Trabajo Práctico N°7

**Administración de la memoria (2da parte)**

1. Un sistema operativo tiene un esquema de memoria virtual basado en paginación simple en el que el mapeo se realiza mediante una Tabla de Páginas de longitud F (*F: número de páginas de un proceso que residen en memoria física*).
   1. Si la máxima memoria virtual que se puede asignar es 4 GB y el tamaño de la página es 2048 Bytes

¿cuál es la cantidad máxima de páginas que puede tener un proceso en memoria virtual?

(22 \* 230 bytes)/(2\*210 bytes) = 2 Mpaginas

* 1. Si la cantidad máxima de memoria física que se puede asignar es 512 MB ¿cuál es el tamaño de la Tabla de Páginas?

(29 \* 220 bytes)/ 211 = 218 paginas= 256 Kpaginas

* 1. ¿Cuántos bits se necesitan para numerar las páginas en memoria física?

Se necesitan 18 bits.

* 1. Calcule la cantidad de memoria que ocupa la Tabla de Páginas.

21 bits se representan en 3 bytes.

El tamaño de la tabla = cantidad de páginas \* cantidad de bytes para representar el número de página virtual.

Tamaño de la tabla = 218 paginas \* 3 bytes = 768 Kbytes.

1. Los sistemas de paginado simple A, B y C tienen, respectivamente, los siguientes parámetros:

* Dirección virtual: 16, 32, 24 bits.
* Tamaños de página: 512, 512, 1024 palabras.

Calcule para cada uno de los sistemas:

* 1. Cantidad de páginas en memoria virtual.

Sabemos que la cantidad de bits para la dirección virtual se calcula como:

|dv| = |p| + |w|, donde |p| es la cantidad de bits que se utilizan para representar el número de página y |w| es la cantidad de bits que se utilizan para representar el número de palabra dentro de la página.

El tamaño de una página se calcula como: W = 2|w|.

Luego, |w| = log2W.

Conociendo |w| y |dv|, obtenemos |p| = |dv| - |w|.

Finalmente, la cantidad de páginas en memoria virtual se define como 2|p|.

Para A con |dv| = 16 bits y W = 512 palabras.

|w| = log2W = log2 512 = 9 bits

Ahora obtengamos |p|.

|p| = |dv| - |w| = 16 bits – 9 bits = 7 bits.

Finalmente, el número de páginas virtuales de A es 2|p| = 27 = 128 páginas.

Para B con |dv| = 32 bits y W = 512 palabras.

|w| = log2W = log2 512 = 9 bits

Ahora obtengamos |p|.

|p| = |dv| - |w| = 32 bits – 9 bits = 23 bits.

Finalmente, el número de páginas virtuales de A es 2|p| = 223 = 8 Mpáginas.

Para C con |dv| = 24 bits y W = 1024 palabras.

|w| = log2W = log2 1024 = 10 bits

Ahora obtengamos |p|.

|p| = |dv| - |w| = 24 bits – 10 bits = 14 bits.

Finalmente, el número de páginas virtuales de A es 2|p| = 214 = 16 Kpáginas.

* 1. Si el número de páginas que pueden residir en memoria principal es 64 páginas, ¿cuál es la cantidad de memoria (en Bytes) ocupada por la Tabla de Páginas en cada caso?.

Para A, |p| = 7 bits. En bytes, se necesita 1 byte para representar el número de página en memoria virtual.

Luego, el tamaño de la tabla de páginas es: n° de páginas en memoria principal \* |p| = 64\*1 Byte = 64 Bytes.

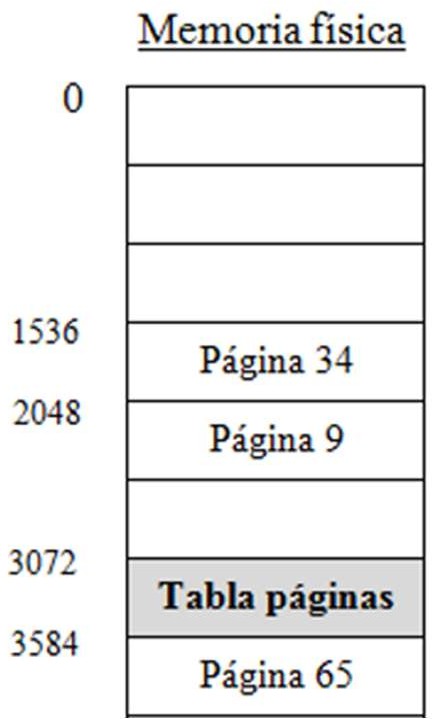
Para B, |p| = 23 bits. En bytes, se necesitan 3 bytes para representar el número de página en memoria virtual.

Luego, el tamaño de la tabla de páginas es: n° de páginas en memoria principal \* |p| = 64\*3 Byte = 192 Bytes.

Para C, |p| = 14 bits. En bytes, se necesitan 2 bytes para representar el número de página en memoria virtual.

Luego, el tamaño de la tabla de páginas es: n° de páginas en memoria principal \* |p| = 64\*2 Byte = 128 Bytes.

1. Un sistema de memoria virtual de paginado simple usa páginas de 512 palabras, un tamaño de memoria virtual de 512 páginas (0 a 511) y una memoria física con capacidad de 8 páginas (0 a 7). En un momento dado, el contenido de la memoria física es el que muestra la figura inferior.



El tamaño de una página se calcula como: W = 2|w|, donde |w| es la cantidad de bits que se utilizan para representar el número de palabra dentro de la página.

Luego, |w| = log2W. |w| = log2512 = 9.

TMV (Tamaño de memoria virtual) = 512 páginas = 512 \* 512 = 218.

Sabemos que el TMV = 2|dv|. Luego, 218 = 2|dv|. Entonces, |dv| = 18 bits.

|dv| = |p| + |w|, donde |p| es la cantidad de bits que se utilizan para representar el número de página y |w| es la cantidad de bits que se utilizan para representar el número de palabra dentro de la página.

Conociendo |w| y |dv|, obtenemos |p| = |dv| - |w| = 18 bits – 9 bits = 9 bits.

En bytes, se necesitan 2 bytes para representar el número de página virtual.

* 1. Muestre el contenido de la tabla de páginas.

|  |  |
| --- | --- |
| 0 |  |
| 1 |  |
| 2 |  |
| 3 | 34 |
| 4 | 9 |
| 5 |  |
| 6 |  |
| 7 | 65 |

* 1. Muestre el contenido de la tabla de páginas luego de que se carga la página 49 en la ubicación 0 y la página 34 es reemplazada por la página 12.

|  |  |
| --- | --- |
| 0 | 49 |
| 1 |  |
| 2 |  |
| 3 | 12 |
| 4 | 9 |
| 5 |  |
| 6 |  |
| 7 | 65 |

* 1. ¿Qué ocurre cuando se referencia la dirección virtual 33000?

Para saber en que página se encuentra la dirección virtual 33000, debemos dividir (división entera) la dirección virtual en el tamaño de página. N°página = 33000/512 = 64.

Esa dirección se encuentra en la página 64 de memoria virtual. Como esa página no se encuentra en memoria física, se produce un fallo de página.

1. Considere tres procesos X, Y y Z con tamaños de 700, 200, y 500 palabras respectivamente. Con paginado simple se desperdicia algo de espacio debido a que es poco probable que el tamaño de los procesos sea un múltiplo exacto del tamaño de la página. Para cada uno de los procesos determine la cantidad de memoria desperdiciada debido a la fragmentación interna para los siguientes tamaños de página: a) 200 palabras, b) 500 palabras, c) 600 palabras .
2. Tamaño de página es 200 palabras.

El proceso X de tamaño de 700 palabras, se dividirá en 4 páginas de 200 palabras. En la última página no se ocupará el total de la página ya que solo restan 100 palabras. Luego, se desperdiciará un 50% de la página.

El proceso Y de tamaño de 200 palabras no se dividirá. Se encontrará en su totalidad en una página. En este caso no hay memoria desperdiciada por fragmentación interna.

El proceso Z de tamaño de 500 palabras se dividirá en 3 páginas de 200 palabras. En la última página no se ocupará el total de la página ya que solo restan 100 palabras. Luego, se desperdiciará un 50% de la página.

1. Tamaño de página es 500 palabras.

El proceso X de tamaño de 700 palabras, se dividirá en 2 páginas de 500 palabras. En la última página no se ocupará el total de la página ya que solo restan 200 palabras. Luego, se desperdiciará un 60% de la página.

El proceso Y de tamaño de 200 palabras no se dividirá. Se encontrará en su totalidad en una página. Como el proceso tiene de tamaño 200 palabras y la página 500 palabras se desperdiciará un 60% de la página.

El proceso Z de tamaño de 500 palabras se encontrará en su totalidad en una página. En este caso no hay memoria desperdiciada por fragmentación interna ya que el proceso tiene el tamaño de la página.

1. Tamaño de página es 600 palabras.

El proceso X de tamaño de 700 palabras, se dividirá en 2 páginas de 600 palabras. En la última página no se ocupará el total de la página ya que solo restan 100 palabras. Luego, se desperdiciarán un espacio equivalente a 500 palabras lo que representa casi un 83% de la página.

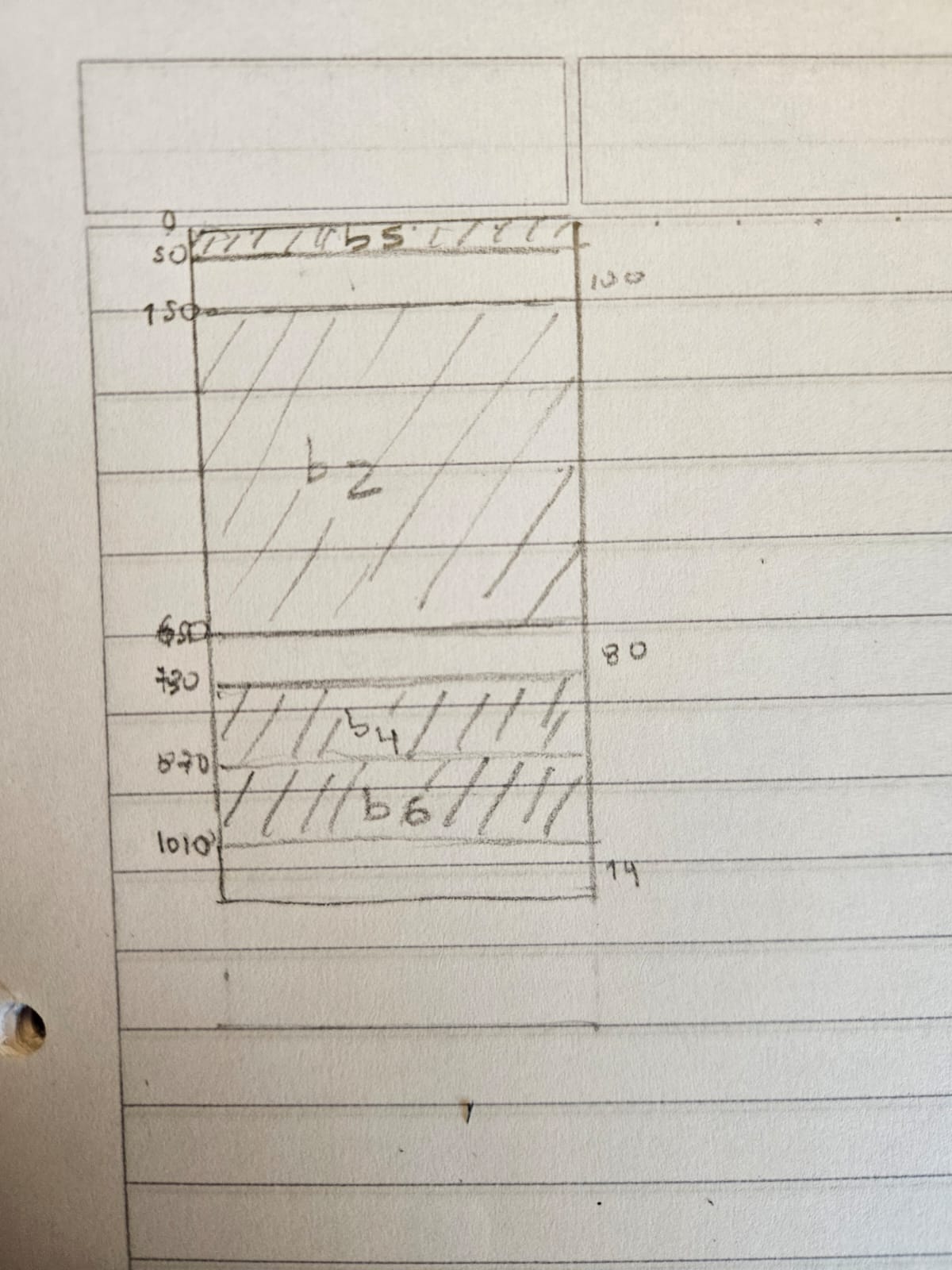
El proceso Y de tamaño de 200 palabras se encontrará en su totalidad en una página. Como el proceso tiene de tamaño 200 palabras y la página 600 palabras se desperdiciará dos tercios de la página.

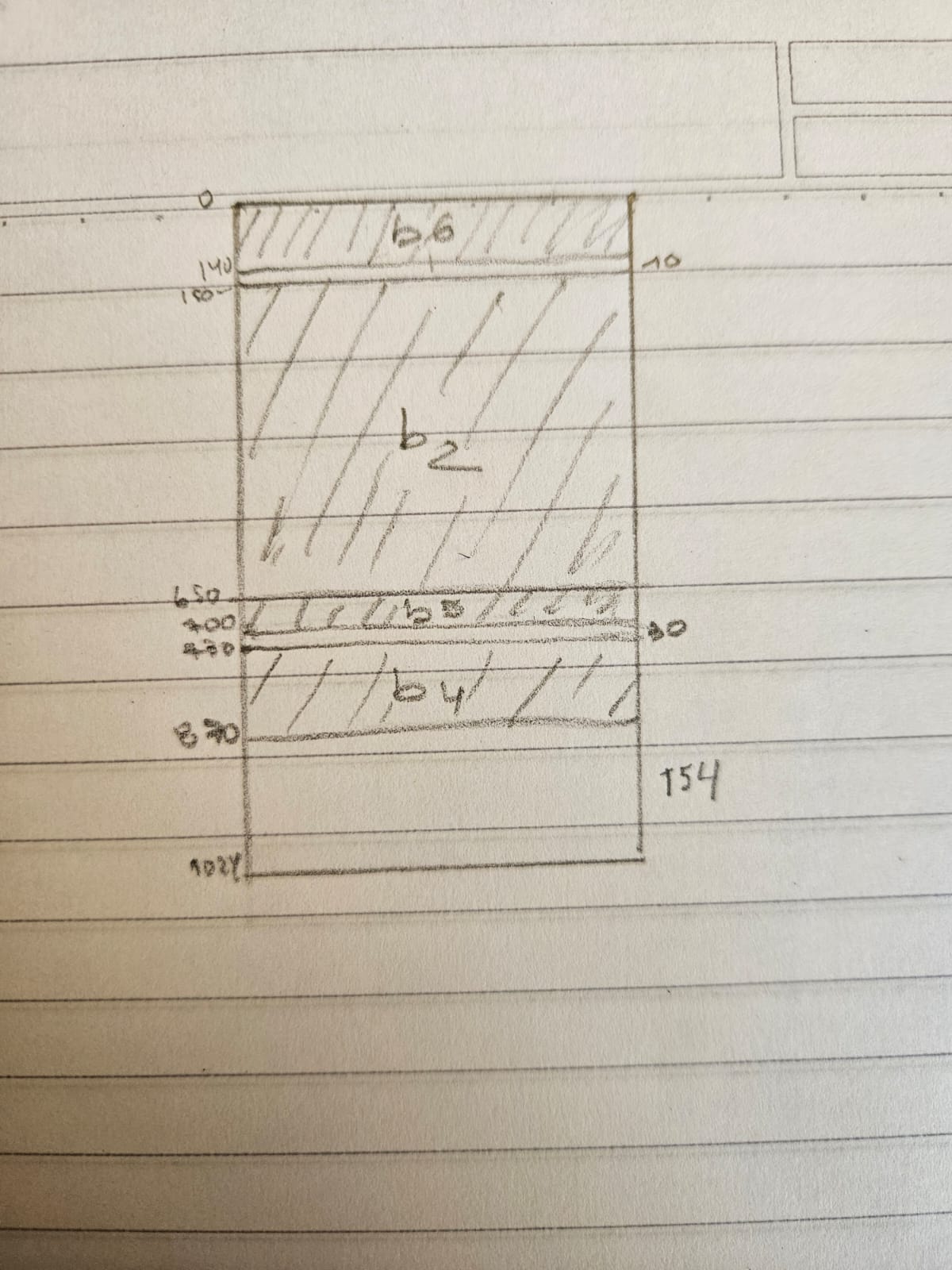
El proceso Z de tamaño de 500 palabras se encontrará en su totalidad en una página. Como el proceso ocupa 100 palabras menos que el tamaño de la página, se desperdiciará aproximadamente un 17% del tamaño de ésta.

1. En un esquema de asignación de memoria de un sistema no paginado ocurre la siguiente secuencia de solicitudes (“request”) al módulo administrador de memoria:

* Asignar bloque b1 de tamaño 150
* Asignar bloque b2 de tamaño 500
* Asignar bloque b3 de tamaño 80
* Asignar bloque b4 de tamaño 140
* Liberar bloque b1
* Liberar bloque b3
* Asignar bloque b5 de tamaño 50
* Asignar bloque b6 de tamaño 140

Suponiendo un tamaño total de memoria es de 1024 palabras, muestre las direcciones de comienzo y los tamaños de los espacios libres para los siguientes algoritmos de asignación de memoria, luego que todas las solicitudes hayan sido procesadas:

* 1. “First Fit”
  2. “Best Fit”



1. La memoria física de un sistema acepta 4 páginas, como máximo. Suponiendo que inicialmente no tiene cargada ninguna página, encontrar el número de fallos de página para la secuencia de referencias *Z = b a d e g b a b d e g a d e*, para los siguientes algoritmos de reemplazo de páginas
   1. Óptimo

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Página/tiempo | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Z |  | b | a | d | e | G | b | a | b | d | e | g | a | d | e |
| 0 |  | b | b | b | b | B | b | b | b | b | e | e | e | e | e |
| 1 |  |  | a | a | a | A | a | a | a | a | a | a | a | a | a |
| 2 |  |  |  | d | d | D | d | d | d | d | d | d | d | d | d |
| 3 |  |  |  |  | e | G | g | g | g | g | g | g | g | g | g |
| Fallo |  | \* | \* | \* | \* | \* |  |  |  |  | \* |  |  |  |  |
| P |  | b | a | d | e | G |  |  |  |  | e |  |  |  |  |
| Q |  |  |  |  |  | E |  |  |  |  | b |  |  |  |  |

* 1. FIFO

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Página/tiempo | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Z |  | b | a | d | e | g | b | a | b | d | e | g | a | d | e |
| 0 |  | - b | -b | -b | -b | g | g | g | g | -g | e | e | e | e | e |
| 1 |  |  | a | a | a | -a | b | b | b | b | -b | g | g | g | g |
| 2 |  |  |  | d | d | d | -d | a | a | a | a | -a | -a | -a | -a |
| 3 |  |  |  |  | e | e | e | -e | -e | d | d | d | d | d | d |
| Fallo |  | \* | \* | \* | \* | \* | \* | \* |  | \* | \* | \* |  |  |  |
| P |  | b | a | d | e | g | b | a |  | d | e | g |  |  |  |
| Q |  |  |  |  |  | b | a | d |  | e | g | b |  |  |  |

* 1. LRU

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Página/tiempo | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| **Z** |  | **b** | **a** | **d** | **e** | **g** | **b** | **a** | **b** | **d** | **e** | **g** | **a** | **d** | **e** |
| 0 |  | b | b | b | b | g | g | g | g | g | e | e | e | e | e |
| 1 |  |  | a | a | a | a | b | b | b | b | b | b | g | g | g |
| 2 |  |  |  | d | d | d | d | a | a | a | a | g | a | a | a |
| 3 |  |  |  |  | e | e | e | e | e | d | d | d | d | d | d |
| Fallo |  | \* | \* | \* | \* | \* | \* | \* |  | \* | \* | \* | \* |  |  |
| P |  | b | a | d | e | g | b | a |  | d | e | g | a |  |  |
| Q |  |  |  |  |  | b | a | d |  | e | g | a | b |  |  |
| STACK |  | b | a  b | d  a  b | e  d  a  b | g  e  d  a | b  g  e  d | a  b  g  e | b  a  g  e | d  b  a  g | e  d  b  a | g  e  d  b | a  g  e  d | d  a  g  e | e  d  a  g |

* 1. RELOJ

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| página/tiempo | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| z |  | b | a | d | e | g | b | a | b | d | e | g | a | d | e |
| 0 |  | -b/1 | -b/1 | -b/1 | -b/1 | g/1 | g/1 | g/1 | g/1 | -g/1 | e/1 | e/1 | e/1 | e/1 | e/1 |
| 1 |  |  | a/1 | a/1 | a/1 | -a/0 | b/1 | b/1 | b/1 | b/1 | -b/0 | g/1 | -g/1 | -g/1 | -g/1 |
| 2 |  |  |  | d/1 | d/1 | d/0 | -d/0 | a/1 | a/1 | a/1 | a/0 | -a/0 | a/1 | a/1 | a/1 |
| 3 |  |  |  |  | e/1 | e/0 | e/0 | -e/0 | -e/0 | d/1 | d/0 | d/0 | d/0 | d/1 | d/1 |
| fallo |  | \* | \* | \* | \* | \* | \* | \* |  | \* | \* | \* |  |  |  |
| p |  | b | a | d | e | g | b | a |  | d | e | g |  |  |  |
| q |  |  |  |  |  | b | a | d |  | e | g | b |  |  |  |

1. Un sistema que tiene un esquema de memoria virtual de segmentación con paginado acepta, como máximo, 512 páginas en memoria física por proceso. Una dirección virtual *(s, p, w)* se especifica con los siguientes valores:

|s| = 8

|p| = 12

|w| = 10 Encuentre:

* 1. Cantidad máxima de segmentos que puede asignarse a un proceso.

La cantidad máxima de segmentos que puede asignarse a un proceso es igual a 2|s| = 28 = 256 segmentos.

* 1. Cantidad máxima de páginas que puede tener un segmento.

La cantidad máxima de páginas que puede asignarse a un segmento es igual a 2|p| = 212 = 4 Kpaginas.

* 1. Tamaño de la Tabla de segmentos.

La tabla de segmentos posee una entrada por cada segmento de un proceso. Luego, el tamaño máximo de la tabla de segmentos es 256 entradas.

* 1. Tamaño de la Tabla de páginas.

La tabla de páginas tiene una entrada por cada página de un segmento. Luego, el número máximo de entradas es 4 Kpáginas.

* 1. Cantidad de memoria que ocupa una Tabla de segmentos asumiendo que el contenido es número de Tabla de páginas.

La tabla de segmentos tiene 256 entradas. Como cada entrada contiene el número de tabla de página, veamos cuantas tablas de pagina están asociadas al proceso. Hay una tabla de páginas por segmento. Entonces, hay un total de 256 tablas de página. Luego, para representar el número de tabla de página es 8 bits = 1 byte. Entonces el tamaño de tabla de segmentos es 256 entradas \* 1 byte/entrada = 256 Bytes.

* 1. Cantidad de memoria que ocupa una Tabla de páginas asumiendo que el contenido es el número de página.

Sabemos que las tablas de páginas pueden tener 4 K entradas y cada una de ellas almacenará el número de página física, que se representa con 9 bits, es decir, 2 bytes. Luego, la cantidad de memoria que ocupa una tabla de páginas es 8 Kbytes.

* 1. Cantidad de memoria que ocupan todas las Tablas de Páginas.

La cantidad de memoria que ocupan todas las tablas de páginas de un proceso es igual a la cantidad de segmentos por el tamaño de la tabla de páginas ya que cada segmento de un proceso posee una tabla de páginas. Luego, ocupan 256\* 8 Kbytes = 2048 Kbytes = 2 Mbytes.

Suponiendo que en un momento dado en el sistema están activos 700 procesos, encuentre:

* 1. Cantidad de memoria que ocupan todas las Tablas de segmentos.

Como vimos anteriormente, el tamaño de la tabla de segmentos es igual 256 bytes.

Si el número de procesos activos es 700, cada proceso tendrá asociada una tabla de segmentos de dicho tamaño. Luego, la cantidad de espacio que ocupan las tablas de segmentos es igual a: 256 bytes \* 700 = 179200 bytes.

* 1. Cantidad de memoria que ocupan todas las Tablas de páginas.

Vimos que la cantidad de memoria que ocupan todas las tablas de páginas es igual al número de procesos por la cantidad de memoria que ocupan todas las tablas de páginas de los procesos:

700 \* 2 Mbytes = 1.400 Mbytes.

A efectos de reducir la cantidad de memoria ocupada por las Tablas de páginas:

* 1. ¿Qué modificación propondría sin disminuir la cantidad de memoria física asignada a cada proceso?
  2. Recalcule el punto **i.** con la modificación propuesta e indique la cantidad de memoria ahorrada con el cambio.

1. Considere un sistema de paginado con las siguientes medidas de utilización:

* Utilización de CPU: 30% para I/O y 70% para cómputos.
* Disco de paginado: 97,7 %.
* Otros dispositivos de I/O: 5%.

¿Con cuál o con cuáles de las siguientes acciones se puede aumentar la utilización de la CPU en cómputos? Señale la acción y explique por qué toma esa decisión:

* 1. Instalar una CPU más rápida.
  2. Instalar un disco de paginado más grande.
  3. Aumentar el grado de multiprogramación.
  4. Instalar dispositivos de I/O adicionales.
  5. Aumentar el tamaño de la memoria primaria.

Aumentar el tamaño de la memoria primaria ayudaría a mejorar la utilización de la CPU ya que el tiempo que ocupa en operaciones de entrada-salida se debe a la cantidad de fallos de página que ocurren y que requieren de operaciones de este tipo con el disco. Podemos ver que la cantidad de fallos de páginas es muy grande ya que la utilización del disco de paginado es alta. Incrementar el tamaño de memoria principal permitiría llevar una mayor cantidad de páginas de un proceso a RAM, reduciendo la cantidad de fallos de página.

El resto de acciones decido descartarlas ya que la opción de instalar una CPU más rápida no mejoraría la situación porque el problema no está en la velocidad de procesamiento, sino en el tiempo que la CPU pasa inactiva esperando por operaciones de entrada/salida causadas por los fallos de página. Tampoco resulta útil instalar un disco de paginado más grande, ya que el problema no es la falta de espacio en el disco sino la cantidad de accesos que se realizan a él. Aumentar el grado de multiprogramación empeoraría la situación porque generaría más competencia por la memoria, lo cual incrementaría aún más los fallos de página. Finalmente, instalar dispositivos de I/O adicionales no tendría impacto, ya que el problema principal está en el uso intensivo del disco de paginado y no en los demás dispositivos de entrada/salida, que apenas se están utilizando.

Belu =)