

OBJETIVOS

- Describir los beneficios del sistema de memoria virtual
- Explicar los conceptos de paginado por demanda, algoritmos de reemplazo de páginas y alocación de cuadros de páginas
- Discutir el principio del modelo de working-set

KMC © 2019

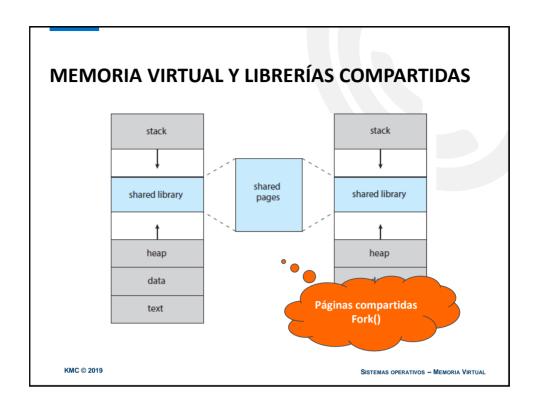
BASE

- Memoria Virtual separación de la memoria lógica del usuario de la memoria física.
 - OSolo parte del programa necesita estar en memoria para su ejecución.
 - oEl espacio de direcciones lógicas puede ser más grande que el espacio de direcciones físicas.
 - Permite a varios procesos compartir el espacio de direcciones.
 - OLa creación de procesos sea más eficiente
- La memoria virtual puede ser implementada vía:
 - oPaginado por demanda
 - OSegmentación por demanda

KMC © 2019

SISTEMAS OPERATIVOS - MEMORIA VIRTUAL

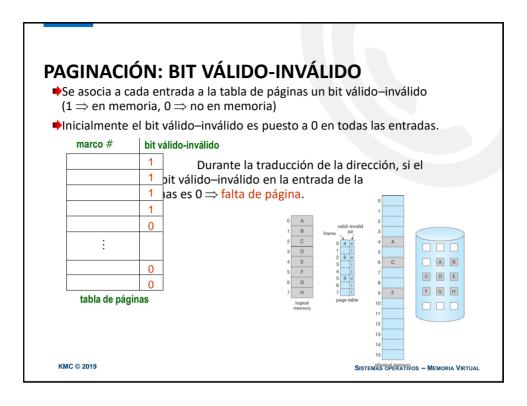
MEMORIA VIRTUAL MÁS GRANDE QUE LA MEMORIA FÍSICA page 0 page 1 page 2 virtual memory MIC © 2019 MEMORIA VIRTUAL MÁS GRANDE QUE LA MEMORIA VIRTUAL SISTEMAS OPERATIVOS – MEMORIA VIRTUAL



PAGINADO POR DEMANDA

- Traer una página a la memoria solo cuando es necesario.
 - Son necesarias menos E/S
 - o Es necesaria menos memoria
 - o Respuesta más rápida
 - Más usuarios
- Cuando una página se necesita ⇒ se la referencia
 - o referencia inválida ⇒ aborto
 - o no está en memoria ⇒ se la trae a memoria
- Intercambiador "perezoso" nunca intercambia en memoria hasta que la página se necesite.
 - El intercambiador (swapper) que trata con páginas es un paginador (pager)

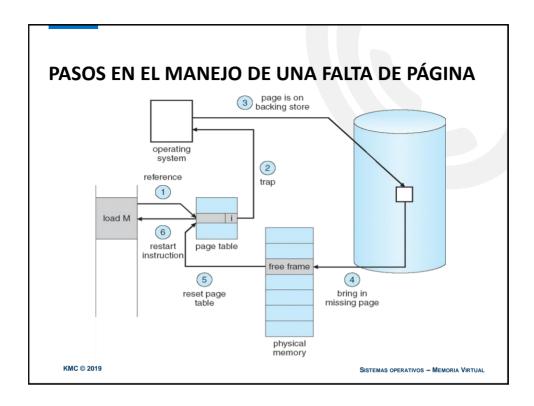
KMC © 2019



FALTA DE PÁGINA

- Si hay una referencia a una página, la primer referencia hace un trap al SO ⇒ falta de página
- El SO mira la tabla para decidir:
 ○Referencia Inválida ⇒ aborto.
 ○No está en memoria.
- Toma un marco libre.
- Lleva la página al marco.
- Reestablece las tablas, bit de validación = 1.
- Reinicio de la instrucción:
 - Movimiento de bloque
 - oLocación con auto incremento/decremento

KMC © 2019



PERFORMANCE DEL PAGINADO POR DEMANDA

- Ritmo de falta de páginas $0 \le p \le 1.0$
 - » si p = 0 no hay falta de páginas
 - » si p = 1, cada referencia es una falta de página
- Tiempo Efectivo de Acceso (TEA)

TEA = (1 - p) x acceso a memoria + p x (sobrecarga de falta de página

+ salida de la página + entrada de la página + sobrecarga de reinicio)

Ejemplo

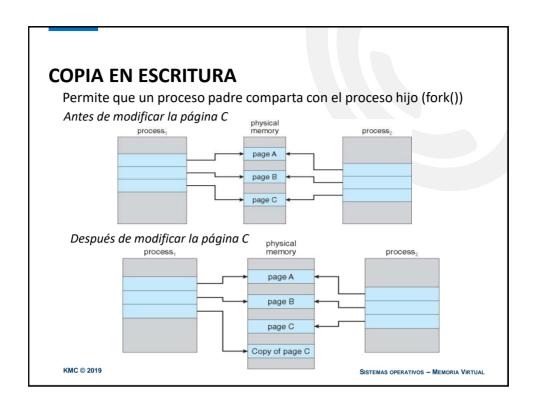
- Tiempo de acceso a memoria = 200 nanosegundos
- Tiempo promedio de servicio de una falta de página = 8 milisegundos.

TEA =
$$(1 - p) \times 200 + p \times (8 \text{ milisegundos})$$

= $(1 - p) \times 200 + p \times 8000000$
= $200 + p \times 7999800$

 Si uno de 1000 causa una falta de página, entonces TEA = 8.2 Microseconds.

KMC ESTO significa una reducción de 40!!



¿QUÉ OCURRE CUANDO NO HAY MARCOS LIBRES?

- Reemplazo de páginas se busca alguna página en memoria que no está en uso y se la intercambia.
 - oalgoritmo
 - operformance se requiere un algoritmo que resulte en un mínimo número de falta de páginas.
- Algunas páginas pueden ser quitadas o volcadas en memoria varias veces.

KMC © 2019

REEMPLAZO DE PÁGINAS

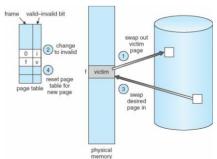
- Se previene sobrealocación de memoria por modificación de la rutina de servicio de falta de páginas para incluir el reemplazo de páginas.
- Uso del bit de modificado ("sucio") para reducir la sobrecarga de la transferencia de páginas – solo las páginas modificadas son escritas en disco.
- El reemplazo de páginas completa la separación entre memoria lógica y memoria física puede ser provista una gran memoria lógica en una pequeña memoria física.

KMC © 2019

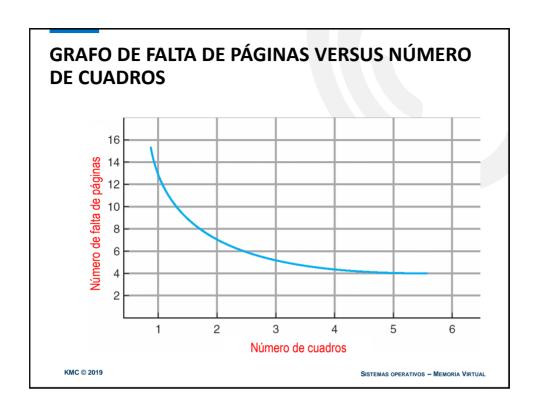
SISTEMAS OPERATIVOS - MEMORIA VIRTUAL

REEMPLAZO DE PÁGINAS

- 1. Encontrar la locación de la página deseada en el disco
- 2. Encontrar un marco libre:
 - Si hay un marco libre, usarlo
 - Si no hay marco libre usar un algoritmo de reemplazo de página para seleccionar el marco **víctima.**
- Traer la página deseada al marco libre, modificar la tabla de páginas y la tabla de marcos.
- 4. Reiniciar el proceso



KMC © 2019



ALGORITMOS DE REEMPLAZO DE PÁGINAS

- Procuran un ritmo de falta de páginas bajo.
- Se evalúa el algoritmo ensayándolo sobre una secuencia particular de referencias a memoria (secuencia de referencia) y computando el número falta de páginas en la secuencia.
- En el ejemplo, la secuencia es

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1

KMC © 2019

ALGORITMO PRIMERO EN ENTRAR—PRIMERO **EN SALIR (FIFO)**

- Secuencia de referencia: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5
- 3 marcos (hay 3 páginas en memoria al mismo tiempo por proceso)

```
1
                                      9 faltas de páginas
2
   2
      2
3
   3
             2 2 2 2 2
```

4 marcos

```
2
                                   10 faltas de páginas
                3 2 2 2 2
3
                            3
4
```

- Reemplazo FIFO Anomalía de Belady
 - más marcos ⇒ menos faltas de páginas

KMC © 2019

SISTEMAS OPERATIVOS - MEMORIA VIRTUAL

ALGORITMO ÓPTIMO

- Reemplace la página que no será usada por un período largo de tiempo.
- Ejemplo con 4 marcos

- 1 1 1 1 1 1 4 4 2 2 2 2 2 2 2 2
- 6 faltas de páginas
- 3 3 3 3 3 3 3
- 4 4 5 5 5 5 5 5
- •¿Cómo se conoce esto?
- Usado para medir como se comporta un algoritmo.

KMC @ 2019

ALGORITMO MENOS RECIENTEMENTE USADO (LRU)

• Secuencia de referencia: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

1 1 1 1 1 1 1 5 2 2 2 2 2 2 2 2 2 2 3 3 3 5 5 5 5 4 3

4 4 4 4 4 3 3 4

8 faltas de páginas

- Implementación del contador
 - Cada entrada a la tabla de páginas tiene un contador; cada vez que la página es referenciada se copia el reloj en el contador.
 - Cuando la página necesita ser cambiada, mira los contadores para determinar cuales hay que cambiar.

KMC © 2019

SISTEMAS OPERATIVOS - MEMORIA VIRTUAL

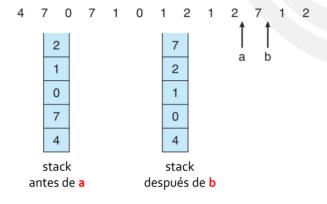
ALGORITMO LRU

- Implementation por Stack mantiene un stack de números de páginas en forma de una lista doblemente enlazada:
 - Página referenciada:
 - se mueve al tope
 - Requiere cambios de punteros
 - No se necesita buscar para realizar el reemplazo

KMC © 2019

USO DEL STACK PARA REGISTRAR LAS REFERENCIAS A PÁGINAS MÁS RECIENTES

Secuencia de referencia



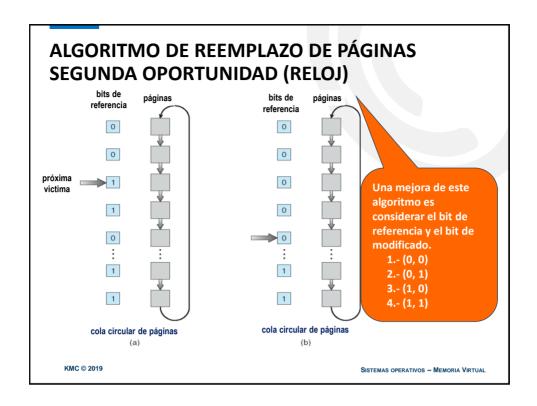
KMC © 2019

SISTEMAS OPERATIVOS - MEMORIA VIRTUAL

ALGORITMOS DE APROXIMACIÓN A LRU

- Bit de referencia. Con cada página se asocia un bit, inicialmente= 0
 - Cuando la página es referenciada el bit es puesto a 1.
 - Reemplace aquella en la cual es 0 (si existe). No se conoce el orden.
- Segunda oportunidad
 - Necesita el bit de referencia.
 - Reemplazo circular (Clock).
 - Si la página a ser reemplazada (en orden circular) tiene el bit de referencia = 1 entonces:
 - Se pone el bit de referencia en 0.
 - Se deja la página en memoria.
 - Se reemplaza la siguiente página (en el orden circular), sujeta a las mismas reglas.

KMC © 2019



ALGORITMOS DE CUENTA

- Se mantiene un contador del número de referencias que han sido hechas a la misma página.
- Algoritmo LFU: reemplaza la página con la menor cuenta.
- Algoritmo MFU: reemplaza la página con la mayor cuenta, basado en el argumento que la página con la cuenta más chica fue recién puesta y todavía tiene que ser usada.

KMC © 2019

ALOCACIÓN DE MARCOS

- Cada proceso necesita un número mínimo de páginas.
- Ejemplo: IBM 370 6 páginas para manejar la instrucción SS MOVE:
 - la instrucción es de 6 bytes, puede expandirse a 2 páginas.
 - 2 páginas para manejar desde.
 - 2 páginas para manejar hacia.
- Dos esquemas de alocación.
 - alocación fija
 - alocación con prioridad

KMC © 2019

SISTEMAS OPERATIVOS - MEMORIA VIRTUAL

ALOCACIÓN FIJA

- Alocación igualitaria p.e., si hay 100 marcos y 5 procesos, a cada uno se les da 20 páginas.
- Alocación Proporcional Aloca de acuerdo al tamaño del proceso.

$$S = \sum S_i$$

m = número de marcos

$$a_i = alocación para $p_i = \frac{s_i}{s} \times m$$$

$$m = 64$$

$$s_{i} = 10$$

$$s_2 = 127$$

$$a_1 = \frac{10}{137} \times 64 \approx 5$$

$$a_2 = \frac{127}{137} \times 64 \approx 59$$

KMC © 2019

ALOCACIÓN CON PRIORIDAD

- Se usa un esquema de alocación proporcional usando prioridades antes que tamaño.
- Si el proceso P_i genera una falta de página
 - Se selecciona para reemplazar uno de sus marcos.
 - Se selecciona para reemplazar un marco de un proceso con menor número de prioridad.

KMC © 2019

SISTEMAS OPERATIVOS - MEMORIA VIRTUAL

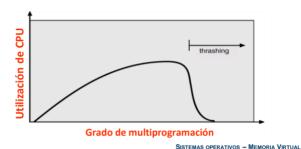
REEMPLAZO GLOBAL VS. LOCAL

- Reemplazo global el proceso selecciona un marco de reemplazo de todos los marcos; un proceso puede tomar los marcos de otro.
- Reemplazo local cada proceso selecciona de su propio conjunto el marco a reemplazar.

KMC © 2019

THRASHING

- Si un proceso no tiene suficientes páginas, el ritmo de falta de páginas es muy alto. Esto lleva a:
 - baja utilización de CPU.
 - el SO piensa que es necesario incrementar el grado de multiprogramación.
 - otro proceso se agrega al sistema.
- Thrashing = un proceso está ocupado haciendo solamente intercambio de páginas.



KMC © 2019

THRASHING

- ¿Por qué trabaja el paginado? Modelo de Localidad
 - El proceso migra desde una localidad a otra.
 - Las localidades se pueden solapar.
- ¿Por qué ocurre el thrashing?
- Σ tamaño de la localidad > tamaño total de memoria

KMC © 2019

MODELO DE CONJUNTO DE TRABAJO (WORKING-SET)

El modelo está basado en la localidad.

- Δ = ventana working-set = un número fijo de referencias de páginas. Ejemplo: 10,000 instrucciones
- WSS_i (working set del proceso P_i) = número total de páginas referenciadas en el más reciente Δ (varía en el tiempo)
 - \Rightarrow si \triangle es demasiado chico no acompaña la localidad.
 - \Rightarrow si \triangle es demasiado grande acompaña varias localidades.
 - \Rightarrow si $\Delta = \infty \Rightarrow$ acompañará al programa entero.
- $D = \Sigma WSS_i \equiv$ demanda total de marcos
- si $D > m \Rightarrow$ Thrashing

m=# de marcos de memoria

Política: si D > m, entonces suspende uno de los procesos.

KMC © 2019

SISTEMAS OPERATIVOS - MEMORIA VIRTUAL

MODELO DE WORKING-SET

Referencias a la tabla de páginas

...2615777751623412344434344413234443444...

 $WS(t_1) = \{1,2,5,6,7\}$

 $WS(t_2) = \{3,4\}$

 $\Delta = 10$

KMC © 2019

CONTROL DEL WORKING SET

- Aproximar con un intervalo de tiempo + bit de referencia
- Ejemplo: $\Delta = 10,000$
 - Las interrupciones del Timer se producen cada 5000 unidades de tiempo.
 - Se mantienen en memoria 2 bits por cada página.
 - Siempre que el timer interrumpe copia e inicializa los valores de todos los bits de referencia a 0.
 - Si uno de los bits en memoria = 1 ⇒ página en el working set.
- ¿Por qué no es completamente preciso?
- Mejora = 10 bits e interrupción cada 1000 unidades de tiempo.

KMC © 2019

SISTEMAS OPERATIVOS - MEMORIA VIRTUAL

ESQUEMA DE FRECUENCIA DE FALTA DE PÁGINA



Número de cuadros

- La idea es establecer un ritmo "aceptable" de falta de páginas.
 Si el ritmo actual es demasiado bajo, los procesos pierden marcos.
 - OSi el ritmo actual es demasiado alto, el proceso gana marcos.

KMC © 2019

OTRAS CONSIDERACIONES - PREPAGINADO

- Prepaginado
 - Para reducir el gran número de falta de páginas que ocurren en el inicio del proceso
 - Se necesitará prepaginar todas o algunas páginas del proceso antes de ser referenciadas
 - Pero si las páginas prepaginadas no son usadas se incurrió en gasto de E/S y memoria
 - Suponga que s páginas son prepaginadas y α de esas páginas son usadas
 - Es el costo de salvar s* α faltas > ó < qué el costo de prepaginar s* (1- α) páginas no necesarias?
 - α cercano a cero \Rightarrow prepaginado pierde

KMC © 2019

SISTEMAS OPERATIVOS - MEMORIA VIRTUAL

OTRAS CONSIDERACIONES – TAMAÑO DE PÁGINA

- Selección del tamaño de página
 - fragmentación
 - tamaño de tabla
 - sobrecarga de E/S
 - localidad

KMC © 2019

OTRAS CONSIDERACIONES – ESTRUCTURA DEL PROGRAMA

Estructura de programa
 Arreglo de enteros A[1024, 1024]
 Cada fila está almacenada en una página
 Un cuadro

■ Programa 1 **for** j := 1 **to** 1024 **do for** i := 1 **to** 1024 **do** A[i,j] := 0;

1024 x 1024 faltas de páginas

■ Programa 2 **for** i := 1 **to** 1024 **do for** j := 1 **to** 1024 **do** A[i,j] := 0;

1024 faltas de páginas

• Fijación para E/S y direccionamiento

KMC © 2019

SISTEMAS OPERATIVOS - MEMORIA VIRTUAL

OTRAS CONSIDERACIONES - FIJACIÓN DE E/S

- Fijación de E/S Algunas veces las páginas deben ser fijadas en la memoria
- Considere E/S Las páginas que son usadas para copiar un archivo desde un dispositivo deben ser fijadas para no ser seleccionadas por el algoritmo de reemplazo de páginas

KMC © 2019

EJEMPLOS DE SISTEMAS OPERATIVOS

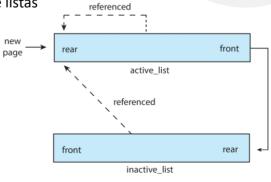
- Linux
- Windows
- UNIX-SVR4
- Solaris

KMC © 2019

SISTEMAS OPERATIVOS - MEMORIA VIRTUAL

LINUX

- Usa demanda de páginas, aloca las páginas de una lista de marcos libres.
- Usa política de reemplazo global, similar a la aproximación de LRU utilizando el algoritmo del Reloj.
- Mantiene dos tipos de listas
 - Lista de Activa
 - Lista de Inactive



KMC © 2019

WINDOWS

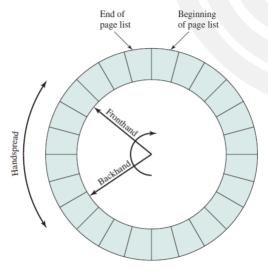
- ✓ Demanda de páginas con **clustering**. Es decir trae las páginas que rodean a la página demandada.
- ✓ Política de reemplazo global ó local, similar a la aproximación de LRU utilizando el algoritmo del Reloj.
- ✓ Mantiene una lista de marcos libres.
- ✓A los procesos se les asigna un working set mínimo y un working set máximo.
- ✓ Cuando la cantidad de memoria libre en el sistema cae bajo determinado umbral, se activa en forma automática hasta restaurar la cantidad de memoria libre.
- ✓ Este ajuste automático remueve páginas de los procesos que están excedidos de su working set mínimo.

KMC © 2019

SISTEMAS OPERATIVOS - MEMORIA VIRTUAL

UNIX-SVR4 Formato Administración de Memoria Entrada en la tabla de páginas Mod- Refe-Pro-Page frame number Age Valid rence tect Descriptor de bloque de disco Device block number Swap device number Type of storage Entrada en la tabla de dato de página frame Block Pfdata Reference Logical Page state number count device pointer Entrada en la tabla de swap-use Reference Page/storage unit number count KMC @ 2019 SISTEMAS OPERATIVOS - MEMORIA VIRTUAL

UNIX- SVR4: ALGORITMO REEMPLAZO DE PÁGINA BASADO EN EL RELOJ



KMC © 2019

SISTEMAS OPERATIVOS - MEMORIA VIRTUAL

UNIX-SOLARIS-LINUX

- √ Mantiene una lista de páginas libres para asignar a procesos en falta
- ✓ Lotsfree parámetro umbral (cantidad de memoria libre) para comenzar a paginar
- ✓ Desfree parámetro umbral para incrementar el paginado
- √ Minfree parámetro umbral para ser intercambiadas las páginas
- √ El paginado es realizado por un proceso pageout
- ✓ Pageout barre las páginas usando un algoritmo de reloj modificado
- √ Scanrate es la frecuencia con que las páginas son barridas. Estos rangos varían entre slowscan y fastscan
- ✓ La frecuencia de llamado a *pageout* depende de la cantidad de memoria libre disponible.

KMC © 2019

Bibliografía:

- Silberschatz, A., Gagne G., y Galvin, P.B.; "*Operating System Concepts*", 7^{ma} Edición. 2009, 9^{na}. Edición 2012, 10^{ma}. Edición 2018.
- Stallings, W. "Operating Systems: Internals and Design Principles", Prentice Hall, 7^{ma}Edición, 2011; 8^{va.} Edición, 2014, 9^{na.} Edición 2018.

KMC © 2019