

Carrera: **Analista Programador**

## Arquitectura de Sistemas Operativos

### Módulo II

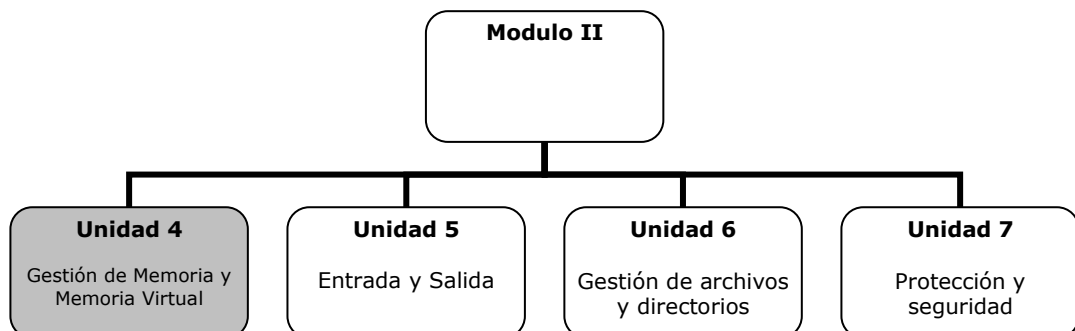
Comprender cómo el Sistema Operativo realiza la administración de recursos del Sistema de Computación.

#### Unidad 4

Gestión de memoria y memoria virtual.

Profesor Titular: Martín Mario.

Autor de contenidos: Lic. Juan Carlos Romero.





---

## Presentación

---

Para iniciar esta unidad lo invitamos a imaginarse en un gran recipiente que almacena datos y tareas, repasemos el sentido de tarea en este contexto, la tarea es un proceso en actividad, no es un programa. El autor Deitel en su libro Sistemas Operativos, expresó lo siguiente "Un proceso es un perro vivo y un programa es un perro muerto".

Sigamos con lo nuestro, estamos hablando de un espacio físico que almacena datos y tareas, pero pensemos en ese espacio como un espacio comunitario, si lo queremos llamar de otra forma, digamos que se trata de un espacio centralizado donde se almacenan temporalmente los datos y tareas de todos. Detengámonos un momento en la palabra todos, ¿quiénes son todos? o ¿a qué se refiere todos?

Comencemos desde el principio.

Cuando iniciamos el sistema de computación, se carga el SO automáticamente, y cuando terminó de cargarse, el SO indica que ya se puede interactuar con él, entonces los usuarios, que tienen distintas jerarquías y privilegios desde el administrador del SO hasta los usuarios comunes, comienzan a utilizar el sistema en su conjunto (Sistema de Computación administrado por el SO).

¿Qué hacen ahora los usuarios?

Necesitan obtener el resultado de tareas que se deben realizar en esta sesión de cómputo, para ello deben mandar a ejecutar los procesos que deberán obtener esos resultados y muchos de esos procesos para obtener los resultados esperados necesitan datos.

Presentemos un ejemplo simple:

Un usuario necesita obtener un listado de saldos de clientes, otro usuario necesita obtener un listado de cheques rechazados, otro usuario esta usando un procesador de texto y una planilla de cálculo y el administrador del sistema, esta ejecutando procesos de resguardo de datos, aquí vemos que tenemos usuarios ejecutando procesos y la mayoría de los procesos necesitan o usan datos, por ejemplo el procesador de texto y la planilla de cálculo. Esto ya nos da una idea de que significa la palabra todos.

De lo afirmado surge una problemática que el SO tendrá que resolver. Algunas preguntas que nos podemos hacer para identificar dicha problemática son las siguientes:

¿Qué ubicación toman en memoria los procesos y los datos?, ¿qué estrategia usa el SO para permitirle al proceso encontrar los datos que necesita?, ¿qué estrategias usa el SO para asignar espacio o memoria cuando un usuario lanza



a ejecutar un proceso?, ¿qué sucede cuando el tamaño de los procesos supera el espacio libre de memoria física?

En esta unidad estudiaremos las distintas estrategias que utiliza el SO para administrar el recurso memoria, obteniendo las respuestas a estas preguntas.

Por todo lo expresado hasta aquí es que esperamos que usted, a través del estudio de esta unidad, adquiera capacidad para:

- Analizar e identificar, dentro del ámbito del SO, la diferencia entre direccionamiento lógico y físico.
- Comprender la importancia de la administración o gestión del recurso memoria física temporaria, estudiando las distintas estrategias de administración.
- Identificar la o las estrategias más relevantes en los Sistemas Operativos actuales.
- Realizar ejercicios aplicados a cada una de las estrategias, para internalizar los conceptos teóricos.

A continuación, le presentamos un detalle de los contenidos y actividades que integran esta unidad. Usted deberá ir avanzando en el estudio y profundización de los diferentes temas, realizando las lecturas requeridas y elaborando las actividades propuestas, algunas de desarrollo individual y otras para resolver en colaboración con otros estudiantes y con su profesor tutor.

#### Contenidos y Actividades

##### 1. Espacio de direcciones, lógico y físico.



##### Lectura requerida

- **Silberschatz A. y Galvin P.**; Capítulo 8 Gestión de memoria. En su: **Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. Página 239



##### Lectura Sugerida

- Stallings W.; Capítulo 7 Gestión de memoria **En su: Sistemas Operativos –Aspectos Internos y principios de diseño–**; 5ta Edición; España Prentice Hall, 2007. Página 307.
- Silberschatz A. Galvin P. Gagne G.; Capítulo 8 Memoria principal; **En su: Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Garw Hill 2007.



- Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 4 Gestión de memoria **En su: Sistemas Operativos –Una visión aplicada-**; España Mc Graw Hill, 2001. Página 163.

## 2. Intercambio.



### Lectura requerida

- Silberschatz A. y Galvin P.; Capítulo 8 Gestión de memoria. **En su: Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. Página 239



### Lectura Sugerida

- Stallings W.; Capítulo 7 Gestión de memoria. **En su: Sistemas Operativos –Aspectos Internos y principios de diseño-**; 5ta Edición; España Prentice Hall, 2007. Página 307
- Silberschatz A. Galvin P. Gagne G.; Capítulo 8 Memoria principal; **En su: Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Garw Hill 2007.
- Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 4 Gestión de memoria. **En su: Sistemas Operativos –Una visión aplicada-**; España Mc Graw Hill, 2001. Página 163

## 3. Asignación contigua.



### Lectura requerida

- Silberschatz A. y Galvin P.; Capítulo 8 Gestión de memoria. **En su: Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. Página 239.



### Lectura Sugerida

- Stallings W.; Capítulo 7 Gestión de memoria **En su: Sistemas Operativos –Aspectos Internos y principios de diseño-**; 5ta Edición; España Prentice Hall, 2007. Página 307.
- Silberschatz A. Galvin P. Gagne G.; Capítulo 8 Memoria principal; **En su: Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Garw Hill 2007.
- Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 4 Gestión de memoria **En su: Sistemas Operativos –Una visión aplicada-**; España Mc Graw Hill, 2001. Página 163.



#### 4. Paginación.



##### Lectura requerida

- Silberschatz A. y Galvin P.; Capitulo 8 Gestión de memoria En su: **Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. Página 239.



##### Lectura Sugerida

- Stallings W.; Capítulo 7 Gestión de memoria **En su: Sistemas Operativos –Aspectos Internos y principios de diseño–**; 5ta Edición; España Prentice Hall, 2007. Página 307.
- Silberschatz A. Galvin P. Gagne G.; Capítulo 8 Memoria principal; **En su: Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Garw Hill 2007.
- Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 4 Gestión de memoria **En su: Sistemas Operativos –Una visión aplicada–**; España Mc Graw Hill, 2001. Página 163.



##### Trabajo Práctico Sugerido

- Trabajo Práctico Nº 10: **Memoria Paginada**

#### 5. Segmentación.



##### Lectura requerida

- Silberschatz A. y Galvin P.; Capitulo 8 Gestión de memoria **En su: Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. Página 239.



##### Lectura Sugerida

- Stallings W.; Capítulo 7 Gestión de memoria **En su: Sistemas Operativos –Aspectos Internos y principios de diseño–**; 5ta Edición; España Prentice Hall, 2007. Página 307.
- Silberschatz A. Galvin P. Gagne G.; Capítulo 8 Memoria principal; **En su: Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Garw Hill 2007.
- Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 4 Gestión de memoria **En su: Sistemas Operativos –Una visión aplicada–**; España Mc Graw Hill, 2001. Página 163.



## 6. Paginación por demanda.



### Lectura requerida

- Silberschatz A. y Galvin P.; Capitulo 8 Gestión de memoria **En su: Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. Página 239.



### Lectura Sugerida

- Stallings W.; Capítulo 7 Gestión de memoria **En su: Sistemas Operativos –Aspectos Internos y principios de diseño–**; 5ta Edición; España Prentice Hall, 2007. Página 307.
- Silberschatz A. Galvin P. Gagne G.; Capítulo 8 Memoria principal; **En su: Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Garw Hill 2007.
- Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 4 Gestión de memoria **En su: Sistemas Operativos –Una visión aplicada–**; España Mc Graw Hill, 2001. Página 163.

## 7. Reemplazo de páginas.



### Lectura requerida

- Silberschatz A. y Galvin P.; Capitulo 8 Gestión de memoria **En su: Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. Página 239.



### Lectura Sugerida

- Stallings W.; Capítulo 7 Gestión de memoria. **En su: Sistemas Operativos –Aspectos Internos y principios de diseño–**; 5ta Edición; España Prentice Hall, 2007. Página 307
- Silberschatz A. Galvin P. Gagne G.; Capítulo 8 Memoria principal; **En su: Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Garw Hill 2007.
- Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 4 Gestión de memoria **En su: Sistemas Operativos –Una visión aplicada–**; España Mc Graw Hill, 2001. Página 163.



## 8. Asignación de marcos



### Lectura requerida

- Silberschatz A. y Galvin P.; Capítulo 8 Gestión de memoria **En su: Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. Página 239.



### Lectura Sugerida

- Stallings W.; Capítulo 7 Gestión de memoria **En su: Sistemas Operativos –Aspectos Internos y principios de diseño–**; 5ta Edición; España Prentice Hall, 2007. Página 307.
- Silberschatz A. Galvin P. Gagne G.; Capítulo 8 Memoria principal; **En su: Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Garw Hill 2007.
- Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 4 Gestión de memoria **En su: Sistemas Operativos –Una visión aplicada–**; España Mc Graw Hill, 2001. Página 163.

## 9. Hiperpaginación.



### Lectura requerida

- Silberschatz A. y Galvin P.; Capítulo 8 Gestión de memoria. **En su: Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. Página 239.



### Lectura Sugerida

- Stallings W.; Capítulo 7 Gestión de memoria. **En su: Sistemas Operativos –Aspectos Internos y principios de diseño–**; 5ta Edición; España Prentice Hall, 2007. Página 307
- Silberschatz A. Galvin P. Gagne G.; Capítulo 8 Memoria principal; **En su: Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Garw Hill 2007.
- Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 4 Gestión de memoria. **En su: Sistemas Operativos –Una visión aplicada–**; España Mc Graw Hill, 2001. Página 163.

## Cierre de la unidad

## ANEXO

Para el estudio de estos contenidos usted deberá consultar la bibliografía que aquí se menciona:



## BIBLIOGRAFÍA OBLIGATORIA

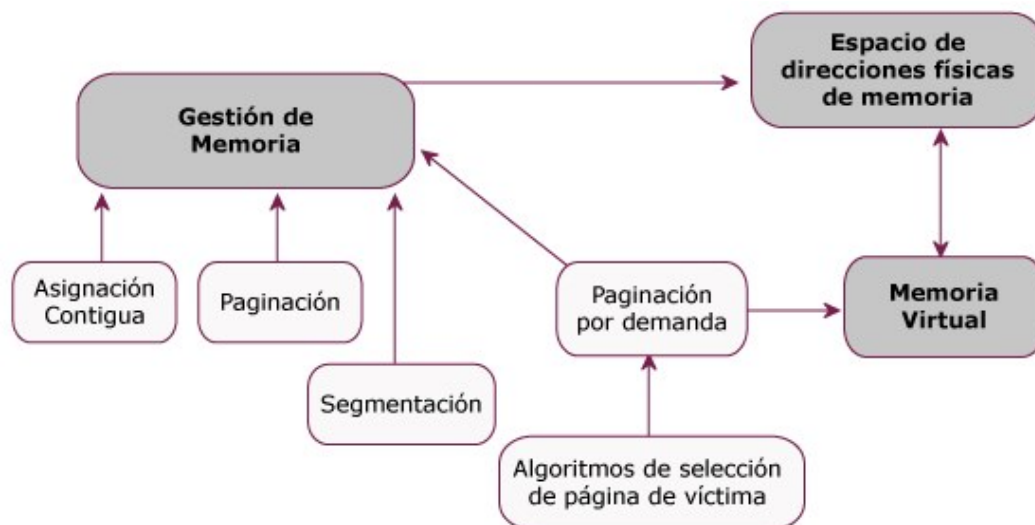
- Silberschatz A. y Galvin P.; **Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999.
- Stallings W.; **Sistemas Operativos –Aspectos Internos y principios de diseño-**; 5ta Edición; España Prentice Hall, 2007.
- Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; **Sistemas Operativos –Una visión aplicada-**; España Mc Graw Hill, 2001.

## BIBLIOGRAFIA AMPLIATORIA

- Tanenbaum A. Woodhull A.; **Sistemas Operativos –Diseño e implementación-**; 2da Edición; México Prentice Hall 1997.
- Silberschatz A. Galvin P. Gagne G.; **Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Garw Hill 2007.

## Organizador Gráfico

El siguiente esquema le permitirá visualizar la interrelación entre los conceptos que a continuación abordaremos. Le sugerimos que vuelva a este organizador una vez completado el estudio de la unidad, le ayudará a ordenar sus ideas.





Lo/a invitamos ahora a comenzar con el estudio de los contenidos que conforman esta unidad.

## 1. Espacio de direcciones, lógico y físico.

Para comenzar a entender la diferencia entre el espacio lógico y físico, presentaremos este tema desde el punto de vista de un programador.

Cuando programamos y necesitamos que nuestro programa utilice datos en memoria definimos una variable, seguramente a dicha variable en función del lenguaje de programación que estemos utilizando, tendrá un nombre determinado, pero ese nombre seguro que representa el tipo y semántica del dato y será fácil de recordar, al nombre que el programador le asigna a la variable se llama nombre simbólico y es utilizando únicamente por el programador.

El programador ya escribió su programa y ahora tiene que compilarlo y vincularlo para obtener una versión del mismo en lenguaje de máquina ejecutable, entonces pensemos en qué se transformó el nombre de nuestra variable. Nuestra variable se transformó en una dirección lógica de memoria, ahora la referencia a la variable no es un nombre sino un número que apunta a un espacio de memoria lógica que contiene el dato. Aún el programa no se lanzó a ejecutar por eso no tiene una ubicación real en memoria física, cuando el programa se manda a ejecutar y está almacenado en memoria física la dirección del dato es una dirección física de memoria.

Resumiendo cuando estamos en el programa fuente nos referimos de nombre simbólico de variable, cuando hablamos de programa ejecutable hablamos de dirección lógica de memoria y cuando hablamos de proceso almacenado en memoria real hacemos mención a la dirección física de memoria.



### Lectura requerida

Silberschatz A. y Galvin P.; Capítulo 8 Gestión de memoria **En su: Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. Página 239.



### Lectura Sugerida

Stallings W.; Capítulo 7 Gestión de memoria **En su: Sistemas Operativos –Aspectos Internos y principios de diseño-**; 5ta Edición; España Prentice Hall, 2007. Página 307.

Silberschatz A. Galvin P. Gagne G.; Capítulo 8 Memoria Principal; **En su: Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Garw Hill 2007.

Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 4 Gestión de memoria **En su: Sistemas Operativos –Una visión aplicada-**; España Mc Graw Hill, 2001. Página 163.

### Guía para la lectura

Durante o luego de la lectura de la bibliografía responda las siguientes preguntas:

- Diferencia entre dirección física y lógica
- ¿Qué entiende por espacio de direcciones lógicas?
- ¿Qué entiende por espacio de direcciones físicas?
- ¿Qué es la MMU y cuál es su función?
- ¿Interprete y describa el gráfico de la Figura 8.3 Reubicación dinámica empleando un registro de reubicación del libro de Silberschatz Galvin –Sistemas Operativos- 5ta Edición Página 246?

## 2. Intercambio

Recuerdo que cuando era niño miraba en mi televisor de válvulas en color blanco y negro un dibujito animado cuyo nombre no recuerdo, pero su temática tenía una relación con este tema.

Se trataba de un adolescente que trabajaba de agente secreto y siempre estaba involucrado en una misión, su padre era un científico que había inventado una máquina que tenía un almacenamiento de conocimientos y habilidades de los mejores representantes de todas las disciplinas deportivas, académicas y profesionales, entonces cuando su hijo tenía que ir a una misión



secreta le cargaba en su memoria la disciplina que necesitaba y a partir de ese momento tenía la capacidad necesaria para determinado trabajo, cuando cambiaba de misión o necesitaba otro conocimiento, nuevamente la máquina que había construido el padre realizaba un intercambio de conocimiento entre la memoria del hijo y el almacenamiento.

El tema que planteamos en este punto es la funcionalidad del SO que permite realizar el intercambio de procesos entre la memoria del sistema de computación y el dispositivo de almacenamiento magnético, hay que dejar claro que en el dibujo animado del agente secreto se intercambian conocimientos y habilidades y en el intercambio que realiza el SO es de procesos. En la bibliografía que presentamos a continuación encontraran la información necesaria para entender esta metodología, solo hay que leerla.



### Lectura requerida

Silberschatz A. y Galvin P.; Capitulo 8 Gestión de memoria. **En su: Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. Página 239



### Lectura Sugerida

Stallings W.; Capítulo 7 Gestión de memoria **En su: Sistemas Operativos –Aspectos Internos y principios de diseño-**; 5ta Edición; España Prentice Hall, 2007. Página 307.

Silberschatz A. Galvin P. Gagne G.; Capítulo 8 Memoria principal; **En su: Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Garw Hill 2007.

Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 4 Gestión de memoria **En su: Sistemas Operativos –Una visión aplicada-**; España Mc Graw Hill, 2001. Página 163.

### Guía para la lectura

Durante o luego de la lectura de la bibliografía responda las siguientes preguntas:

- ¿En qué situación el SO realizaría un intercambio?



- ¿Cómo se lleva adelante el intercambio en un algoritmo por prioridades?
- ¿Cuáles deben ser las características del almacenamiento auxiliar para que se pueda realizar un intercambio eficiente?
- ¿Cuáles son las restricciones que presenta el intercambio?
- ¿Cuál es la gran desventaja del intercambio?

### 3. Asignación contigua

Ahora tenemos que comenzar a pensar en la forma en que se puede asignar o distribuir el espacio de almacenamiento entre distintos procesos que son mandados a ejecutar por los usuarios y los procesos que ejecuta el SO.

Una de las formas es la asignación contigua de memoria, que se basa en asignar un espacio contiguo de direcciones de memoria al SO y otro espacio contiguo de direcciones de memoria a los procesos de usuario.

A su vez la **asignación contigua** tiene distintas formas. Una de ellas es la asignación con una sola partición. Podemos decir que es la más simple de todas porque existe una partición o una porción de direcciones contiguas de memoria para el SO y una porción de direcciones contiguas de memoria para un único proceso de usuario. Para administrar esta estrategia el SO no tiene que realizar mucho trabajo.

Otra forma es la **asignación con múltiples** particiones que a su vez se divide en fijas y reubicables, la asignación con múltiples particiones fijas asigna una partición al SO y distintas particiones a distintos procesos de usuarios donde cada proceso toma una única partición y las particiones no se pueden modificar, no se puede cambiar su tamaño o sus direcciones de memoria durante una sesión de trabajo. La asignación con múltiples particiones reubicables asigna una partición al SO y el resto de memoria queda como una gran partición que se reparte a medida que ingresan los procesos a memoria ajustando en tamaño exacto, la partición de memoria al tamaño del proceso y dejando el espacio libre restante si lo hubiera como una nueva partición libre, esta imagen de la memoria se va modificando a medida que los procesos van terminando y se van ingresando nuevos procesos a memoria, esta estrategia tiene una característica destacable que es la posibilidad de reubicar a los procesos en memoria para obtener espacios libres de memoria que puedan ser asignados a otros procesos.

La característica más importante de la asignación contigua de memoria es que los procesos para poder ser ejecutados necesitan estar almacenados en memoria en un espacio de direcciones contiguo, esto quiere decir que no se puede poner parte de un proceso en una porción de memoria y otra parte del mismo proceso en un espacio de direcciones de memoria que no sea contiguo



al anterior respetando la lógica del proceso y, por supuesto, todo el proceso tiene que estar almacenado en memoria.

Es interesante observar la evolución de las estrategias de asignación de memoria, cada estrategia tiene sus ventajas y sus desventajas, cuando se crea una nueva estrategia esta trata de mantener las ventajas de la anterior y resolver las desventajas, generando una nueva estrategia con mayores ventajas e introduciendo desventajas menores. Lo invitamos a realizar la siguiente actividad de lectura con el propósito de seguir profundizando este tema.



### Lectura requerida

Silberschatz A. y Galvin P.; Capítulo 8 Gestión de memoria **En su: Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. Página 239.



### Lectura Sugerida

Stallings W.; Capítulo 7 Gestión de memoria **En su: Sistemas Operativos –Aspectos Internos y principios de diseño–**; 5ta Edición; España Prentice Hall, 2007. Página 307.

Silberschatz A. Galvin P. Gagne G.; Capítulo 8 Memoria principal; **En su: Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Garw Hill 2007.

Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 4 Gestión de memoria. **En su: Sistemas Operativos –Una visión aplicada–**; España Mc Graw Hill, 2001. Página 163

### Guía para la Lectura

Durante o luego de la lectura de la bibliografía responda las siguientes preguntas:

- Describa la administración de memoria asignación contigua con una sola partición.
- Describa la administración de memoria asignación contigua con múltiples particiones fijas.



- Describa la administración de memoria asignación contigua con múltiples particiones reubicables.
- ¿Qué entiende por primer ajuste, mejor ajuste y peor ajuste?
- ¿Qué es la fragmentación externa e interna?
- Describa un algoritmo de compactación.

#### 4. Paginación

Una de las desventajas de la asignación contigua es poder encontrar en memoria un hueco libre de memoria contigua que permita almacenar un proceso que se quiere ejecutar, otra desventaja es la cantidad de memoria libre que no se puede usar porque es tan pequeña que no permite almacenar un proceso y además esos pequeños espacios de memoria no son contiguos, alguien podría preguntar entonces, ¿porqué no se reubican esos espacios pequeños que están libres y discontinuos en un espacio contiguo?, la respuesta es simple, la reubicación consume mucho tiempo de procesador y se tendrían que dejar de ejecutar los procesos que en ese momento se están ejecutando.

Para solucionar este problema, deberíamos poder asignar los espacios libres de memoria aunque no estuvieran contiguos sin hacer reubicación de memoria, para eso tendríamos que permitir que el proceso se pueda repartir entre los distintos huecos libres de memoria no contigua, cortando el proceso en pequeñas trozos, aunque parezca imposible así lo hicieron, por supuesto, de una forma muy técnica.

Para llevar esta idea a la práctica la memoria se dividió en pequeñas partes, todas del mismo tamaño a cada parte se le dio el nombre de frame, bloque o marco de memoria y al proceso también se lo dividió en pequeñas partes todas del mismo tamaño y a cada parte se le dio el nombre de página, cabe aclarar que los frame y las páginas también son del mismo tamaño, de esta manera se almacenaba cada página en un frame.

El proceso no necesariamente tenía que estar almacenado en forma contigua en la memoria para ser ejecutado, pero si tenía que estar todo el proceso en la memoria, el SO se encargaría de administrar esta estrategia.

Podemos hacernos una pregunta muy interesante que será contestada más adelante: ¿qué sucede si el proceso que se quiere ejecutar demanda más memoria física de lo que tenemos disponible en este momento y se tiene que ejecutar sí o sí?

Para responder a este interrogante lo invitamos a seguir investigando mediante la siguiente actividad de lectura.



### Lectura requerida

Silberschatz A. y Galvin P.; Capitulo 8 Gestión de memoria **En su: Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. Página 239.



### Lectura Sugerida

Stallings W.; Capítulo 7 Gestión de memoria. **En su: Sistemas Operativos –Aspectos Internos y principios de diseño–**; 5ta Edición; España Prentice Hall, 2007. Página 307

Silberschatz A. Galvin P. Gagne G.; Capítulo 8 Memoria principal; **En su: Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Garw Hill 2007.

Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 4 Gestión de memoria. **En su: Sistemas Operativos –Una visión aplicada–**; España Mc Graw Hill, 2001. Página 163

### Guía para la Lectura

Durante o luego de la lectura de la bibliografía responda las siguientes preguntas:

- Describa el soporte de hardware para la paginación.
- ¿Cómo se compone una dirección lógica y una dirección física?
- Realice un modelo de paginación para la memoria lógica y física.
- ¿Qué es la tabla de marcos?
- ¿Qué es la tabla de páginas?
- ¿Cómo se logra la protección de la memoria en un entorno paginado?
- ¿Qué es la paginación multinivel?
- ¿Qué es la tabla de páginas invertida?
- ¿Qué son las páginas compartidas?



A continuación le proponemos realizar el siguiente trabajo práctico.



### Trabajo Práctico Sugerido

#### **Trabajo Práctico N° 10:** Memoria Paginada

*Usted encontrará las consignas de este Trabajo Práctico en el Anexo que incluimos al final de este Orientador.*

*Comparta sus dudas e inquietudes con sus pares y con su tutor a través de los medios de comunicación disponibles en el Campus.*

## 5. Segmentación

Para comenzar la presentación de la administración de memoria usando segmentación no encuentro mejores palabras que las de Silberschatz y Galvin en su libro "Sistemas Operativos" 5ta Edición, "Un aspecto importante de la gestión de memoria que se hizo inevitable al surgir la paginación es la separación entre la visión que el usuario tiene de la memoria y la memoria física real. La visión del usuario no es igual a la memoria física real; sólo tiene una correspondencia con ella. La correspondencia permite diferenciar entre la memoria lógica y la física."

La segmentación tiene una semejanza con la paginación ya que el proceso no necesariamente tiene que estar almacenado en forma contigua en la memoria, la diferencia que más se destaca es que el proceso se divide en partes de tamaño variable llamadas segmentos donde cada segmento se identifica con un número y en la paginación los procesos se dividen en partes del mismo tamaño llamadas páginas.

A continuación, para facilitar la comprensión de este tema lo/la invitamos a realizar la siguiente actividad.



### Lectura requerida

Silberschatz A. y Galvin P.; Capítulo 8 Gestión de memoria **En su: Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. Página 239.





### Lectura Sugerida

Stallings W.; Capítulo 7 Gestión de memoria. **En su: Sistemas Operativos –Aspectos Internos y principios de diseño-**; 5ta Edición; España Prentice Hall, 2007. Página 307

Silberschatz A. Galvin P. Gagne G.; Capítulo 8 Memoria principal; **En su: Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Garw Hill 2007.

Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 4 Gestión de memoria **En su: Sistemas Operativos –Una visión aplicada-**; España Mc Graw Hill, 2001. Página 163.

### Guía para la lectura

Durante o luego de la lectura de la bibliografía responda las siguientes preguntas:

- Describa el hardware de segmentación.
- ¿Qué es la STBR y la STLR?
- ¿Cuáles son las ventajas de la segmentación con respecto a la protección y al compartimiento?
- Describa el problema de la fragmentación en la segmentación.
- ¿Qué es la segmentación con paginación?

## 6. Paginación por demanda

La paginación por demanda ofrece una gran ventaja, permite la ejecución de procesos sin que los mismos se encuentren almacenados en su totalidad en la memoria principal, mantiene de la paginación simple los conceptos de página y de marco o frame y la relación que existe entre una página y un frame, ambas son del mismo tamaño almacenan la misma cantidad de bytes y siempre la página cuando se lleva a memoria principal se almacena en un frame libre.

Para ofrecer esta nueva posibilidad es de suma importancia el uso de memoria virtual, es donde el SO mantiene almacenado los procesos paginados y es

desde allí donde se lleva a memoria principal la página que contiene la instrucción o el dato solicitado cuando se produce un direccionamiento a memoria.

Recordemos que en la administración de memoria de paginación simple se intercambia todo el proceso entre el disco y la memoria principal, y en la administración de memoria de paginación por demanda se intercambia la página.

Voy a hacer un pequeño comentario con respecto a la palabra intercambio, cuando se dice intercambio, inmediatamente pensamos que algo va y algo que viene, en el caso de la administración paginada simple podríamos pensar que un proceso sale del disco y llega a memoria principal y otro sale de la memoria principal y llega al disco, en el caso de la administración paginada por demanda podríamos pensar que una página sale del disco y llega a memoria principal y otra página sale de la memoria principal y llega al disco, esto no siempre es cierto, en el siguiente ítem "Reemplazo de páginas", volveremos a esta problemática.

Si la administración de memoria paginada por demanda permite ejecutar procesos sin que ellos estén almacenados completamente en memoria, entonces tenemos más memoria principal disponible, entonces podemos llevar a memoria principal partes (páginas) de otros procesos para que se puedan ejecutar, entonces aumenta el nivel de multiprogramación (cantidad de procesos en memoria que compiten por el uso del procesador), y además se podrían ejecutar procesos que en tamaño superan la capacidad de memoria física existente.

Vamos a terminar esta breve introducción planteando dos problemáticas:

¿Qué sucede si ocurre lo siguiente?

### **Problemática 1**

En un momento determinado luego que se hallan cargado en memoria muchas páginas de distintos procesos, ya no quedan frames libres en memoria que puedan almacenar una página, y el administrador del SO manda a ejecutar un proceso de muy alta prioridad, este proceso se página en memoria virtual pero no se puede ejecutar porque no hay frames libres.

### **Problemática 2**

La memoria principal esta llena de páginas de distintos procesos, pero faltan cargar páginas de algunos procesos que no están cargados completamente en memoria, aprovechando los beneficios de la administración paginada por demanda. Un proceso se esta ejecutando pero necesita datos que están almacenados en una página que no esta cargada en memoria principal y no queda ningún frame libre que pueda ser asignado.



### Lectura requerida

Silberschatz A. y Galvin P.; Capitulo 8 Gestión de memoria. **En su: Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. Página 239



### Lectura Sugerida

Stallings W.; Capítulo 7 Gestión de memoria. **En su: Sistemas Operativos –Aspectos Internos y principios de diseño-**; 5ta Edición; España Prentice Hall, 2007. Página 307

Silberschatz A. Galvin P. Gagne G.; Capítulo 8 Memoria principal; **En su: Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Garw Hill 2007.

Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 4 Gestión de memoria. **En su: Sistemas Operativos –Una visión aplicada-**; España Mc Graw Hill, 2001. Página 163.

### Guía para la lectura

Durante o luego de la lectura de la bibliografía responda las siguientes preguntas:

- ¿Cuál es la función de la memoria virtual?
- Describa las características principales de la administración de memoria paginada por demanda.
- ¿Qué hardware apoya a la paginación por demanda?
- ¿Cuál es el efecto que tiene la administración paginada por demanda sobre el sistema de computación?
- ¿Qué es un fallo de página?



## 7. Reemplazo de páginas.

Partiendo de las problemáticas planteadas en el punto anterior, podemos decir que ahora el SO tiene que resolver otro problema, ya que los procesos siempre tienen que tener los recursos a su disposición o tienen que creer que los tienen, para poder realizar la tarea para la cual fueron creados o para resolverla.

No es un mal momento para recordar uno de los objetivos del SO, "Brindar un entorno de ejecución para los procesos", y recordemos que los procesos son los que trabajan y hacen las cosas dentro de un Sistema de Computación.

La solución es simple, el SO tiene que encontrar una página que esta cargada en memoria principal que en este momento no este siendo usada, por supuesto que para seleccionar una página para ser descartada de memoria principal y obtener un marco para almacenar la página del proceso que se tiene que ejecutar, el SO deberá seguir los pasos de alguna estrategia, estas estrategias reciben el nombre de algoritmos de selección de página víctima o algoritmos de reemplazo de páginas.

La idea es encontrar en memoria principal una página que hace mucho tiempo que no se usa o que no se usará más, esta última situación es muy difícil de resolver ya que el SO aún no puede adivinar el futuro.

La implementación de estas soluciones es más compleja, y para darnos cuenta de ello nos haremos las siguientes preguntas:

*1.- ¿Qué página debería seleccionar como víctima el SO, una página que pertenece al proceso que demanda un marco de memoria o una página de cualquier proceso?*

*2.- ¿Qué página debería seleccionar como víctima el SO, una página que haya sufrido una modificación durante su estadía en memoria principal o una página que no haya sido modificada?*

Las respuestas a estas preguntas presentan distintas formas de implementar la solución e iremos encontrando las respuestas a medida que vayamos avanzando en la lectura de la bibliografía.



### Lectura requerida

Silberschatz A. y Galvin P.; Capitulo 8 Gestión de memoria **En su: Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. Página 239.



### Lectura Sugerida

Stallings W.; Capítulo 7 Gestión de memoria. **En su: Sistemas Operativos –Aspectos Internos y principios de diseño-**; 5ta Edición; España Prentice Hall, 2007. Página 307.

Silberschatz A. Galvin P. Gagne G.; Capítulo 8 Memoria principal; **En su: Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Garw Hill 2007.

Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 4 Gestión de memoria **En su: Sistemas Operativos –Una visión aplicada-**; España Mc Graw Hill, 2001. Página 163.

### Guía para la lectura

Durante o luego de la lectura de la bibliografía responda las siguientes preguntas:

- ¿Cuáles son los algoritmos de selección de página víctima?
- Describa brevemente cada uno de los algoritmos.
- ¿Qué es la anomalía de Belady?
- Describa los algoritmos de conteo.

## 8. Asignación de marcos

Como se darán cuenta nada queda librado al azar en un sistema, entonces cuando el SO tiene que asignar marcos libres a los procesos en el momento que se mandan a ejecutar, utiliza estrategias definidas para la asignación de memoria. Estas estrategias por ejemplo, determinan la cantidad de marcos libres que se asignarán. Si el SO es muy generoso asignará todos los marcos libres que tiene en ese momento, suponiendo que no se realizarán solicitudes de marcos a futuro, pero ustedes saben que, “el que guarda siempre tiene”, esto, se convierte en algo muy importante a la hora de recibir un pedido de página que no se encuentra almacenada en memoria principal, problemática que hemos planteado en el punto anterior, si existe un marco libre para asignar no habrá que poner en ejecución la estrategia que selecciona una página víctima, de esta manera la asignación será mucho más rápida.



### Lectura requerida

Silberschatz A. y Galvin P.; Capitulo 8 Gestión de memoria. **En su: Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. Página 239



### Lectura Sugerida

Stallings W.; Capítulo 7 Gestión de memoria **En su: Sistemas Operativos –Aspectos Internos y principios de diseño-**; 5ta Edición; España Prentice Hall, 2007. Página 307.

Silberschatz A. Galvin P. Gagne G.; Capítulo 8 Memoria principal; **En su: Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Garw Hill 2007.

Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 4 Gestión de memoria **En su: Sistemas Operativos –Una visión aplicada-**; España Mc Graw Hill, 2001. Página 163.

### **Guía para la lectura**

Durante o luego de la lectura de la bibliografía responda las siguientes preguntas:

- ¿Cuál es la ventaja de mantener un porcentaje de marcos libres para futuras peticiones?
- ¿Cuáles son las restricciones que limitan las estrategias de asignación de marcos?
- Describa los algoritmos de asignación.
- Describa la asignación de marcos global y local.



**SEÑALADOR**

## 9. Hiperpaginación

Pensemos que pasaría si la cantidad de procesos que se mandan a ejecutar va aumentando a medida que pasa el tiempo y que los procesos tienen un tiempo de ejecución muy alto, esto significa que el nivel de multiprogramación (cantidad de procesos que compiten por el procesador) aumenta, si a esto le agregamos una memoria que tiene baja capacidad de almacenamiento, entonces estamos en un serio problema.

Con una administración de memoria paginada bajo demanda, se asignan a los procesos una cantidad mínima de frames que almacenan algunas de sus páginas y las páginas restantes quedan en memoria virtual, de esta manera el proceso puede comenzar su ejecución, cuando la memoria ya no tiene frames libres para ofrecer es cuando todo comienza.

Supongamos que en este mismo momento se manda a ejecutar un proceso pero no hay un mínimo de frames libres para ejecución, entonces hay que seleccionar páginas víctimas y realizar un intercambio o swapping de páginas de memoria virtual a memoria principal en el mejor de los casos y en el peor de los casos de memoria principal a memoria virtual y de memoria virtual a memoria principal, es así como el sistema comienza a trabajar mucho más tiempo en el intercambio de páginas que en la propia ejecución de los procesos, provocando que los procesos no avancen en su ejecución y esto para el usuario significaría un ataque de nervios.

Por todo lo dicho hasta aquí lo/la invitamos a realizar la siguiente actividad de lectura.



### Lectura requerida

Silberschatz A. y Galvin P.; Capítulo 8 Gestión de memoria. **En su: Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. Página 239



### Lectura Sugerida

Stallings W.; Capítulo 7 Gestión de memoria **En su: Sistemas Operativos –Aspectos Internos y principios de diseño-**; 5ta Edición; España Prentice Hall, 2007. Página 307.

Silberschatz A. Galvin P. Gagne G.; Capítulo 8 Memoria principal; **En su: Fundamentos de Sistemas Operativos;** 7ma Edición; España Mc Garw Hill 2007.

Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 4 Gestión de memoria. **En su: Sistemas Operativos –Una visión aplicada-**; España Mc Graw Hill, 2001. Página 163

### Guía para la lectura

Durante o luego de la lectura de la bibliografía responda las siguientes preguntas:

- ¿Qué es la hiperpaginación o Thrashing?
- ¿Cuáles son las causas de la hiperpaginación?
- ¿Qué es el working set?
- ¿Cómo se evita la Hiperpaginación?

### Cierre de la unidad

Con esta unidad seguimos analizando y estudiando una de las funciones más importantes del SO, la administración de recursos, en este caso la administración del recurso memoria principal, un recurso relativamente caro y escaso.

Actualmente, el sistema de computación se convirtió en una necesidad imprescindible para todo tipo de organización, hoy sin un sistema de computación una empresa no podría brindar un buen servicio o su producción no sería competitiva en el mercado, pero más aún en la mayoría de los hogares hay un sistema de computación y cuántas veces los usuarios sintieron en carne propia la problemática de la escasez de memoria principal en su sistema, la necesidad de agregar memoria?, si el sistema se lo permitía o cambiar el sistema por un sistema que permita administrar una memoria principal con mayor capacidad de almacenamiento.





Día a día la tecnología de software nos provee de programas cada vez más sofisticados, con sonido, imágenes y mayor funcionalidad, pero todo esto equivale a decir mayor capacidad de memoria principal para que dichos programas puedan funcionar eficientemente. Cuando decimos mayor capacidad de memoria principal decimos SO que tengan las estrategias de administración de memoria necesarias para gestionar dicho recurso.

Hemos visto como han evolucionado las administraciones de memoria desde la más simple, la administración de memoria contigua simple hasta las administraciones de memoria paginada bajo demanda, segmentada y segmentada con paginación, entendiendo cada una de ellas y maravillándonos con los pasos de la evolución del SO.

Sigamos pensando y creando nuevas formas de administración de memoria.

***Fin de la unidad 4***

## Anexo

---



### Trabajo práctico sugerido

---

#### Trabajo práctico Nº 10: **Memoria Paginada**

### Presentación

---

Este trabajo tiene el propósito de orientarlo/a para la comprensión de los temas desarrollados en la cuarta unidad de esta asignatura.

Estos ejercicios lo/la ayudarán a entender una de las más interesantes formas de administrar la memoria de un sistema de computación, la paginación simple: permite almacenar un proceso en memoria de forma no necesariamente contigua. La administración de memoria paginada bajo demanda permite además ejecutar un proceso sin necesidad de que éste esté almacenado en la memoria en su totalidad.

Este trabajo intenta favorecerle el acceso a las siguientes metas de aprendizaje:

- Entender el mecanismo que permite ejecutar un proceso que no necesariamente está almacenado en forma contigua y en su totalidad en memoria.
- Repasar los conceptos de la administración de memoria paginada.
- Conocer cómo el sistema operativo administra el recurso memoria Principal.

Le presentamos a continuación, las consignas de trabajo:

### Consignas

---

1. Considere un espacio de direcciones lógicas de 8 páginas de 1024 Bytes cada una, y un espacio de direcciones físicas de 32 frames.
  - a) ¿Cuántos bits se necesitan para especificar una dirección lógica?
  - b) ¿Cuántos bits se necesitan para especificar una dirección física?
2. Considere un sistema con un espacio lógico de memoria de 128 K páginas con 8 KB cada una y direccionamiento al nivel de Byte. ¿Cuántos bits son necesarios para especificar una dirección lógica?



3. Dado un sistema de paginación donde cada frame direcciona 1024 palabras de 64 bits, en el que deben residir simultáneamente en memoria principal los procesos A, B, C y D con los requerimientos de memoria que se indican en el cuadro siguiente. Se requiere calcular cuantos frames demanda cada proceso y el mínimo de frames que debe poseer la memoria para satisfacer las necesidades de estos procesos.

4.

Proceso	Memoria requerida en KBytes	Memoria requerida en cantidad de Frames
A	1248	
B	6240	
C	2600	
D	2080	

5. Sea un sistema de memoria virtual paginada con direcciones lógicas de 32 bits que proporcionan un espacio lógico virtual de 220 páginas y con una memoria física de 32 MB. ¿Cuánto ocupará la tabla de marcos de página si cada entrada de la misma ocupa 32 bits?
6. Considere un sistema que trabaja con un esquema de paginación a demanda. El tamaño de la página es de 4K, y las direcciones lógicas son de 32 bits. Se pide determine cuántas páginas se pueden direccionar y a qué página hace referencia la dirección 12345 (base 10).
7. Considere un sistema que trabaja con un esquema administración de memoria paginada a demanda en donde están en ejecución 3 procesos: A, B y C con los siguientes requerimientos de memoria real: 200 K, 150 K y 300 K respectivamente. La longitud de cada página es de 1K.
- Cual es la cantidad de frames de la memoria real si las instrucciones tienen direcciones de 32 bits?
  - Determine el contenido de las tablas de páginas para los 3 procesos considerando que los frames de 0 a 7 contienen las páginas: A-0, B-0, C-5, A-1, C-3, C-1, B-2 y C-4.
  - En la dirección lógica X'0680' del proceso A se encuentra una instrucción de bifurcación condicional. Utilizando las tablas definidas en el punto anterior, determine la dirección física de dicha instrucción.

---

Al finalizar, compare su producción con la grilla que incluimos a continuación.



## Grilla de Autocorrección Nº 10: Memoria Paginada

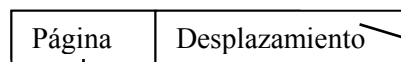
### Orientaciones para la corrección



*Recuerde que estas son sólo orientaciones para que usted pueda comenzar a desarrollar el trabajo práctico.  
Usted puede ampliar cualquiera de estos conceptos utilizando bibliografía adecuada, imaginación y creatividad.*

### Respuesta 1

#### a) Dirección lógica

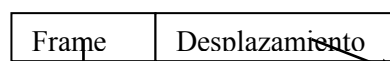


Cada página 1024 bytes =  $2^{10}$  bytes = 10 bits.

8 páginas =  $2^3$  páginas = 3 bits.

Para especificar una dirección lógica se necesitan 13 bits.

#### b) Dirección física



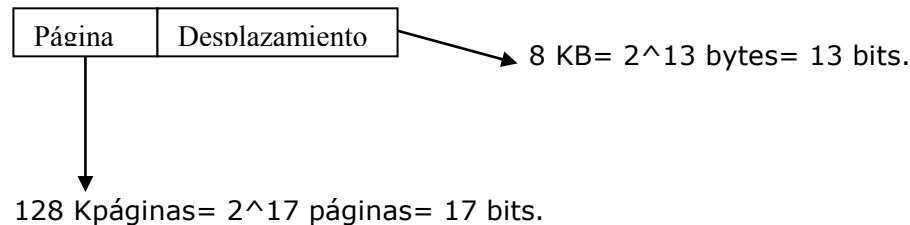
Cada frame = cada página = 10 bits.

32 frames =  $2^5$  frames = 5 bits.

Para especificar una dirección física se necesitan 15 bits.

## Respuesta 2

Dirección lógica



En la dirección lógica hay 30 bits.

## Respuesta 3

1024 palabras de 64 bits =  $2^{10}$  palabras \* 8 bytes = 8 KBytes.

Proceso A = 1248 KBytes / 8 KBytes = 156 frames.

Proceso B = 6240 KBytes / 8 KBytes = 780 frames.

Proceso C = 2600 KBytes / 8 KBytes = 325 frames.

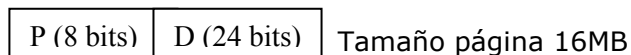
Proceso D = 2080 KBytes / 8 KBytes = 260 frames.

Total = 1521 frames.

El mínimo de frames que debe poseer la memoria para satisfacer a todos los procesos es de 1521.

## Respuesta 4

Dirección lógica



Memoria física = 32 MB = 2 frames

TDP

pagina	marco
8 bits	1 bits

Tabla de Marcos de Página

Marco	estado	Tamaño
0	0	2 bits
1	0	2 bits
		4 bits

## Respuesta 5

Desplazamiento = 4096 Bytes (de 0 a 4095) necesita 12 bits para el desplazamiento

Para el Número de página son 32 bits – 12 bits = 20 bits

Con 20 Bits se pueden direccional 1.048.576 páginas = 1024K páginas

Para saber a que página hace referencia la dirección 12345 (10) hay que pasarla a binario y tomar de dicho número binario los bits que corresponden al número de página.

12345(10) = 00000000000000000011

Página

000000111001 (2)

Desplazamiento

La dirección 1234(10) hace referencia a la página 3

Respuesta 6

El desplazamiento es de 1024 bytes = necesito 10 bits de desplazamiento

32 bits – 10 bits = 22 bits para el frame = 4194304 marcos

- a. Determine el contenido de las tablas de páginas para los 3 procesos considerando que los frames de 0 a 7 contienen las páginas: A-0, B-0, C-5, A-1, C-3, C-1, B-2 y C-4.

TDPA		TDPB		TDPC	
Pagina	frame	pagina	frame	pagina	frame
0	0	0	1	0	-
1	3	1	-	1	5
2	-	2	6	2	-
3	-	3	-	3	4
4	-	4	-	4	7
5	-	5	-	5	2
continua		continua		continua	

- b. En la dirección lógica X’0680’ del proceso A se encuentra una instrucción de bifurcación condicional. Utilizando las tablas definidas en el punto anterior, determine la dirección física de dicha instrucción

hexadecimal 0

binario 0000

6

0110

8

1000

0

0000

Dirección lógica:

Desplazamiento  
1010000000

