

Resumen 2do Parcial

SOP Teórico

U3: Administración de procesos. Continuación

Recursos:

Es un elemento utilizado por un solo proceso en un momento determinado, ya sea hardware o software.

Es cualquier elemento que puede ser solicitado, utilizado o liberado.

Hay 2 tipos:

- Apropiativo: Puede liberarse del proceso que lo está utilizando sin dificultades
- No Apropiativo: No puede liberarse del proceso que lo está utilizando sin provocar un fallo en el computo

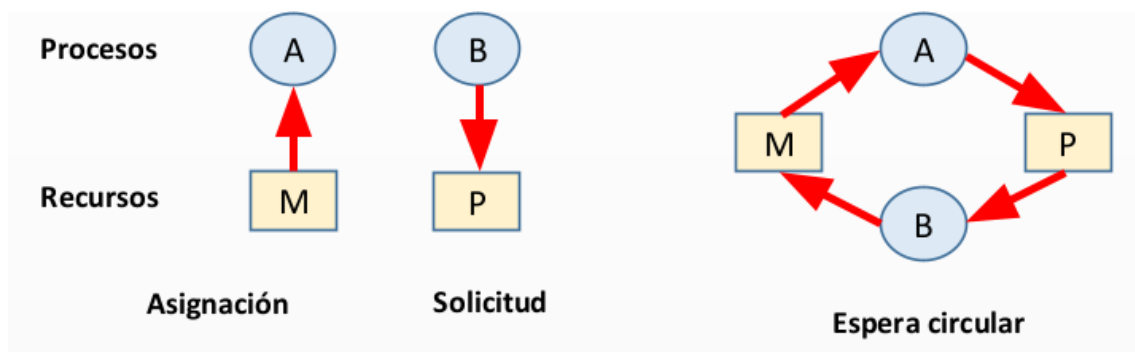
Interbloqueos

Es una situación en la que se encuentran un conjunto de procesos cuando todos los procesos están esperando un suceso que solo puede ser ocasionado por otro proceso que también está esperando otro suceso.

Como todos los procesos están en espera ninguno podrá devolver recursos y seguirán bloqueados indefinidamente.

Modelado de Interbloqueos:

Los procesos serán círculos, los recursos cuadrados y las acciones flechas direccionadas:



Condiciones para que se dé un interbloqueo:

- Exclusión mutua: Un recurso está asignado a un proceso o está disponible.
- Posesión y espera: Un proceso posee algunos recursos y los que no los tiene los solicita.
- No Apropiativo: El recurso no se puede quitar y debe ser liberado.
- Espera circular: Un conjunto de procesos esperan recursos formando un bucle.

Estrategias para Enfrentar los Interbloqueos:

a) Ignorar el problema: Algoritmo Avestruz

Se ignora el problema, prefiriendo que se produzca el interbloqueo esporádicamente o no regularmente y no gastar recursos en tratar de evitarlo.

b) Detección-Recuperación del Interbloqueo

El SOP permite que se produzca el interbloqueo para detectar el algoritmo que lo produjo.

Se implementa de 3 formas:

- Apropiación: Se quita el recurso al proceso que provoca el interbloqueo temporalmente para asignarlo a otro proceso.
- Vuelta Atrás: Se genera un registro con los estados del proceso y si se llega a un interbloqueo se vuelve al estado anterior antes de que se produjera.
- Eliminar el Proceso: Se matan los procesos que pueden ocasionar el ciclo.

c) Evitar los interbloqueos

El sistema deberá decidir si es seguro asignar un recurso o no teniendo información pertinente de antemano, suministrados por los procesos determinando previamente que y cuanto de cada recurso necesita utilizar.

d) Prevención de Interbloqueos

Se implementa cuando el sistema evita que se den las 4 condiciones necesarias para que se produzca el interbloqueo

- Exclusión mutua: Se delega el uso y acceso del recurso a través de procesos demonio
- Posesión y espera: Si un proceso no puede tener todos los recursos que necesita, los que ya tiene los debe liberar
- No apropiativo: Es difícil anular esta situación

- Espera circular: Los procesos pueden solicitar procesos en orden creciente, y si solicita un recurso de orden menor deberá liberar el actual para adquirir el otro.

U4: Administración de Memoria.

Estrategias de administración de memoria:

Administrador de memoria:

Sus funciones son

- Registrar las partes libres y ocupadas de la memoria
- Asignar y liberar espacio en memoria para los procesos
- Administra el swapping de los procesos

Técnicas de Administración de memoria

- Mono programación
- Multi programación
 - a. Con particiones fijas
 - b. Con particiones variables
- Memoria Virtual
 - a. Paginación
 - b. Segmentación

Mono Programación

Consiste en cargar un solo programa de usuario en memoria por ejecución

Desventajas:

- Se ejecuta un proceso a la vez
- Solo se asigna memoria
- Baja utilización de memoria (probable desperdicio de recursos)

Multi programación

a) Multi programación Con particiones Fijas (PF):

Consiste en dividir la memoria en N particiones de diferente tamaño y cuando se carga un proceso en una partición de memoria pertenece ahí hasta que finaliza.

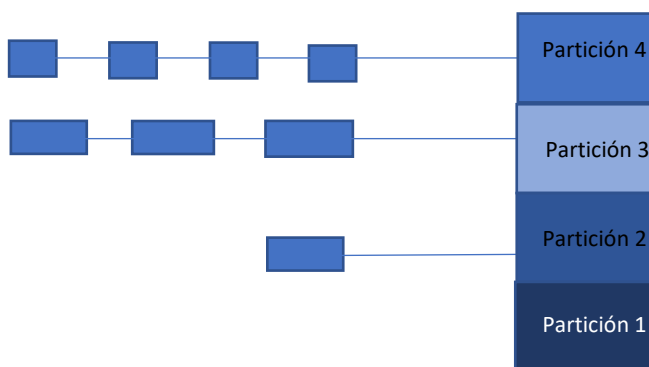
Tiene 2 variantes:

- PF con cola independiente
- PF con cola única

PF con cola independiente:

Cada partición tiene su propia cola de procesos y solo puede cargar en una partición los procesos de la cola correspondiente.

Inconveniente: Si una partición se queda sin procesos en la cola, no podrá ser aprovechada para cargar otros procesos por lo que se desperdician recursos.



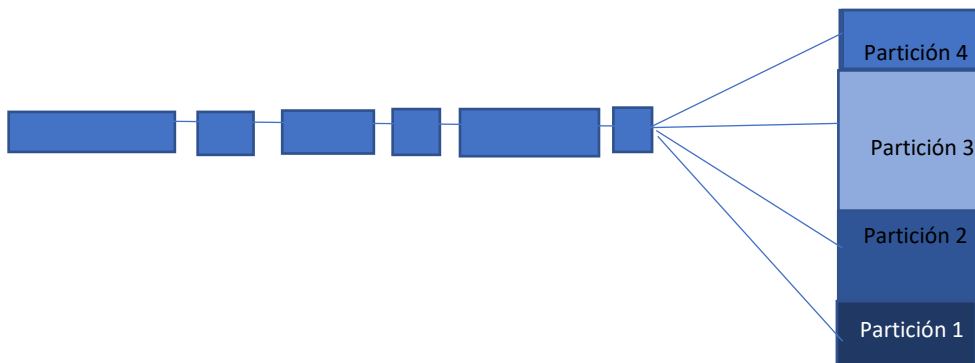
Importante:

Cuando decimos que hay espacios de memoria que quedan sin poder utilizarse decimos que ocurre una fragmentación, y hay 2 tipos de esta:

Fragmentación Interna	Fragmentación Externa
Tenemos un tamaño de memoria que podría ser utilizado, pero por un motivo X no se usa. Se le asigna a un proceso mas memoria de la que necesita.	Tenemos un espacio de memoria libre, pero es demasiado pequeño para cargar un proceso. Se le asigna a un proceso menos memoria de la que necesita.
Ejemplo: Guardar un auto en un hangar	Ejemplo: Guardar un auto en una caja de zapatos.

PF con cola única:

Este algoritmo busca en la cola el proceso que mejor se ajuste al tamaño de la partición.



Inconveniente: los procesos pequeños no son seleccionados.

Solución: Un proceso no puede ser ignorado una cantidad mayor a X veces.

Grado de multiprogramación: Hace referencia a la cantidad de procesos que pueden ser cargados en memoria al mismo tiempo, coincide con el número de particiones de la memoria.

Overflow: Suceso en el cual un proceso crece a un tamaño mayor que la partición a la cual fue asignado.

Abstracción de Memoria

- Espacio de Direcciones: Conjunto de direcciones que puede utilizar un proceso para direccionar la memoria. Cada proceso tiene su propio espacio de direcciones.
- Registro base: Es donde se carga la dirección física donde empieza el programa en memoria. Indica desde donde se puede usar el programa.
- Dirección en Memoria = Reg Base + Dirección Absoluta
- Registro Limite: Se carga con la longitud del programa, es decir, cuantas direcciones de memoria va a ocupar.

b) Multi programación Con Particiones Variables

Se introduce el intercambio:

- Se lleva el proceso completo a memoria, se ejecuta un tiempo y se baja al disco
- Los procesos inactivos se almacenan en el disco, para liberar memoria mientras no se ejecutan
- Cuando un proceso vuelve a subir a la memoria, puede cargarse en otro espacio de direcciones respecto del que estaba antes
- El número y tamaño de las particiones varía en forma dinámica y se crean a medida que se cargan los procesos
- Las particiones que se crean se adaptan al tamaño de los procesos

Desventajas de las particiones dinámicas:

- Complejidad de asignación y liberación de memoria
- Hay que buscar memoria libre para asignar a un proceso que llega
- Hay que tener un margen de espacio libre por si el proceso crece
- El que las particiones se creen adaptándose a los procesos provoca mucha fragmentación externa, debido a que los espacios libres no se pueden reorganizar y quedan muy pequeños e imposibilita utilizarlos para asignar ese espacio a un proceso. Necesidad de desfragmentar la memoria, lo cual gasta tiempo de procesamiento.

Técnicas de Administración de Memoria

- 1) Mapa de Bits
- 2) Lista enlazada

1) Mapa de Bits:

La memoria RAM se divide en unidades de asignación (UA)

Bit = 0 → UA libre

Bit = 1 → UA ocupada

¿Cómo afecta la fragmentación al mapa de bits?

- Si las UA son muy pequeñas hacen falta más cantidad para asignar procesos y el mapa de bits crece. Se produce fragmentación externa.
- Si las UA son muy grandes si el proceso no llena las UA se desperdiciará espacio de memoria. Se produce fragmentación interna.

¿De Que depende el tamaño del Mapa de Bits?

Depende de la cantidad de UA.

¿Cómo se asigna memoria a un proceso mediante el Mapa de Bits?

Se buscan UA libres, y se las asigno al proceso. Cambiar los bits de 0 a 1

¿Cómo se registra la liberación de memoria cuando un proceso finaliza?

Se ubican las UA que el proceso ocupaba, se liberan y se cambian los bits de 1 a 0.

2) Listas Enlazadas:

Consta de organizar los procesos en segmentos enlazados mediante direcciones de memoria.

Cada segmento consta de 4 partes:

Proceso o Hueco	Dirección de Comienzo	Longitud	Puntero al próximo segmento
-----------------	-----------------------	----------	-----------------------------

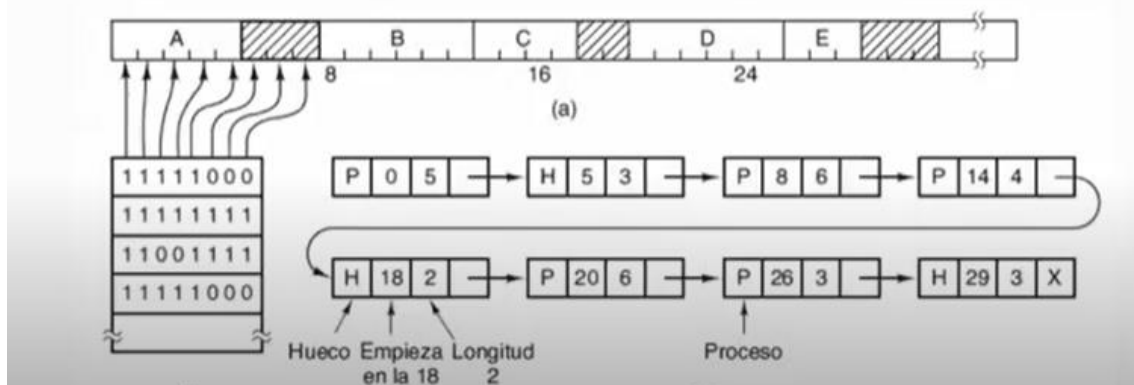
La primera parte indica si el segmento lo ocupa un hueco (memoria libre) o un proceso.

La segunda indica la dirección desde la cual el segmento es usado en memoria.

La tercera indica cuantas direcciones de memoria ocupa el segmento iniciando el conteo desde la dirección inicial. Por ejemplo, si la dirección inicial es 15 y la longitud 4, el segmento ocupa las direcciones 15, 16, 17 y 18.

La cuarta parte indica cual es la dirección donde está el próximo segmento.

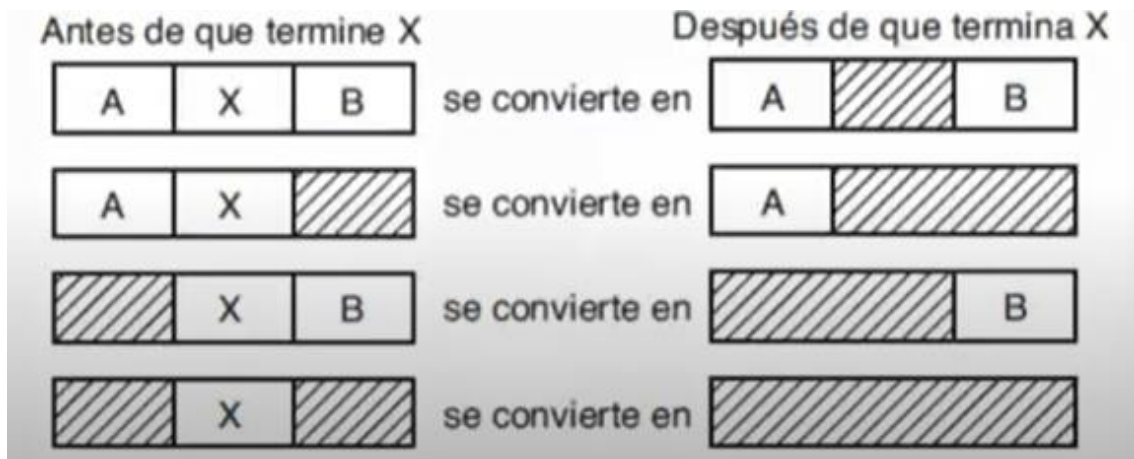
• Ejemplo de lista enlazada:



Liberación de memoria en lista enlazada:

Cuando el proceso cargado en un segmento de la lista finaliza pueden ocurrir 2 situaciones:

- Si los segmentos contiguos al segmento liberado son procesos, ese segmento se queda como un Hueco ocupando el espacio de direcciones del proceso.
- Si el segmento liberado tiene algún hueco contiguo, el hueco resultante tendrá la longitud combinada de ambos segmentos e iniciará en la dirección inicial más pequeña.



Algoritmos de Asignación de Memoria:

- Primer ajuste: Busca el primer hueco en el cual quepa el proceso.
 - Siguiendo ajuste: Funciona igual que el primer ajuste, pero inicia la búsqueda del hueco donde quedó antes.
 - Mejor ajuste: Busca el hueco mas pequeño donde quepa el proceso.
 - Peor Ajuste: Busca el hueco mas grande en el cual quepa el proceso. El objetivo de esto es que el hueco resultante de la asignación de memoria sea más grande, y por ello utilizable, y provoque menos fragmentación externa.
 - Ajuste Rápido: Organiza los huecos en categorías de espacio libre y usa una categoría estándar para asignar ese espacio a procesos que usualmente tienen el mismo tamaño.
- El problema es que luego de realizar la asignación hay que reubicar el hueco resultante.

Inconvenientes de la administración de memoria con particiones fijas y variables:

- Un proceso debe estar cargado totalmente en memoria y en direcciones continuas.
- No se podrá ejecutar un proceso si su tamaño es mayor al de la RAM disponible.
- El grado de multiprogramación es limitado al espacio de la RAM.
- Solución: **Memoria Virtual**

Memoria Virtual:

Es una técnica de administración de memoria en la cual se reserva un espacio en el Disco para usarlo como la RAM

El SOP crea un espacio de direcciones virtuales en el disco

Permite ejecutar programas cuyo tamaño es mayor al de la RAM disponible

Se implementa dividiendo el programa en partes en las que algunas estarán cargadas en la RAM y otras en el Disco

Aumenta el grado de multiprogramación



Se implementa por 2 técnicas:

- Paginación
- Segmentación

Paginación

Cada programa genera y tiene su propio **espacio de direcciones virtuales**, el cual está formado por el proceso dividido en partes iguales llamadas **páginas**, las cuales cuando se cargan en la RAM se cargan en un espacio llamado **marco de página** que tiene el mismo tamaño que estas.

Traducción de direcciones:

Es una tarea que realiza la unidad de administración de memoria (MMU), en la cual cuando un proceso se ejecuta a medida que se necesitan paginas se cargan en memoria y las que no se bajan al disco, la MMU se encarga de traducir las direcciones de las páginas que deben subirse a memoria que están en el espacio de direcciones virtuales a la dirección efectiva en la memoria RAM.

Fallo de página: Suceso que ocurre cuando un programa intenta utilizar una página que no está cargada en la RAM.

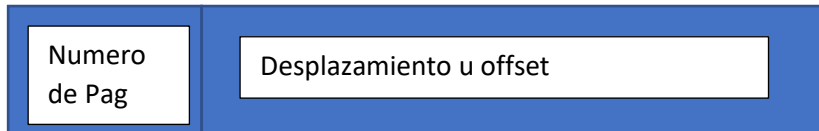
El MMU detecta la página no asociada y hace que el CPU libere espacio en

memoria bajando alguna página al disco.

Se carga la página solicitada en la RAM y se asocia en la tabla de páginas, y luego se continua con la ejecución de la instrucción que generó el fallo.

Procedimiento para la traducción de direcciones

Si la dirección virtual posee 16 bits se compondrá de la siguiente forma.

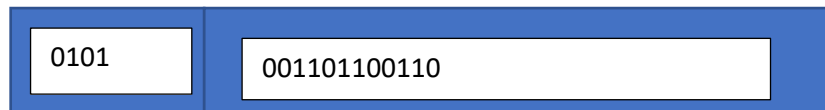


4 bits

12 bits

1) Calcular el tamaño de las páginas en bytes, ejemplo páginas de 4kb = 4096 bytes

2) Calcular el decimal del número de página y del desplazamiento de la dirección virtual, ejemplo de dirección virtual



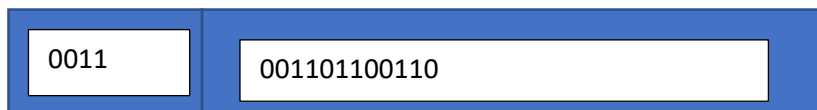
número página: $0101 = 5$

offset = $2^1 + 2^2 + 2^5 + 2^6 + 2^8 + 2^9 = 870$

Dirección virtual = número de página x tamaño de página + offset

Dirección Virtual = $5 \times 4096 + 870 = 21350$

3) Calcular la dirección física a la cual apunta la dirección virtual, en el ejemplo la página 5 virtual apunta a:



Se procede igual que con la dirección virtual, se calcula la página de la memoria a la que apunta y el offset:

Número de página: $0011 = 3$

Offset = 870

Dirección Física: número de página x tamaño de página + offset

Dirección Física: $3 \times 4096 + 870 = 13158$

Ósea que la dirección 21350 virtual está cargada en memoria en la dirección 13158.