**1*. Si un computador no posee hardware de reubicación, e implementa intercambio (swapping), entonces el gestor de memoria necesita utilizar un cargador para recalcular las direcciones físicas de un proceso. ¿Sería posible para el sistema de intercambio reubicar los segmentos de datos y pila? Explica cómo funcionaría este sistema, o si es imposible que funcione.***

La única forma de implementar el swapping sería cargar las páginas en las mismas posiciones de memoria en que se cargan inicialmente. Sería posible que el cargador recalculase las direcciones en base al desplazamiento respecto a las posición cuando se cargó la página por última vez. Cualquier referencia a una posición de memoria, por ejemplo, un puntero, quedaría invalidado por el SO, debido a que este conoce las direcciones y puede recalcularlas, pero no su contenido que podría ser una referencia a memoria que quedaría sin recalcular.

***2. Considere un sistema con un espacio lógico de memoria de 128K páginas (máximo espacio de memoria virtual) con 8 KB cada una, una memoria física de 64 MB y direccionamiento al nivel de byte. ¿Cuántos bits hay en la dirección lógica? ¿Y en la física?***

128 KB páginas ---> 217 bytes => 17 direcciones (páginas)

Tamaño de las páginas ---> 8 KB / página = 213 bytes / página => 13 bits de desplazamiento

Memoria física = 64 MB ---> 226 bytes => 26 bits en la dirección física.

Solución:

Bits en la dirección lógica => 30 bits que se descomponen de la siguiente forma:

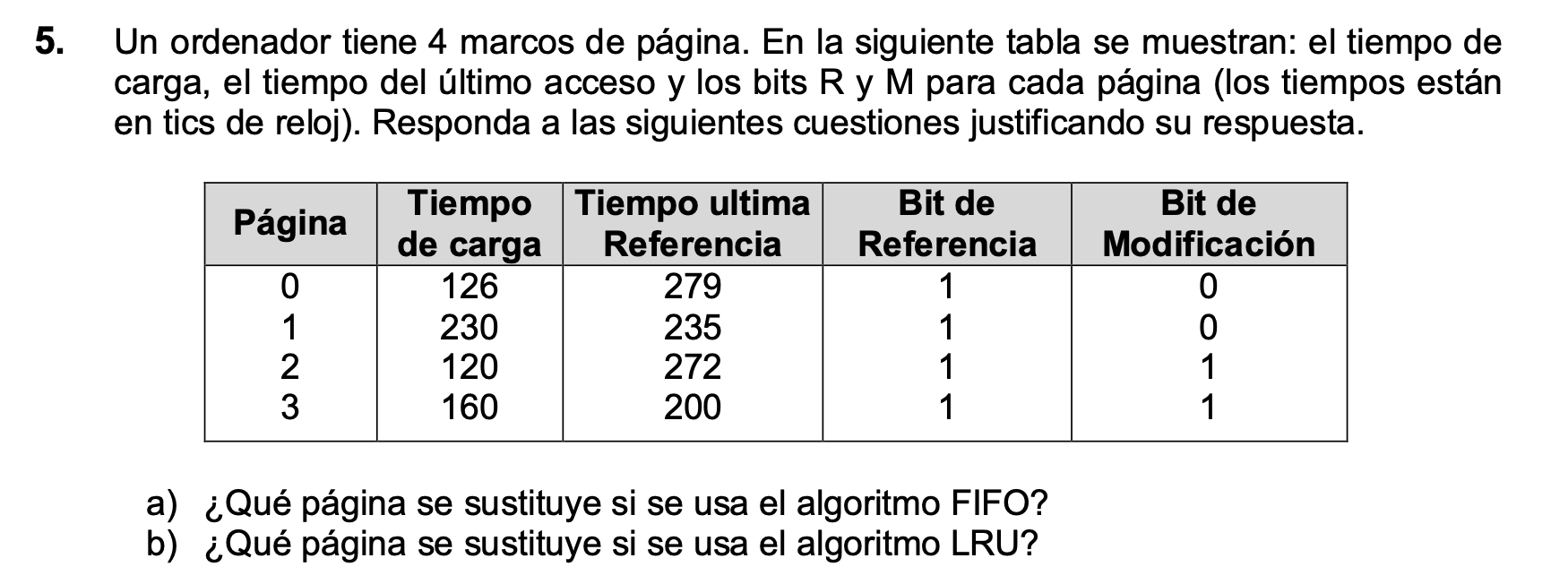
|  |  |
| --- | --- |
| 17 bits : nº páginas / marco | 13 bits : desplazamiento |

Bits en la dirección física => 26 bits

***3. Considérese un sistema con memoria virtual en el que el procesador tiene una tasa de utilización del 15% y el dispositivo de paginación está ocupado el 97% del tiempo, ¿qué indican estas medidas? ¿Y si con el mismo porcentaje de uso del procesador el porcentaje de uso del dispositivo de paginación fuera del 15%?***

Si el dispositivo de paginación está ocupado el 97% del tiempo, estaría el procesador paginando continuamente por lo que nos encontraremos en un estado de hiperpaginación. Es decir, que el proceso pasa más tiempo en la cola de servicio del dispositivo de paginación que ejecutando lo que afectaría al proceso y a todo el sistema. Si el porcentaje de ambos es bajo, hay espacio libre por lo que podríamos ejecutar varios procesos a la vez aumentando el grado de multiprogramación.

***4. Sea un sistema de memoria virtual paginada con direcciones lógicas de 32 bits que proporciona un espacio virtual de 2^20 páginas y con una memoria física de 32 Mbytes ¿cuánta memoria requiere en total un proceso que tenga 453Kbytes, incluida su tabla de páginas cuyas entradas son de 32 bits?***

******

*a) Miramos el tiempo de carga, por tanto, la pagina con menor tiempo de carga la sustituimos en este caso,*

***6. ¿Depende el tamaño del conjunto de trabajo de un proceso directamente del tamaño del programa ejecutable asociado a él? Justifique su respuesta.***

No, ya que es posible que el proceso, sea grande o sea pequeño, sólo necesite acceder a una parte del segmento de datos con regularidad (por el principio de proximidad) y no al segmento completo, por lo que el conjunto del trabajo variaría en tamaño respecto al ejecutable.

***7. ¿Por qué una cache (o la TLB) que se accede con direcciones virtuales puede producir incoherencias y requiere que el sistema operativo la invalide en cada cambio de contexto y, en cambio, una que se accede con direcciones físicas no lo requiere?***

Porque las direcciones virtuales son relativas a cada proceso y las físicas se mantienen siempre en la misma posición en memoria.

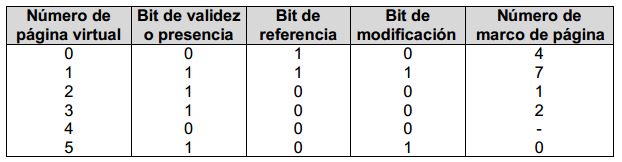
***8. Un ordenador proporciona un espacio de direccionamiento lógico (virtual) a cada proceso de 65.536 bytes de espacio dividido en páginas de 4096 bytes. Cierto programa tiene un tamaño de región de texto de 32768 bytes, un tamaño de región de datos de 16386 bytes y tamaño de región de pila de 15878. ¿Cabría este programa en el espacio de direcciones? (Una página no puede ser utilizada por regiones distintas). Si no es así, ¿cómo podríamos conseguirlo, dentro del esquema de paginación?***

***9. Analice qué puede ocurrir en un sistema que usa paginación por demanda si se re-compila un programa mientras se está ejecutando. Proponga soluciones a los problemas que pueden surgir en esta situación.***

En la paginación por demanda. Las páginas de los procesos residen en memoria secundaria. Cuando un proceso se reincorpora, el paginador lleva a memoria las páginas necesarias. Evitando, así, tener en memoria páginas que no se utilizarán.

Un posible problema que puede surgir en la situación descrita es que al recompilar cambien las páginas del proceso (aquellas que están en memoria secundaria) y que por tanto, por la lógica seguida en la paginación por demanda, en memoria principal queden las páginas antiguas. Una posible solución sería matar el proceso e inmediatamente crear un proceso nuevo, ahora sí, cuando el paginador lleve a memoria las páginas de este proceso llevará las páginas nuevas. Otra posible solución para el mismo problema sería eliminar las páginas antiguas de la memoria principal de manera que cuando el proceso requiera de esas páginas se producirá una falta de página y se llevarán a memoria principal las páginas actualizadas.

***11. Suponga que la tabla de páginas para el proceso actual se parece a la de la figura. Todos los números son decimales, la numeración comienza en todos los casos desde cero, y todas las direcciones de memoria son direcciones en bytes. El tamaño de página es de 1024 bytes.***

****

Si tenemos un tamaño de página de 1024 bytes(2^10), el tamaño del campo desplazamiento es de 10 bits.

**¿Qué direcciones físicas, si existen, corresponderán con cada una de las siguientes direcciones virtuales? (no intente manejar ninguna falta de página, si las hubiese)**

**a) 999** → nº de página virtual 0, la página no está presente

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Número de página virtual → 0   |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | | Desplazamiento (10 bits) = 999   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | |

**b) 2121** → nº de página virtual 2, la página se encuentra en el marco 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Número de página virtual → 2   |  |  |  | | --- | --- | --- | | 0 | 1 | 0 | | Desplazamiento (10 bits)   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | |

Por lo tanto…

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Número de marco → 1   |  |  |  | | --- | --- | --- | | 0 | 0 | 1 | | Desplazamiento (10 bits)   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | |

Dirección física = 1097

**c) 5400** → nº de página virtual 5, la página se encuentra en el marco 0

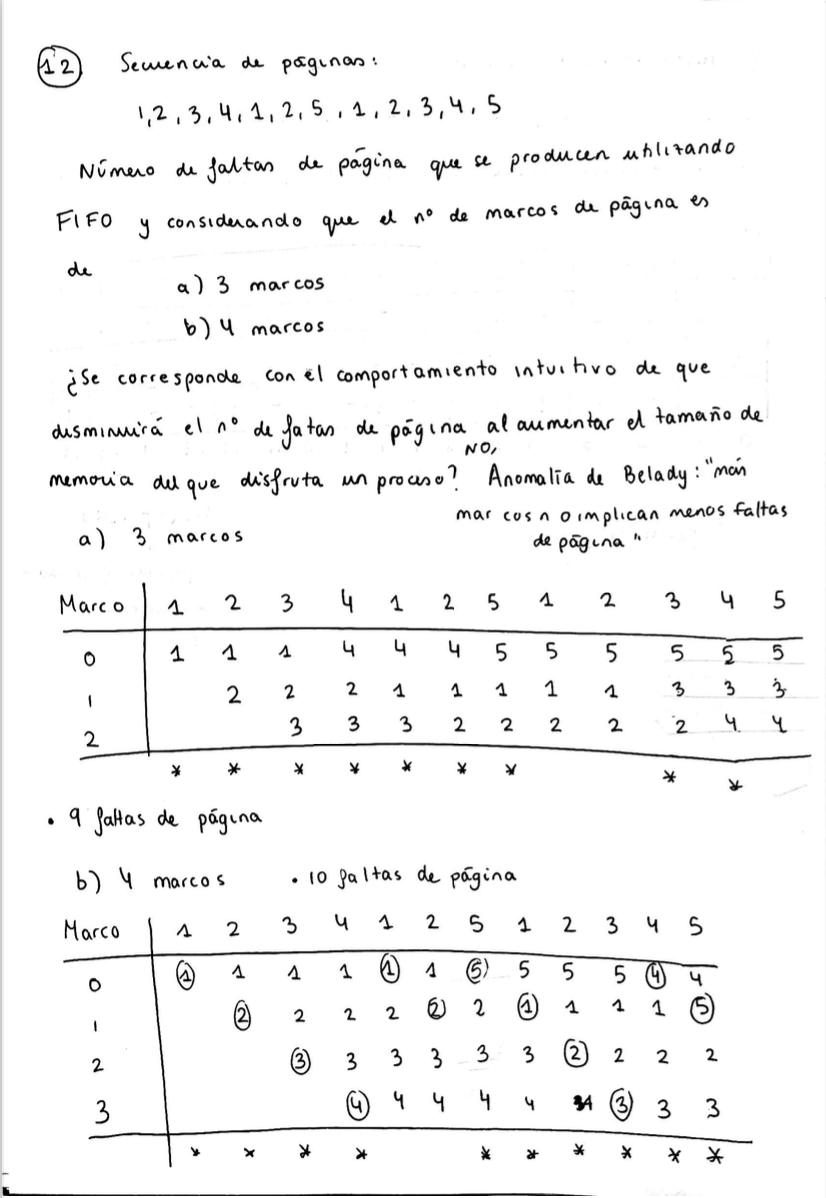
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Número de página virtual → 5   |  |  |  | | --- | --- | --- | | 1 | 0 | 1 | | Desplazamiento (10 bits)   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | |

Por lo tanto…

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Número de marco →   |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | | Desplazamiento (10 bits)   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | |

Dirección física = 280

***12. Sea la siguiente secuencia de números de página referenciados: 1,2,3,4,1,2,5,1,2,3,4,5 Calcula el número de faltas de página que se producen utilizando el algoritmo FIFO y considerando que el número de marcos de página de que disfruta nuestro proceso es de  a) 3 marcos b) 4 marcos  ¿Se corresponde esto con el comportamiento intuitivo de que disminuirá el número de faltas de página al aumentar el tamaño de memoria de que disfruta el proceso?***

******

***13. ¿Por qué la localidad no es un factor que se tiene en cuenta en los sistemas con segmentación? (SPOCK)***

¿Es porque en segmentación no estan contiguos los bloques?

***14. En la gestión de memoria en un sistema paginado, ¿qué estructura/s de datos necesitará mantener el Sistema Operativo para administrar el espacio libre?***

La estructura de la tabla de páginas que tiene cada proceso ya que contiene información sobre el marco de página donde se almacena en memoria y toda la información correspondiente

***16. Situándonos en un sistema paginado, donde cada proceso tiene asignado un número fijo de marcos de páginas. Supongamos la siguiente situación: existe un proceso con 7 páginas y tiene asignados 5 marcos de página. Indica el contenido de la memoria después de cada referencia a una página si como algoritmo de sustitución de página utilizamos el LRU (la página no referenciada hace más tiempo). La secuencia de referencias es la indicada en la figura.***

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Referencias | 2 | 1 | 3 | 4 | 1 | 5 | 6 | 4 | 5 | 7 | 4 | 2 |
| Marcos de página | 2 | 2 | 2 | 2 | 2 | 2 | 6 | 6 | 6 | 6 | 6 | 6 |
|  |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 |
|  |  |  | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 7 | 7 | 7 |
|  |  |  |  | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
|  |  |  |  |  |  | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
|  | \*\* | \* | \* | \* |  | \* | \* |  |  | \* |  | \* |

\*\****En la primera columna hay falta de página pero no me deja poner el asterisco***\*\*

Se producen 8 faltas de página.

***17. ¿Cuál es la ventaja del algoritmo de faltas de página sobre el algoritmo basado en el modelo del conjunto de trabajo utilizando el tamaño de ventana w? ¿Cuál es la desventaja?***

Si se producen muchas faltas de página, se realizan comprobaciones de más, lo que resulta en una sobrecarga hardware innecesaria, pues hay muchos fallos seguidos y no superarán el intervalo.

Si hay pocos fallos de página, se pueden tener páginas que no se han referenciado desde hace mucho en memoria, y por tanto están ocupando memoria.

***19. Describa el funcionamiento del algoritmo de sustitución basado en la frecuencia de faltas de página, con los siguientes datos: 4 marcos de página, en t = 0 la memoria contiene a la página 2. El tamaño de la ventana es τ= 3 y se produce la secuencia de referencias de páginas, 1 4 2 2 2 4 5 5 3 3 5 1 1 1 1 4***

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 1 | 1 | 1 | 1 | 1 | 5 | 5 | 5 | 5 | 5 | 5 | - | - | - | - |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | - | - | - | - |
|  | - | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | - | - | - | - |
|  | - | - | - | - | - | - | - | - | 3 | 3 | 3 | 3 | - | - | - | - |

t= 0 1\* 2\* 3 4 5 6 7\* 8 9\* 10 11 12!\* 13 14 15 16

\* → falta de página, se activa algoritmo de sustitución

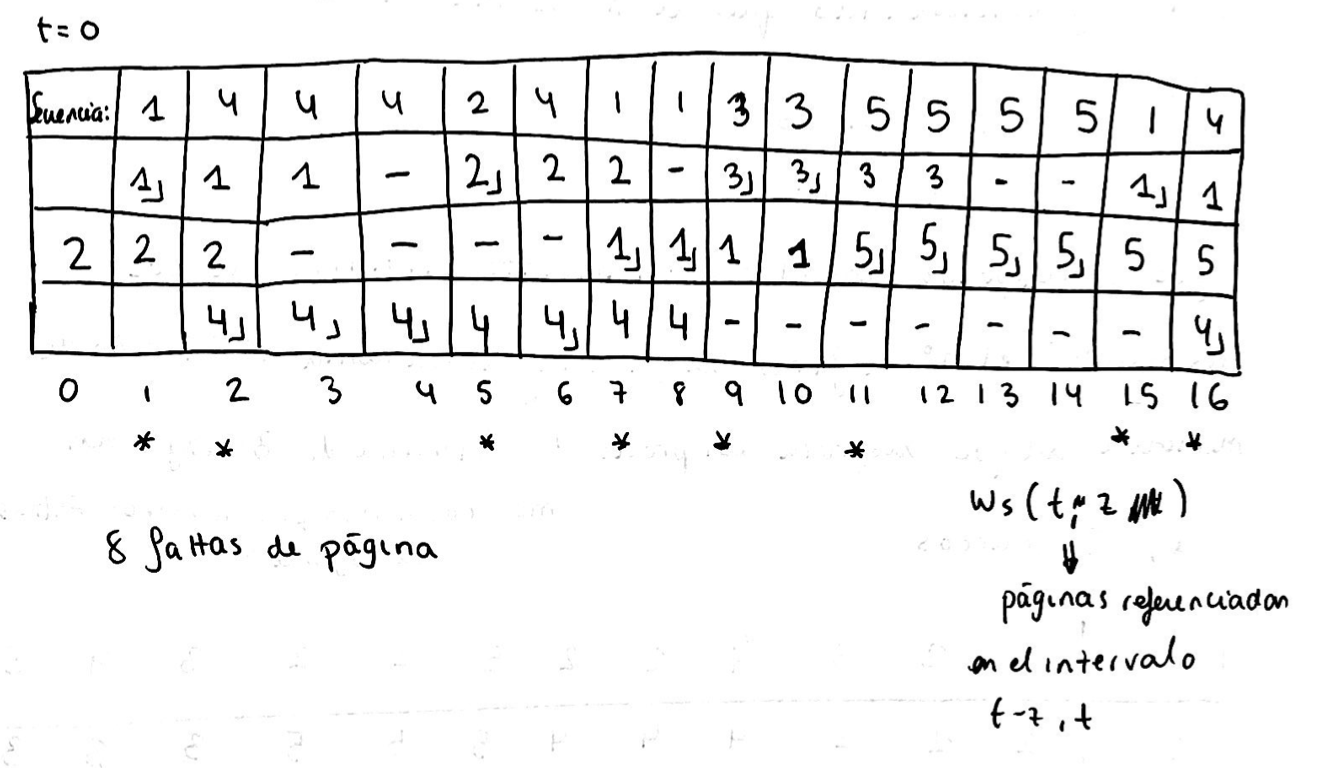
! → bloqueo, no disponemos de marcos suficientes. El proceso debe esperar hasta tener los recursos necesarios para ampliar su tamaño

**** Z (tc-1, tc) si tc - tc-1 > **τ**

R(tc,Y)

**** R (tc-1, **τ**) + Z(tc) en otro caso

***20. Describa el funcionamiento del algoritmo de sustitución global basado en el algoritmo basado en el modelo del conjunto de trabajo, con los siguientes datos: 4 marcos de página, en t= 0 la memoria contiene a la página 2 que se referenció en dicho instante de tiempo. El tamaño de la ventana es τ= 3 y se produce la secuencia de referencias de páginas,1 4 4 4 2 4 1 1 3 3 5 5 5 5 1 4***



***21. Una computadora con memoria virtual paginada tiene un bit U por página virtual, que se pone automáticamente a 1 cuando se realiza un acceso a la página. Existe una instrucción limpiar\_U (dir\_base\_tabla) que permite poner a 0 el conjunto de los bits U de todas las páginas de la tabla de páginas cuya dirección de comienzo pasamos como argumento. Explica cómo puede utilizarse este mecanismo para la implementación de un algoritmo de sustitución basado en el modelo del conjunto de trabajo.***

En el modelo del conjunto de trabajo, todas las paginas que no han sido referenciadas en el intervalo de tiempo (t-z, t] son desreferenciadas (eliminadas). Es decir, sólo mantiene en MP, las paginas referenciadas en ese intervalo. Para añadir la instrucción, ***limpiar\_U (dir\_base\_tabla),*** podemos proceder de la siguiente forma. Cuando accedemos a una página, el bit U se activa y se pone a 1. Tenemos que desactivar el bit de página cuando pasa la ventana de tiempo, y en cada referencia eliminar los conjuntos que estén en memoria con el bit activo. Es decir, cada x referencias la función limpiar\_U, borrará las páginas cuyo bit U esté a 0, es decir que no hayan sido referenciadas en ese intervalo.

***22. Un Sistema Operativo con memoria virtual paginada tiene el mecanismo fijar\_página(np) cuyo efecto es proteger contra la sustitución al marco de página en que se ubica la página virtual np. El mecanismo des\_fijar (np) suprime esta protección.*** Podemos interpretar ambos mecanismos como un lock/unlock.

***a) ¿Qué estructura/s de datos son necesarias para la realización de estos mecanismos?***

Para poder realizar estos mecanismos, necesitaríamos la tabla de marcos de página ya que es donde se encuentra la información sobre la protección (bit de protección).

***b) ¿En qué caso puede ser de utilidad estas primitivas?***

Pueden ser útiles cuando varios usuarios están compartiendo una página, o se puede usar como un buffer de E/S que debe estar bloqueado durante la transferencia (no se referencia en mucho tiempo y se podría eliminar). Podemos solucionarlo con fijar\_pagina, ya que la protegemos frente a la sustitución, evitando que sea eliminada.

***c) ¿Qué riesgos presentan y qué restricciones deben aportarse a su empleo?***

Se puede dar la hiperpaginación, pues si se bloquean demasiadas páginas se estará constantemente realizando trabajo de swapping. Para solucionar este problema se puede poner un límite de páginas bloqueadas, restriingir el número de marcos que se pueden fijar.

***24. Se tiene un sistema de memoria virtual con paginación a dos niveles que permite agrupar las páginas en “directorios de páginas”. Cada tabla de páginas puede contener hasta 1024 páginas. Los espacios de direcciones lógicas de este sistema son de 4Gbytes y el tamaño de página es de 4Kbytes. El espacio de direcciones físicas puede tener hasta 1Gb. Describa la estructura de las direcciones lógicas y de las direcciones físicas de este sistema de memoria virtual.***

***27. Respecto a la gestión de memoria que se hace en Linux: suponga que un proceso realiza una llamada al sistema fork creando un proceso hijo. Represente gráficamente como quedan las estructuras de datos relacionadas con ambos procesos.***

***28. ¿Qué información comparten un proceso y su hijo en un sistema Linux después de ejecutar el siguiente código? Justifique su respuesta e indique qué hace este trozo de código.***

***if ( fork() != 0 )***

***wait (&status)***

***else***

***exec (B);***

***// usamos una llamada al sistema exec genérica con un único argumento, el nombre //de un archivo ejecutable***

En linux, un proceso hijo es un clon del padre (salvo por el PID y la memoria). Ambos procesos disponen de las mismas variables (aunque estas son independientes). Además, el hijo hereda los descriptores de fichero del padre.

En ese trozo de código lo que ocurre es lo siguiente: Un padre crea un proceso hijo y se pone a esperar. El hijo realiza la llamada al sistema exec y ejecuta el proceso B. Cuando el proceso hijo termine desbloqueará al proceso padre.