Tema VII

Memoria Virtual

Memoria virtual

- Es un esquema de gestión de memoria en donde puede que sólo una parte del espacio de direcciones virtuales de un proceso residente sea cargada realmente en memoria física
  - La memoria virtual permite la ejecución de procesos parcialmente cargados
- Esto se consigue manteniendo una imagen del espacio de direcciones virtuales completo de un proceso en memoria secundaria, y trayendo a memoria principal parte de esa imagen cuando sea necesaria. La elección de que sección traer, cuando y donde es efectuada por el s.o.
- La memoria virtual puede implementarse como extensión de la gestión de memoria paginada, segmentada o como una combinación de ambas
- 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**
- Por lo tanto 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
- La implementación de la memoria virtual requiere que el hardware del computador soporte el reinicio de las instrucciones de su repertorio.
- Además cuando se implementa la memoria virtual también se suele utilizar un componente hardware denominado unidad de gestión de memoria (Memory Managemeru 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 denomina *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

Páginas	Otros	r	m	$p_2p_1p_0$	v	Marco j
i=0		1	0	101	1	30
i=1		0	0	110	0	
i=2		0	0	110	0	
i=3		1	0	101	1	36

Tabla de páginas proceso A

Páginas	Otros	r	m	$p_2p_1p_0$	v	Marco j
i=0		0	0	100	0	
i=1		0	0	100	0	
i=2		0	0	101	0	
i=3		1	0	101	1	40
i=4		1	1	110	1	33
i=5		1	0	100	1	34

Tabla de páginas proceso B

Marcos	
j=30	Página i=0 (A)
j=31	Página i=45 (F)
j=32	Libre
j=33	Página i=4 (B)
j=34	Página i=5 (B)
j=35	Página i=67 (F)
j=36	Página i=3 (A)
j=37	Libre
j=38	Libre
j=39	Página i=21 (G)
j=40	Página i=3 (B)

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

- 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**
  - *Bloque de disco de un archivo ejecutable*
  - *Bloque de disco del área de intercambio*

Ubicación de todo el espacio virtual en el area de intercambio

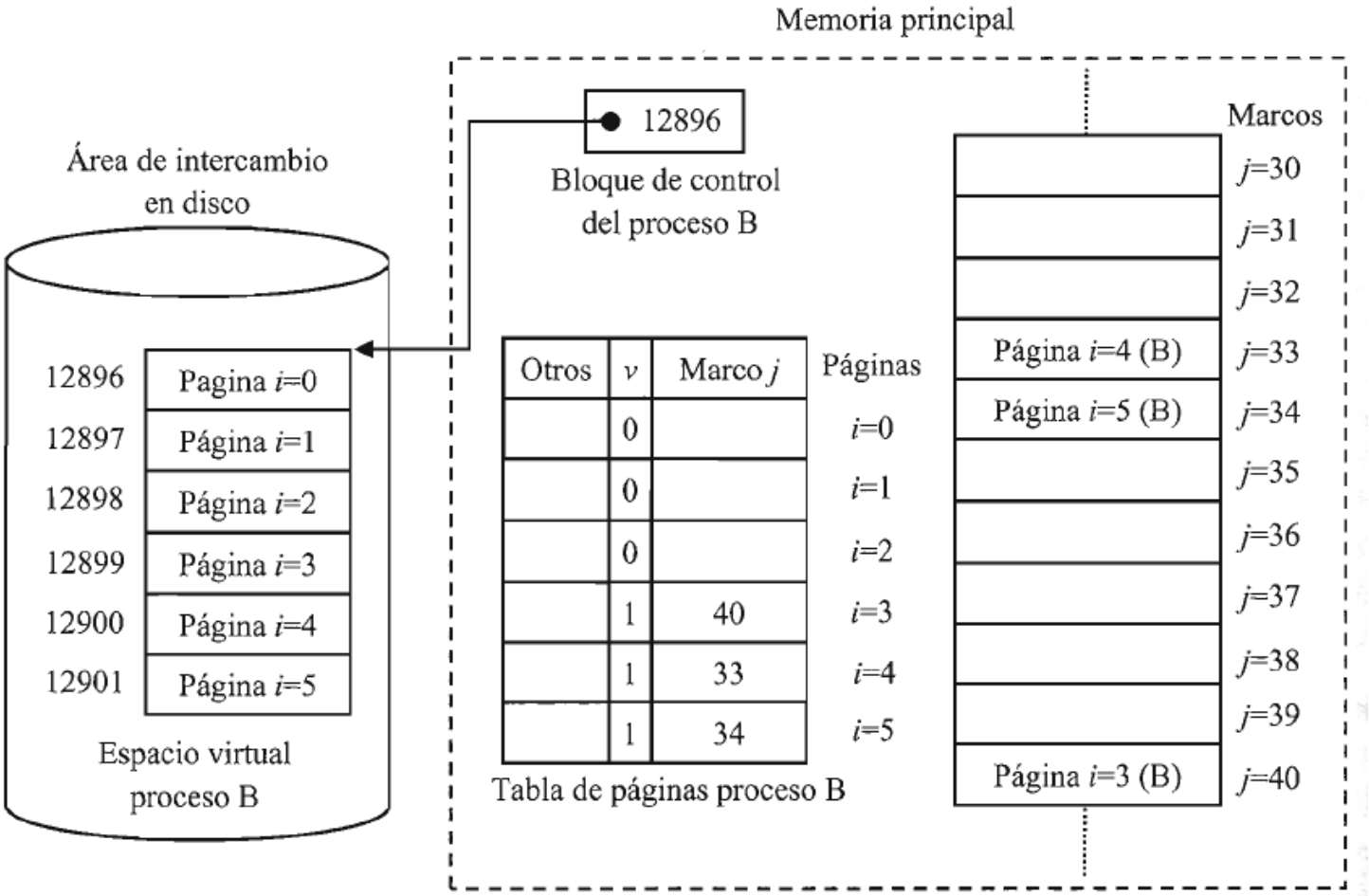

Figura 7.2 – Localización de las páginas de un proceso en el área de intercambio mediante una dirección de disco base ubicada en el bloque de control del proceso

Uso de un mapa de disco

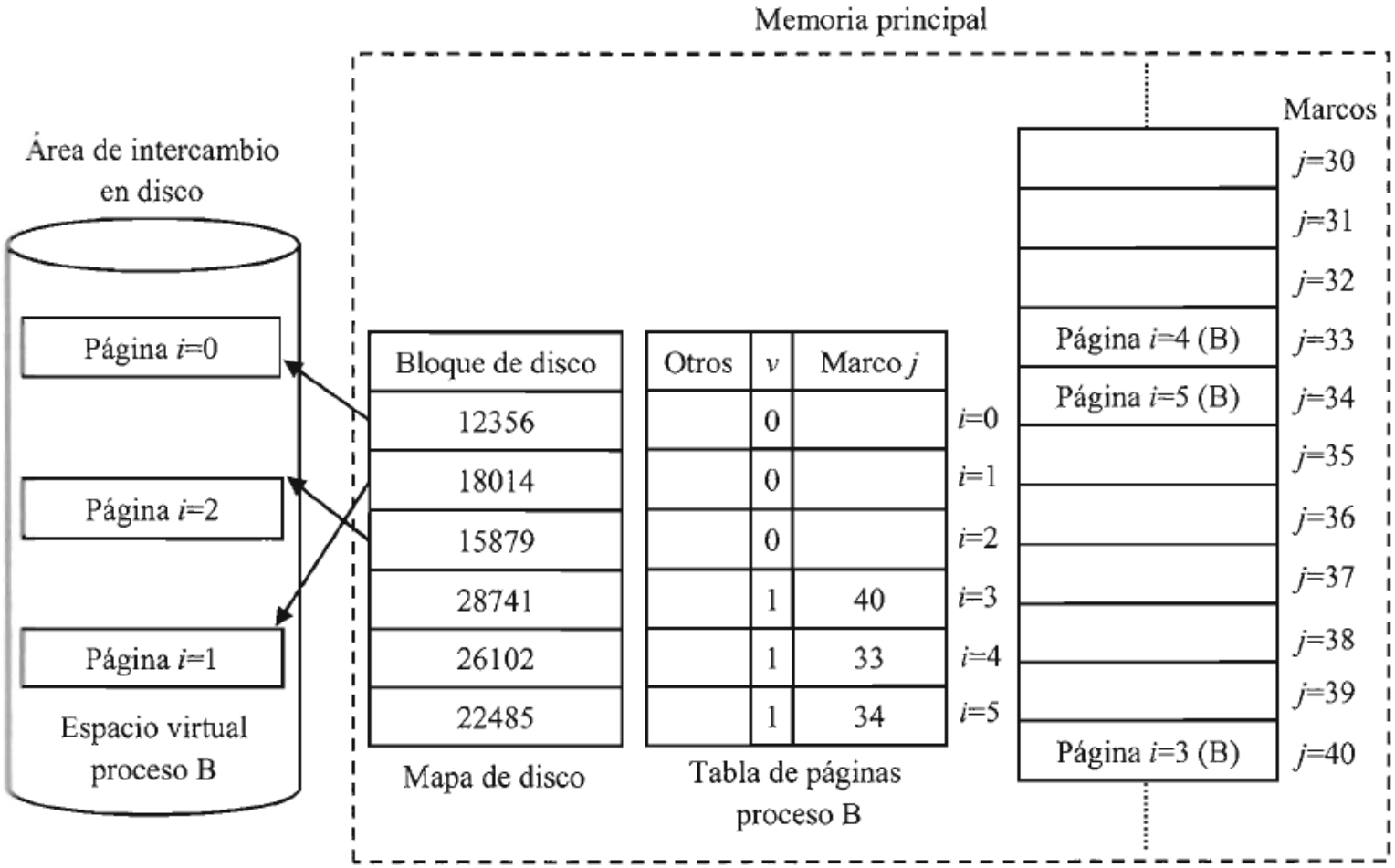

Figura 7.3 – Localización de las páginas de un proceso en al área de intercambio mediante el uso de un mapa de disco

- 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 una excepción por elemento ausente para anunciar el hecho al s.o.
  - A esta excepción se denomina fallo de página
- Por lo tanto 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, así el procedimiento aseguir 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

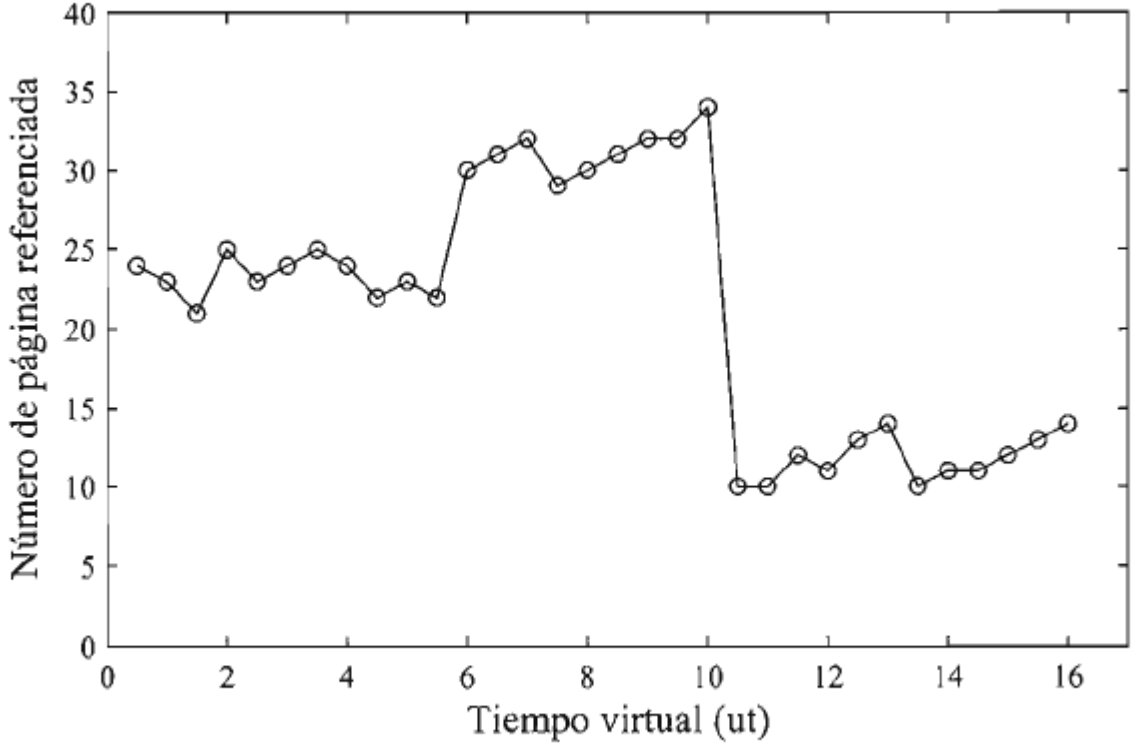

Figura 7.4 – Números de páginas referenciados durante la ejecución del proceso del Ejemplo 7.5

Conjunto d etrabajo

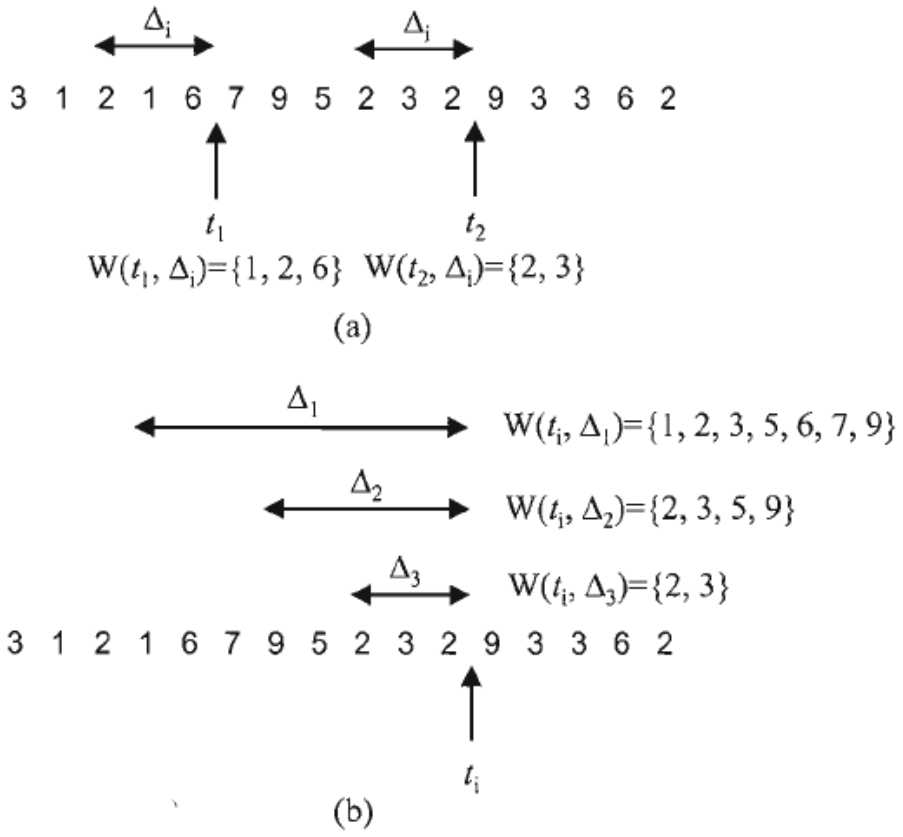

Figura 7.5 – Conjunto de trabajo de un proceso con distintos instantes de tiempo virtual (a) y distintas ventanas de tiempo virtual (b)

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

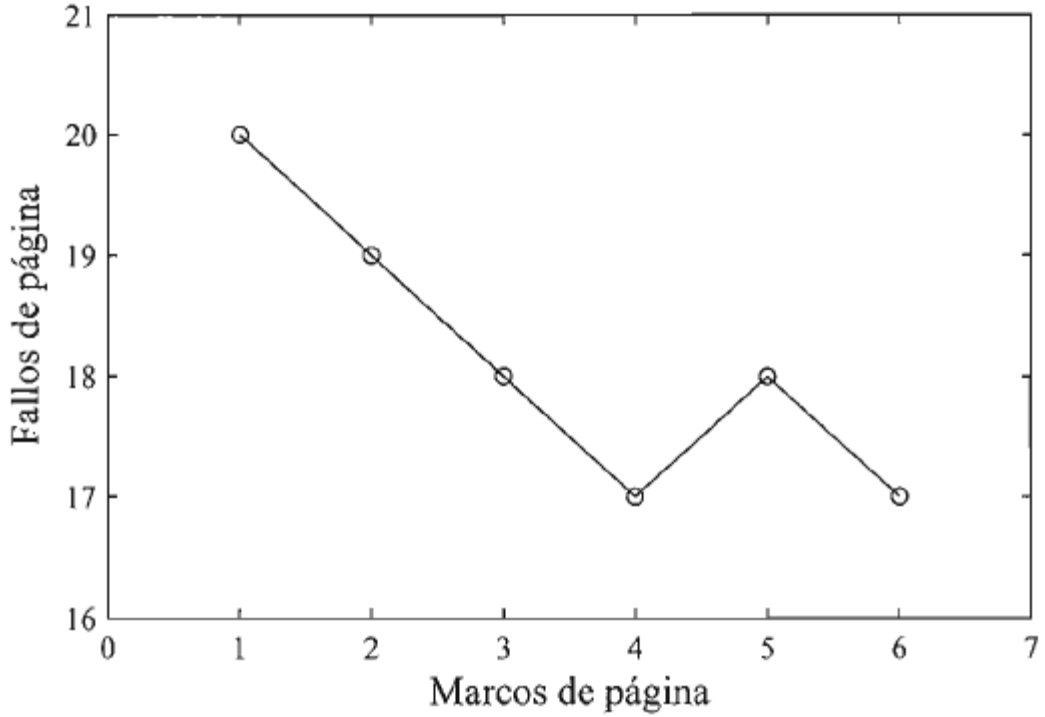

Figura 7.6 – Ilustración de la anomalía de Belady

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.

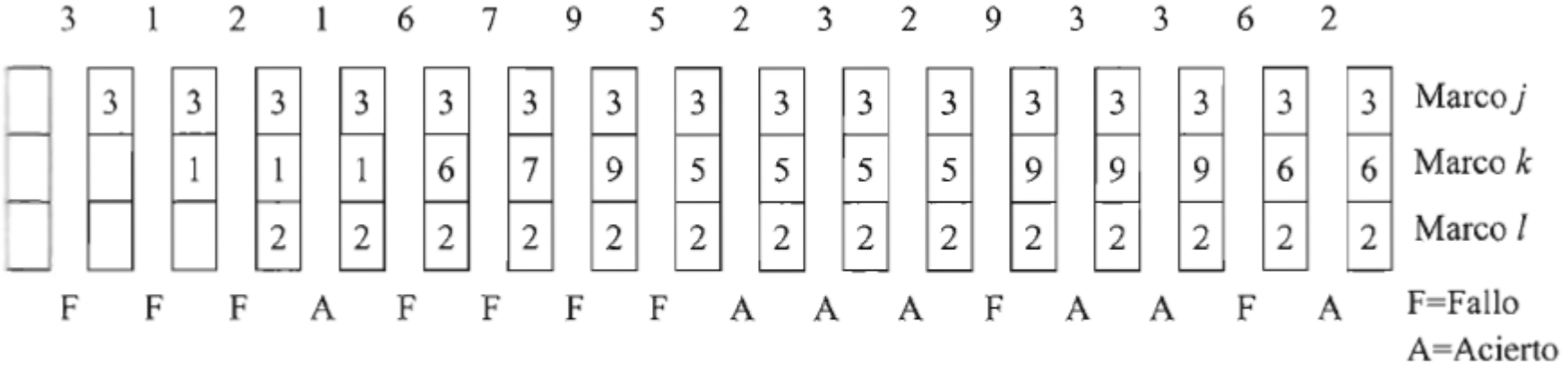

Figura 7.7 – Ejemplo de aplicación del algoritmo de reemplazamiento de página óptimo

Algoritmo de reemplazamiento LRU

- El algoritmo de reemplazamiento de la página usada menos recientemente o algoritmo de reemplazamiento LRU (Least Recently Used) selecciona para ser reemplazada aquella página del conjunto de páginas candidatas a ser reemplazadas que lleva más tiempo sin ser referenciada
- Dos posibles realizaciones:
  - Mediante una pila que mantiene los números de las páginas, cada vez que una página se referencia, su número se elimina de la pila y se coloca en la cumbre de la misma. De esta forma, en la parte superior dela pila se tiene siempre el número de la última página usada y en el fondo el de la página que hace más tiempo que se usó
  - Mediante registros contadores

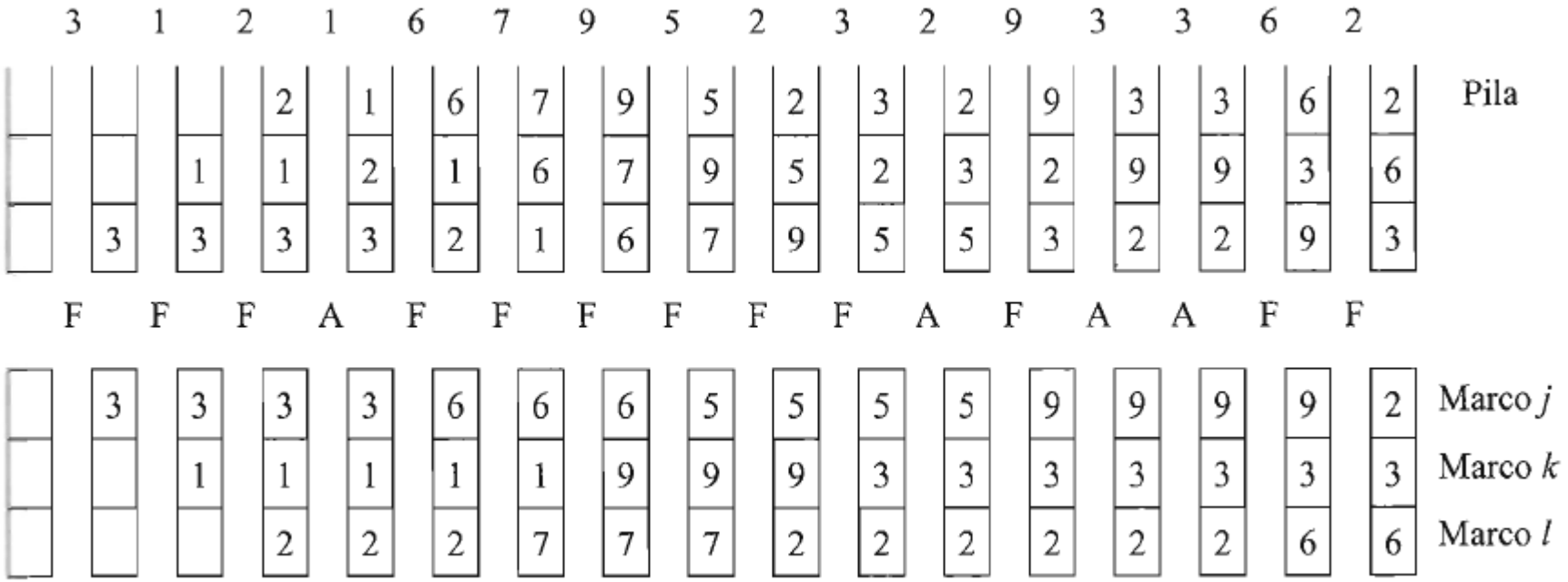

Figura 7.8 – Ejemplo de aplicación del algoritmo de reemplazamiento LRU

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

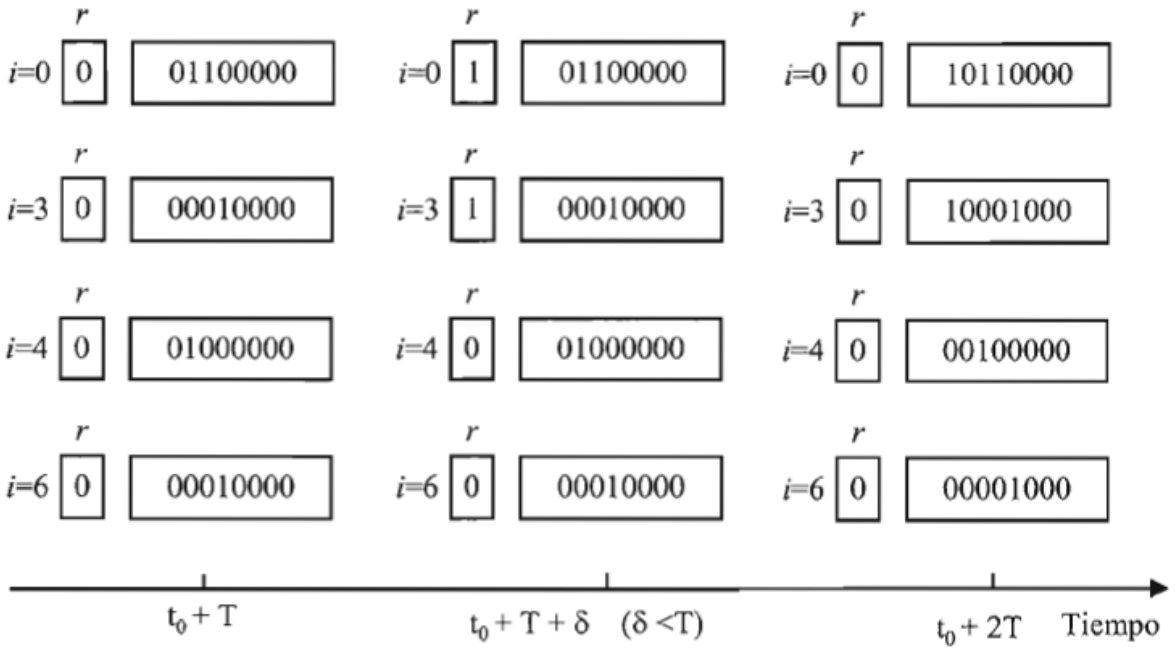

Figura 7.9 – Ejemplo de aplicación del algoritmo de envejecimiento

Algoritmo de reemplazamiento FIFO

- Se sustituye la página que lleva más tiempo en memoria. En realidad, no es necesario guardar el tiempo de entrada, ya que 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

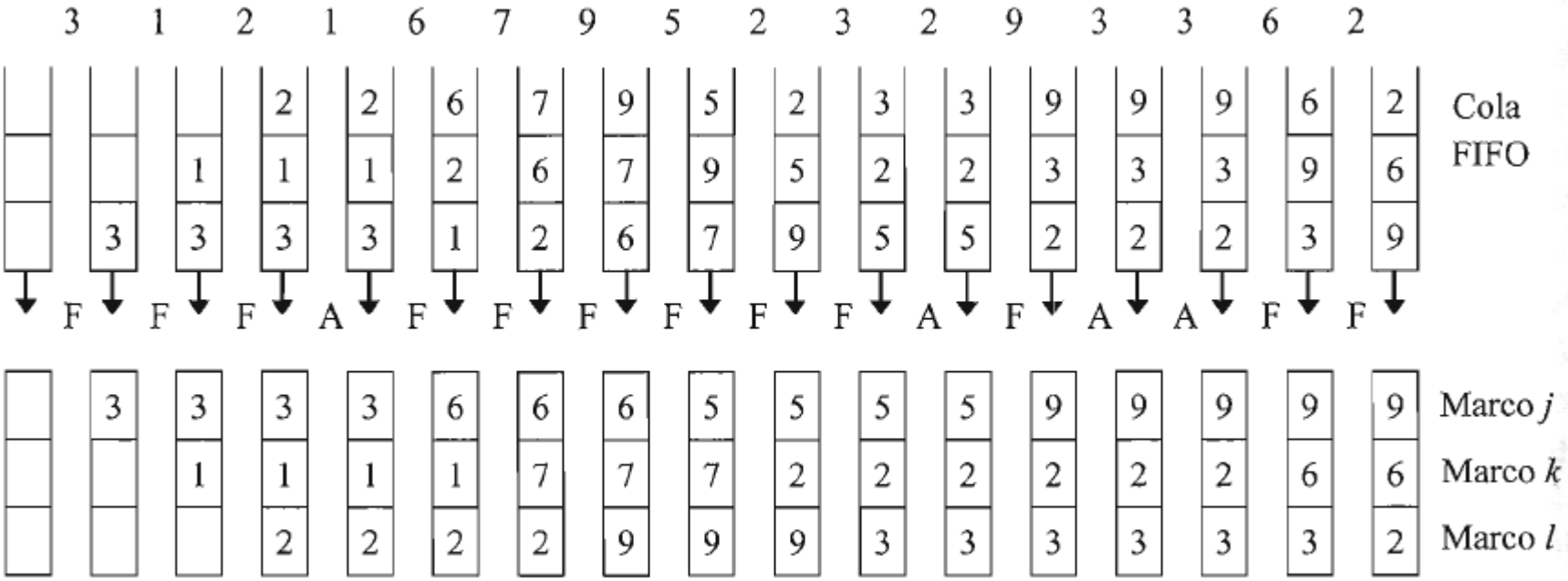

Figura 7.10 – Ejemplo de aplicación del algoritmo de reemplazamiento FIFO

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* = O; 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 O y coloca el número de la página al de la cola, es decir, se le da una segunda oportunidad.

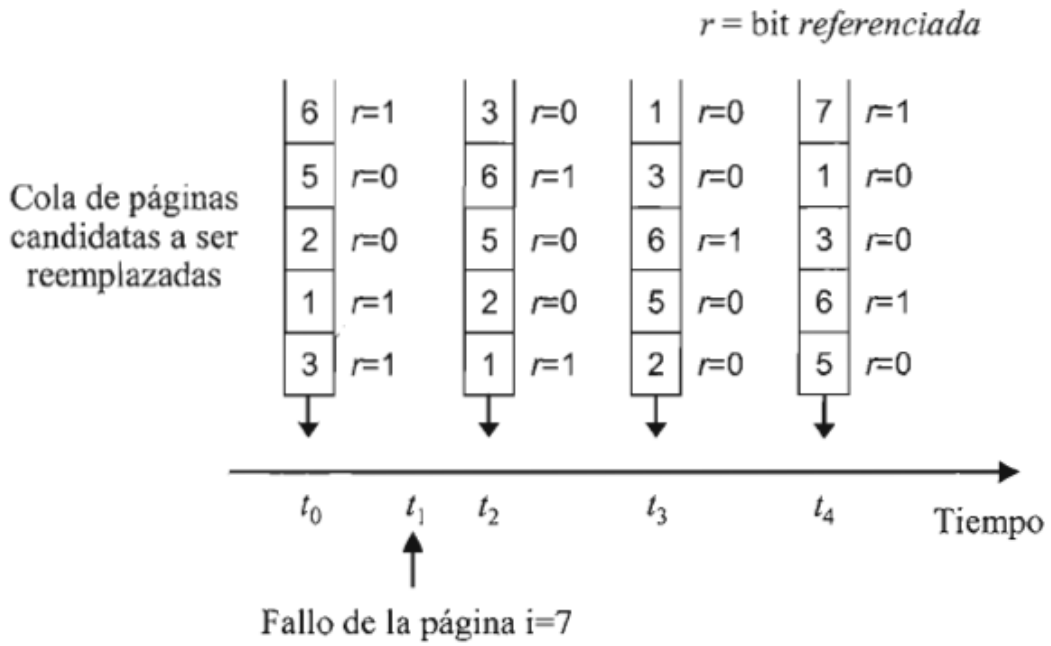

Figura 7.11 – Ejemplo de aplicación del algoritmo de la segunda oportunidad

Segunda oportunidad o Reloj

- 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

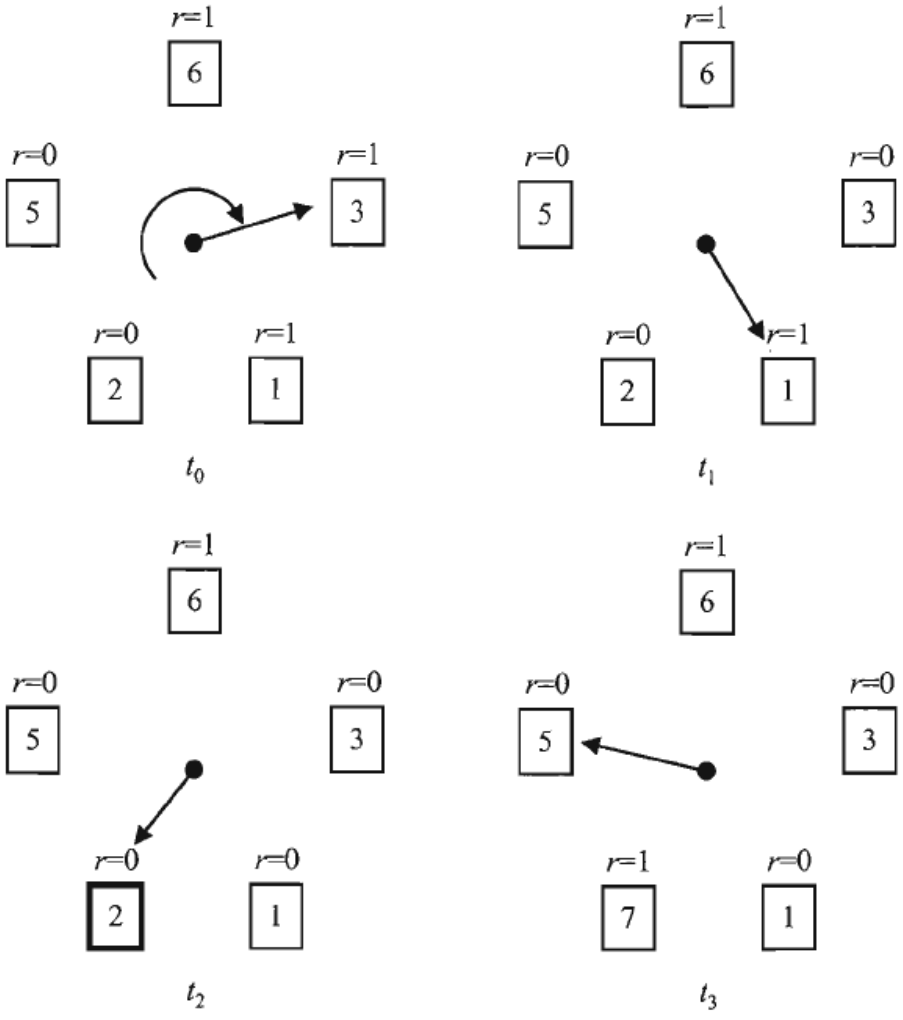

Figura 7.12 – Ejemplo de aplicación del algoritmo del reloj

Algoritmo mejorado

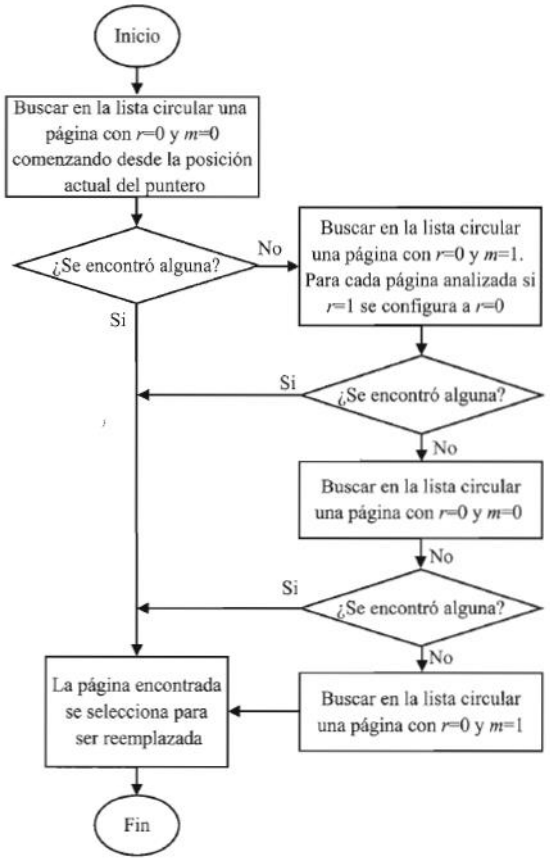

Figura 7.13 – Algoritmo del reloj mejorado

Algoritmo de reemplazamiento del reloj considerando el

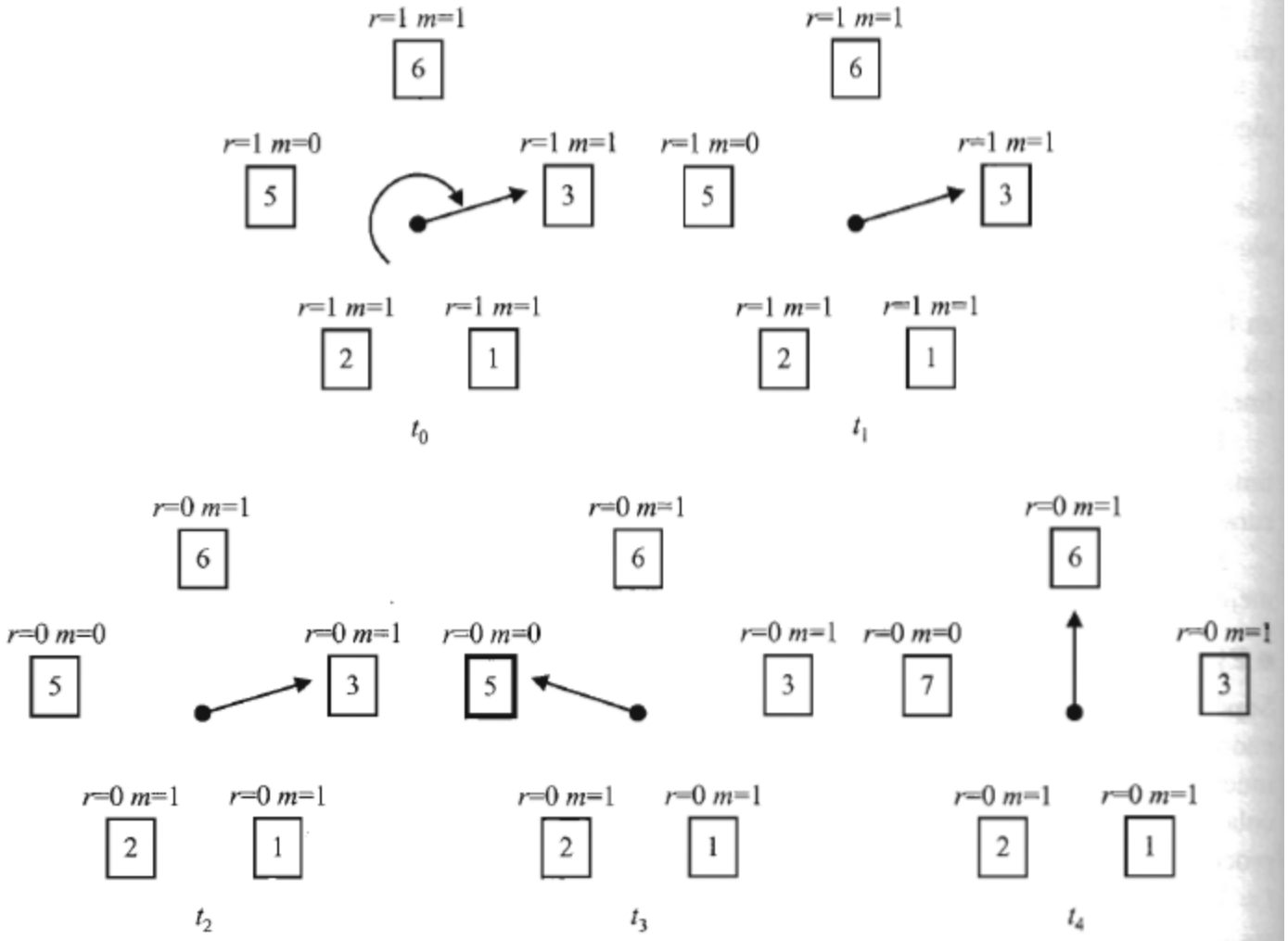

Figura 7.14 – Ejemplo de aplicación del algoritmo del reloj mejorado

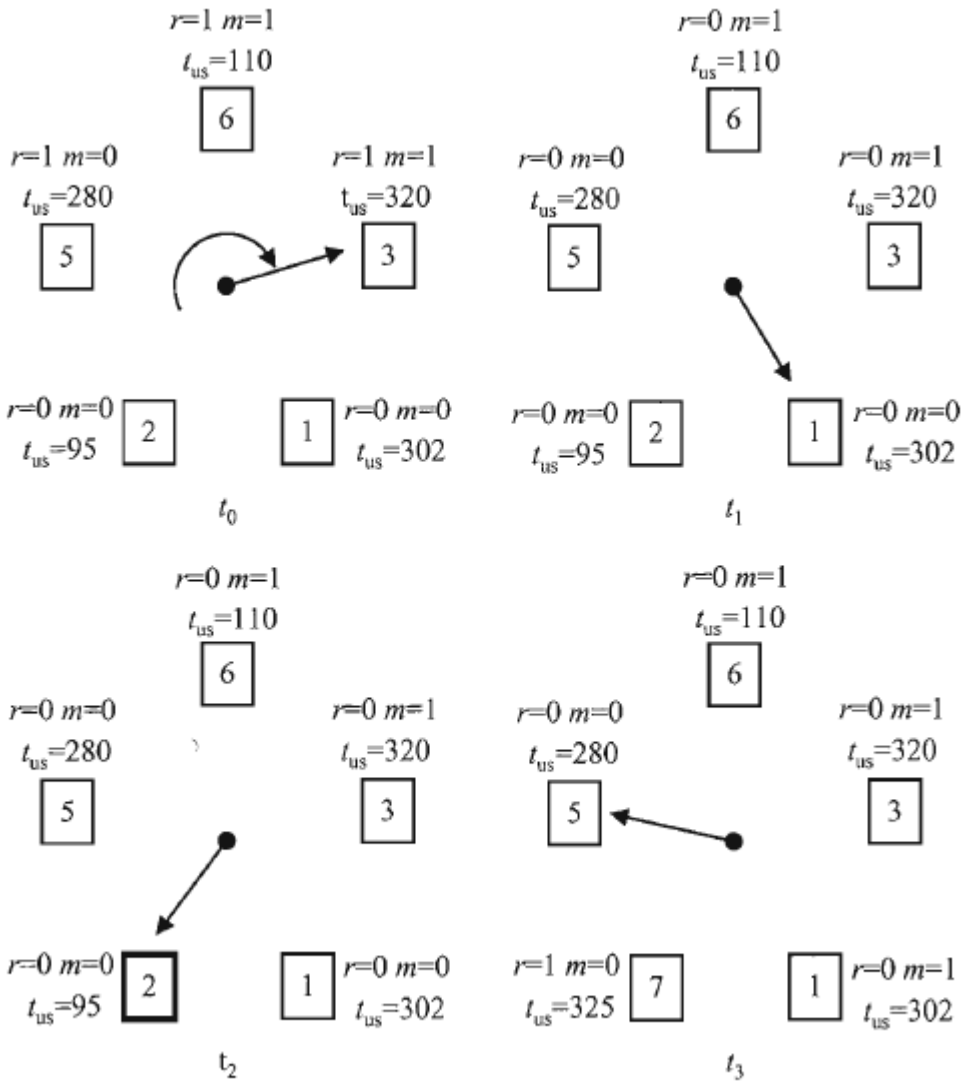

Figura 7.15 – Ejemplo de aplicación del algoritmo WSClock para t_v = 325 ut y Δ = 25 ut

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 P FF*

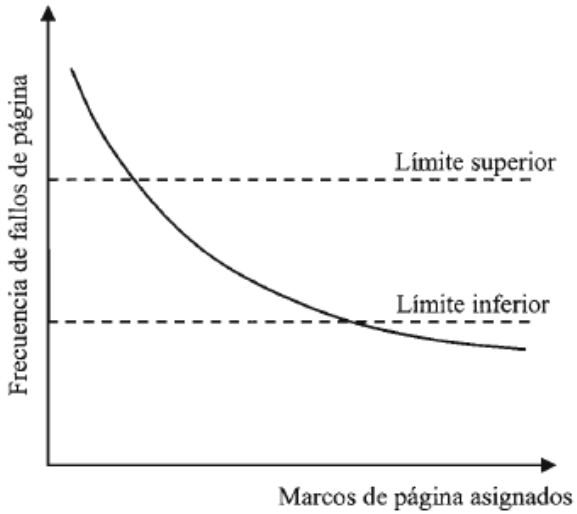

Figura 7.16 – Representación gráfica de la frecuencia de fallos de página en función del número de marcos asignados

Control de carga

- debe controlar el *grado de multiprogramación del sistema* GM.

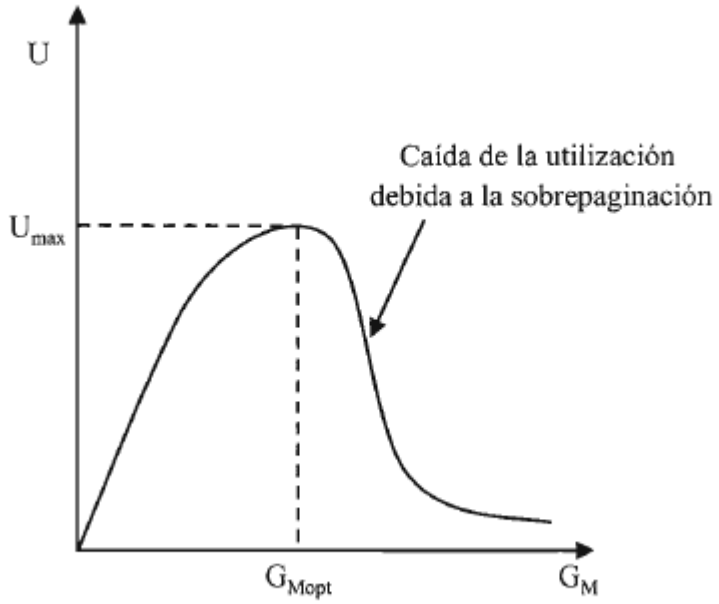

Figura 7.17 – Influencia de grado de multiprogramación sobre la utilización del procesador

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 algoritmode 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 NpA asociadas a un proceso antes de iniciar o continuar con su ejecución.
- Cargar el conjunto de trabajo del proceso
- Cargar un conjunto de páginas ubicadas de forma contigua en el área de intercambio

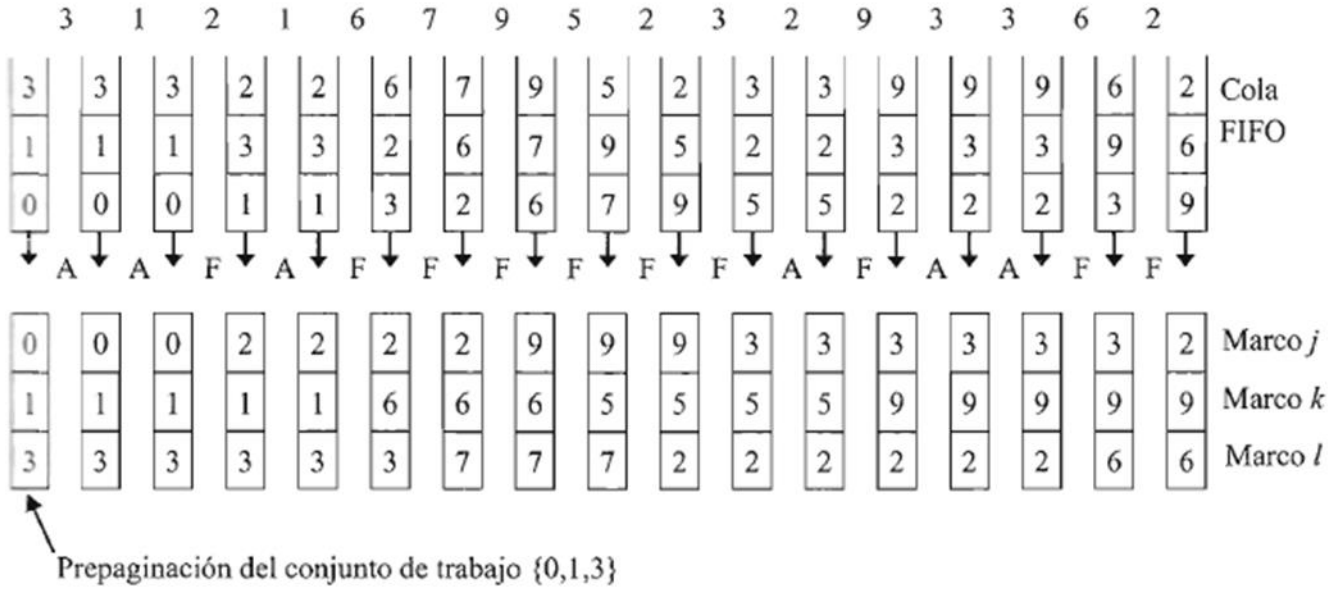

Figura 7.18 – Ejemplo de aplicación del algoritmo de reemplazamiento FIFO con prepaginación del conjunto de trabajo