

Tema VII

Memoria Virtual

UNED

Manuel Fernández Barcell

<http://www.mfbarcell.es>

Blog: <http://prof.mfbarcell.e>

Memoria Virtual

- Sólo una parte del espacio de direcciones virtuales de un proceso residente se cargada en memoria física
 - La memoria virtual permite la ejecución de procesos parcialmente cargados
- Mantiene una imagen del espacio de direcciones virtuales completo de un proceso en memoria secundaria, y trae a memoria principal parte de esa imagen cuando es necesaria.
 - La elección de que sección traer, cuando y donde es efectuada por el S.O.
- La tarea adicional del hardware de traducción de direcciones en sistemas virtuales es detectar si el elemento referenciado esta en memoria real o no.
 - Para ello se añade **un bit de presencia**, a cada entrada de la TP (en el caso de gestión paginada)
- En el caso de que en el momento de la traducción el bit este a cero, el hardware genera una excepción por elemento ausente para anunciar el hecho al S.O.
 - A esta excepción se denomina **fallo de página**
- El fallo de página se produce cuando en la tabla de páginas del proceso no hay asociado un marco de página a la página que se quiere acceder

Memoria virtual

- La memoria virtual permite aumentar el grado de multiprogramación del sistema.
 - En memoria principal caben más procesos si solo hay algunas partes de cada proceso
- Cuando se implementa la memoria virtual también se suele utilizar un componente hardware denominado unidad de gestión de memoria (*Memory Managemer Unit*, MMU) que se encarga de realizar la traducción de una dirección virtual en una dirección física.
- La memoria virtual se puede implementar usando
 - *Paginación por demanda* (UNIX, Linux y Wins),
 - *Segmentación por demanda* (OS/2) o
 - *Segmentación con paginación por demanda* (Multics).

Paginación por demanda

- Únicamente cargan las páginas que se van referenciando durante la ejecución del proceso.
- Cuando se referencia una página que no está cargada en la memoria principal el hardware produce una excepción denominada *fallo de página*
- Tareas
 - *Reemplazamiento de páginas*
 - Debe seleccionar mediante la utilización de algún algoritmo de reemplazamiento el marco j de página donde se va cargar la página i que traiga desde memoria secundaria.
 - *Asignación de marcos de memoria principal*
 - El sistema operativo tiene que decidir cuánto marcos asigna inicialmente para cada uno.
 - *Control de carga*
 - El sistema operativo debe controlar el grado de multiprogramación del sistema
 - *Copia en la memoria secundaria de páginas modificadas*
 - El sistema operativo debe decidir en qué momento y de qué forma copiará las páginas que han sido modificadas de la memoria principal a la memoria secundaria

Estructuras

- La *Tabla de Marcos de Página*, la *Lista de Marcos Libres* y las *Tablas de Páginas*.
 - Las dos primeras estructuras son similares a las utilizadas en la paginación simple
- Tabla de página
 - *Número de marco de página*
 - *Validez o presencia*
 - *Protección*
 - Usualmente este campo suele constar de tres bits ($P2P1P0$) como máximo que permiten establecer los permisos de
 - Acceso lectura ($p2$) . Escritura ($P1$) y Ejecución (Po) de una página.
 - *Referenciada*
 - *Modificada*

Ejemplo

Memoria principal

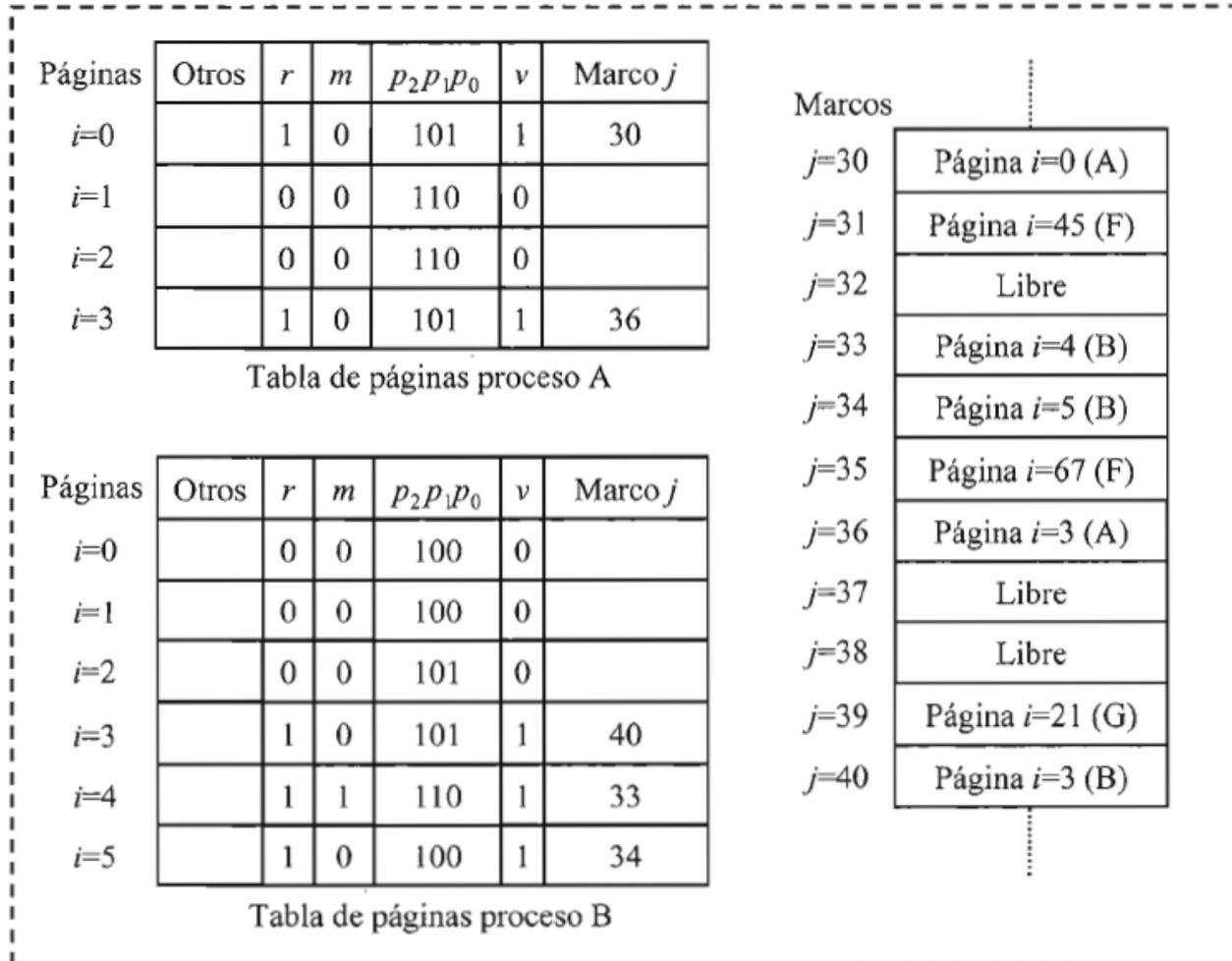


Figura 7.1 – Ejemplo de tablas de páginas utilizadas en paginación por demanda

Memoria Virtual

- Reinicio de instrucciones
 - Es necesario que la arquitectura del computador permita reiniciar cualquier instrucción después de un fallo de página
- **Localización de las páginas en memoria secundaria**
- El sistema operativo puede buscar una página en dos posibles localizaciones de memoria secundaria
 - *Bloque de disco de un archivo ejecutable*
 - *Bloque de disco del área de intercambio*

Ubicación de todo el espacio virtual en el área de intercambio

- Copia todas las páginas al área de intercambio

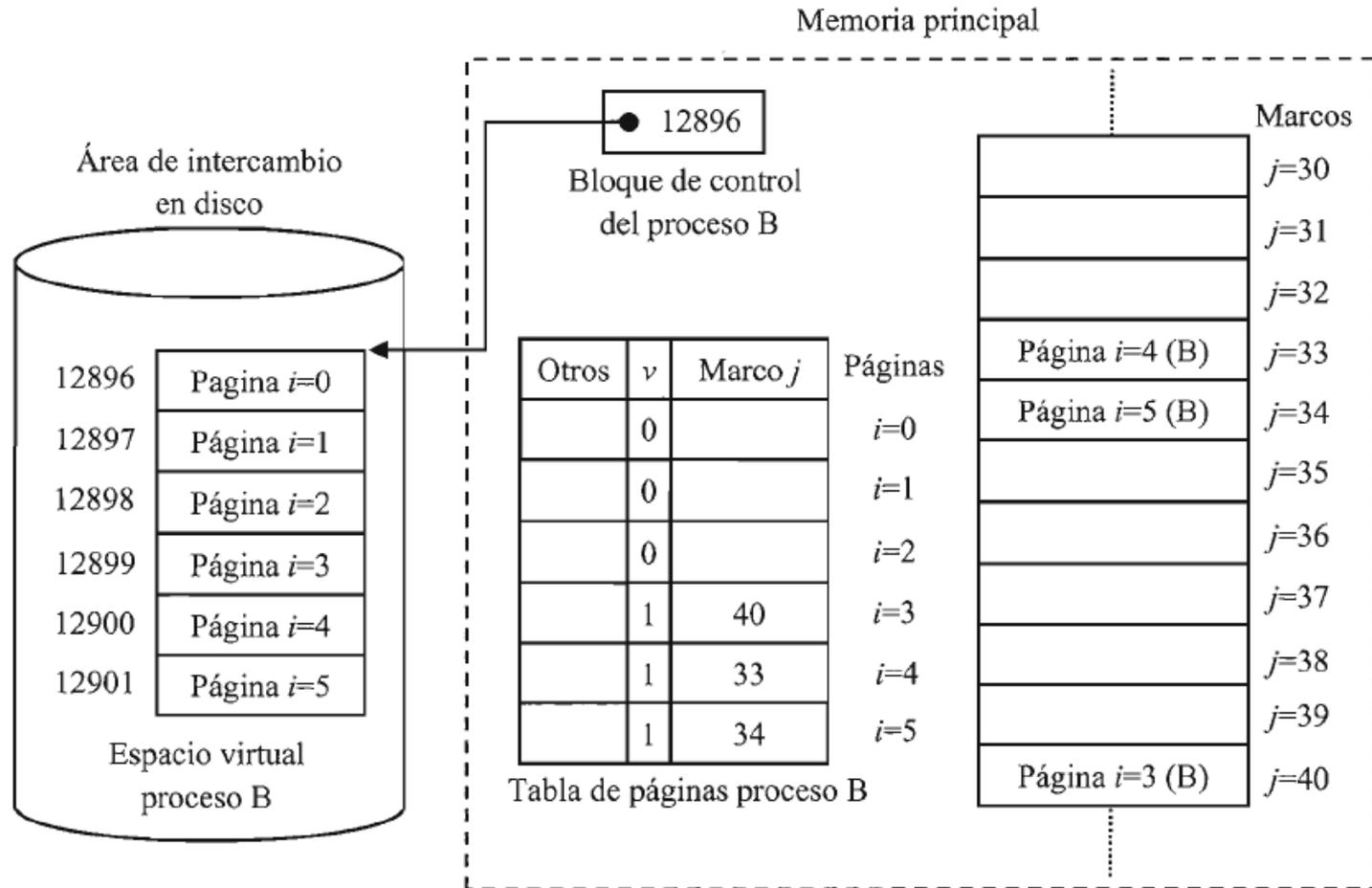


Figura 7.2 – Localización de las páginas de un proceso en el área de intercambio mediante una dirección de disco base ubicada en el bloque de control del proceso

Uso de un mapa de disco

- No copia todas las páginas en el área de intercambio, solo las reemplazadas

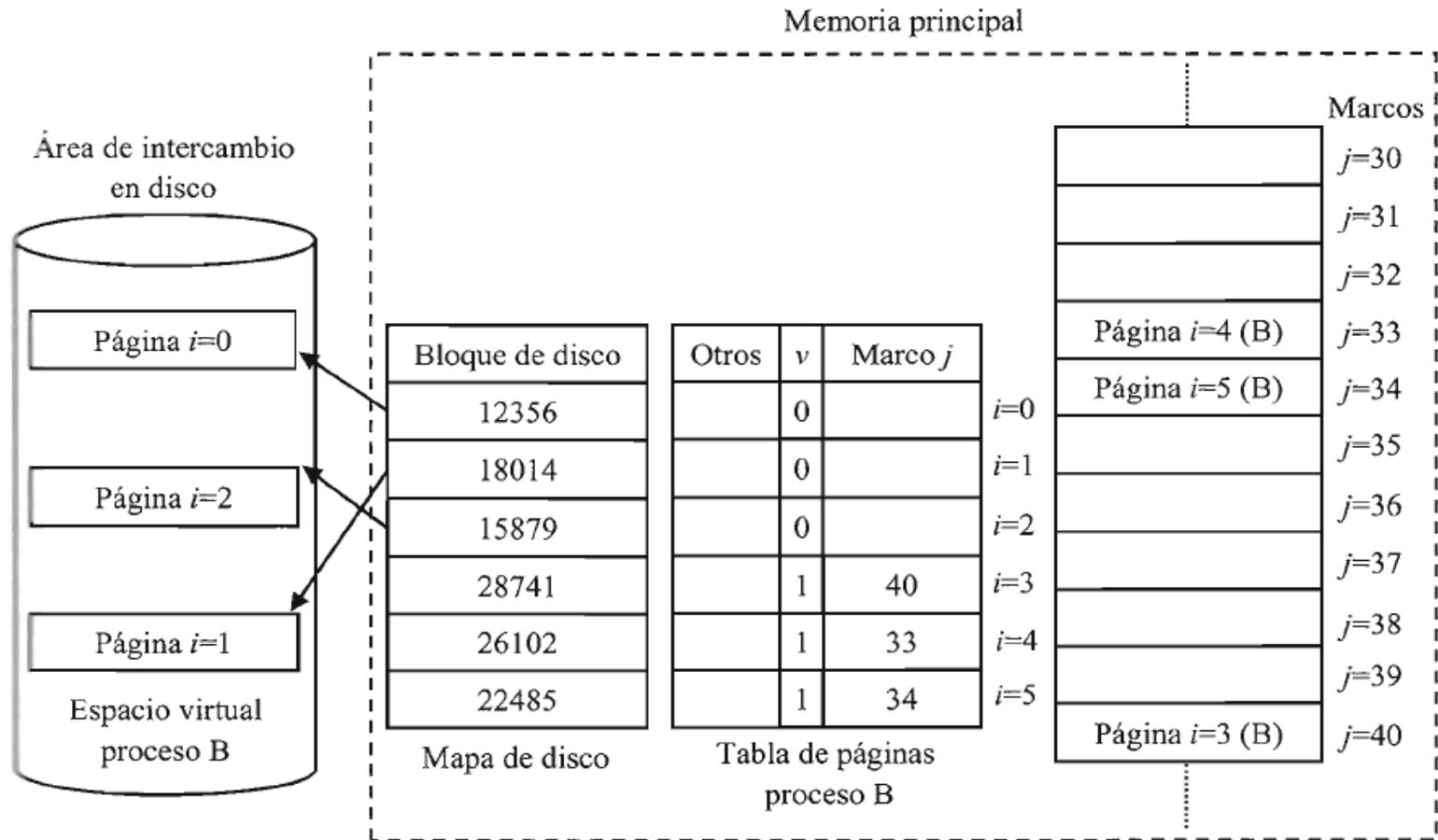


Figura 7.3 – Localización de las páginas de un proceso en el área de intercambio mediante el uso de un mapa de disco

- Bloqueo de marcos de página
 - Se dice que un marco de memoria está *bloqueado* o *pinchado (pinning)*, si su contenido no puede ser reemplazado

Tratamiento de un fallo de página

- En el caso de que en el momento de la traducción el bit este a cero, el hardware genera un fallo de página
 - El fallo de página se produce cuando en la tabla de páginas del proceso no hay asociado un marco de página a la página que se quiere acceder
- Cuando el proceso en ejecución experimenta un fallo de página queda suspendido hasta que la página que falta sea incorporada en memoria principal,
- El procedimiento a seguir es el siguiente:
 - Se verifica si la dirección es válida.
 - Si la dirección fuera inválida se enviará una señal al proceso o se abortaría
 - Si es una referencia válida, pero aún no se ha traído esta página, el S.O. detecta un fallo de página y determina la página virtual requerida para traerla.
 - En este caso la instrucción queda interrumpida y se guarda el estado del proceso, para poder continuarlo en el mismo lugar y estado
 - Se selecciona un marco libre.
 - Si no existe un marco libre, se tendría que ejecutar un algoritmo
- Cuando el marco queda limpio, el S.O. examina la dirección en el disco donde se encuentra la página necesaria y planifica una operación de lectura de la misma. Mientras se carga la página, el proceso sigue suspendido y se permite ejecutar otro proceso
 - Cuando se completa la lectura del disco, la tabla de páginas se actualiza para indicar que ya se dispone de la página en memoria
 - La instrucción que produjo el fallo regresa al estado de comienzo y se planifica su ejecución, pudiéndose acceder a la página como si siempre hubiese estado en memoria

Conjunto de trabajo de un proceso

- Conjunto corresponde al conjunto de páginas referenciadas por el programa durante un intervalo reciente de tiempo, dicho intervalo es una ventana cuyo tamaño depende del diseño.
- Esta basado en **el principio de localidad**:
- Del estudio del comportamiento de los programas, se observa que existe una fuerte tendencia de los programas a favorecer ciertos subconjuntos de sus espacios de direcciones durante la ejecución.
 - A este fenómeno se le conoce como **localidad de referencia**, y puede ser:
 - **Localidad de referencia espacial**:
 - Tendencia a referenciar posiciones agrupadas (*arrays*)
 - **Localidad de referencia temporal**:
 - Tendencia a referenciar la misma posición o grupo de posiciones varias veces durante breves intervalos de tiempo (bucles)
 - Una **localidad** es un pequeño grupo de páginas no necesariamente adyacentes a las cuales aluden la mayoría de las referencias a memoria durante un periodo de tiempo

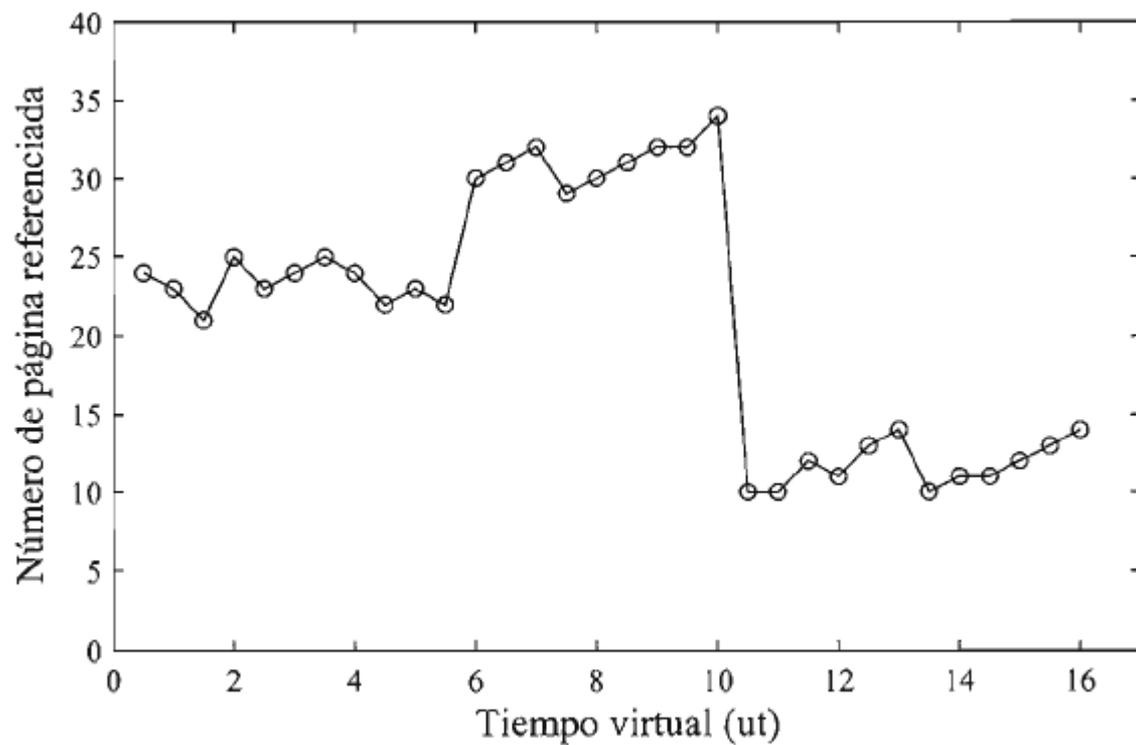


Figura 7.4 – Números de páginas referenciados durante la ejecución del proceso del Ejemplo 7.5

Conjunto de trabajo

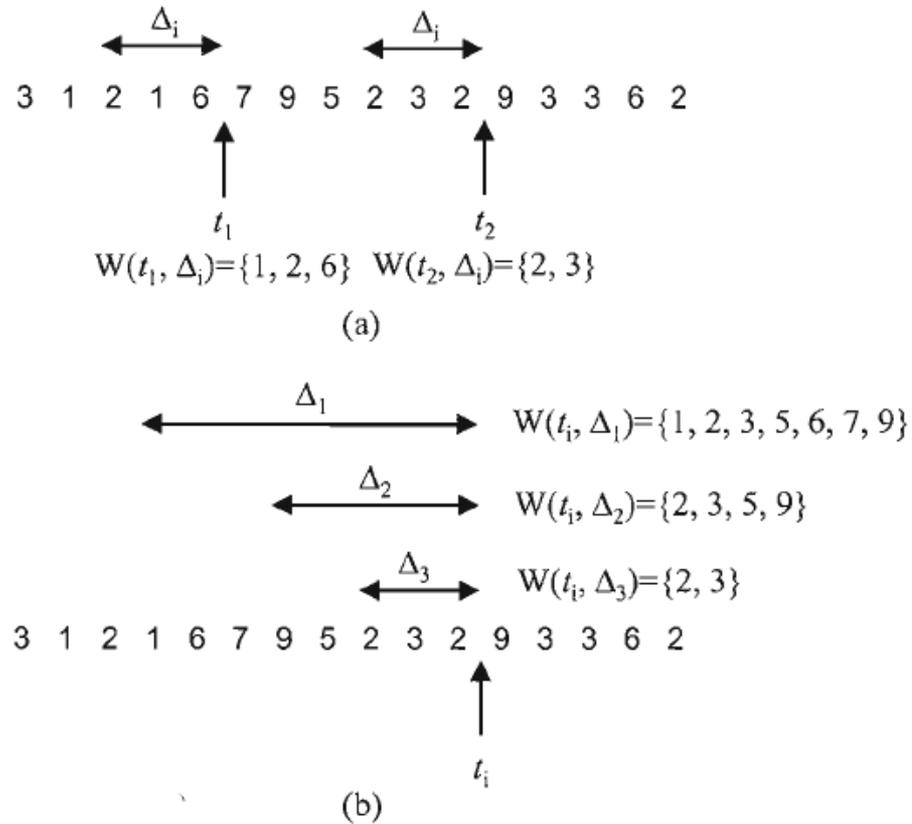


Figura 7.5 – Conjunto de trabajo de un proceso con distintos instantes de tiempo virtual (a) y distintas ventanas de tiempo virtual (b)

Reemplazamiento de páginas

- Consiste en seleccionar una página k cargada en un marco j de memoria principal para ser reemplazada por la página i a la que hacía referencia la dirección virtual que produjo el fallo de página.
- El término *conjunto de marcos candidatos*
 - Hace referencia a los marcos de memoria principal donde se encuentran almacenadas las páginas candidatas.
- *Estrategia de reemplazamiento de páginas local*
- *Estrategia de reemplazamiento de páginas global*

Anomalía de Belady

- Algoritmos de pila no la tienen

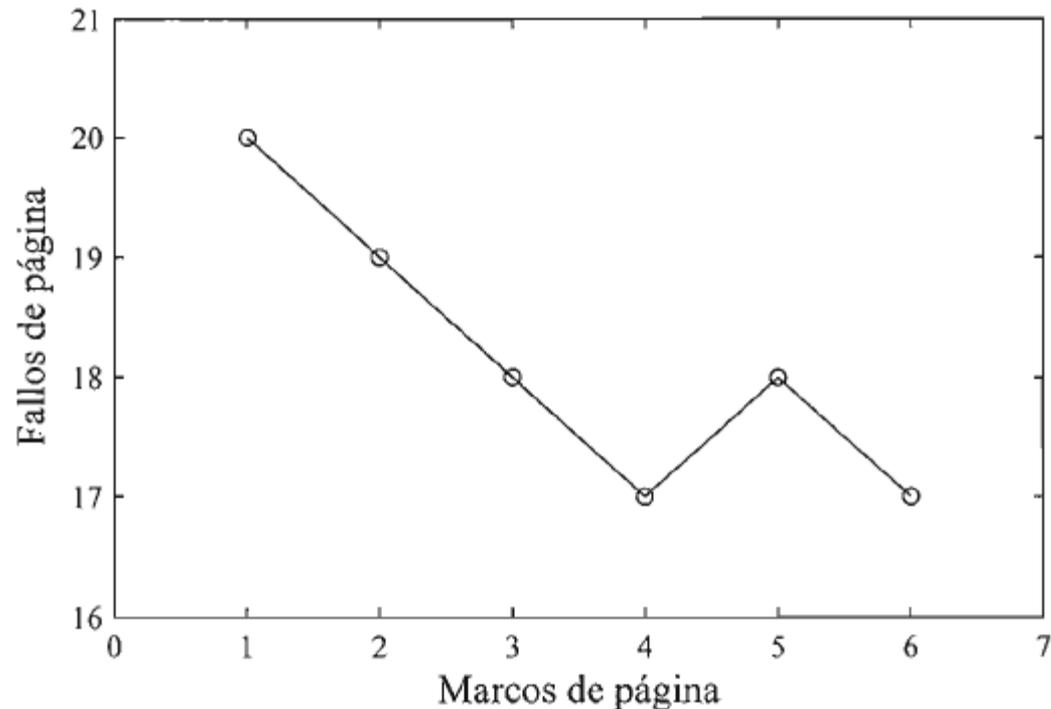


Figura 7.6 – Ilustración de la anomalía de Belady

Algoritmo de reemplazamiento óptimo

- El algoritmo de reemplazamiento óptimo selecciona para ser reemplazada aquella página del conjunto de páginas candidatas a ser reemplazadas que tardará más en volver a ser referenciada por una dirección virtual.

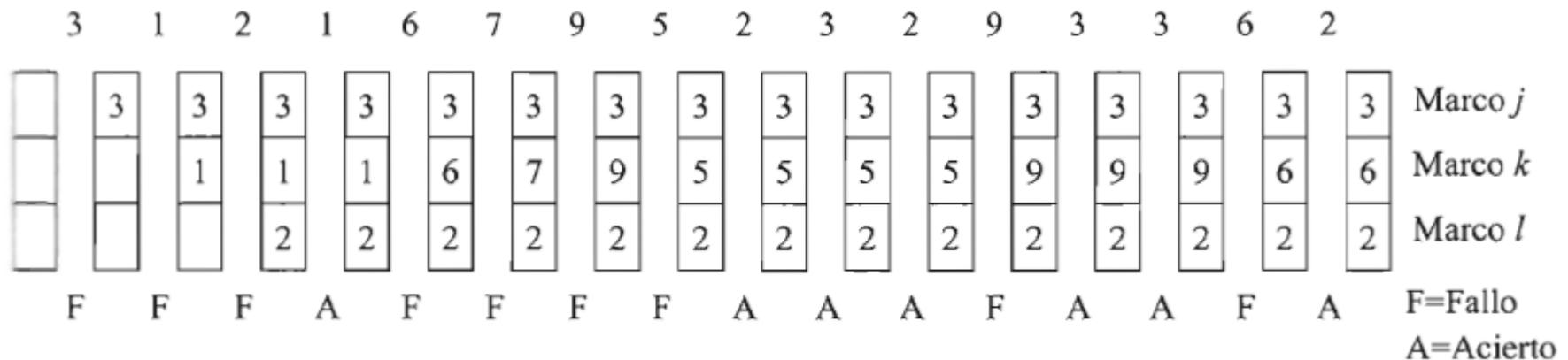


Figura 7.7 – Ejemplo de aplicación del algoritmo de reemplazamiento de página óptimo

Algoritmo de reemplazamiento mediante envejecimiento

- Se asigna un registro de desplazamiento software de n bits a cada página cargada en memoria principal.
- El registro se inicializa a 0 cuando la página es cargada en memoria.
 - Cada cierto tiempo T preestablecido, el sistema operativo desplaza un bit a la derecha el contenido del registro asignado a cada página cargando el bit referenciada en el bit más significativo de cada registro.
 - Además pone a 0 el bit referenciada de cada página.
- La página que se selecciona para ser reemplazada es aquella cuyo registro de desplazamiento contiene el número binario más pequeño, ya que será la página menos usada recientemente.

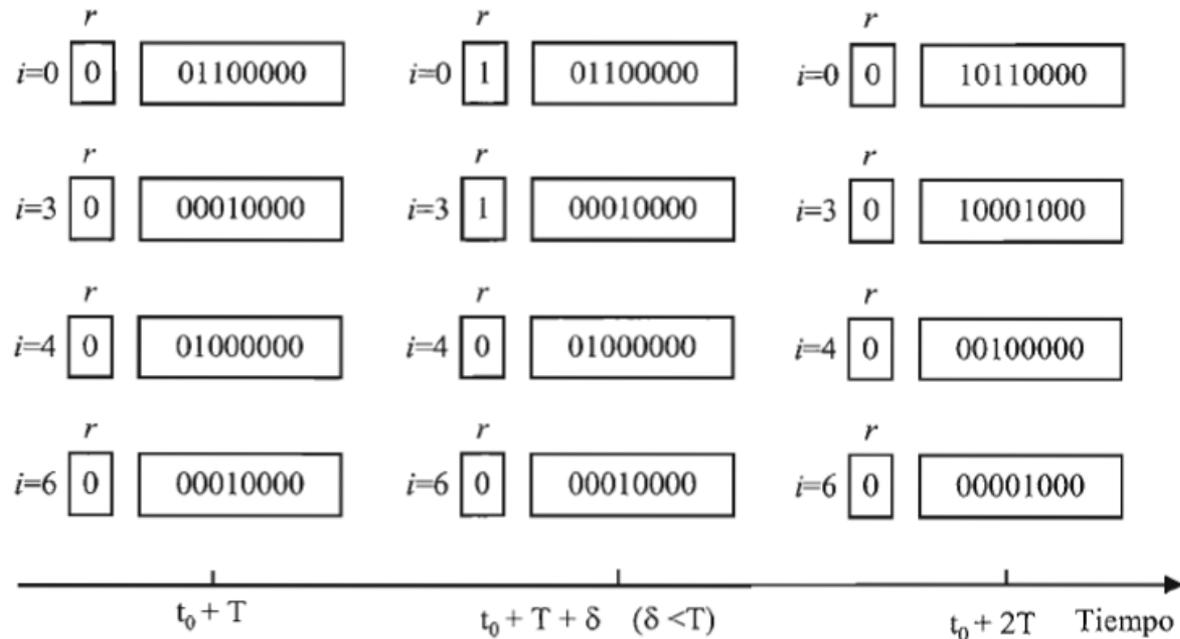


Figura 7.9 – Ejemplo de aplicación del algoritmo de envejecimiento

Algoritmo de reemplazamiento FIFO

- Se sustituye la página que lleva más tiempo en memoria.
 - Se puede crear una cola, según el orden de entrada, con todas las páginas de la memoria, cuando hay que sustituir una página, se elige la primera de la cola y la que se trae se inserta al final de la cola.
- Sufre la anomalía de Belady que consiste en que aumentan los fallos de página al aumentar el número de marcos de página para asignación

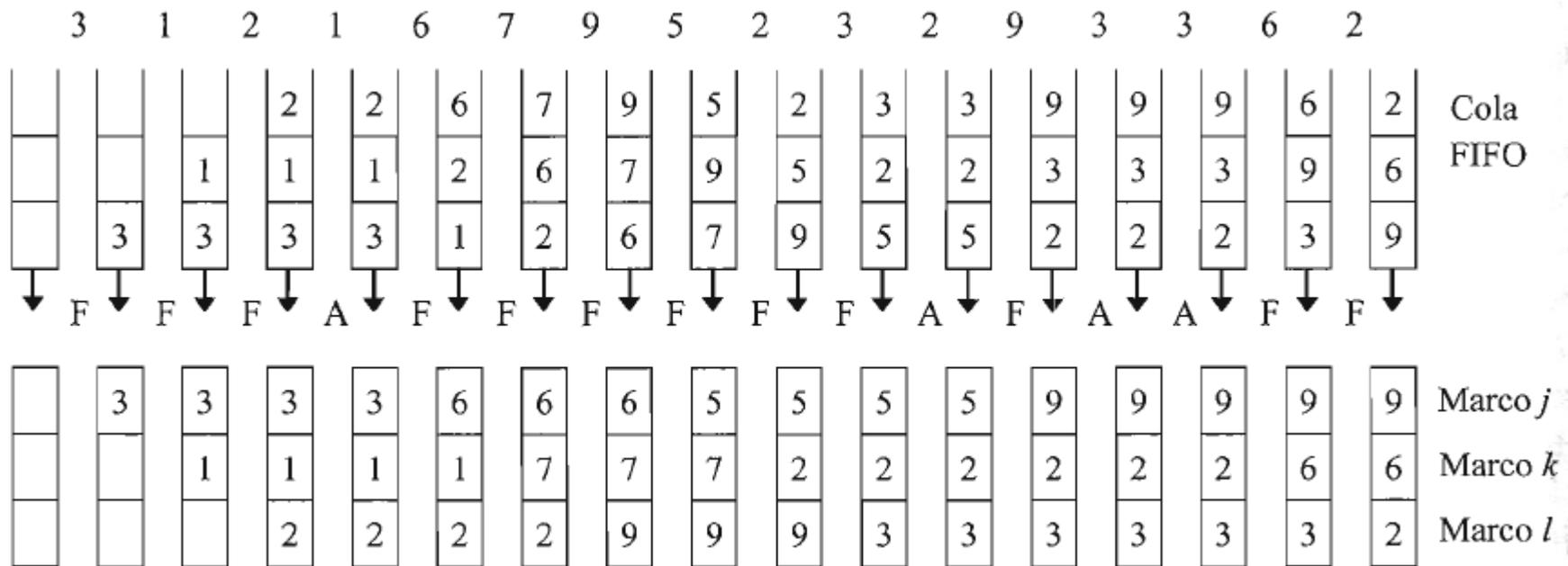


Figura 7.10 – Ejemplo de aplicación del algoritmo de reemplazamiento FIFO

Algoritmo de reemplazamiento de la segunda oportunidad (algoritmo del reloj)

- Es una variante del algoritmo FIFO que busca la página que lleva más tiempo cargada en memoria y no ha sido referenciada recientemente
- Cada vez que el procesador referencia a una dirección virtual contenida en una página i , el bit referenciada de la entrada i de la tabla de página del proceso en ejecución es activado ($r = 1$).
- Busca la página que lleva más tiempo cargada en memoria y no ha sido referenciada recientemente.
- Básicamente este algoritmo consulta el bit referenciada (r) de la página que se encuentra al principio de la cola FIFO.
 - Si $r = 0$; entonces la página es seleccionada para ser reemplazada y el algoritmo finaliza.
 - Por el contrario, si $r = 1$ entonces el algoritmo pone el bit a 0 y coloca el número de la página al de la cola, es decir, se le da una segunda oportunidad.

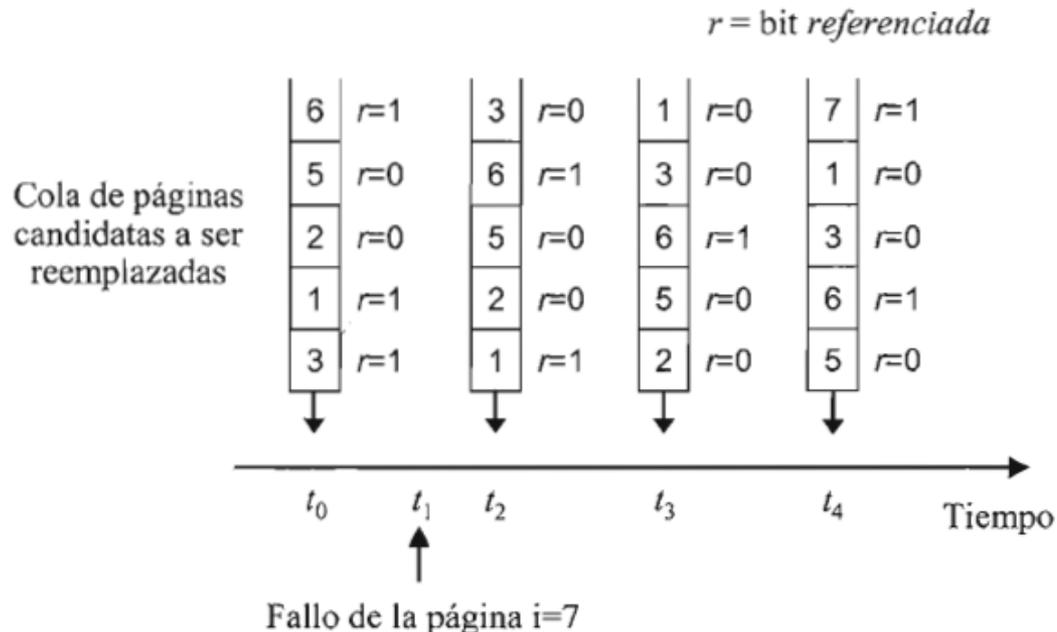


Figura 7.11 – Ejemplo de aplicación del algoritmo de la segunda oportunidad

Segunda oportunidad o Reloj como cola circular

- Es una cola circular, con un puntero que indica cual es la página que se sustituirá a continuación
- Cuando se necesita un marco de página el puntero avanza hasta que encuentra una página con un bit de referencia a cero,
- Según avanza el puntero se ponen a cero los bits de referencia

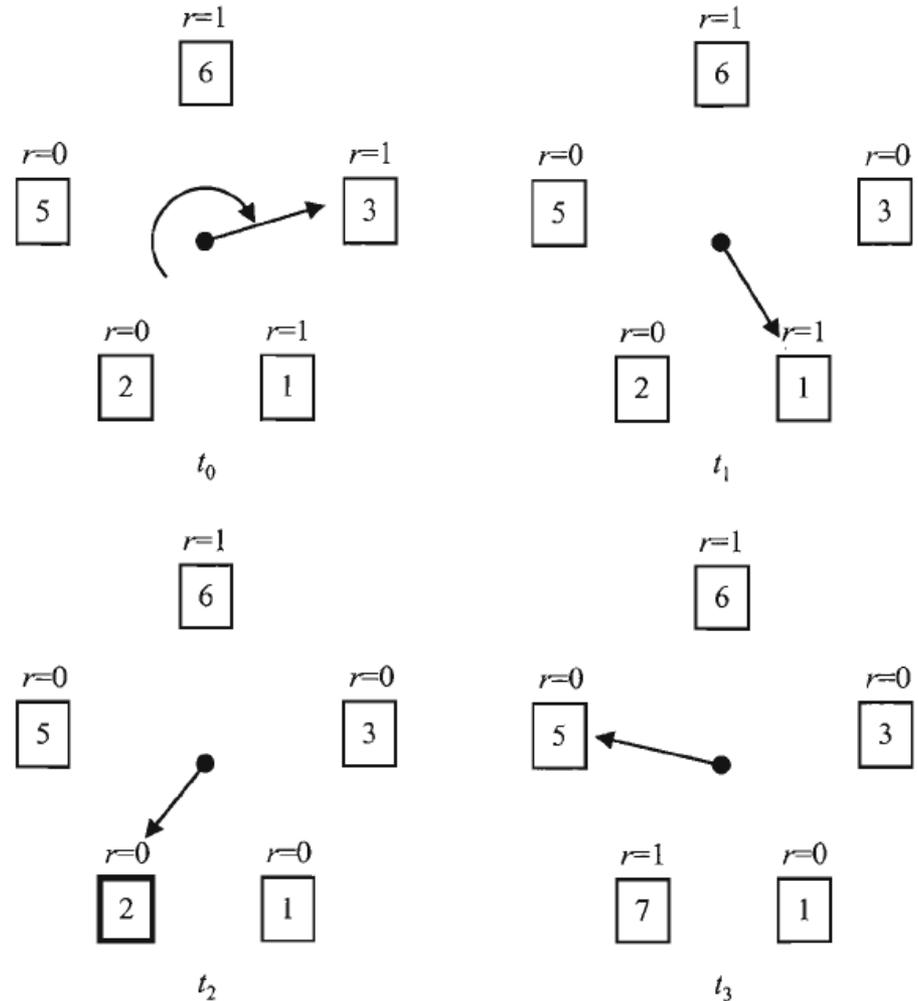


Figura 7.12 – Ejemplo de aplicación del algoritmo del reloj

Algoritmo del reloj mejorado

- Además del bit de referencia r considera el bit de modificación m

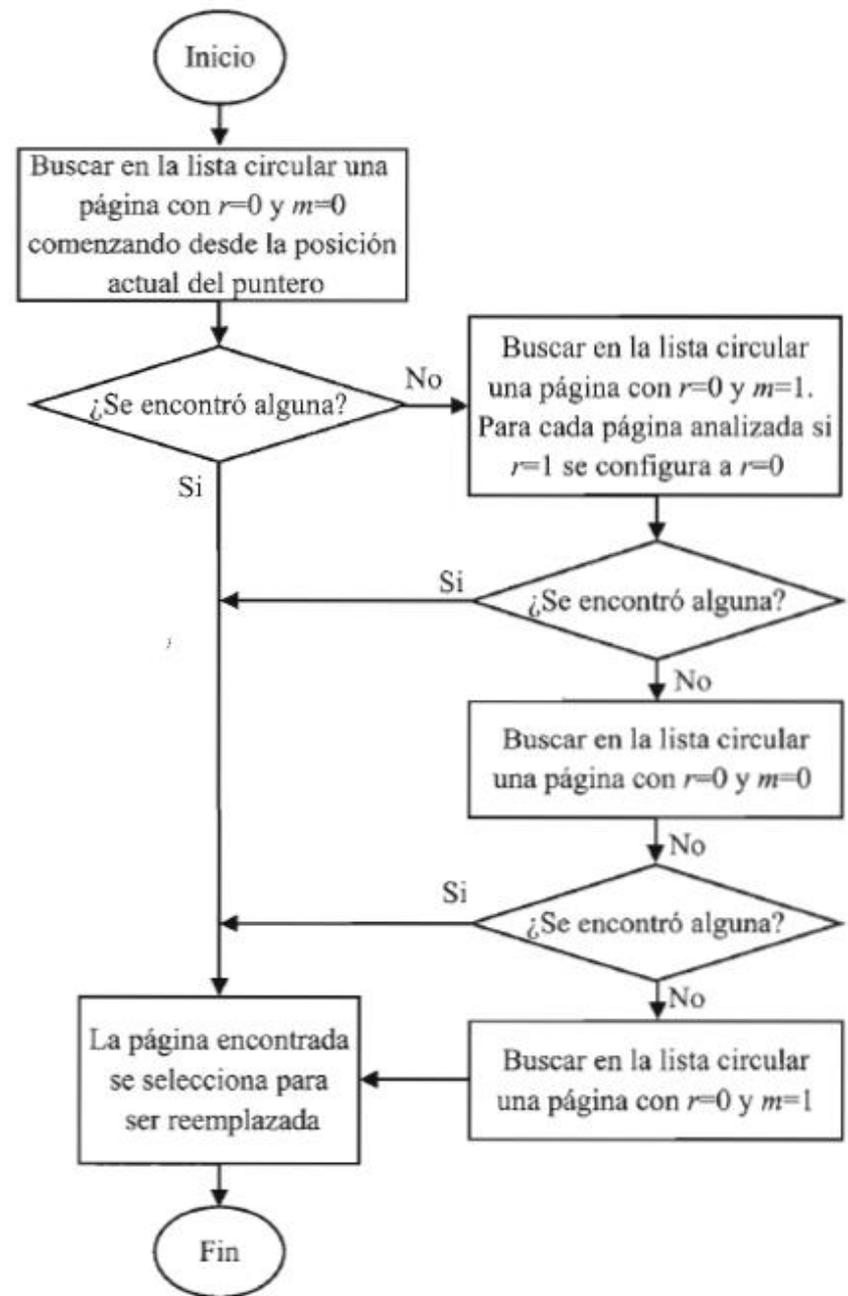


Figura 7.13 – Algoritmo del reloj mejorado

mejorado

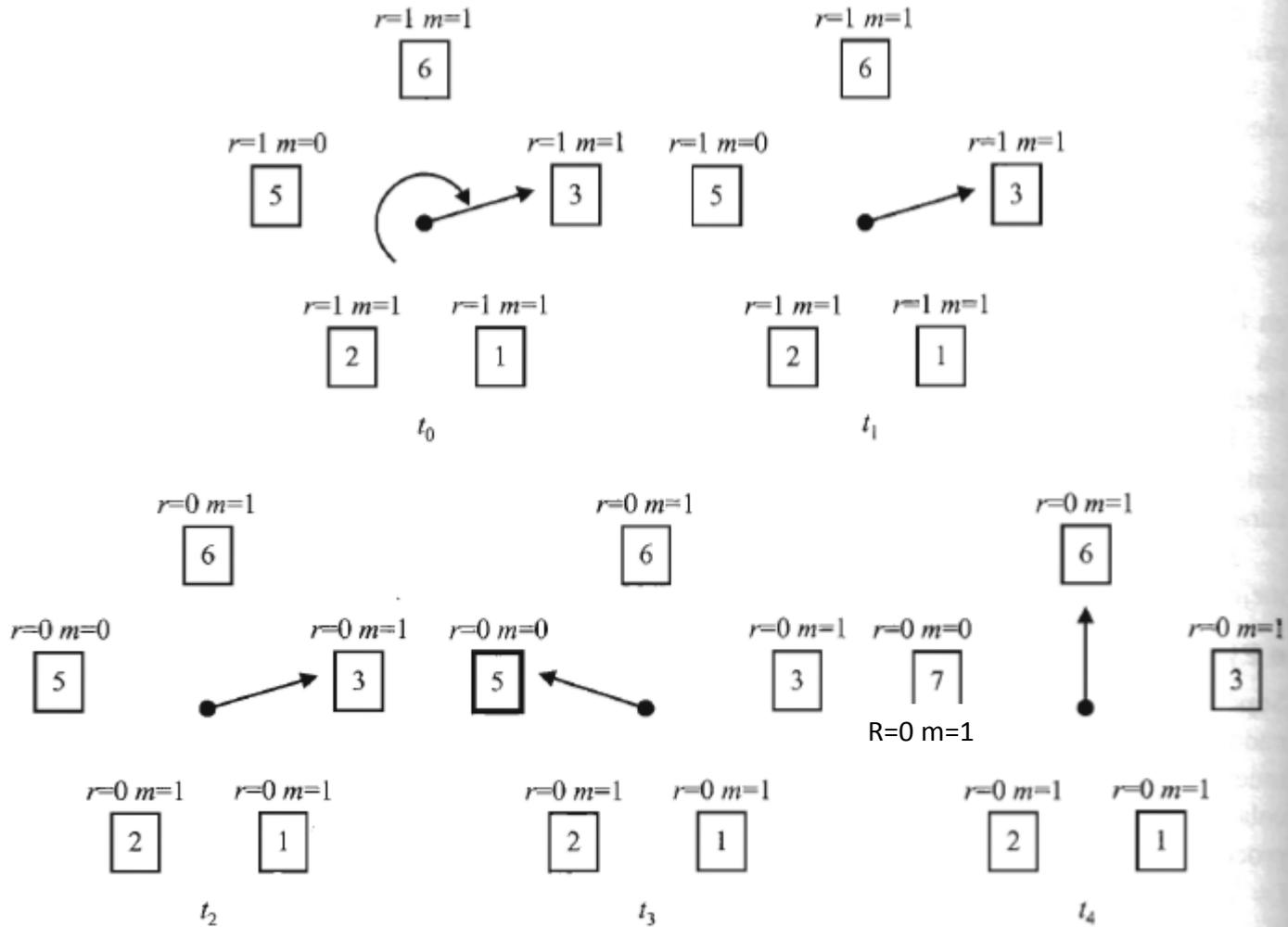
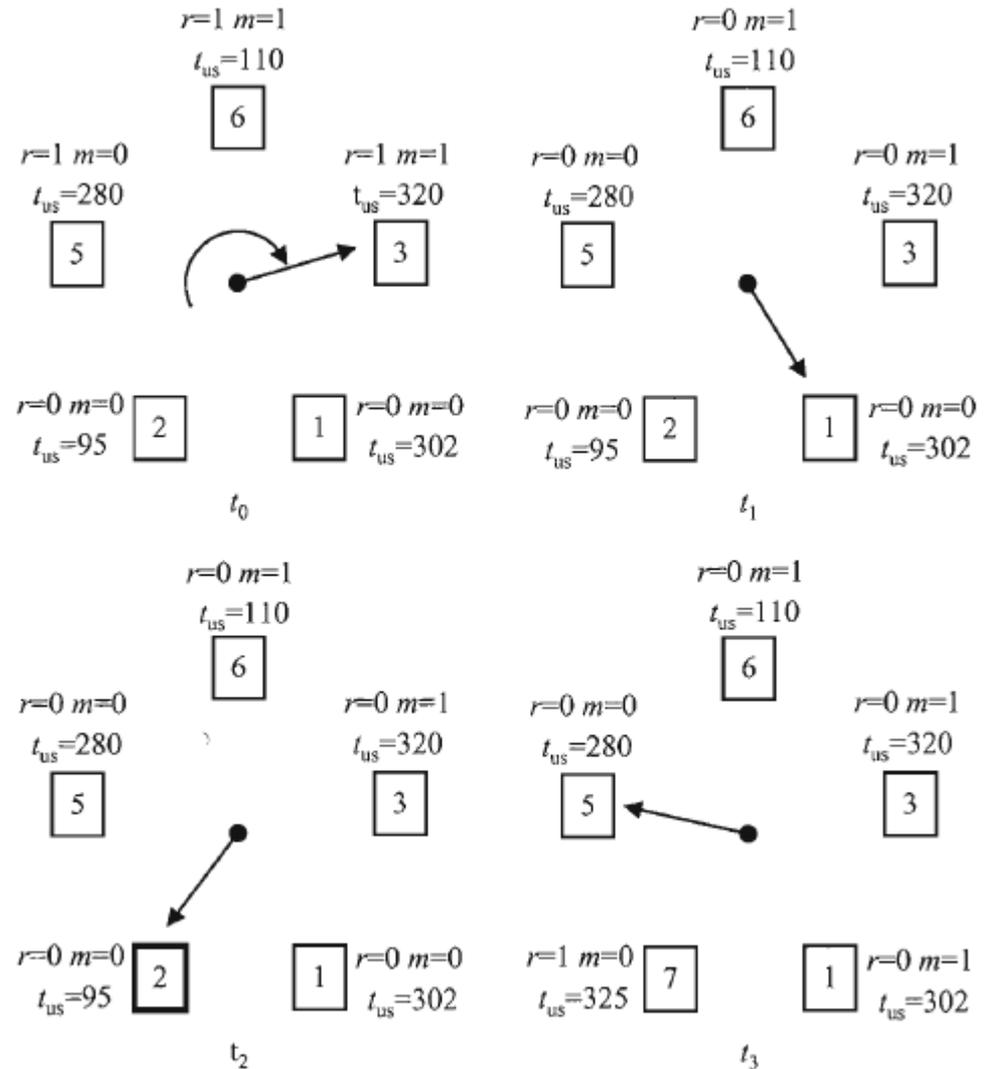


Figura 7.14 – Ejemplo de aplicación del algoritmo del reloj mejorado

Algoritmo de reemplazamiento del reloj considerando el conjunto de trabajo (WSClock)

- Tiempo de último uso: tiempo virtual del último uso de la página
- La diferencia entre el tiempo virtual actual y el tiempo de último uso, establece la edad de la página
- Si la edad de la página es menor o igual que Δ entonces la página pertenece al conjunto de trabajo del proceso.



Asignación de memoria principal

- Decidir cuántos marcos de página de la memoria principal reserva para un proceso que se tiene que ejecutar.
- Asignación fija
- *Asignación variable*
 - *Algoritmo de asignación equitativa*
 - *Algoritmo de asignación proporcional*
 - *Algoritmo de asignación por la frecuencia de fallos de página o algoritmo PFF (frecuencia de Fallo de Página)*

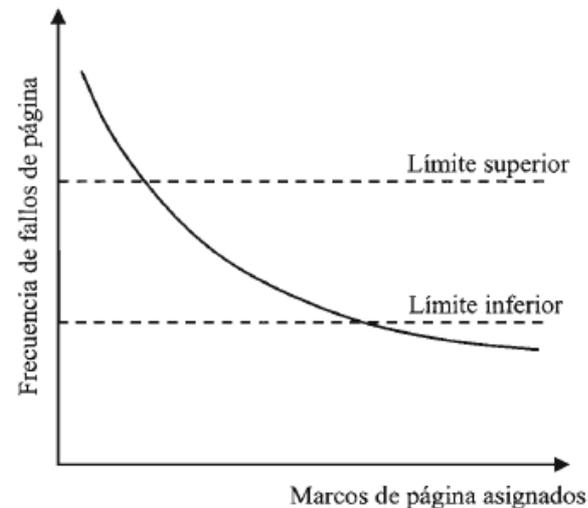


Figura 7.16 – Representación gráfica de la frecuencia de fallos de página en función del número de marcos asignados

Control de carga

- Debe controlar el *grado de multiprogramación del sistema* G_M .
- Por encima de G_{Mopt} puede haber sobrepaginación

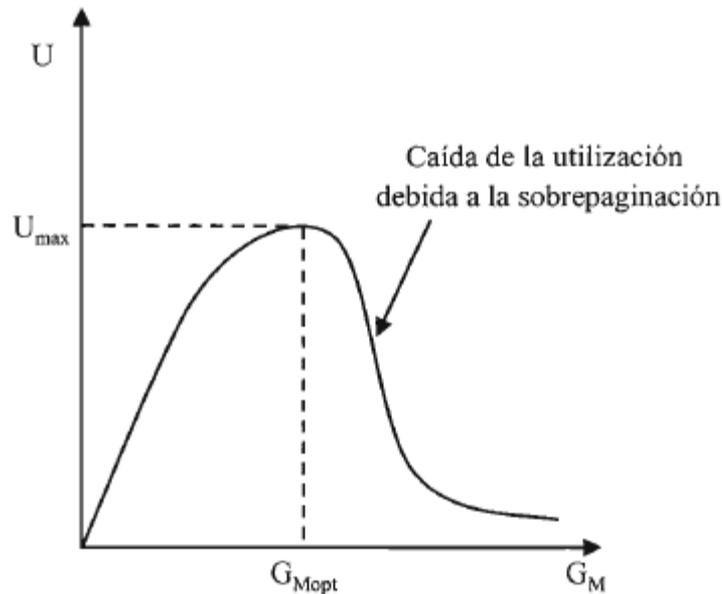


Figura 7.17 – Influencia de grado de multiprogramación sobre la utilización del procesador

Copia en la memoria secundaria de páginas modificadas

- El sistema operativo debe decidir en qué momento y de qué forma copiará las páginas que han sido modificadas de la memoria principal a la memoria secundaria
- Limpieza por demanda
 - La página modificada se escribe en memoria cuando es seleccionada como página víctima para ser reemplazada por otra página que ha producido un fallo de página
- *Limpieza por adelantado*
 - Las páginas que han sido modificadas se agrupan en lotes y se planifica cada cierto tiempo su escritura en memoria secundaria antes de que sean elegidas por el algoritmo de reemplazamiento.

Consideraciones adicionales sobre la paginación por demanda

- Tamaño de página
 - Fragmentación interna
 - *Tamaño de una tabla de páginas*
 - *Número de fallos de página*
 - *Tiempo de uso de E/S.*

Paginación por Adelantado

- Para evitar este conjunto de fallos iniciales se puede usar la estrategia conocida como paginación por adelantado (*prepaging*) que consiste en cargar un cierto número de páginas N_{PA} asociadas a un proceso antes de iniciar o continuar con su ejecución.
- Estrategias
 - Cargar el conjunto de trabajo del proceso
 - Cargar un conjunto de páginas ubicadas de forma contigua en el área de intercambio

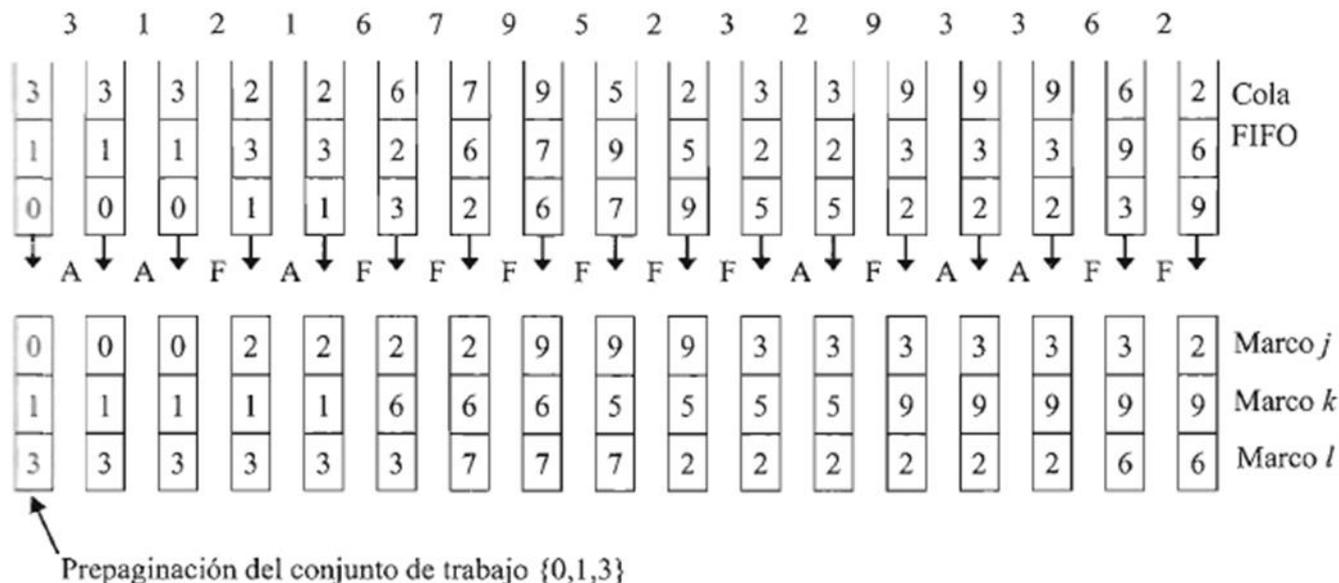


Figura 7.18 – Ejemplo de aplicación del algoritmo de reemplazamiento FIFO con prepaginación del conjunto de trabajo

Reserva de Marcos Libres

- El sistema operativo puede mantener una reserva de marcos libres implementada en la forma de una lista de marcos libres.
- Cuando se necesita un marco libre se coge el situado en la cabecera de la lista de marcos libre .
- Si la reserva de marcos libre cae por debajo de un cierto límite se utiliza el algoritmo de reemplazamiento para seleccionar páginas reemplazables, cuyos marcos se agregan a la lista hasta alcanzar un cierto límite máximo.

2. (2 p) Enumerar y describir **brevemente** los cuatro principales factores que hay que tener en cuenta a la hora de seleccionar el tamaño de página en la técnica de gestión de la memoria principal mediante paginación por demanda.

Los factores que hay que tener en cuenta a la hora de seleccionar el tamaño de página son:

- *Fragmentación interna.* En promedio la fragmentación interna es de media página por proceso. En consecuencia, cuanto mayor sea el tamaño de la página mayor será la fragmentación interna y mayor será el desperdicio de espacio de la memoria principal.
- *Tamaño de una tabla de páginas.* Cada proceso tiene una tabla de páginas cargada en memoria principal que posee un número de entradas igual al número de páginas en que se descompone el espacio de direcciones virtuales del proceso. Cuanto menor sea el tamaño de una página, en más páginas se descompondrá el espacio virtual del proceso, y las tablas de páginas serán de mayor tamaño. Por lo tanto, consumirán más espacio de memoria. Además, si el tamaño de la tabla de páginas aumenta, el tiempo de cambio de proceso será mayor ya que el sistema operativo tendrá que cargar o vaciar más registros si se tiene una MMU con banco de registros o con TLB.
- *Número de fallos de página.* Si el tamaño de página es pequeño el espacio de direcciones virtual constará de más páginas por lo que la probabilidad de que se produzca un fallo de página será mayor.
- *Tiempo de uso de E/S.* El tiempo de una lectura o escritura en un disco se descompone en tiempo de búsqueda, tiempo de latencia rotacional y tiempo de transferencia. De estos tres tiempos el tiempo de transferencia es mucho más pequeño que los tiempos de búsqueda y de latencia. Por ello se tarda menos tiempo en transferir una página de tamaño grande que en transferir varias páginas de tamaño más pequeño. Por otra parte las páginas se escriben o se leen en memoria secundaria una a una. Si el tamaño de página es pequeño, el espacio de direcciones virtuales del proceso se descompondrá en un mayor número de páginas, por lo que la probabilidad de realizar operaciones de E/S será mayor. En consecuencia el tiempo empleado en E/S por el sistema operativo será mayor, tiempo de E/S que no puede ser utilizado por la ejecución de los procesos.

4. (2 p) El sistema operativo en colaboración con el hardware gestiona la memoria principal usando la técnica de demanda de página con un tamaño de página de 4 KiB. La memoria principal del computador tiene una capacidad de 256 MiB con un tamaño de palabra de 16 bits. La unidad direccionable es la palabra. Por otra parte el espacio de direcciones virtuales de un proceso A ocupa 128 MiB. Determinar el tamaño en bits de cada uno de los campos en que se descompone una dirección física y una dirección virtual del proceso A.

Nota: 1 KiB= 1024 bytes.

Cuando se utiliza la técnica de paginación una dirección física se descompone en los campos número de marco de página de f bits y desplazamiento dentro del marco de d bits. Por su parte una dirección lógica se descompone en los campos número de página de p bits y desplazamiento dentro de la página de d bits.

- *Dirección Física:*

Para obtener el tamaño del campo número de marco de página se va a calcular en primer lugar el número de marcos N_{MP} en que se descompone la memoria principal si se considera un tamaño de página $S_p = 4 \text{ KiB}$:

$$N_{MP} = \text{floor}\left(\frac{C_{MP}}{S_p}\right) = \text{floor}\left(\frac{256 \text{ MiB}}{4 \text{ KiB}}\right) = \text{floor}\left(\frac{2^{28}}{2^{12}}\right) = 2^{16} \text{ marcos}$$

Conocido N_{MP} el tamaño de este campo se obtiene resolviendo la siguiente desigualdad:

$$\min_f \{N_{MP} \leq 2^f\}$$

Luego como $N_{MP} = 2^{16}$ entonces $f = 16$ bits.

Para obtener el tamaño del campo desplazamiento primero hay que expresar el tamaño de una página en palabras:

$$S_p = \frac{4 \text{ KiB}}{2 \text{ bytes/palabra}} = \frac{2^{12} \text{ bytes}}{2 \text{ bytes/palabra}} = 2^{11} \text{ palabras}$$

Conocido S_p el tamaño de este campo se obtiene resolviendo la siguiente desigualdad:

$$\min_d \{S_p \leq 2^d\}$$

Luego como $S_p = 2^{11}$ entonces $d = 11$ bits.

- *Dirección Lógica:*

Para obtener el tamaño del campo número de página se va a calcular en primer lugar el número de páginas N_p en que se descompone el espacio de direcciones lógicas del proceso A si se considera un tamaño de página $S_p = 4 \text{ KiB}$:

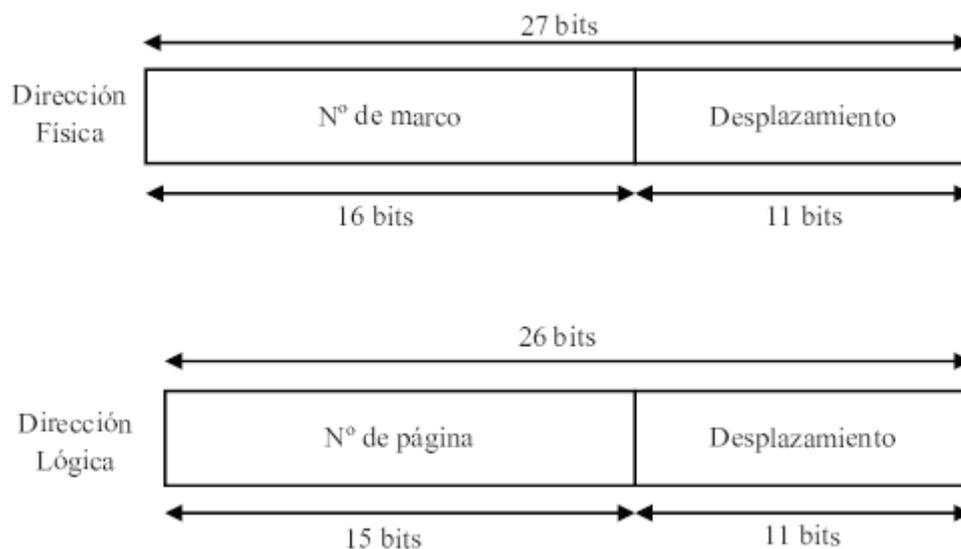
$$N_p = \text{ceil}\left(\frac{C_A}{S_p}\right) = \text{ceil}\left(\frac{128 \text{ MiB}}{4 \text{ KiB}}\right) = \text{ceil}\left(\frac{2^{17} \text{ KiB}}{4 \text{ KiB}}\right) = \text{ceil}(2^{15}) = 2^{15} \text{ páginas}$$

Conocido N_p el tamaño de este campo se obtiene resolviendo la siguiente desigualdad:

$$\underset{p}{\text{mín}} \{N_p \leq 2^p\}$$

Luego $p = 15$.

El tamaño del campo desplazamiento de una dirección lógica es igual al de una dirección física, en este caso, $d = 11 \text{ bits}$. En la figura se representa el formato de una dirección física y de una dirección lógica.



5. Supóngase que un determinado sistema operativo asigna cinco marcos de página para la ejecución de un determinado proceso. Además utiliza el **algoritmo de reemplazamiento del reloj** y para implementarlo utiliza una lista enlazada o cola circular de cinco entradas. Cada entrada contiene el número de una página del proceso cargada en memoria. En la Figura 1 se muestra el estado de la cola circular en el instante de tiempo t_0 y un puntero a una página de la cola. Además se muestra el estado del bit referenciada r de la tabla de páginas del proceso para cada página de la cola. Supóngase que en el instante de tiempo t_0 una referencia a la página $i = 7$ produce un fallo de página. Se pide:

- (1 p) Explicar el funcionamiento del algoritmo de reemplazamiento del reloj.
- (0.5 p) Determinar **razonadamente** la página que sería seleccionada para ser reemplazada al aplicar este algoritmo.
- (0.5 p) Dibujar el estado final de la cola circular y del puntero tras aplicarse el algoritmo y realizarse el reemplazamiento.

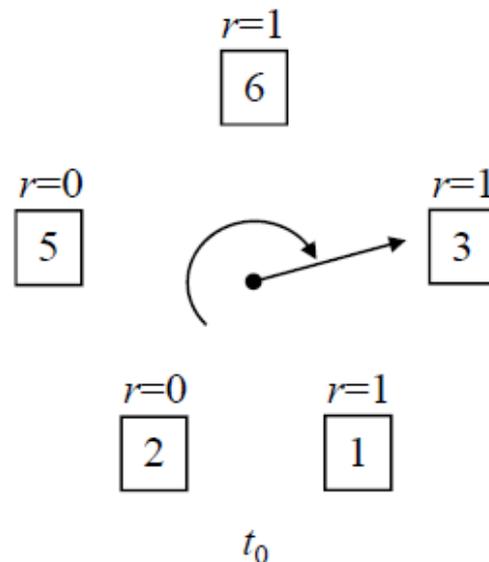


Figura 1

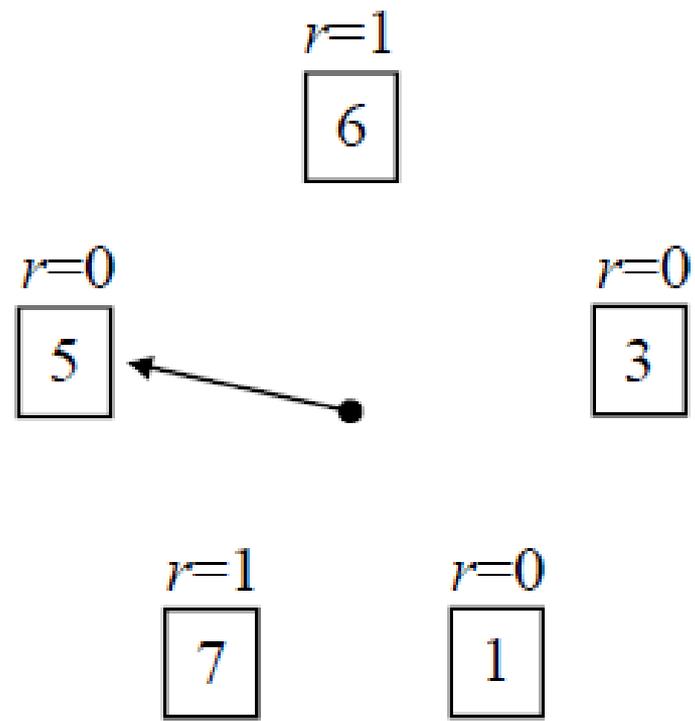


Figura 2

5. (2 p) El sistema operativo en colaboración con el hardware gestiona la memoria principal mediante paginación por demanda. La traducción de direcciones se realiza usando una MMU con un TLB. El tiempo medio de acceso al TLB es despreciable y su tasa de aciertos es del 95 %. Determinar el tiempo medio de despacho de una referencia a memoria si se tienen: una tasa de fallos de página del 8 %, un tiempo medio de acceso a memoria de 90 ns y un tiempo medio de gestión de un fallo de página de 20 ms. Despreciar la existencia de memoria caché.

Solución Ejercicio 5

La MMU dispone de un TLB en el cual se cargan un cierto número de entradas de la tabla de páginas del proceso en ejecución. Cuando se va a traducir una dirección virtual que referencia a una determinada página i se busca en el TLB la entrada i de la tabla de páginas. Si se encuentra se produce un acierto, en caso contrario se produce un fallo y hay que acceder a la memoria principal a leer dicha entrada de la tabla de páginas. Luego el tiempo medio de lectura t_l de la entrada de la tabla de páginas asociada a la página referenciada es:

$$t_l = h t_{ga} + (1 - h) t_f$$

En la expresión anterior h es la tasa de aciertos del TLB, t_{ga} es el tiempo medio de gestión de un acierto en el TLB y t_f es el tiempo medio de gestión de un fallo en el TLB. Del enunciado se sabe que $h = 0,95$ y $t_{ga} = 0$. Además se puede considerar que $t_f = t_{am} = 90$ ns. Luego sustituyendo valores y operando se obtiene:

$$t_l = 0,95 \cdot 0 + 0,05 \cdot 90 \cdot 10^{-9} = 4,5 \cdot 10^{-9} = 4,5 \text{ ns}$$

El tiempo medio de despacho de una referencia a memoria t_R se calcula de la siguiente forma:

$$t_R = t_l + (1 - p) t_{am} + p t_{gf}$$

donde p es la tasa de fallos de página y t_{gf} es el tiempo medio de gestión de un fallo de página. Sustituyendo valores y operando se obtiene:

$$t_R = 4,5 \cdot 10^{-9} + 0,92 \cdot 90 \cdot 10^{-9} + 0,08 \cdot 20 \cdot 10^{-3} = 1,6000873 \text{ ms}$$