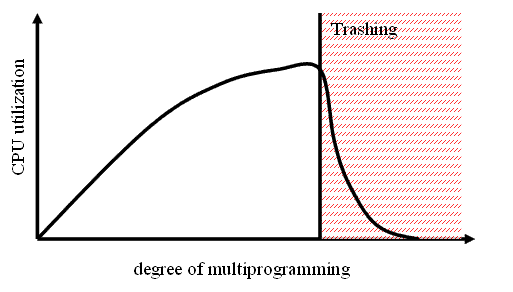
Hiper-paginación

Es un problema en la administración de memoria que ocurre cuando se utilizan demasiados recursos, y cada ver en aumento, para realizar una tarea demasiado pequeña o que va necesitando menos.

# Causas de hiper-paginación

Cuando el uso del procesador es muy bajo, el sistema operativo introduce otro proceso nuevo para aumentar la multiprogramación. Si entra un proceso nuevo, que no tiene suficientes marcos para ejecutarse, provocará un error en la memoria y entonces tomará prestados marcos a otro proceso. Pero ese proceso necesitaba los marcos que tenía asignados, por lo que también generará un fallo y seguirá repitiendo el ciclo de esa manera. Cuando esos procesos se finalizan, entran a la cola de realizados, entonces el procesador se libera; y al liberarse, el sistema operativo vuelve a introducir un proceso nuevo.

El proceso recién añadido vuelve a empezar todo el ciclo. Este ciclo provoca que los procesos tarden mucho más a entrar en memoria y que el procesador baje mucho su rendimiento.



En la imagen se muestra cómo aumenta el uso del procesador incrementa junto al número de procesos en ejecución. Pero en cierto momento el uso del procesador se estanca en un límite máximo y decae mucho hasta no ser usado. El problema es que siguen habiendo muchos procesos en ejecución, por lo que no tienen recursos suficientes para operar. Ese estado (marcado en rojo) es el de hiper-paginación.

# Cómo prevenir la hiper-paginación

Para prevenir la sobrepaginación, es necesario darle a los procesos la cantidad de marcos que necesiten. Una forma de saber la cantidad adecuada de marcos es utilizar la Working-Set o de Conjunto de Trabajo. Funciona examinando cuántos marcos está utilizando un proceso realmente; esta técnica se basa en el concepto de localidad de ejecución del proceso.

La localidad se refiere a un conjunto de páginas que se utilizan activamente de forma combinada. Todo programa está compuesto de un conjunto de localidades distintas, que pueden traslaparse unas con otras, y a medida que un proceso se ejecuta, se va desplazando de una localidad a otra.

Este modelo usa un parámetro numérico que define la ventana del conjunto de trabajo activo (localidad). Este parámetro indica las páginas que están siendo usadas, la idea es examinar el número de referencias más recientes a las páginas. Si una página está activa, se encontrará dentro del conjunto de trabajo, mientras que si la página ya no es utilizada, será eliminada de éste número de unidades de tiempo después de su última referencia.

De esta manera, el conjunto de trabajo es una aproximación a la localidad del programa. El sistema operativo supervisa el conjunto de trabajo de cada proceso y le asigna marcos suficientes para proporcionarle el tamaño del conjunto activo. Si hay suficientes marcos iniciales, se puede iniciar otro proceso. Si aumenta la suma de los tamaños de los conjuntos activos, excediendo el número total de marcos disponibles, el sistema operativo selecciona un proceso y lo suspende.

Si la demanda es superior a la cantidad de marcos disponibles se produce hiper-paginación y en este caso el sistema operativo suspende un proceso.

El sistema operativo es quien también monitoriza los conjuntos de trabajo de cada proceso y le asigna los suficientes marcos para que no se produzca hiper-paginación.

# Herramientas para detectar la hiper-paginación

En sistemas Windows

El monitor de recursos incluido en Windows permite ver todos los procesos en ejecución y la carga de trabajo que representan para el sistema.

Otra herramienta es el Process Explorer de Microsoft. Permite ver todos los procesos en ejecución, mostrando la cantidad de hilos, tareas de escritura/lectura, entradas/salidas e incidencias.

En Linux

En Linux existe la particularidad de que el gestor “básico” de procesos permite ver la cantidad de hilos en ejecución para cada programa y monitorear por completo su influencia en el sistema. Ello, junto a las propiedades de control completo sobre la incidencia de los procesos en el kernel, hacen que no sean necesarias herramientas externas más allá del uso del sistema.

Aun así, existen algunas herramientas especializadas como **thrash-protect** o **clean-thrash**.

Resumen capítulo 3

# Entrada/salida

Una de las principales funciones del sistema operativo es gestionar las entradas y salidas en el equipo sobre el que opera. Independientemente del dispositivo que se tenga que gestionar, el sistema debe ser capaz de poder hacerlo sin demasiada dificultad.

En hardware, los dispositivos de E/S se pueden dividir en dos categorías:

* **Por bloques:** almacena información en bloques de tamaño fijo. Esto permite leer o escribir la información en cada bloque y que sea independiente de los otros bloques.
* **Por caracteres:** envía o recibe una cadena de información sin separarla en bloques. Eso significa que no hay una forma específica de buscar las cosas y que la información debe pasar toda hasta llegar al punto que interesa.

La parte electrónica que hace funcionar a un dispositivo se llama controlador de dispositivo. Las partes mecánicas necesarias para que funcione son el dispositivo en sí mismo. El sistema se encarga de operar el controlador. Lo hace mediante ciertos comandos que el dispositivo acepta según sus especificaciones propias.

## Acceso directo a memoria (DMA)

Permite que el controlador no tenga que ejecutar una acción de entrada/salida a la vez. Ordena al dispositivo qué orden debe ejecutar, con cuánta memoria y en qué dirección de memoria debe ubicarse para hacerlo. Además, esta información se almacena en un buffer de memoria para evitar que las operaciones atasquen y saturen la entrada a la memoria principal del sistema.

A veces no se puede transferir un bloque de orden de una sola vez a través del bus principal. Cuando eso sucede se debe saltar un bloque y luego seguir dos bloques adelante. A ese proceso se le llama intercalación.

# Software de E/S

El principal objetivo del software de E/S es que el sistema operativo pueda dar órdenes a los dispositivos utilizando un lenguaje común sin tener que cambiarlo para cada caso específico. Otro objetivo importante es que si hay errores en un dispositivo, sea su controlador quien lo resuelva sin afectar al sistema principal. También debe poder decidir si las transferencias se hacen de forma sincrónica o si son controladas por interrupciones.

El software de E/S se compone de cuatro capas:

1. **Manejadores de interrupciones (capa inferior):** es el encargado de controlar los errores en los dispositivos y de mantenerlos a nivel tan bajo como sea posible.
2. **Controladores de dispositivos de software:** es todo el software que depende del dispositivo para ejecutarse y también aquel sin el cual el dispositivo no puede interactuar con el sistema operativo. Debe poder aceptar peticiones abstractas del sistema operativo y ver que el dispositivo las cumpla.
3. **Software de sistema independiente al dispositivo:** es el encargado de realizar las funciones de E/S comunes a todos los dispositivos y de traducir las instrucciones a una interfaz uniforme al software de nivel de usuario.
4. Software de usuario (capa superior): son todos los procedimientos de bibliotecas especializadas que ejecuta el usuario desde aplicaciones que no están en el núcleo del sistema.

## Bloqueo mutuo

Se genera cuando un proceso solicita cierto recurso que otro proceso tiene ocupado, mientras que ese segundo proceso también espera otro recurso que el primero tiene ocupado. Ambos necesitan un recurso que el otro tiene, por lo que ambos se bloquean; y como están bloqueados, ninguno libera el recurso que el otro necesita. Permanecerán en espera para siempre porque ninguno liberará lo que el otro necesita.

Un conjunto de procesos está en bloqueo mutuo si cada proceso del conjunto está esperando un evento que sólo otro proceso del conjunto puede causar.

### Recursos

Existen dos tipos de recursos:

* **Expropiables:** aquellos que se pueden arrebatar a un proceso sin que hayan efectos negativos.
* **No expropiables:** aquellos que no se pueden quitar a un proceso sin que la ejecución de ese proceso se detenga.

Para que un proceso haga uso de los recursos se debe:

1. Solicitar el recurso.
2. Usar el recurso.
3. Liberar el recurso.

Para que haya un bloqueo mutuo se deben cumplir las siguientes condiciones:

1. **Exclusión mutua:** cada recurso está asignado únicamente a un proceso.
2. **Retener y esperar:** Los procesos que ya tienen recursos pueden solicitar más.
3. **No expropiación:** No se pueden liberar recursos a la fuerza.
4. **Espera circular:** Hay una cadena de procesos en espera de que otros liberen recursos.

Hay cuatro estrategias para manejar el bloqueo mutuo:

1. **Ignorar el problema (algoritmo de avestruz):** se actúa como si el problema no existiera. Algunos sistemas prefieren ignorar el bloqueo mutuo y esperar a que pase antes de restringir recursos a otros procesos.
2. **Detección y recuperación:** se realiza revisando constantemente que no existan bloqueos. Cada vez que un recurso se solicita o se libera, se actualiza la información relevante para que los demás puedan solicitar o liberar de manera adecuada.
3. **Evitar que suceda, asignando bien los recursos:** funciona imponiendo restricciones y políticas sobre la forma en que los procesos solicitan y liberan recursos. Eso evita que se cumplan las condiciones para bloqueos mutuos.
4. **Evitar mediante algoritmos especializados:** Se analiza cada situación en la que se podría dar un bloqueo para comprobar cuál es la mejor forma de operar en cada caso. Algunos de los algoritmos son:
   1. **Algoritmo del banquero:** consiste en evaluar una petición cada vez que se realiza y evaluar si cumplirla conduce o no a un estado seguro del sistema en el que no se producirá un bloqueo mutuo. Si conduce a un estado seguro, se le otorga; si no, se pone en una lista de espera.
   2. **Trayectorias de recursos:** se establecen trayectorias de ejecución para cada proceso. En base a esas trayectorias se establece un orden para brindar recursos. Las trayectorias son modeladas de tal manera que ambas puedan llegar al punto final de su ejecución sin que se intercalen en ningún momento.
   3. **Algoritmo del banquero con recursos de diversos tipos:** consiste en planificar el uso de recursos por adelantado para cada proceso. Se crea una matriz de comparación para saber qué recursos necesita cada proceso y de cuántos dispone el sistema. Todas las matrices se superponen y se comparan para calcular la diferencia entre lo que todos solicitan y el balance total de recursos disponibles. Se crea una nueva matriz que establece qué proceso utilizará qué recurso en el orden necesario para que sean liberados adecuadamente.

# Discos en RAM