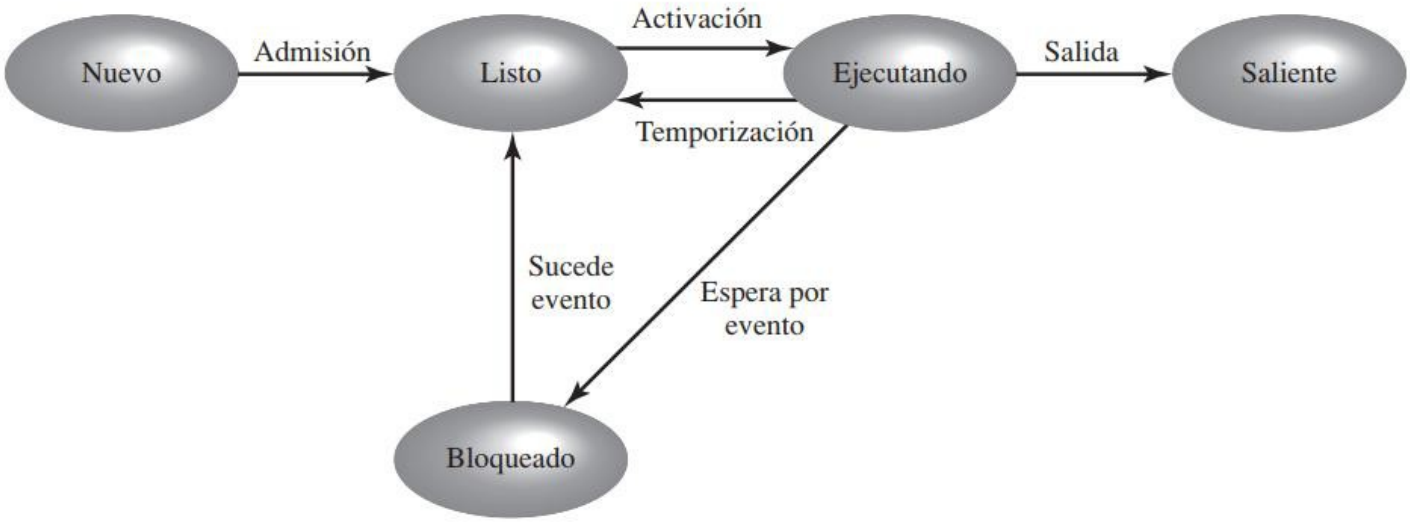
**Procesos**

*- Defina proceso.*

Entidad que consiste de un código de programa y un conjunto de datos asociados a dicho código, y si el procesador comienza a ejecutar este código de programa, nos referimos a esta entidad en ejecución como un proceso.

*- Dibuje un modelo de procesos de 5 estados. Explique cada estado. ¿Cuáles son los eventos por los cuales se pasa de un estado a otro?*
*- Dibuje un diagrama de estados con los estados principales de un proceso. Explique cada uno de los estados.*




**Ejecutando**: El proceso está actualmente en ejecución.
**Listo**: Un proceso que se prepara para ejecutar cuando tenga oportunidad.
**Bloqueado**: Un proceso que no puede ejecutar hasta que se cumpla un evento determinado o se complete una operación E/S.
**Nuevo**: Un proceso que se acaba de crear y que aún no ha sido admitido en el grupo de procesos ejecutables por el sistema operativo. Típicamente, se trata de un nuevo proceso que no ha sido cargado en memoria principal, aunque su bloque de control de proceso (BCP) si ha sido creado.
**Saliente**: Un proceso que ha sido liberado del grupo de procesos ejecutables por el sistema operativo, debido a que ha sido detenido o que ha sido abortado por alguna razón.

*Cuando una interrupción o llamada al sistema transfiere el control al SO, por lo general se utiliza un área de la pila distinta a la pila del proceso. Explique cuál es la razón.*

Cuando ocurre una interrupción o llamada al sistema, el procesador se pone en modo núcleo y el control se pasa al sistema operativo. Para este fin, el contexto se salva (pila kernel) y se cambia de modo a una rutina del sistema operativo. Sin embargo, la ejecución continúa dentro del proceso de usuario actual. De esta forma, no se realiza un cambio de proceso, sino un cambio de modo dentro del mismo proceso. Si el sistema operativo, después de haber realizado su trabajo, determina que el proceso actual debe continuar con su ejecución, entonces el cambio de modo continúa con el programa interrumpido, dentro del mismo proceso.

*- ¿Cuáles son las actividades principales que el SO realiza en la gestión de procesos?*

Creación y terminación de procesos. Planificación y activación de procesos. Intercambio de procesos. Sincronización de procesos y soporte para comunicación entre procesos. Gestión de los bloques de control de proceso.

*- Un proceso está en estado suspendido: ¿Cuáles son sus características y cómo pudo haber llegado a dicho estado?*
*- Explique la razón de incorporar un estado suspendido a un modelo de estados de procesos.*

Características (4):
- •No está listo para ser ejecutado
- •La condición de bloqueo es independiente a la de suspensión (si está esperando un evento y este ocurre, esto no habilita al proceso a ser ejecutado)
- •Fue suspendido por un agente (él mismo, el padre o el SO)
- •No puede salir de suspendido hasta que el agente le permita hacerlo

Razones (5):
- •Swapping (se necesita memoria para traer otro proceso)
- •Temporización (proceso que se ejecuta cada cierto intervalo)
- •Solicitud del padre (pone en espera al hijo para modificarlo o coordinar algo con otros hijos)
- •Solicitud del usuario (para debuggear o porque usa muchos recursos)
- •Otras razones del SO (sospecha que anda mal)

*- Explique detalladamente qué es el intercambio de procesos.*

Mover parte o todo el proceso de memoria principal a disco. Cuando ningún proceso está en estado listo, el SO manda uno de los procesos bloqueados a disco, en la cola de suspendidos. Luego, el SO trae un proceso de la cola de suspendidos o admite uno nuevo.

*- ¿Cuáles son las estructuras de control más importantes del SO? ¿Para qué se utilizan cada una de ellas?*
*- Plantee un esquema general de las tablas del SO y detalle la estructura de los procesos.*

Las estructuras de control más importantes del SO son las tablas de memoria, E/S, ficheros y procesos.
•Las **tablas de memoria** se usan para mantener un registro tanto de la memoria principal (real) como de la secundaria (virtual).
•Las **tablas de E/S** gestionan los dispositivos de E/S y los canales del computador.
•Las **tablas de ficheros** proporcionan información sobre la existencia de ficheros, su posición en almacenamiento secundario, su estado actual y otros atributos.
•Las **tablas de procesos** se utilizan para gestionar procesos.

Un proceso está compuesto por un programa (o varios), un conjunto de posiciones de memoria donde se encuentran los datos (variables locales y globales, constantes), una pila para registrar los llamados a procedimientos junto con sus parámetros y un conjunto de atributos que usa el SO para controlar el proceso (BCP).

*- ¿Cuándo se puede producir un intercambio de procesos? Explique cada opción.*

Ocurre cuando el SO toma control sobre el proceso actualmente en ejecución (interrupciones, trap, llamadas al sistema)

Interrupciones
- •Int de reloj: proceso que excede el time slice, pasa a listo.
- •Int de E/S: El SO determina que operación de E/S ocurrió, si muchos procesos estaban esperando esa operación, el SO los mueve a listo (o de bloq/susp a listo/susp). Después el SO decide si continúa la ejecución del proceso actual o si mete otro de mayor prioridad.
- •Fallo de memoria: el procesador se encuentra con una referencia a una dirección de memoria virtual, que no se encuentra en memoria principal. El sistema operativo debe traer el bloque (página o segmento) que contiene la referencia, desde memoria secundaria a memoria principal.

Trap
- •El SO conoce si un error/excepción es irreversible o no. Si es irreversible, el proceso en ejecución pasa a saliente y se hace un cambio de proceso. Si es reversible, el SO actúa en base al diseño y naturaleza del error (puede continuar ejecutando el mismo proceso o cambiar).

Llamada al sistema:
- •El proceso actual realiza una llamada al sistema (abrir un archivo). Ésta llamada provoca la ejecución de una rutina del SO que a veces provoca que el proceso actual pase a bloqueado.

*- ¿A qué se le llama traza de un proceso? ¿y traza combinada?*

La traza de un proceso es la ruta de ejecución del mismo a través de uno o más programas. La traza combinada se refiere el cambio de procesos en ejecución en una traza en particular.

*Dibuje y explique un diagrama de procesos de dos estados. Dibuje y explique cómo queda conformado un diagrama de colas para ejecutar procesos con dos estados.*

Dos estados, ejecutando y no ejecutando; cuatro eventos: entrada, activación, detención y salida. Cuando el SO crea un nuevo proceso, crea el BCP y mete el proceso en estado no ejecutando.

Los procesos no ejecutando tienen su BCP en una cola.

*- ¿Cuáles son los eventos que producen la creación de procesos? De un ejemplo de cada uno.*

**Nuevo proceso de lotes**: el SO dispone de lotes de trabajos, cuando este está listo para procesar un nuevo trabajo, leerá el siguiente y lo ejecutará.
**Sesión interactiva**: usuario desde una terminal interactúa con el sistema.
**Creado por el SO para proporcionar un servicio**: proceso para controlar la impresión.
**Creado por un proceso existente**: para usar el paralelismo o modularizar.

*- De ejemplos de terminación de procesos.*

**Finalización normal**
**Límite de tiempo excedido**
**Memoria no disponible**
**Violaciones de frontera**
**Error de protección**
**Error aritmético**
**Límite de tiempo**
**Fallo de E/S**
**Instrucción no válida**
**Instrucción privilegiada**
**Uso inapropiado de datos**

**Intervención del operador**
**Terminación del proceso**
**Solicitud del proceso padre**

*- Al pasar de un modelo de dos estados a uno de 5 estados uno de los estados se divide en dos, Listo y Bloqueado. ¿Cuál es ese estado? ¿Cuál es la razón por la cual se divide este estado en dos?*

Se refiere al estado No Ejecutado. Algunos procesos que están en el estado de No Ejecutando están listos para ejecutar, mientras que otros están bloqueados, esperando a que se complete una operación de E/S. Por tanto, utilizando una única cola (para los No Ejecutados), el activador no puede seleccionar únicamente los procesos que lleven más tiempo en la cola. En su lugar, debería recorrer la lista buscando los procesos que no estén bloqueados y que lleven en la cola más tiempo. Una forma más natural para manejar esta situación es dividir el estado de No Ejecutando en dos estados, Listo y Bloqueado.

*- ¿Cómo está formada la "información del estado del proceso"?*
La información de estado de proceso indica los contenidos de los registros del procesador. Cuando un proceso está ejecutando, esta información está, por supuesto, en los registros. Cuando un proceso se interrumpe, toda la información de los registros debe guardarse de forma que se pueda restaurar cuando el proceso continúe con su ejecución. Normalmente el conjunto de registros incluye: registros visibles por usuario, registros de control y de estado, y punteros de pila (imagen del proceso).

*- ¿Que realiza el SO cuando crea un proceso?*

1. **Asignar un identificador de proceso único al proceso**: se añade una nueva entrada a la tabla 1ria de procesos.

2. **Reservar espacio para el proceso**: todo lo que contiene la imagen del proceso.

3. **Inicialización del BCP**: PID y PPID según corresponda, en el estado se pone todo en 0 menos el PC (se pone donde arranca) y los punteros de pila de sistema.

4. **Establecer los enlaces apropiados**: se mete en la cola correspondiente (Listos, Listos/Suspendidos).

5. **Creación o expansión de otras estructuras de datos**.

*- ¿Qué son "unidad propietaria de recursos" y "unidad de ejecución"? ¿Cuál es la diferencia entre un proceso y un hilo?*

Hasta ahora el concepto de un proceso abarcaba dos características:

•Propiedad de recursos: un proceso incluye un espacio de direcciones virtuales para el manejo de la imagen del proceso (colección de programa, datos, pila y atributos definidos en el bloque de control del proceso). De vez en cuando a un proceso se le puede asignar control o propiedad de recursos tales como la memoria principal, canales E/S, dispositivos E/S y archivos. El sistema operativo realiza la función de protección para evitar interferencias no deseadas entre procesos en relación con los recursos.

•Planificación/ejecución: la ejecución de un proceso sigue una traza o una traza combinada. De esta manera, un proceso tiene un estado de ejecución (Ejecutando, Listo, etc.) y una prioridad de activación y ésta es la entidad que se planifica y activa por el sistema operativo.

Estas dos características son la esencia de un proceso, sin embargo son independientes y podrían ser tratadas como tales por el sistema operativo. Para distinguir estas dos características, la unidad que se activa **(unidad de ejecución)** se suele denominar **hilo** (thread), o proceso ligero, mientras que la **unidad de propiedad de recursos** se suele denominar **proceso** o tarea.

*- ¿Qué significa multihilo? ¿Cuáles son las diferencias entre modelo multihilo y monohilo?*

Multihilo se refiere a la capacidad de un sistema operativo de dar soporte a múltiples hilos de ejecución en un solo proceso. El enfoque tradicional de un solo hilo de ejecución por proceso, en el que no se identifica con el concepto de hilo, se conoce como estrategia monohilo.

En un modelo de proceso monohilo, la representación de un proceso incluye su BCP y el espacio de direcciones de usuario, además de las pilas de usuario y núcleo para gestionar el comportamiento de las llamadas/retornos en la ejecución de los procesos. Mientras el proceso está ejecutando, los registros del procesador se controlan por ese proceso y, cuando el proceso no se está ejecutando, se almacena el contenido de estos registros.

En un entorno multihilo, sigue habiendo un único BCP y un espacio de direcciones de usuario asociado al proceso, pero ahora hay varias pilas separadas para cada hilo, así como un bloque de control para cada hilo que contiene los valores de los registros, la prioridad, y otra información relativa al estado del hilo. De esta forma, todos los hilos de un proceso comparten el estado y los recursos de ese proceso, residen en el mismo espacio de direcciones y tienen acceso a los mismos datos.

*- ¿Un hilo puede estar en estado suspendido? Explique.*

No. Porque no tiene sentido aplicarlo, ya que dicho estado es un concepto de nivel de proceso. En particular, si se expulsa un proceso, todos sus hilos se deben expulsar porque comparten el espacio de direcciones del proceso.

*- ¿Qué tipos de hilos existen? Detalle cada uno (a nivel de núcleo y a nivel de usuario).*

Existen dos amplias categorías de implementación de hilos: hilos de nivel de usuario (user-level threads, **ULT**) e hilos de nivel de núcleo (kernel-level threads, **KLT**).

**ULT**: la aplicación gestiona todo el trabajo de los hilos y el núcleo no sabe que estos existen, sino que para el núcleo sólo existe un proceso. Se pueden crear nuevos hilos mientras se ejecuta el proceso.

**KLT**: el núcleo gestiona todo el trabajo de los hilos, las aplicaciones no controlan los hilos. Las aplicaciones interaccionan con una API del kernel que gestiona los hilos.

## Memoria

*- ¿Cómo se divide la memoria en un sistema monoprogramado, y en uno multiprogramado?*

En un sistema monoprogramado, la memoria se divide en dos partes: una parte para el sistema operativo (monitor residente, núcleo) y una parte para el programa actualmente en ejecución.

En un sistema multiprogramado, la parte de «usuario» de la memoria se debe subdividir posteriormente para acomodar múltiples procesos.

*- ¿Quién se encarga de la subdivisión de la memoria? ¿Cómo se denomina esta tarea?*

El sistema operativo es el encargado de la tarea de subdivisión, y a esta tarea se le denomina **gestión de la memoria**.

*- ¿Cuáles son los requisitos que debe satisfacer la gestión de memoria? Explique cada uno de ellos.*

•**Reubicación**: una vez que un programa se ha llevado al disco, sería bastante limitante tener que colocarlo en la misma región de memoria principal donde se hallaba anteriormente, cuando éste se trae de nuevo a la memoria. Por el contrario, podría ser necesario reubicar el proceso a un área de memoria diferente. Por tanto, no se puede conocer de forma anticipada dónde se va a colocar un programa y se debe permitir que los programas se puedan mover en la memoria principal, debido al intercambio o swap.

•**Protección**: cada proceso debe protegerse contra interferencias no deseadas por parte de otros procesos, sean accidentales o intencionadas. Por tanto, los programas de otros procesos no deben ser capaces de referenciar sin permiso posiciones de memoria de un proceso, tanto en modo lectura como escritura.

•**Compartición**: cualquier mecanismo de protección debe tener la flexibilidad de permitir a varios procesos acceder a la misma porción de memoria principal. Por ejemplo, si varios programas están ejecutando el mismo programa, es ventajoso permitir que cada proceso pueda acceder a la misma copia del programa en lugar de tener su propia copia separada.

•**Organización Lógica**: si el sistema operativo y el hardware del computador pueden tratar los programas de usuarios y los datos en la forma de módulos de algún tipo, entonces se pueden lograr varias ventajas:

- Los módulos se pueden escribir y compilar independientemente.

-Proporcionar diferentes grados de protección a los módulos (sólo lectura, sólo ejecución).

- Es posible introducir mecanismos por los cuales los módulos se pueden compartir entre los procesos.

•**Organización Física**: La memoria del computador se organiza en al menos dos niveles, conocidos como memoria principal y memoria secundaria. La memoria principal proporciona acceso rápido a un coste relativamente alto, aunque no proporciona almacenamiento permanente. La memoria secundaria es más lenta y más barata que la memoria principal, aunque proporciona almacenamiento para programas y datos a largo plazo.

*- ¿En qué consiste el particionamiento fijo de memoria? Explique las dos alternativas que existen.*

La memoria principal se divide en particiones estáticas en tiempo de generación del sistema. Un proceso se puede cargar en una partición con igual o superior tamaño.

Virtudes: sencilla de implementar, poca sobrecarga para el sistema operativo.

Defectos: uso ineficiente de la memoria, debido a la fragmentación interna; debe fijarse el número máximo de procesos activos.

Existen dos alternativas para el particionamiento fijo: una posibilidad consiste en hacer uso de particiones del mismo tamaño. En este caso, cualquier proceso cuyo tamaño es menor o igual que el tamaño de partición, puede cargarse en cualquier partición disponible; otra posibilidad es, si todas las particiones están llenas y no hay ningún proceso en estado Listo o ejecutando, el sistema operativo puede mandar a swap un proceso de cualquiera de las particiones y cargar otro proceso, de forma que el procesador tenga trabajo que realizar.

*- ¿Qué dificultades existen con el uso de particiones fijas del mismo tamaño?*

Un programa podría ser demasiado grande para caber en una partición. En este caso, el programador debe diseñar el programa con el uso de overlays. Super ineficiente, cualquier programa ocupa una partición entera.

*- Explique cómo funcionaría el algoritmo de ubicación en particiones fijas y en particiones de diferente tamaño.*

**Algoritmo de ubicación en particiones fijas**: como todos los tamaños de particiones son iguales lo manda al primero que encuentra. Si todas están usadas por procesos no listos, se saca uno y se mete el nuevo.

**En particiones de diferente tamaño**: hay dos formas posibles de asignar los procesos a las particiones. Meterlo en la partición más chica en la que entre, acá se necesita una cola para cada partición donde los procesos se destinan a una partición ya estando en disco (no óptima porque pueden quedar particiones sin utilizar). La otra forma más óptima es tener una sola cola para los procesos y en el momento de cargar el proceso se mete en la partición más pequeña disponible. Si no hay espacio se swappea (preferentemente) el proceso con el tamaño más aproximado al proceso que está por entrar.

*- ¿En qué consiste la técnica de particionamiento dinámico?*

Cantidad de particiones variable y su longitud también. Fragmentación externa compactación. La compactación consume mucho tiempo.

*- ¿Cuántos algoritmos de ubicación se consideran en el particionamiento dinámico? Explique cada uno de ellos.*

3 algoritmos:

Mejor-ajuste: carga el proceso ubicándolo en el espacio de memoria disponible que mejor se ajuste al tamaño del proceso.

Primer-ajuste: carga el proceso ubicándolo en el primer espacio de memoria disponible que encuentra.

Peor-ajuste: carga el proceso ubicándolo en el espacio de memoria de mayor tamaño que encuentra.

*- ¿Cuál es la diferencia entre dirección lógica y física?*

Una dirección lógica es una referencia a una ubicación de memoria independiente de la asignación actual de datos a la memoria; se debe llevar a cabo una traducción a una dirección física antes de que se alcance el acceso a la memoria.

Una dirección física, o dirección absoluta, es una ubicación real de la memoria principal.

*- Explique cómo funciona paginación de memoria.*

La memoria principal se divide en muchos marcos pequeños de igual tamaño. Cada proceso se divide en páginas de igual tamaño; los procesos más pequeños requieren menos páginas, los procesos mayores requieren más. Cuando un proceso se trae a la memoria, todas sus páginas se cargan en los marcos disponibles, y se crea una tabla de páginas.

*- Explique cómo funciona la segmentación de memoria.*

Cada proceso se divide en segmentos, donde cada segmento no requiere ser del mismo tamaño, pero tiene un tamaño máximo. La dirección lógica está compuesta por dos partes: número de segmento y desplazamiento. Cuando un proceso se trae a memoria, todos sus segmentos se cargan en regiones de memoria disponibles, y se crea una tabla de segmentos.

*- ¿Cuáles son las claves más importantes de los esquemas de paginación y segmentación sencilla?*

•Todas las referencias a la memoria dentro de un proceso se hacen con direcciones lógicas, que se traducen dinámicamente durante la ejecución.

•Un proceso se puede dividir en varias porciones (páginas/segmentos) y estas porciones no tienen que estar ubicadas contiguamente.

*- ¿Qué es el conjunto residente del proceso? ¿Qué beneficios trae gestionar la carga de procesos de esta manera?*

•Es el conjunto de instrucciones y datos que está cargado en memoria en un instante determinado.

Ventajas:

•Se puede tener una cantidad mayor de procesos en la memoria principal ya que no se carga el proceso completo.

•Un proceso puede ser más grande que toda la memoria. El programador puede hacer de cuenta que trabaja con una memoria gigante ya que es el SO el que se encarga de virtualizar la memoria y de buscar las páginas/segmentos que se requieran en un momento determinado.

*- ¿Para qué se utiliza la región de swap? ¿qué es el thrashing?*

El thrashing ocurre cuando el sistema ocupa la mayor parte del tiempo llevando y trayendo porciones de swap en lugar de ejecutar instrucciones.

*- ¿Qué es el principio de proximidad?*

Es el principio que indica que las referencias al programa y a los datos dentro de un proceso tienden a agruparse, lo que nos permite hacer suposiciones sobre cuáles son las porciones del proceso que se necesitarán en un futuro próximo (evita thrashing).

*- En un esquema de memoria virtual con paginación, ¿Cómo es la estructura de una tabla de páginas? ¿Para qué sirven los bits de control P y M?*

La tabla de páginas consiste de varias filas, donde el número de la fila indica el número de la página. Cada fila tiene los bits de control P y M, además de otros bits de control y el número de marco correspondiente a la página. El bit P es el bit de “presente” y el bit M es el de “modificado”.

*- ¿Dónde y cómo se almacena la tabla de páginas?*

Las tablas de páginas se almacenan en la memoria virtual (disco), por lo que éstas también están sujetas a paginación como cualquier página. Cuando se ejecuta un proceso, al menos un pedazo de su tabla tiene que estar en memoria, incluyendo la entrada de tabla de páginas de la página actualmente en ejecución.

*- ¿Qué es la paginación multinivel? ¿Por qué es útil?*

La paginación multinivel se logra teniendo tablas en diferentes niveles, formando así una jerarquía. Existe una tabla que es del nivel más alto, esta apunta a otras tablas de nivel inferior, estas de nivel inferior a otras más inferiores aún y así sucesivamente hasta llegar a las tablas que hacen referencia a los marcos. Esto permite ahorrar memoria porque si todas las referencias de una tabla son inválidas, directamente pones la entrada de la tabla superior que hace referencia a esta como inválida.

*- ¿Cómo funcionan las tablas de páginas invertidas?*

Contiene una entrada por cada marco de página (por lo que no va a ser dinámica la tabla). Cada entrada en la tabla tiene qué página está almacenada y un identificador del proceso al que pertenece la página. Como la tabla está organizada por marcos no se puede hacer una búsqueda directa, se debería buscar secuencialmente por lo que usualmente, para agilizar este proceso, la tabla se organiza como una tabla hash.

*- ¿Qué es la TLB? ¿Cuál es su utilidad? ¿Cómo funciona?*

La TLB (Translation Lookaside Buffer) es una caché especial de alta velocidad para entradas de la tabla de páginas. Esta caché funciona parecido a una caché normal y contiene aquellas entradas de las tablas de página que han sido usadas de forma más recientes.

El funcionamiento es el siguiente: dada una dirección lógica, el procesador primero busca en la TLB, si la entrada de la tabla de páginas buscada está ahí (hit), entonces se recupera el marco y se arma la dirección física. Si no se encuentra (miss), se fija en la tabla con el número de página dado. Si el bit P está en 1 entonces se calcula la dirección física y posteriormente se agregará en la TLB, pero si el bit está en 0 entonces la página no se encuentra en memoria por lo que se produce un fallo de acceso a memoria, llamado fallo de página.

*- ¿Qué es necesario considerar a la hora de definir el tamaño de página?*

Páginas grandes menos fragmentación interna; Páginas chicas más páginas para cada proceso (y por ende más tablas). Además usualmente se usan discos giratorios para la virtualización, y como este lee por sectores se favorece el tamaño grande de páginas.

*- ¿Qué implicancias tiene el esquema de segmentación de memoria?*

*1. ¿Qué implicancias tiene el esquema de segmentación de memoria?*

Implicancias del esquema de segmentación de memoria:

a) Simplifica el tratamiento de estructuras de datos que pueden crecer. Si el programador no conoce a priori el tamaño que una estructura de datos en particular puede alcanzar es necesario hacer una estimación salvo que se utilicen tamaños desegmento dinámicos. Con la memoria virtual segmentada, a una estructura de datos se le puede asignar su propio segmento, y el sistema operativo expandirá o reducirá el segmento bajo demanda. Si un segmento que necesita expandirse se encuentre en la memoria principal y no hay suficiente tamaño, el sistema operativo poder mover el segmento a un área de la memoria principal mayor, si se encuentra disponible, o enviarlo a swap. En este último caso el segmento al que se ha incrementado el tamaño volverá a la memoria principal en la siguiente oportunidad que tenga.

b) Permite programas que se modifican o recopilan de forma independiente, sin requerir que el conjunto completo de programas se re-enlacen y se vuelvan a cargar. De nuevo, esta posibilidad se puede articular por medio de la utilización de múltiples segmentos.

c) Da soporte a la compartición entre procesos. El programador puede situar un programa de utilidad o una tabla de datos que resulte útil en un segmento al que pueda hacerse referencia desde otros procesos.

*d) Soporta los mecanismos de protección. Esto es debido a que un segmento puede definirse para contener un conjunto de programas o datos bien descritos, el programador o el administrador de sistemas puede asignar privilegios de acceso de una forma apropiada.*

*- ¿Cómo se implementa?*

*En la exposición de la segmentación sencilla, indicamos que cada proceso tiene su propia tabla de segmentos, y que cuando todos estos segmentos se han cargado en la memoria principal, la tabla de segmentos del proceso se crea y se carga también en la memoria principal. Cada entrada de la tabla de segmentos contiene la dirección de comienzo del correspondiente segmento en la memoria principal, así como la longitud del mismo. El mismo mecanismo, una tabla segmentos, se necesita cuando se están tratando esquemas de memoria virtual basados en segmentación. De nuevo, lo habitual es que haya una única tabla de segmentos por cada uno de los procesos. En este caso sin embargo, las entradas en la tabla de segmentos son un poco más complejas. Debido a que sólo algunos de los segmentos del proceso pueden encontrarse en la memoria principal, se necesita un bit en cada entrada de la tabla de segmentos para indicar si el correspondiente segmento se encuentra presente en la memoria principal o no. Si indica que el segmento está en memoria, la entrada también debe incluir la dirección de comienzo y la longitud del mismo.*

*- ¿Qué es la Segmentación paginada?*

*Segmentación paginada: En un sistema combinado de paginación/segmentación, el espacio de direcciones del usuario se divide en un número de segmentos, a discreción del programador. Cada segmento es, por su parte, dividido en un número de páginas de tamaño fijo, que son del tamaño de los marcos de la memoria principal. Si un segmento tiene longitud inferior a una página, el segmento ocupará únicamente una página. Desde el punto de vista del programador, una dirección lógica sigue conteniendo un número de segmento y un desplazamiento dentro de dicho segmento. Desde el punto de vista del sistema, el desplazamiento dentro del segmento es visto como un número de página y un desplazamiento dentro de la página incluida en el segmento.*

*- ¿Qué diferencias hay entre paginación bajo demanda y prepaginación?*

Con paginación bajo demanda, una página se trae a memoria sólo cuando se hace referencia a una posición en dicha página. Si el resto de elementos en la política de gestión de la memoria funcionan correctamente, ocurriría lo siguiente. Cuando un proceso se arranca inicialmente, va a haber una ráfaga de fallos de página. Según se van trayendo más y más páginas a la memoria, el principio de proximidad sugiere que las futuras referencias se encontrarán en las páginas recientemente traídas. Así, después de un tiempo, la situación se estabilizará y el número de fallos de página caerá hasta un nivel muy bajo.

*Con paginación adelantada (prepaging), se traen a memoria también otras páginas, diferentes de la que ha causado el fallo de página. Si las páginas de un proceso se encuentran almacenadas en la memoria secundaria de forma contigua, es mucho más eficiente traer a la memoria un número de páginas contiguas de una vez, en lugar de traerlas una a una a lo largo de un periodo de tiempo más amplio. Por supuesto, esta política es ineficiente si la mayoría de las páginas que se han traído no se referencian a posteriori.*

*- ¿Es importante la ubicación en un esquema de paginación?*

*Para sistemas que usan o bien paginación pura o paginación combinada con segmentación, la ubicación es habitualmente irrelevante debido a que el hardware de traducción de direcciones y el hardware de acceso a la memoria principal pueden realizar sus funciones en cualquier combinación de página-marco con la misma eficiencia.*

*- ¿Por qué es necesario el bloqueo de marcos?*

*Bloqueo de marcos: Cuando un marco está bloqueado, la página actualmente almacenada en dicho marco no puede reemplazarse. Gran parte del núcleo del sistema operativo se almacena en marcos que están bloqueados, así como otras estructuras de control claves. Adicionalmente, los buffers de E/S y otras áreas de tipo critico también se ponen en marcos bloqueados en la memoria principal. El bloqueo se puede realizar asociando un bit de bloqueo a cada uno de los marcos. Este bit se puede almacenar en la tabla de marcos o también incluirse en la tabla de páginas actual.*

*- Compare los algoritmos básicos de reemplazo.*

Algoritmos básicos de reemplazo:

- La política óptima de selección tomará como reemplazo la página para la cual el instante de la siguiente referencia se encuentra más lejos. Esta política es imposible de implementar, porque requiere que el sistema operativo tenga un perfecto conocimiento de los eventos futuros. Sin embargo, se utiliza como un estándar a partir del cual contrastar algoritmos reales.

- La política de reemplazo de la página usada menos recientemente (LRU) seleccionará como candidata la página de memoria que no se haya referenciado desde hace más tiempo. Debido al principio de proximidad referenciada, esta página sería la que tiene menos probabilidad de volverá a tener referencias en un futuro próximo. El problema con esta alternativa es la dificultad en su implementación.

- La política FIFO trata los marcos de página ocupados como si se tratase de un buffer circular, y las páginas se remplazan mediante una estrategia cíclica de tipo round-robin. Todo lo que se necesita es un puntero que recorra de forma circular los marcos de página del proceso. Por tanto, se trata de una de las políticas de reemplazo más sencilla de implementar. El razonamiento tras este modelo, además de su simplicidad, es el reemplazo de la página que lleva en memoria más tiempo: una página traída a la memoria hace- mucho tiempo puede haber dejado de utilizarse. Este razonamiento a menudo es erróneo, debido a que es habitual que en los programas haya una zona del mismo o regiones de datos que son utilizados de forma intensiva durante todo el tiempo de vida del proceso. Esas páginas son expulsadas de la memoria y traídas de nuevo de forma repetida por un algoritmo de tipo FIFO.

- La política del reloj, da vueltas a través de todas las páginas del buffer buscando una que no se haya modificado desde que se ha traído y que no haya sido accedida recientemente. Esta página es una buena opción para reemplazo y tiene la ventaja que, debido a que no se ha modificado, no necesita escribirse de nuevo en la memoria secundaria. Si no se encuentra una página candidata en la primera vuelta, el algoritmo da una segunda vuelta al buffer, buscando una página modificada que no se haya accedido recientemente.