos de **paginación bajo demanda**, estamos suponiendo que un proceso arranca con solo cargar la primer pagina. Esto no es así.

- Por arquitectura hay una mínima cantidad de paginas que necesita un proceso
- Por los niveles de **indireccion** que puede haber (serian los direccionamientos indirectos o indexados)

Memoria Virtual

Cantidad de marcos mínimos asignado a los procesos

Ejemplos

Variable A : esto es una dirección de memoria

Vector A(I): debo resolver I, y luego A(I)

(Acá mínimo 2 niveles de indireccion)

Pero podemos armar una matriz de este tipo

A(b(c(d))) En este caso tenemos varios niveles de indireccion (Técnica de prog que se llama vectores de agrupamiento)

El compilador determina la cantidad de marcos de acuerdo a los niveles de indireccion que hay.

Memoria Virtual

Inconvenientes que puede ocasionar la M.V.

Sobrepaginacion – Thrashing – Hiperpaginacion

La idea de la M.V. cual es??

Si un proceso necesita mas marcos, y si hay??

Si no hay marcos libres??

Si hay muchos Fallos de Paginas ??

**Si los procesos se lo pasan mas tiempo paginando
que procesando, caemos en lo que se denomina
Sobrepaginacion**

Memoria Virtual

Porque podemos caer en Thrashing?

Vamos a ver primero el contexto donde estamos

Mainframe (procesos batch): El que define el nacimiento es el planificador de largo plazo
Si tenemos procesos orientados a la E/S, por ej.
EL Planificador de Corto Plazo puede decir, tenemos mucho tiempo en wait la CPU, x lo tanto se pueden hacer nacer mas procesos, puede pasar que los procesos entren en F.P. si el P.M.P. sigue viendo mucho en wait, así podemos entrar en sobrepagincion.

Memoria Virtual

Mainframe o no (procesos interactivos):

- Acá el problema puede ser una mala decisión del usuario cuando desarrollo (sobredimensiono)
- No hay suficientes recursos para lo que quiero hacer (trabajar con un video, abrir 20 sitios)
- Falla de algún software de base
- En cuanto a ciertos problemas que tiene que ver con las decisiones del usuario los S.O. mucho no pueden hacer.

Memoria Virtual

Como podemos evitar la Sobrepaginacion

Deberíamos poder garantizar que los procesos tengan los marcos necesarios.

Recordar que esto se basa en los modelos de programación (paradigma estructurado, objeto).

Tienen el concepto de que trabajan dentro de una rutina, esto se llama **Criterio de Localidad**.

Tener toda la **rutina** cargada con su **Var. Locales**.

Ahora si uso **Var Globales** seguramente no voy a tener todo lo que necesito.

Memoria Virtual

Por lo tanto usando este Criterio de Localidad.

- El S.O. tiene una referencia de la cantidad de marcos que necesita para el CS y el DS (Arqui.)
- Si la segmentación respeta el punto de vista del programador, cada rutina tiene su segmento y como estos están paginados, el compilador sabe de antemano cuantos marco necesita esa rutina. Y si usa variables locales también sabe los marcos que necesita para el DS.

Memoria Virtual

El problema se plantea cuando hay paginación pura.

Debemos definir el **Modelo de Localidad**:

Significa que a medida que se ejecuta un proceso este se mueve de localidad en localidad, seria un conjunto de paginas que se usan activamente juntas dentro de una rutina.

El Modelo de **Conjunto de Trabajo** se basa en la suposición de localidad.

Memoria Virtual

Por ejemplo

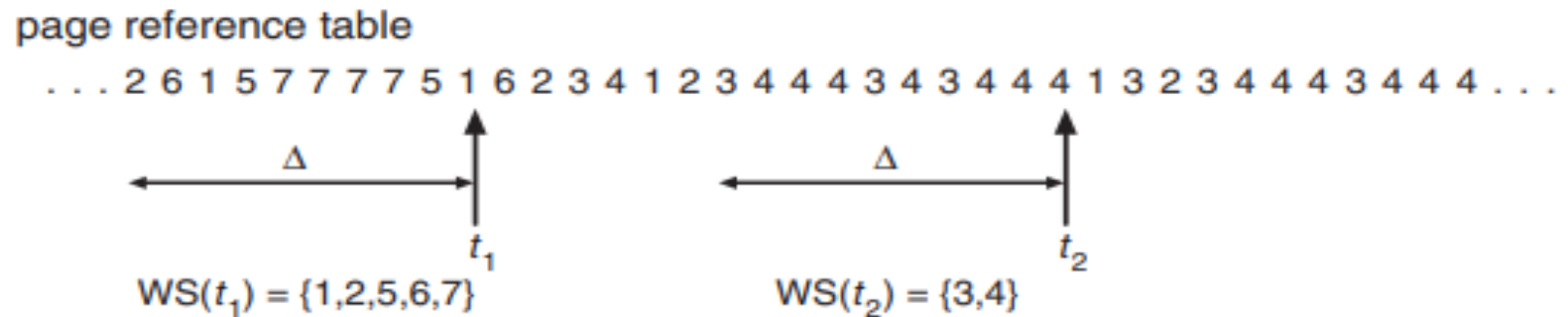
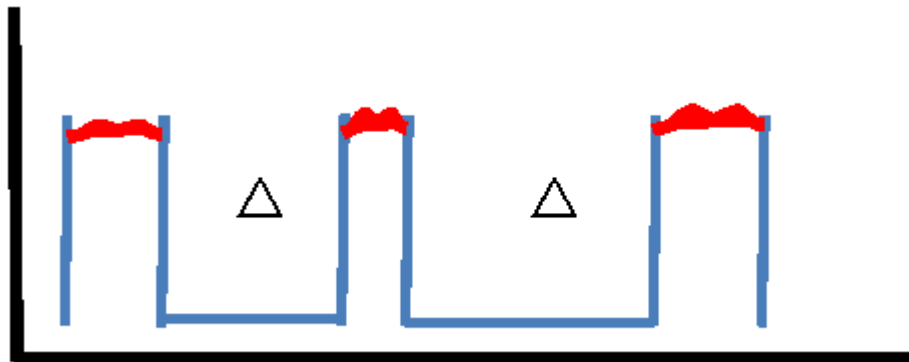


Figure 8.20 Working-set model.

Este modelo define un Δ , para definir la ventana del conjunto de trabajo



Memoria Virtual

Otro modelo es poner una tope inferior y un tope superior a la cantidad de fallos de pagina.

- Si tengo muchos fallos de pagina quiere decir que no le di lo suficiente.
- Si casi no tengo fallos quiere decir que le di demás

Por lo tanto debemos medir la **Tasa de Fallos**

Ej. Si cada 100 direccionamientos tengo 10 fallos, quiere decir que le di poco, si tengo 1 al menos quiere decir que puedo sacrificar los de la misma rutina.

Memoria Virtual

Tamaños de Paginas

Hasta ahora veníamos trabajando con un solo tamaño.

La realidad de los S.O. modernos es que tratan otros tipos de archivos, los cuales son mas difícil de paginar por su tamaño.

Por ej. los archivos multimedias que son de tamaños considerados.

Por lo tanto los S.O. manejan distintos tamaños de paginas

Memoria Virtual

Tamaños de Paginas

Una zona de la RAM es para paginas fijas.
La otra zona de la RAM es para paginas variables.

Los tamaños no varían por ejemplo para la zona del CS (1,2,4, u 8K)

Para la zona del DS si pueden varias, y llegar a Megas de tamaño, siempre respetando que sean potencia de base 2, **porque?**

Memoria Virtual

➤ **Prepaginacion**

En este caso voy a cargar todas las paginas de código de una rutina determina, como se cuales paginas usa, se usa un **método estadístico**, el S.O. guarda en un archivo las referencias usadas.

➤ **Presacrificio**

En este caso tengo que evitar quedarme con la pila de M.L. vacía.

Para esto debo presacrificar próximas P.V. Esto el S.O. lo hace en momentos de cambio de contexto.

Esto no significa sacarla de la RAM, sino marcarla, porque si se vuelve a usar no hace falta ir a buscarla al disco.

Memoria Virtual

Memoria RAM

Hoy en día los equipos prácticamente les sobra memoria

Entonces porque la M.V.?, ya dijimos que tiene enormes ventajas

En algunos casos podemos jugar con algunos parámetros del S.O. para usar + o – swap

Pero el S.O. al tener mucha RAM de sobra usa muchas veces esta memoria como cache del disco rígido.

Fin del Tema