Los programas y datos deben estar en memoria principal para poder poderlos ejecutar y referenciarlos directamente.

El SO debe:

* Llevar un registro de las partes de memoria que se están utilizando y de aquellas que no
* Asignar espacio en memoria principal a los procesos cuando estos la necesitan
* Liberar espacio de memoria asignada a procesos que han terminado

Ademas debe

* Lograr que el programador se abstraiga de la alocacion de los programas
* Brindar seguridad entre procesos para que no accedan a donde no deben
* Brindar acceso compartido a determinadas secciones como librerías, código común, etc.
* Garantizar la perfomance del sistema.

**Administración de memoria**

Es de los factores más importantes del diseño de los SO. El objetivo del diseño de la administración de la memoria es abstraer al programador, brindar seguridad, brindar acceso compartido, garantizar la performance del sistema, etc.

Se deben tener en cuenta algunos requisitos:

**Reubicación**: Mientras un proceso se ejecuta, puede ser sacado y traído a la memoria (swap) y, posiblemente, colocarse en diferentes direcciones.

**Protección**: los procesos NO deben referenciar (acceder) a direcciones de memoria de otros procesos (salvo que tengan permiso).

**Compartición**: es para que 2 procesos puedan compartir determinada parte de memoria (como por ejemplo rutinas comunes, librerías, espacios explícitamente compartidos, etc).

Las direcciones de memoria pueden ser: Lógicas (son las que utilizan los procesos, es independiente al lugar de la RAM donde se almacena) o Físicas (espacio físico de la memoria, es la dirección absoluta). Si se usan direcciones lógicas, se necesita una conversión, llamada resolución de direcciones (address-binding), que generalmente se hace en tiempo de ejecución. Esta resolución es realizada por el **MMU** que es un dispositivo HW que mapea direcciones virtuales a físicas a través de un registro base y límite.

Mecanismos de asignación de memoria

Con el fin de que varios procesos usen la memoria, se realizan particiones a las mismas. Estas pueden ser Fijas si se realizan de antemano o Dinámicas si se realizan a medida que llegan los procesos. La desventaja de las fijas es que puede ocurrir FI (Fragmentación Interna) y en las dinámicas puede ocurrir FE (Fragmentación Externa). FI es que quede memoria libre dentro de la partición, y FE es que quede memoria libre entre particiones.

Existen diferentes **técnicas de administración de memoria**:

Paginación: el espacio de los procesos se divide en partes iguales llamadas páginas. La memoria RAM se divide en partes iguales llamadas marcos. Cualquiera de las páginas puede ocupar cualquiera de los marcos. Existe una tabla de páginas para organizar, que indica en qué marco está cada página y sirve para que el MMU pueda traducir. Cada proceso tiene su tabla de páginas y sirve para proteger la memoria. Provoca FI ya que como el tamaño de las páginas es fijo y, en general, el último bloque de datos o instrucciones de un proceso no coincide exactamente con el tamaño de la página.

La principal ventaja de la paginación es evitar los problemas de la fragmentación.

Segmentación: cada parte de un espacio de direcciones (librerías, código, datos, pilas) se le llama segmento y estos se pueden cargar en la memoria de manera no continua. Se tiene una tabla de segmentos, que indica dirección base y límite del mismo. Puede provocar FE, pero es más sencillo respetar los principios de compartir y proteger la memoria.

En ambas no existe continuidad. (refiere a que el espacio de direcciones lógicas (el que usa el proceso) no tiene que corresponderse de manera directa y continua con el espacio de direcciones físicas (la memoria RAM) en el que se almacenan los datos o instrucciones).

Segmentación paginada: Se combinan ambas, primero se segmenta y después se página. Para cada segmento hay una tabla de páginas. Si se comparte un segmento, se comparten todas sus páginas.