Memoria Virtual

Eloy Anguiano Rey
eloy.anguiano@uam.es

Ana González
ana.marcos@uam.es

Escuela Politécnica Superior
Universidad Autónoma de Madrid

Parte I

Memoria virtual

Introducción
Conceptos

- Concepto de **Memoria Virtual**: Método para conseguir que la suma de los espacios de pila, datos y texto de un programa pueda ser mayor que el tamaño físico de la memoria disponible para él. (Fotheringham, 1961)
- Cada proceso se asigna un área de direcciones contiguo.
- El SO mantiene en memoria solamente las partes del programa que se están utilizando y mantiene en disco (intercambiadas) el resto.
  - **Conjunto residente**: La parte del proceso que está en la memoria principal o memoria real.
  - Un usuario percibe en potencia una memoria mucho mayor que está en el disco (**Memoria Virtual**)
- Sirve para **sistemas mono y multiprogramados**.

Introducción
Conceptos

Ventajas

- Permite **optimizar el uso de la memoria**:
  - Mantiene **más procesos** en memoria principal.
    - Permite la multiprogramación de forma muy efectiva
  - Mantiene en disco partes del proceso poco usadas (rutinas de atención a errores poco frecuentes, funciones de uso esporádico, datos no usados, ...)
- Permite que un proceso sea más grande que toda la memoria principal.
- Se encarga de todo el sistema operativo.

Introducción
Conceptos

Problemas

- Fallos de direccionamiento
  1. Se genera una interrupción indicando el fallo de acceso a memoria
  2. El proceso pasa a **bloqueado**, y el SO a ejecución
  3. El SO emite una solicitud de E/S al disco
  4. El SO expide otro proceso para que se ejecute
  5. Cuando la interrupción de E/S indica que llegó el fragmento que provocó el fallo, se actualiza la memoria principal.
  6. Se devuelve el control al S.O., que pasa el proceso a **listo**. El planificador decidirá cuándo debe pasar a ejecución.
- Posible hiperpaginación (hyperthrashing)
  - El SO más tiempo intercambiando fragmentos que ejecutando.

Conceptos

Tablas de páginas

Memoria Asociativa

Páginas invertidas

Parte II

Paginación

Conceptos

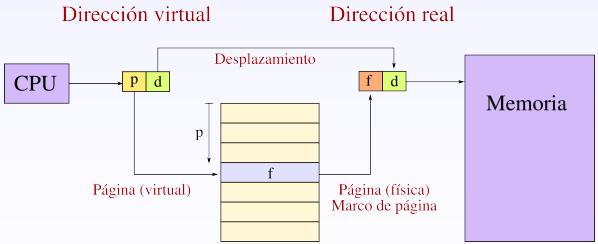
- La **memoria física** se divide en bloques de tamaño fijo que llamamos **marcos**.
- La **memoria virtual** se divide en bloques del mismo tamaño llamados **páginas** (los procesos se dividen en páginas).
- Al ejecutar un proceso se cargan sus páginas en los marcos disponibles. La vinculación de direcciones requiere soporte por **hardware (Manejador de Memoria)**.
- La paginación **remedia la fragmentación externa**, pero no la fragmentación interna.
- Tabla de páginas:
  - Normalmente una por proceso
  - Requiere **soporte por hardware**: manejador de memoria (MMU)
- Un intento de acceso a una página virtual que no esté asociada a un marco produce un señalamiento al SO (trap), llamado **fallo de página**.

Conceptos

Acciones de respuesta del SO en un fallo de página

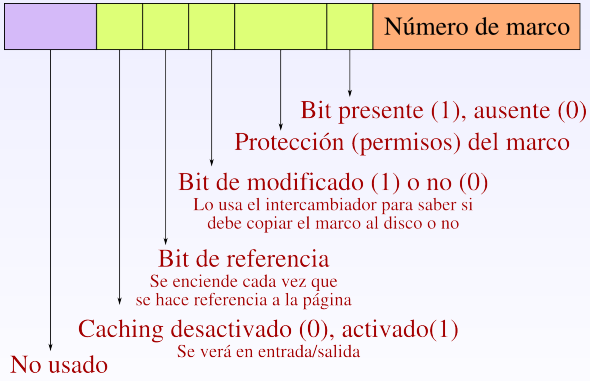
1. Si no hay marcos disponibles se selecciona una página poco usada del proceso.
2. Si no hay marcos disponibles se intercambia la página a disco.
3. Asigna el marco de la página liberada a la página virtual que se intenta acceder.

Esto supone una forma de reasignación dinámica por bloques de las direcciones de memoria del proceso



Tablas de páginas

- Son tablas que contienen (para cada proceso) el número de marco que corresponde a cada página virtual del proceso.
- Estructura de una entrada de la Tabla de Procesos:



Tablas de páginas

Memoria Virtual

Conceptos

Tablas de páginas

Aspectos de diseño

Multinivel

De un nivel: Ejemplo DEC PDP–11

De dos niveles: ejemplo VAX

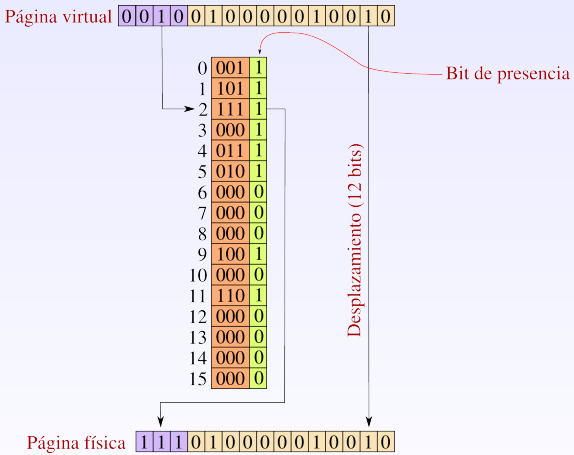
De tres niveles: ejemplo SPARC

Rendimiento de un sistema de paginación

Memoria Asociativa

Páginas invertidas

- El tamaño de página viene definido por el hardware y suele ser una potencia de 2 que varía entre 512 kB y 16 MB.
- Ej. simplificado: Direccionamiento de un espacio virtual de 64 kB, distribuido en 16 páginas de 4 kB.



Tablas de páginas

Conceptos

Tablas de páginas

Aspectos de diseño

Multinivel

De un nivel: Ejemplo DEC PDP–11

De dos niveles: ejemplo VAX

De tres niveles: ejemplo SPARC

Rendimiento de un sistema de paginación

Memoria Asociativa

Páginas invertidas

- Son tablas que contienen (para cada proceso) el número de marco que corresponde a cada página virtual del proceso
- Normalmente **una tabla de páginas por proceso**
- La tabla de página tiene longitud variable (en función del tamaño del proceso)
- Está **cargada en memoria principal** y si es muy grande la tabla, puede estar sujeta a paginación $\Rightarrow$ Tablas de páginas a multiniveles

Al ejecutar el proceso ...

1. La dirección de comienzo de la tabla de páginas para el proceso se mantiene en un registro
2. Se cargan las páginas necesarias en los marcos disponibles

Tablas de páginas
Aspectos de diseño

Memoria Virtual

Conceptos

Tablas de páginas

Aspectos de diseño

Multinivel

De un nivel: Ejemplo DEC PDP–11

De dos niveles: ejemplo VAX

De tres niveles: ejemplo SPARC

Rendimiento de un sistema de paginación

Memoria Asociativa

Páginas invertidas

- El **tiempo de asociación** debe ser reducido
  - Soluciones basadas completamente en hardware (utilizando registros) son las más rápidas, pero esto sólo es válido si las tablas son pequeñas.
- Cuanto menor sea el tamaño de página, menor será la cantidad de fragmentación interna.
- Cuanto menor sea la página, mayor será el número de páginas que se necesitan por proceso.
- Un número mayor de páginas por proceso significa que las tablas de páginas serán mayores.
- Esto puede significar que una gran parte de las tablas de páginas de los procesos activos deben estar en la memoria virtual.

Tablas de páginas
Aspectos de diseño

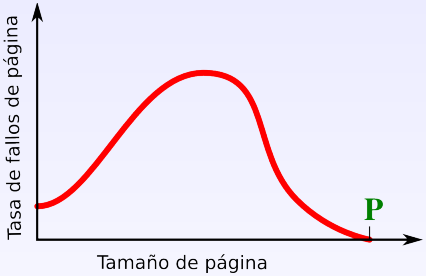
Conceptos
Tablas de páginas
Aspectos de diseño
Multinivel
De un nivel: Ejemplo DEC PDP–11
De dos niveles: ejemplo VAX
De tres niveles: ejemplo SPARC
Rendimiento de un sistema de paginación
Memoria Asociativa
Páginas invertidas

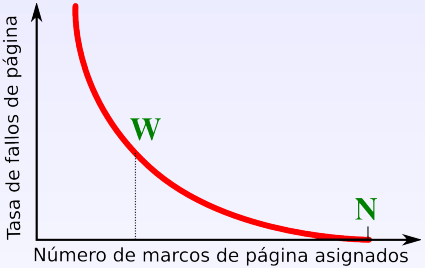
- La memoria secundaria está diseñada para transferir eficazmente los bloques de datos de mayor tamaño, de manera que es propicia para tamaños de página mayores.
- Si el tamaño de página es muy pequeño, estarán disponibles en la memoria principal un gran número de páginas para cada proceso.
- Después de un tiempo, todas las páginas de la memoria contendrán parte de las referencias más recientes del proceso. La tasa de fallos de página será menor.
- Cuando se incrementa el tamaño de la página, cada página individual contendrán posiciones cada vez más distantes de cualquier referencia reciente. La tasa de fallos será mayor.

Tablas de páginas
Aspectos de diseño

Conceptos
Tablas de páginas
Aspectos de diseño
Multinivel
De un nivel: Ejemplo DEC PDP–11
De dos niveles: ejemplo VAX
De tres niveles: ejemplo SPARC
Rendimiento de un sistema de paginación
Memoria Asociativa
Páginas invertidas

El comportamiento típico de la paginación en un programa es el siguiente:





Donde:

- **P**: es el tamaño del proceso completo.
- **W**: es el tamaño del conjunto de trabajo.
- **N**: es el número total de páginas del proceso.

Tablas de páginas
Aspectos de diseño

Conceptos
Tablas de páginas
Aspectos de diseño
Multinivel
De un nivel: Ejemplo DEC PDP–11
De dos niveles: ejemplo VAX
De tres niveles: ejemplo SPARC
Rendimiento de un sistema de paginación
Memoria Asociativa
Páginas invertidas

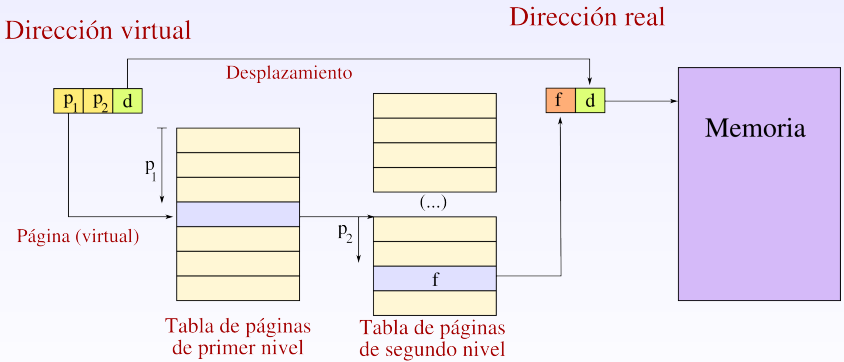
Soluciones al tamaño de tablas grandes

- Paginación multinivel
  - Una parte de las tablas de páginas deben estar en la memoria virtual, sujetas a paginación también
- Tablas de páginas invertidas

Tablas de páginas
Multinivel

Conceptos
Tablas de páginas
Aspectos de diseño
Multinivel
De un nivel: Ejemplo DEC PDP–11
De dos niveles: ejemplo VAX
De tres niveles: ejemplo SPARC
Rendimiento de un sistema de paginación
Memoria Asociativa
Páginas invertidas

- **Objetivo**: evitar tener siempre en memoria tablas de páginas completas.
- **Solución**: dividir la tabla en sub-tablas y mantener en memoria sólo las que sean necesarias en cada momento.



Tablas de páginas
Multinivel

Memoria Virtual

Conceptos

Tablas de páginas

Aspectos de diseño

Multinivel

De un nivel: Ejemplo DEC PDP–11

De dos niveles: ejemplo VAX

De tres niveles: ejemplo SPARC

Rendimiento de un sistema de paginación

Memoria Asociativa

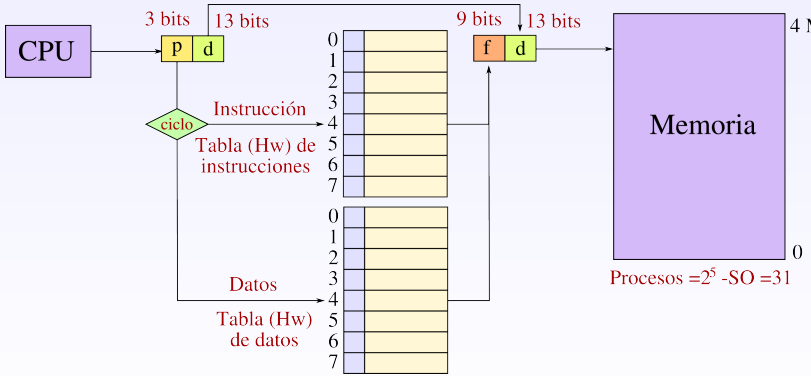
Páginas invertidas

Podemos ver algunos ejemplos de tamaños de página en función del Hardware

Computadora	Tamaño de página
Atlas	512 palabras de 48 bits
Honeywell-Multics	1.024 palabras de 36 bits
IBM 370/XA y 370/ESA	4 Kbytes
Familia VAX	512 bytes
IBM AS/400	512 bytes
DEC Alpha	8 Kbytes
MIPS	de 4 Kbytes a 16 Mbytes
UltraSPARC	de 8 Kbytes a 4 Mbytes
Pentium	de 4 Kbytes a 4 Mbytes
Power Pc	4 Kbytes

Tablas de páginas
De un nivel: Ejemplo DEC PDP–11

- Direccionamiento con 16 bits (memoria lógica de 2^{16}=64Kb).
- Mantiene tablas separadas para instrucciones y datos $\Longrightarrow$ memoria lógica del proceso=128 KB.
- Tamaño de página: 8 KB (2^{13}). Memoria física: 4 MB (2^{22}). 512 páginas.



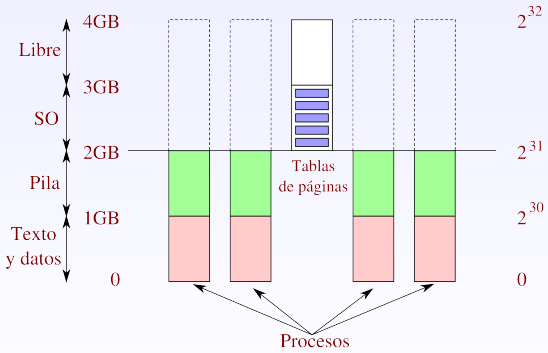
Tablas de páginas
De dos niveles: ejemplo VAX

- Direccionamiento con 32 bits (memoria lógica de $2^{32} = 4$GB).
- Tamaño de página: 512 bytes (2^9).
- Espacio de direcciones lógico divido en cuatro secciones, cada una de 1GB (2^{30} bytes).
- Entrada de la tabla de página: 4 bytes.
- Tamaño de la tabla de páginas ($2^{21} \times 4B$): ¡¡8 MB!!
- Memoria física $<$ 2 MB (en las VAX más pequeñas).
- Direcciones lógicas:

Tablas de páginas
De dos niveles: ejemplo VAX

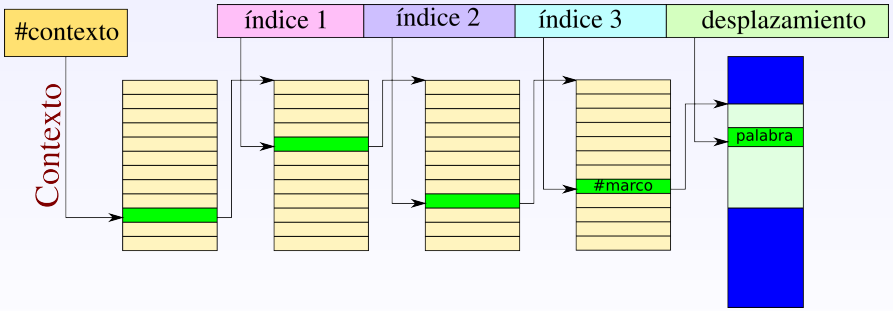
Conceptos
Tablas de páginas
Aspectos de diseño
Multinivel
De un nivel: Ejemplo DEC PDP–11
De dos niveles: ejemplo VAX
De tres niveles: ejemplo SPARC
Rendimiento de un sistema de paginación
Memoria Asociativa
Páginas invertidas

- Para reducir el uso de memoria, las tablas de páginas están también paginadas, y las partes no usadas de la tabla intercambiadas a disco
- Sólo el SO y su tabla de páginas permanece permanentemente en memoria
- Esquema de memoria virtual de los procesos:



Tablas de páginas
De tres niveles: ejemplo SPARC

- Direccionamiento con 32 bits (memoria lógica de 4 GB $=2^{32}$)
- Tamaño de página: 4kB (2^{12}) $\Rightarrow$ 1 millón de páginas.
- Se define un **contexto**, único para cada proceso, y una **tabla de contexto** (hardware) para almacenar un apuntador al comienzo de la tabla de alto nivel del proceso.



Tablas de páginas
Rendimiento de un sistema de paginación

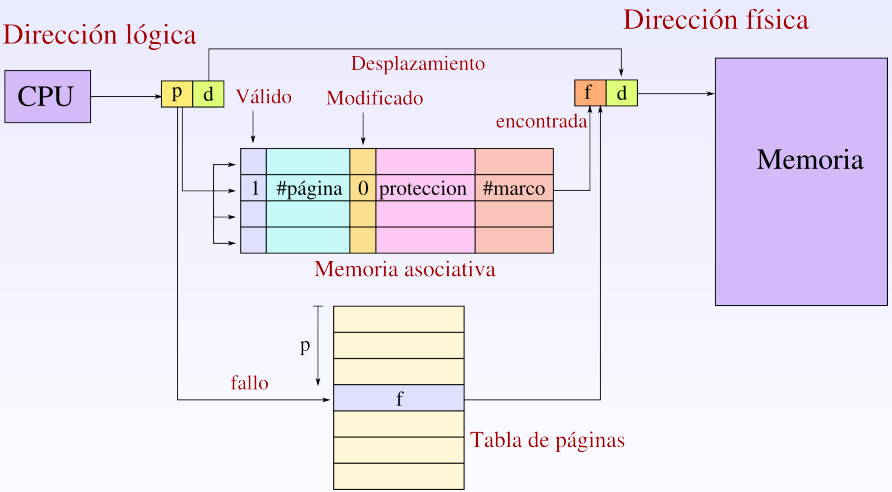
- El tiempo de acceso efectivo a memoria (t_{ae}) para un sistema de paginación de memoria mononivel es $t_{ae} = t_b + (1 - p) \times t_{am} + p \times t_{fallo} + t_{am}$
  donde:
  t_b: tiempo medio de búsqueda en la tabla de páginas
  p: probabilidad de que ocurra un fallo de página
  t_{am}: tiempo de acceso a memoria
  t_{fallo}: tiempo de resolución de un fallo de página. Tiene tres componentes principales:
  1. Atender la interrupción de fallo de página
  2. Traer la página a la memoria
  3. Reiniciar el proceso
- En el caso de tablas multinivel (n niveles):
  $t_{ae} = \sum_{i=1}^{N} [t_{bi} + (1 - p_i) \times t_{am} + p_i \times t_{fallo}] + t_{am}$

Memoria Asociativa
Conceptos

- Solución para acelerar el acceso a los marcos cuando las tablas de procesos son muy grandes (búsqueda lenta) y/o están organizadas en niveles (requiere múltiples accesos a memoria).
- Se basa en la observación de que los procesos acceden normalmente a un número pequeño de páginas (y esporádicamente al resto).
- **Solución**: dotar a los ordenadores con hardware (Memoria Asociativa) para asociar algunas páginas de uso frecuente con páginas físicas sin necesidad de acceder a la tabla de páginas. El tamaño de la memoria asociativa suele ser de 8 a 32 entradas.
- **Proporción de encuentros**: Proporción de accesos a la Memoria Asociativa que son exitosos (la Página Virtual buscada se encuentra en la Memoria).

Memoria Asociativa

Esquema



Memoria Asociativa

Asignación del manejador de memoria

Memoria Virtual

Conceptos

Tablas de páginas

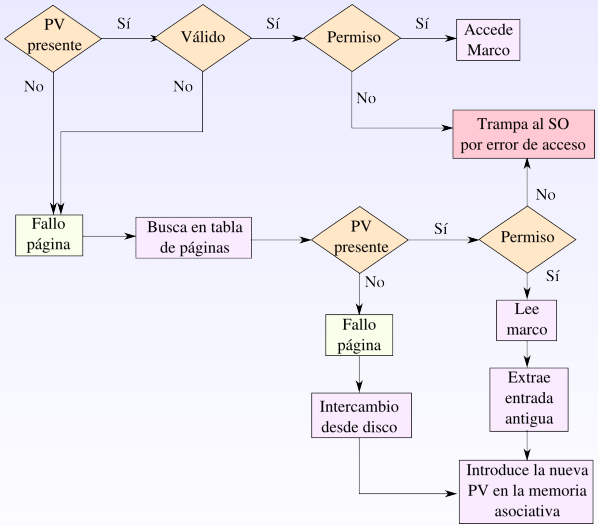
Memoria Asociativa

Conceptos

Esquema

Asignación del manejador de memoria

Páginas invertidas



Páginas invertidas
Conceptos

- En sistemas de direccionamiento por 64 bits (20 Tb= 2^{64}), el número de páginas de (p.ej) 4 KB es 2^{52} . Es impensable poder mantener tablas de páginas de esa longitud, mayor que la memoria física de la mayoría de los sistemas.
- En esos sistemas, el número de marcos físicos es sustancialmente menor, lo que permite organizar la tabla de entradas alrededor de la memoria física en lugar de la memoria virtual (IBM S/38, HP Spectrum).
- La tabla de páginas ahora tiene tantas entradas como páginas físicas (marcos), y cada entrada contiene la dirección virtual que está utilizando dicha página.
- Se utiliza siempre con una memoria asociativa.
- Si una dirección virtual no se encuentra en la tabla invertida, se hace una búsqueda en una tabla convencional, que puede estar en memoria o en disco

Parte III

Políticas del gestor de memoria

Características

Encaminadas a minimizar porcentaje de fallos de páginas para maximizar el rendimiento

¿De qué depende el rendimiento?

- Del tamaño de memoria principal
- De la velocidad relativa de memoria principal y secundaria
- Del tamaño y número de procesos que compiten por los recursos
- Del comportamiento de programas individuales:
  - Aplicación
  - Lenguaje de programación y compilador
  - Estilo de programador
  - Comportamiento dinámico del usuario en programas interactivos

Tipos de políticas

Características

Tipos de políticas

Políticas de Lectura (fetch)

Políticas de Ubicación

Políticas de Reemplazo

Políticas de gestión del conjunto residente

Conjunto de Trabajo

Modelo del conjunto de trabajo

Políticas de Vaciado

Control de carga

No hay una política definitivamente mejor que las otras

- Políticas de Lectura
  - ¿Cuándo cargo una página en memoria principal?
- Políticas de Ubicación
  - ¿Dónde coloco la nueva página?
- Políticas de Reemplazo
  - Si la memoria principal está llena ¿qué página/s reemplazo?
- Gestión del Conjunto Residente
  - ¿Cuánta memoria principal debe asignarse a un proceso cuando se carga?
- Políticas de Vaciado
  - ¿Cuándo escribo una página modificada en memoria secundaria?
- Control de Carga
  - ¿Cuál es el grado de multiprogramación?

Tipos de políticas
Políticas de Lectura (fetch)

Características
Tipos de políticas
Políticas de Lectura (fetch)
Políticas de Ubicación
Políticas de Reemplazo
Políticas de gestión del conjunto residente
Conjunto de Trabajo
Modelo del conjunto de trabajo
Políticas de Vaciado
Control de carga

¿Cuándo cargar una página en memoria principal?

- **Paginación por Demanda**: Se carga una página sólo cuando se produce un fallo en esa página
  - Muchísimos fallos al inicio
  - Estabilización gracias al principio de cercanía
  - Con el tiempo, el número de fallos disminuye hasta un nivel bajo
- **Paginación Previa**: se carga la página que ha producido el fallo y las páginas cercanas a ésta
  - Aprovecha características de los discos

Tipos de políticas
Políticas de Ubicación

Memoria Virtual

Características

Tipos de políticas
Políticas de Lectura (fetch)
Políticas de Ubicación
Políticas de Reemplazo
Políticas de gestión del conjunto residente
Conjunto de Trabajo
Modelo del conjunto de trabajo
Políticas de Vaciado
Control de carga

¿Dónde va a residir una parte de un proceso en la memoria principal?

- Es una decisión importante en sistemas con segmentación
  - Mejor ajuste, primer ajuste, siguiente ajuste, etc
- Carece de importancia en sistemas con paginación (con y sin segmentación)

Tipos de políticas
Políticas de Reemplazo

Si la memoria está ocupada: ¿Qué páginas se eligen para ser reemplazadas?

Gestión del Conjunto Residente

- ¿Número de marcos de página a asignar para cada proceso?
- ¿Reemplazar sólo páginas del propio proceso o de cualquier otro?

Algoritmos de Reemplazo

- Tras un fallo de página y con la memoria principal ocupada al completo, se debe elegir qué página de memoria se lleva a disco para hacer sitio a la nueva página.
- **Objetivo**: reemplazar la página con menor posibilidad de ser referenciada en un futuro cercano:
  - Intentar predecir el comportamiento futuro en función del comportamiento pasado
- Cuanto más elaborada es la política, mayor sobrecarga de Software y Hardware

Tipos de políticas

Políticas de gestión del conjunto residente

Memoria Virtual

Características

Tipos de políticas

Políticas de Lectura (fetch)

Políticas de Ubicación

Políticas de Reemplazo

Políticas de gestión del conjunto residente

Conjunto de Trabajo

Modelo del conjunto de trabajo

Políticas de Vaciado

Control de carga

Tamaño del Conjunto Residente

El S.O. debe decidir cuánta memoria asignar a un proceso (número de páginas a cargar en memoria principal)

Factores positivos que influyen en la decisión

Cuanto menos memoria necesite cada proceso, mayor cantidad de procesos en memoria $\Rightarrow$ mayor probabilidad de procesos listos en memoria

Factores negativos que influyen en la decisión

- Si hay pocas páginas de un proceso en memoria, aumenta la probabilidad de fallos de página
- Por encima de un determinado tamaño, más memoria no tendrá un efecto notable (por principio de cercanía)

Tipos de políticas

Políticas de gestión del conjunto residente

Características
Tipos de políticas
Políticas de Lectura (fetch)
Políticas de Ubicación
Políticas de Reemplazo
Políticas de gestión del conjunto residente
Conjunto de Trabajo
Modelo del conjunto de trabajo
Políticas de Vaciado
Control de carga

Asignación Fija: otorga a cada proceso un número fijo de páginas

- Se decide la cantidad de memoria asignada al proceso en la carga inicial, según el tipo de proceso o directrices del programador o administrador
- Cuando hay fallos de página, siempre se reemplaza una página del mismo proceso

Problemas

- **Asignación por exceso**:
  - Se desperdicia espacio $\Rightarrow$ el proceso no necesita todos los marcos que se le han asignado
  - Procesador ocioso $\Rightarrow$ hay pocos procesos en memoria principal.
- **Asignación por defecto**:
  - Alto porcentaje de fallos de páginas, aunque en el sistema haya marcos vacíos
  - Multiprogramación lenta

Tipos de políticas
Políticas de gestión del conjunto residente

Memoria Virtual

Características
Tipos de políticas
Políticas de Lectura (fetch)
Políticas de Ubicación
Políticas de Reemplazo
Políticas de gestión del conjunto residente
Conjunto de Trabajo
Modelo del conjunto de trabajo
Políticas de Vaciado
Control de carga

Asignación Variable: el número de marcos de un proceso cambia durante su vida

- Cambia en función de la tasa de fallos de página se asignará dinámicamente el número de marcos asignados
- La asignación variable es más potente pero costosa (más trabajo del S.O., que debe evaluar el comportamiento de procesos dinámicamente)

Tipos de políticas

Políticas de gestión del conjunto residente

Características
Tipos de políticas
Políticas de Lectura (fetch)
Políticas de Ubicación
Políticas de Reemplazo
Políticas de gestión del conjunto residente
Conjunto de Trabajo
Modelo del conjunto de trabajo
Políticas de Vaciado
Control de carga

- **Asignación global**: Un proceso puede utilizar marcos que pertenecen a otro proceso (ocurre en asignación con prioridades).
  - Sólo tiene sentido con asignación variable.
- **Asignación local**: Un proceso sólo puede utilizar marcos que le han sido asignados a dicho proceso.
  - Tiene sentido en asignación fija y en asignación variable.
- **Problemas:**
  - Desperdicia espacio si el proceso no necesita todos los marcos que se le han asignado.
  - Si el proceso necesita más marcos de los asignados, el proceso no podrá ejecutarse correctamente, aunque en el sistema haya muchos marcos vacíos.

Tipos de políticas

Políticas de gestión del conjunto residente

Características
Tipos de políticas
Políticas de Lectura (fetch)
Políticas de Ubicación
Políticas de Reemplazo
Políticas de gestión del conjunto residente
Conjunto de Trabajo
Modelo del conjunto de trabajo
Políticas de Vaciado
Control de carga

Asignación fija y de ámbito local

Al cargar un nuevo proceso, se le asigna un número de marcos fijo (depende de aplicación y solicitud del programa)

Desventajas

- Si se asignan pocas páginas se produce un alto porcentaje de fallos de páginas.
- Si se asignan muchas páginas hay pocos procesos en memoria $\Rightarrow$ procesador parado

Tipos de políticas
Políticas de gestión del conjunto residente

Características
Tipos de políticas
Políticas de Lectura (fetch)
Políticas de Ubicación
Políticas de Reemplazo
Políticas de gestión del conjunto residente
Conjunto de Trabajo
Modelo del conjunto de trabajo
Políticas de Vaciado
Control de carga

Asignación variable y de ámbito local

- Al cargar un nuevo proceso, se le asigna un número de marcos (paginación previa o bajo demanda)
- Cuando hay un fallo de página con reemplazo se selecciona una página del proceso que falló
- De vez en cuando, se evalúa la asignación de un proceso y se aumenta o disminuye para mejorar el rendimiento global

Tipos de políticas
Políticas de gestión del conjunto residente

Asignación variable y de ámbito global

Sencilla de implementar

Fallos de página: Nuevo marco libre para el proceso, donde se carga la página. Casi siempre se incrementa la memoria asignada al proceso

Dificultad: La elección del reemplazo se realiza entre todos los marcos

Estrategia: Uso de memoria intermedia para las páginas reemplazadas. Mejora el tiempo recuperación de páginas recientemente reemplazadas.

Cuestiones a resolver

- Tamaño del conjunto residente
- Periodicidad de los cambios del conjunto residente

Solución

Estrategia del **Conjunto de trabajo**

Tipos de políticas
Conjunto de Trabajo

Características
Tipos de políticas
Políticas de Lectura (fetch)
Políticas de Ubicación
Políticas de Reemplazo
Políticas de gestión del conjunto residente
Conjunto de Trabajo
Modelo del conjunto de trabajo
Políticas de Vaciado
Control de carga

- Se utiliza para determinar el tamaño del conjunto residente y el momento de los cambios
- El conjunto de trabajo de un proceso en un instante t es $W(t, \Delta t)$: conjunto de páginas a las que el proceso ha hecho referencia en las últimas Δt unidades de tiempo
- $W(t, \Delta t)$:
  - W crece con Δt
  - W depende del instante t: puede ser 1 o llegar hasta el número total de páginas del proceso
  - Su tamaño puede variar durante la ejecución
  - Crece al iniciarse un proceso
  - Estabilidad (principio de cercanía)
  - Las oscilaciones indican el desplazamiento del programa a otra ubicación

Tipos de políticas
Conjunto de Trabajo

Características
Tipos de políticas
Políticas de Lectura (fetch)
Políticas de Ubicación
Políticas de Reemplazo
Políticas de gestión del conjunto residente
Conjunto de Trabajo
Modelo del conjunto de trabajo
Políticas de Vaciado
Control de carga

- Si el número de marcos disponibles es inferior al tamaño del conjunto de trabajo, se producirán frecuentes fallos de página
- Un proceso hiperpaginado pasa más tiempo intercambiando páginas que ejecutándose, y puede "robar" páginas de otros procesos, provocando su hiperpaginación (hypertrashing o trasiego)
- Consecuencia del hypertrashing: reducción drástica del uso de CPU
- La hiperpaginación se limita con asignación local, y si se asigna a cada proceso un número de marcos suficiente

Problemas

- ¿Cómo calcular el número de marcos que un proceso necesita?
- ¿Cómo calcular tamaño del conjunto de trabajo para un proceso?

Tipos de políticas
Modelo del conjunto de trabajo

Memoria Virtual

Características

Tipos de políticas

Políticas de Lectura (fetch)

Políticas de Ubicación

Políticas de Reemplazo

Políticas de gestión del conjunto residente

Conjunto de Trabajo

Modelo del conjunto de trabajo

Políticas de Vaciado

Control de carga

Finalidad: reducir la tasa de fallos de página

Principio de localidad: las páginas que se utilizan en cada momento se pueden agrupar en grupos que varían a lo largo de la ejecución del proceso. P.e.: una llamada a una función provoca un cambio de localidad

Tipos de políticas
Modelo del conjunto de trabajo

Memoria Virtual

Características
Tipos de políticas
Políticas de Lectura (fetch)
Políticas de Ubicación
Políticas de Reemplazo
Políticas de gestión del conjunto residente
Conjunto de Trabajo
Modelo del conjunto de trabajo
Políticas de Vaciado
Control de carga

Estrategia

- **Supervisar** el conjunto de trabajo de cada proceso
- **Eliminar periódicamente** del conjunto residente las páginas que no pertenezcan a su conjunto de trabajo
- Un proceso **se puede ejecutar sólo si** su conjunto de trabajo esta en memoria principal (en conjunto residente)

Problemas

- El pasado no siempre predice el futuro
- Es impracticable una medida real del conjunto de trabajo de cada proceso
- El valor óptimo de Δt a considerar es desconocido y puede variar

Tipos de políticas
Modelo del conjunto de trabajo

Memoria Virtual

Características
Tipos de políticas
Políticas de Lectura (fetch)
Políticas de Ubicación
Políticas de Reemplazo
Políticas de gestión del conjunto residente
Conjunto de Trabajo
Modelo del conjunto de trabajo
Políticas de Vaciado
Control de carga

Estrategia alternativa

Algoritmo de frecuencia de fallos de página

- Cada página tiene un bit de uso que se pone a 1 cuando accede a la página
- Al producirse fallo de página el S.O. mira el tiempo transcurrido desde el último fallo de página para el proceso
  - Si el tiempo es menor que F (umbral), se añade una página al conjunto residente.
  - En caso contrario:
    - Se descartan las páginas con el bit de uso a 0, reduciéndose el conjunto residente.
    - Se restaura a 0 el valor del bit de uso en las páginas restantes

Problema: periodos de transición entre dos regiones de referencia

El conjunto residente crece antes de que las páginas de la antigua región de referencia hayan sido expulsadas y provoca:

- Aumento del grado de multiprogramación, sobrecargando la CPU.

Tipos de políticas
Modelo del conjunto de trabajo

Características
Tipos de políticas
Políticas de Lectura (fetch)
Políticas de Ubicación
Políticas de Reemplazo
Políticas de gestión del conjunto residente
Conjunto de Trabajo
Modelo del conjunto de trabajo
Políticas de Vaciado
Control de carga

Otra estrategia alternativa

Política de conjunto de trabajo con muestreos en intervalos variables:

Evalúa el conjunto de trabajo mediante muestras temporales:

- Al comienzo de un intervalo de muestreo, se restauran los bits de uso de todas las páginas residentes del proceso
- Al final, sólo las páginas que han sido referenciadas se mantendrán en el conjunto residente para el próximo intervalo de tiempo

Tipos de políticas
Políticas de Vaciado

Memoria Virtual

Características

Tipos de políticas

Políticas de Lectura (fetch)

Políticas de Ubicación

Políticas de Reemplazo

Políticas de gestión del conjunto residente

Conjunto de Trabajo

Modelo del conjunto de trabajo

Políticas de Vaciado

Control de carga

Vaciado por demanda

Vaciado por demanda: Se escribe la página en disco **sólo cuando se va a reemplazar** (antes de leer la siguiente)

Ventaja: Minimiza escrituras en disco

Desventaja: Cada fallo de página produce dos transferencias de página (menor rendimiento)

Vaciado previo

Escribe las páginas modificadas **antes de que se necesiten sus marcos**, escribiéndolas por lotes

Ventaja: Aprovecha las ventajas de escribir en el disco por lotes

Desventaja: Posibles operaciones innecesarias: escritura de páginas que van a ser modificadas antes de ser reemplazadas

Tipos de políticas
Políticas de Vaciado

Memoria Virtual

Características

Tipos de políticas
Políticas de Lectura (fetch)
Políticas de Ubicación
Políticas de Reemplazo
Políticas de gestión del conjunto residente
Conjunto de Trabajo
Modelo del conjunto de trabajo
Políticas de Vaciado
Control de carga

Alternativa: incorporar almacenamiento intermedio

- Páginas reemplazadas a lista de modificadas/no modificadas
- Lista de modificadas a disco periódicamente por lotes y después pasan a la lista de no modificadas

Tipos de políticas
Control de carga

Características
Tipos de políticas
Políticas de Lectura (fetch)
Políticas de Ubicación
Políticas de Reemplazo
Políticas de gestión del conjunto residente
Conjunto de Trabajo
Modelo del conjunto de trabajo
Políticas de Vaciado
Control de carga

Grado de multiprogramación

- Si hay pocos procesos es probable de que todos bloqueados
- Si hay muchos, el tamaño medio del conjunto de trabajo no es el adecuado: aumenta la frecuencia de fallos de página (posible hiperpaginación)

Grado de multiprogramación se debe aumentar cuando el número de fallos de página sea pequeño y disminuir cuando sea grande

Tipos de políticas
Control de carga

Características
Tipos de políticas
Políticas de Lectura (fetch)
Políticas de Ubicación
Políticas de Reemplazo
Políticas de gestión del conjunto residente
Conjunto de Trabajo
Modelo del conjunto de trabajo
Políticas de Vaciado
Control de carga

Criterios de suspensión de procesos

Baja prioridad: Decisión de política de planificación

Muchos fallos de página: Conjunto de trabajo inadecuado

Último activado: Conjunto de trabajo residente no formado

Conjunto residente más pequeño: Penaliza procesos con ubicaciones pequeñas

Proceso mayor: Libera una cantidad de marcos grande

Mayor ventana de ejecución restante: Planificación: primero el proceso más corto

La elección depende de los objetivos y el tipo de programa

Conceptos

Óptimo

LRU (Last Recently Used)

FIFO

Segunda oportunidad

Del reloj

NRU (No Recently Used)

NFU (No Frequently Used)

Almacenamiento Intermedio de páginas

Bloqueo de marcos

Parte IV

Algoritmos de reemplazo

Conceptos

Conceptos
Óptimo
LRU (Last Recently Used)
FIFO
Segunda oportunidad
Del reloj
NRU (No Recently Used)
NFU (No Frequently Used)
Almacenamiento Intermedio de páginas
Bloqueo de marcos

- Tras un fallo de página, el SO debe elegir qué página de memoria deber ser intercambiada disco para hacer sitio a la nueva página que se está solicitando.
- Criterio general: eliminar páginas poco usadas.

Óptimo

Conceptos

Óptimo

LRU (Last Recently Used)

FIFO

Segunda oportunidad

Del reloj

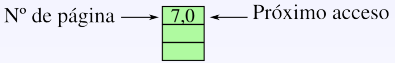
NRU (No Recently Used)

NFU (No Frequently Used)

Almacenamiento Intermedio de páginas

Bloqueo de marcos

- Cada página contiene una etiqueta con el número de instrucciones que transcurrirán hasta el próximo acceso a la página. Se reemplaza la página que tenga la etiqueta más alta.
- **Problema**: Es **irrealizable** ya que el SO no tiene forma de saber cuándo se va a realizar un nuevo acceso a una página. Es posible si se ejecuta en modo simulación, se computan los accesos a las páginas, y esos cómputos se utilizan en sucesivas ejecuciones.



Tiempo	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Página	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0
Marcos	7,x	7,x	7,x	2,5	2,4	2,3	2,2	2,1	2,4	2,3	2,2	2,1	2,2	2,1	2,x	2,x
		0,3	0,2	0,1	0,2	0,1	0,4	4,x	4,x	4,x	0,5	0,4	0,3	0,2	0,1	0,x
			1,B	1,A	1,9	3,4	3,3	3,2	3,1	3,2	3,1	3,x	3,x	1,x	1,x	1,x

LRU (Last Recently Used)

Conceptos

Óptimo

LRU (Last Recently Used)

FIFO

Segunda oportunidad

Del reloj

NRU (No Recently Used)

NFU (No Frequently Used)

Almacenamiento Intermedio de páginas

Bloqueo de marcos

- Implementa una aproximación del algoritmo óptimo, basado en mirar al pasado y a partir de él estimar cuál podría ser el uso de la página.
- Cuando sea necesario, el planificador saca de la tabla que lleva más tiempo sin acceso, de entre las entradas de la tabla.
- Hay varias implementaciones posibles.

LRU (Last Recently Used)

Conceptos
Óptimo
LRU (Last Recently Used)
FIFO
Segunda oportunidad
Del reloj
NRU (No Recently Used)
NFU (No Frequently Used)
Almacenamiento Intermedio de páginas
Bloqueo de marcos

Posibles implementaciones

La tabla se implementa como pila

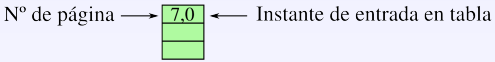
- Pila de números de página que conserva en la salida la página más recientemente usada y en la base la menos recientemente usada
- La actualización de la tabla es muy lenta, aún usando HW especial.

Uso de contadores

- Las entradas de la tabla de páginas tienen un campo de 'tiempo de uso' en el que el Manejador de Memoria escribe el tiempo de cada referencia.
- Cada acceso a memoria requiere una búsqueda en la Tabla de Páginas y una escritura en memoria por cada acceso a memoria.
- Requiere HW especial.

FIFO

- Cada entrada de la tabla tiene un registro asociado con el instante de carga de la página, o (b) el SO mantiene la tabla de páginas en orden de antigüedad (FIFO)
- Cuando se produce un fallo de página y no hay marcos libres, se intercambia a disco la página que lleve más tiempo en la tabla.



Tiempo	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Página	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0
Marcos	7,0	7,0 0,1	7,0 0,1 1,2	2,3 0,1 1,2	2,3 0,1 1,2	2,3 3,5 1,2	2,3 3,5 0,6	4,7 3,5 0,6	4,7 2,8 0,6	4,7 2,8 3,9	0,A 2,8 3,9	0,A 2,8 3,9	0,A 2,8 3,9	0,A 1,D 3,9	0,A 1,D 2,E	0,A 1,D 2,E
Tablas	7	7 0	7 0 1	0 1 2	0 1 2	1 2 3	2 3 0	3 0 4	0 4 2	4 2 3	2 3 0	2 3 0	2 3 0	3 0 1	0 1 2	0 1 2

Segunda oportunidad

Conceptos

Óptimo

LRU (Last Recently Used)

FIFO

Segunda oportunidad

Del reloj

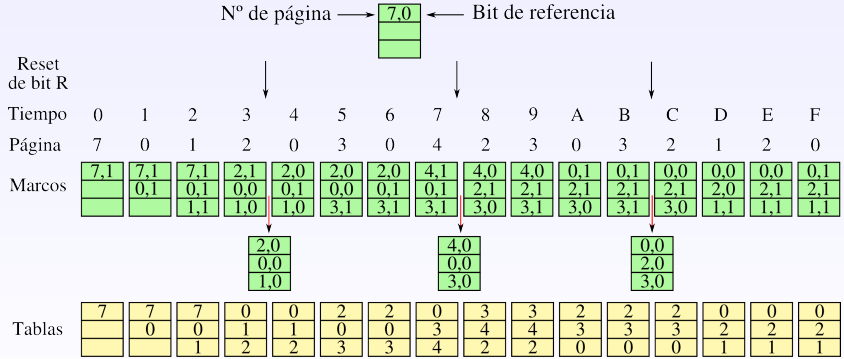
NRU (No Recently Used)

NFU (No Frequently Used)

Almacenamiento Intermedio de páginas

Bloqueo de marcos

- Es una modificación del FIFO, que consiste en revisar el bit de Referencia de la entrada más antigua: si es un 1, se pone a 0 y se sitúa al final de la cola de páginas. Así sucesivamente hasta que se encuentre una página no referenciada.



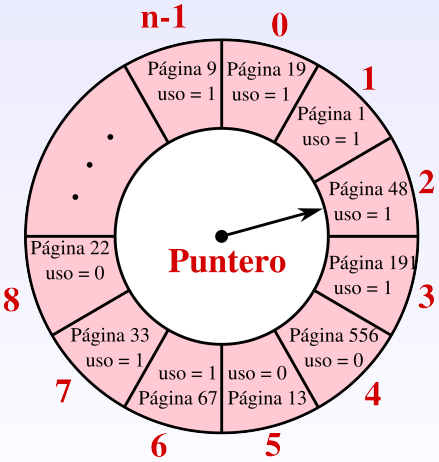
Del reloj

Conceptos
Óptimo
LRU (Last Recently Used)
FIFO
Segunda oportunidad
Del reloj
NRU (No Recently Used)
NFU (No Frequently Used)
Almacenamiento Intermedio de páginas
Bloqueo de marcos

- Las páginas se mantienen en una cola circular, con un apuntador a la páginas más antigua. Si hay un fallo y R=0, la página se retira y se avanza el apuntador. Si R=1 se limpia R se avanza hasta encontrar una página con R=0.
- Es tan sólo una implementación alternativa del algoritmo de la segunda oportunidad, más eficiente por no requerir movimiento de páginas en la tabla de páginas. Son los punteros los que se desplazan, no las entradas de la tabla.

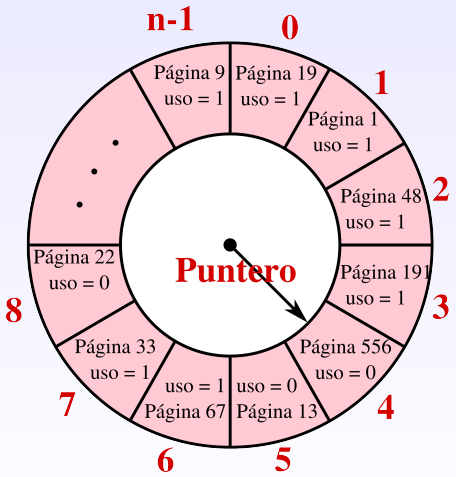
Del reloj

Conceptos
Óptimo
LRU (Last Recently Used)
FIFO
Segunda oportunidad
Del reloj
NRU (No Recently Used)
NFU (No Frequently Used)
Almacenamiento Intermedio de páginas
Bloqueo de marcos



Del reloj

Conceptos
Óptimo
LRU (Last Recently Used)
FIFO
Segunda oportunidad
Del reloj
NRU (No Recently Used)
NFU (No Frequently Used)
Almacenamiento Intermedio de páginas
Bloqueo de marcos



NRU (No Recently Used)

Conceptos

Óptimo

LRU (Last Recently Used)

FIFO

Segunda oportunidad

Del reloj

NRU (No Recently Used)

NFU (No Frequently Used)

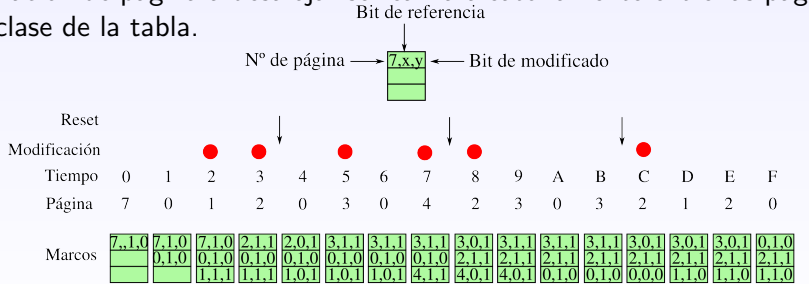
Almacenamiento Intermedio de páginas

Bloqueo de marcos

Utiliza los bits de Referencia (R) y Modificado (M) de las entradas de la tabla de páginas. Se definen cuatro clases de páginas.

- **Clase 0**: Páginas no referenciadas ni modificadas (R=0,M=0).
- **Clase 1**: Páginas no referenciadas, modificadas (R=0,M=1).
- **Clase 2**: Páginas referenciadas no modificadas (R=1,M=0).
- **Clase 3**: Páginas referenciadas, modificadas (R=1,M=1).

La asignación de página a desalojar se realiza aleatoriamente entre las páginas de menor clase de la tabla.



NFU (No Frequently Used)

Conceptos
Óptimo
LRU (Last Recently Used)
FIFO
Segunda oportunidad
Del reloj
NRU (No Recently Used)
NFU (No Frequently Used)
NFU con maduración
Almacenamiento Intermedio de páginas
Bloqueo de

- Implementa una aproximación del algoritmo LRU, utilizando exclusivamente Software:
  - Se activa un contador software por cada página
  - En cada interrupción de reloj, el SO suma el bit R (0 ó 1) al contador.
  - Cuándo existe un fallo de página se selecciona la página con el contador más bajo y se envía a disco.

R	contador	R	contador	R	contador	R	contador
1	00000001	1	00000011	1	00000111	1	00001111
0	00000000	1	00000001	1	00000011	1	00000111
1	00000001	0	00000010	0	00000100	0	00001000
0	00000000	0	00000000	1	00000001	1	00000011
1	00000001	1	00000011	0	00000110	0	00001100
1	00000001	0	00000010	1	00000101	1	00001011
	T=1		T=2		T=3		T=4

- **Problema**: Tiene memoria

NFU (No Frequently Used)
NFU con maduración

- **Solución al problema de memoria**: Maduración.
  - Se activa un contador software por cada página.
  - En cada interrupción de reloj se desplaza el contador un bit a la derecha y luego se suma el bit R al bit más significativo del contador.

R	contador	R	contador	R	contador	R	contador
1	10000000	1	11000000	1	11100000	1	11110000
0	00000000	1	10000000	1	11000000	1	11100000
1	10000000	0	01000000	0	00100000	0	00010000
0	00000000	0	00000000	1	10000000	1	11000000
1	10000000	1	11000000	0	01100000	0	00110000
1	10000000	0	01000000	1	10100000	1	11010000
	T=1		T=2		T=3		T=4

NFU (No Frequently Used)
NFU con maduración

Conceptos
Óptimo
LRU (Last Recently Used)
FIFO
Segunda oportunidad
Del reloj
NRU (No Recently Used)
NFU (No Frequently Used)
NFU con maduración
Almacenamiento Intermedio de páginas
Bloqueo de

Ventaja NRU sobre el reloj

Entre las páginas menos usadas, las no modificadas se reemplazan antes que las modificadas lo que implica un ahorro de tiempo de escritura en disco

Reloj y NRU

Aunque haya que escribir en disco, se supone que la página reemplazada tardará en ser usada de nuevo (principio de cercanía)

Almacenamiento Intermedio de páginas

Conceptos
Óptimo
LRU (Last Recently Used)
FIFO
Segunda oportunidad
Del reloj
NRU (No Recently Used)
NFU (No Frequently Used)
Almacenamiento Intermedio de páginas
Bloqueo de marcos

- Estrategia para mejorar rendimiento de paginación y utilizar política de reemplazo de páginas sencilla
- Creación de dos listas:
  - Lista de páginas libres, para páginas sin modificar
  - Lista de páginas modificadas
- La página a reemplazar no se quita de la memoria principal:
  - Se suprime su entrada de la tabla de páginas (bit ausente = 0)
  - Se pone en la lista de páginas libres o modificadas

Ventajas del almacenamiento intermedio de páginas

- Reduce el coste de cargar páginas que han sido reemplazadas hace poco tiempo
- Lista de páginas libres = lista de marcos disponibles para cargar páginas (se mantiene un mínimo)
- Las páginas modificadas son reescritas por bloques en vez de una a una implica quereduce el número de E/S y la cantidad de tiempo de acceso al disco.

Bloqueo de marcos

Conceptos
Óptimo
LRU (Last Recently Used)
FIFO
Segunda oportunidad
Del reloj
NRU (No Recently Used)
NFU (No Frequently Used)
Almacenamiento Intermedio de páginas
Bloqueo de marcos

Algunos marcos de la memoria principal pueden bloquearse imponiendo una estricción a la política de reemplazo

- La página de un marco bloqueado no puede ser reemplazada
- Utilizado para el núcleo del S.O. y otras áreas críticas
- Se asocia bit de bloqueo a cada marco (en tabla de marcos o tabla de páginas actual)

Asignación de marcos en sistemas monoprogramados

Número de marcos

Tamaño de página

Área de almacenamiento en disco

Administración de fallos de página

Parte V

Criterios de diseño de paginación

Asignación de marcos en sistemas monoprogramados

Asignación de marcos en sistemas monoprogramados

Número de marcos

Tamaño de página

Área de almacenamiento en disco

Administración de fallos de página

- Se asignan páginas al SO y las restantes páginas libres se van asignando tras las correspondientes fallos de página a las páginas del proceso
- Una vez llenas todas las páginas, el manejador utiliza uno de los algoritmos de asignación por demanda para intercambiar páginas de la memoria y habilitar huecos para las nuevas páginas.
- Variación: el SO reserva parte de su espacio (libre) para apoyar la paginación. El espacio reservado sirve para alamacenar temporalmente la página entrante mientras se selecciona la página que se va a intercambiar.

Número de marcos

- **Límite superior**: no se puede asignar más del total de marcos libres.
- **Límite inferior**: Número máximo de referencias necesarias para completar una instrucción: **Límite inferior** $= i_{max} + o_{max}(1 + n_{max})$
  donde:
  - i_{max}: Número máximo de palabras que pueden componer una instrucción.
  - o_{max}: Número máximo de operandos que puede necesitar una instrucción.
  - n_{max}: Numero máximo de referencias indirectas a memoria necesarias para extraer los datos empleados por la instrucción que más referencias utilice.
- Si hay indirección múltiple, se debe limitar el número de referencias indirectas, de modo que si se supera se produzca una trampa al SO.
- **Ejemplo**: Ej. PDP-11 tiene instrucciones de dos palabras con 2 operandos que pueden estar direccionados indirectamente (1 nivel):
  Límite inferior $= 2 + 2 \times (1 + 1) = 6$marcos

Número de marcos

Asignación de marcos en sistemas multiprogramados

- **Asignación equitativa**: Cada proceso de los n existentes reciben el mismo número de marcos (m/n) , de los m marcos del sistema.
- **Asignación proporcional**: El número de marcos asignados a un proceso es proporcional a su tamaño. Si s_i es el tamaño de memoria virtual solicitado por le proceso p_i , entonces el número de marcos asignados al proceso p_i , a_i , de entre los m marcos del sistema será:

$$max(a_i = m\frac{s_i}{\sum s_i}, \textbf{número mínimo de marcos})$$

  - Si varía el nivel de multiprogramación se recalcula (a la alta o a la baja) el número de marcos asignados a cada proceso.
- **Asignación con prioridades**: El número de marcos asignados a un proceso no depende del tamaño del mismo sino de su prioridad.

Tamaño de página

- Razones para escoger un tamaño pequeño:
  - Reduce la fragmentación interna.
  - Favorece la localidad (lo que se carga en memoria se ajusta a lo que se necesita).
- Razones para escoger un tamaño grande:
  - Reduce el tamaño de la tabla de páginas (sólo el 1 % del tiempo de Lectura/Escrituras de/a disco, se debe a la transferencia, el 99 % son los tiempos de latencia y posicionamiento).
  - Reduce el número de fallos de página.
- No hay acuerdo:
  - Intel 80386: 4Kb
  - Motorola 68030: variable entre 256 bytes y 32Kb
  - IBM/370: 2 ó 4Kb

Área de almacenamiento en disco

Lo más simple es asignar un área dedicada (inicialmente vacía), separada del sistema de directorios. Opciones:

1. Cada vez que se crea un proceso se reserva una zona del área de intercambio igual al tamaño de imagen del proceso. A cada proceso se le asigna la dirección en disco de su área de intercambio, que se almacena en la Tabla de Proceso.La lectura se realiza sumando el número de página virtual a la dirección de comienzo del área asignada al proceso.
2. Si los datos y/o la pila pueden crecer, es mejor reservar zonas zonas independientes. A cada zona se le asignan varios bloques.
3. No se asigna nada inicialmente. A cada página se le asigna su espacio en disco cuando se va a intercambiar, y el espacio se libera cuando la página vuelve a memoria. Problema: se debe llevar contabilidad en memoria (página a página) de la localización de las páginas en disco.

Administración de fallos de página

1. El HW hace un señalamiento al núcleo, que guarda el contador de programa (y a veces el estado de instrucción interrumpida)
2. Ejecuta una rutina en ensamblador que resguarda los datos volátiles (registros) y llama al SO como un procedimiento
3. El SO detecta el fallo de página e intenta determinar la página virtual requerida (normalmente contenida en un registro, o a partir del contador de programa almacenado)
4. El SO verifica que la página es válida y los permisos. En caso de que no sea válida o no se disponga de los permisos adecuados el SP envía una señal al proceso o lo elimina. En caso contrario el SO intenta encontrar un marco libre y si no lo hay ejecuta el algoritmo de reemplazo de páginas.
5. Si la página seleccionada ha sido modificada se escribe en disco. Durante ese tiempo el planificador da paso a otro proceso

Administración de fallos de página

6. El SO busca en disco la página solicitada y planifica una operación de disco para recuperarla. Durante el tiempo de carga, el proceso solicitante sigue suspendido, y el planificador puede dar paso a otro proceso de usuario. El marco se pone a "ocupado".
7. Al terminarse la carga de la página el SO actualiza las tablas de páginas del proceso(s) involucrados.
8. La instrucción que cometió el fallo se reinicia (o vuelve al estado en que quedó al ser interrumpida). El Contador de Programa se modifica para que apunte a esa instrucción.
9. El proceso que provocó el fallo se planifica de nuevo y el SO regresa a la rutina que lo llamó.
10. La rutina restaura los registros y demás información volátil y regresa al espacio de usuario para continuar la ejecución.

Administración de fallos de página
Hiperpaginación (Trashing)

- Definimos **conjunto de trabajo** como el número de páginas activas que un proceso tiene en un momento dado. Es el número **suficiente**, mayor al mínimo. Si el número de marcos disponibles es inferior al tamaño del conjunto de trabajo, se producirán frecuentes fallos de página (**hiperpaginación**).
- Un proceso **hiperpaginado** pasa más tiempo intercambiando páginas que ejecutándose, y puedo 'robar' páginas de otros procesos, provocando su hiperpaginación.
- **Consecuencia**: reducción drástica del uso de CPU. El planificador de procesos responde **incrementando el nivel de multiprogramación**. Este procesos se realimenta positivamente hasta que el sistema se desploma.

Administración de fallos de página
Hiperpaginación (Trashing)

Asignación de marcos en sistemas monoprogramados

Número de marcos

Tamaño de página

Área de almacenamiento en disco

Administración de fallos de página

Hiperpaginación (Trashing)

- La hiperpaginación se limita si se limita el número de marcos que el proceso puede utilizar (Asignación local), y si se asigna a cada proceso un número de marcos suficiente.
- **Problema**: Cómo calcular el número suficiente de marcos que un proceso necesita.
- Para calcular el tamaño del conjunto de trabajo, se utiliza el **modelo de conjunto de trabajo**, basado en el **supuesto de localidad**, que dice que en la ejecución de un proceso, las páginas que se utilizan en cada momento se pueden agrupar en grupos que varían a lo largo de la ejecución del proceso. Ej: una llamada a una función provoca un cambio de localidad.

Parte VI

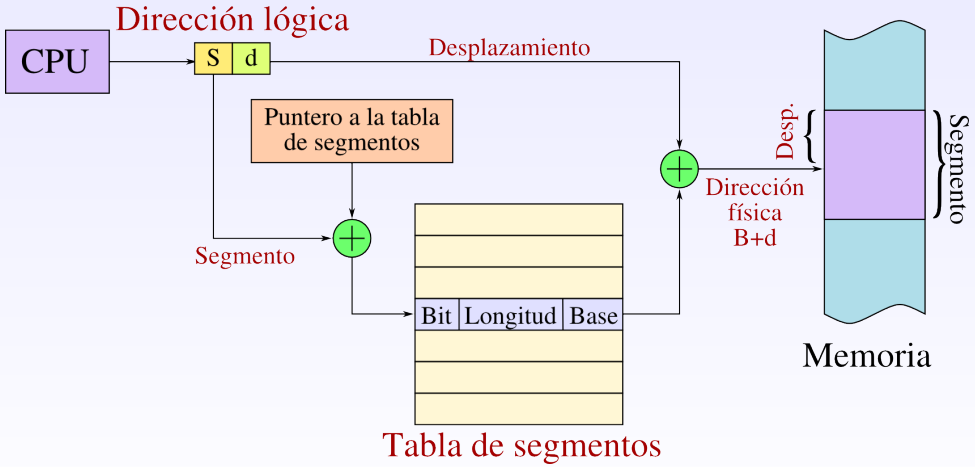
Segmentación

Segmentación

- Técnica para mantener espacios independientes de direcciones virtuales, llamados segmentos. Supone un nuevo nivel de abstracción, en el que distintas partes del proceso tienen espacios virtuales independientes entre sí. El espacio de memoria virtual es multidimensional
- La segmentación permite:
  1. Desentenderse de la administración 'interna' de la memoria virtual del proceso, es decir, del control del crecimiento, decrecimiento y ajuste entre si de las distintas partes del proceso.
  2. Asignar permisos distintos a las partes del proceso.
  3. Compartir datos o procedimientos entre procesos.
- Los segmentos se direccionan desde 0 hasta una dirección máxima, que puede variar.
- La segmentación pura puede producir problemas de fragmentación externa.

Segmentación

Proceso de segmentación

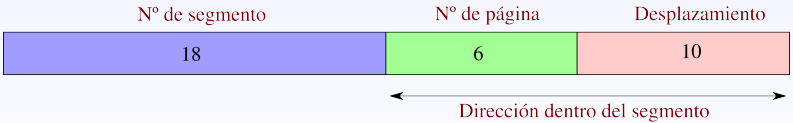


Paginación + Segmentación

- Segmentación paginada
  - Espacio de direcciones dividido en segmentos
  - Cada segmento en varias páginas de tamaño fijo de igual tamaño que el marco de memoria
- Si el segmento es menor que una página entonces ocupa una página completa.
- Direcciones
  - Desde el punto de vista del programador: número de segmento + desplazamiento
  - Desde el punto de vista del sistema: número de segmento + número de página + desplazamiento

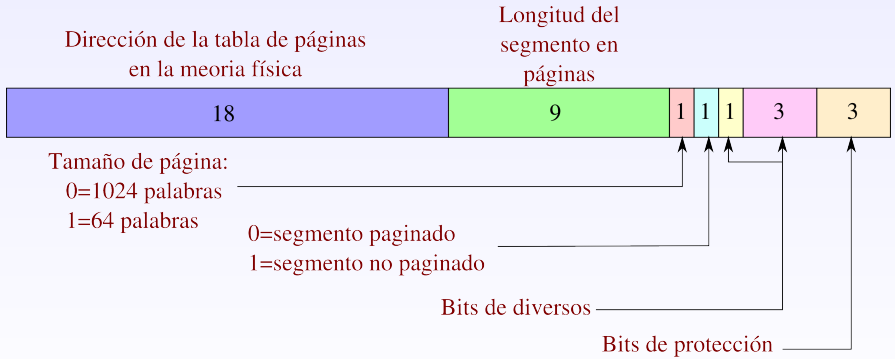
MULTICS (segmentación + paginación)

- Cada programa dispone de una memoria virtual de hasta 2^{18} segmentos, cada uno de hasta 64K (2^{18}) palabras. Cada segmento es un espacio virtual independiente que puede paginarse, con 64 (2^6) páginas de 1K (2^{10})
- Las direcciones físicas son de 24 bits, con las páginas alineadas con fronteras de 64 bytes
- La dirección virtual en MULTICS es:



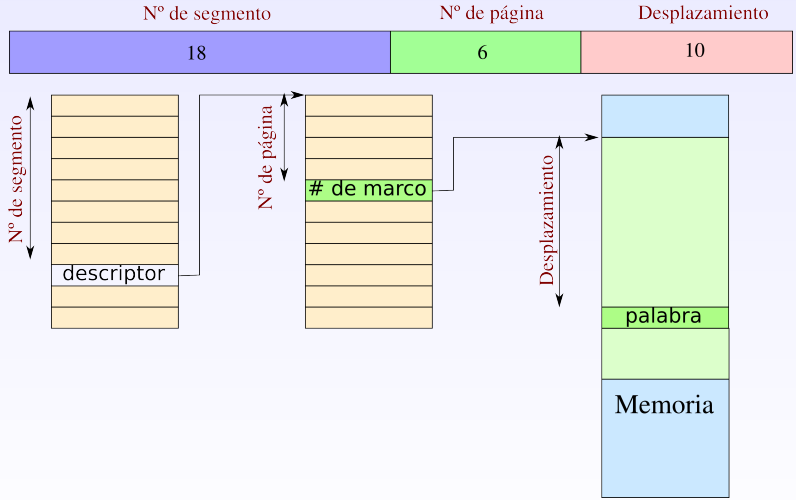
MULTICS (segmentación + paginación)

- Cada programa mantiene una tabla de segmentos, que a su vez reside en un segmento que puede estar paginado.
- Cada entrada de la tabla se segmentos tiene los siguientes campos:



MULTICS (segmentación + paginación)

Conversión de direcciones:



Parte VII

Diseño y políticas del gestor de memoria

Diseño del gestor de memoria

Dependencias

1. Uso o no de memoria virtual (`Si hay hardware disponible`)
2. Uso de paginación, segmentación o ambas (`Si hay hardware disponible`)
3. Algoritmos empleados para los problemas de gestión de memoria

Información

- Actualmente, casi todos los SS.OO. Ofrecen Memoria Virtual (DOS y primeros UNIX no)
- Al usar paginación+segmentación, la mayoría de los problemas surgen en la paginación