

## Hardware básico

**Acceso a la memoria:** La CPU accede a la memoria y los registros para ejecutar instrucciones

## Hardware básico

**Memoria caché:** Importancia de la memoria caché para mejorar la velocidad de acceso y la eficiencia.

## Hardware básico

**Protección del espacio de memoria:** Es esencial evitar accesos no autorizados y asegurar que cada proceso tenga su propio espacio de memoria.

**Registros base y límite:** Se utilizan para implementar la protección de memoria, asegurando que cada proceso opere dentro de sus límites designados.

## Espacios de direcciones lógico y físico

**Traducción dinámica:** Cada referencia a la memoria se traduce en tiempo de ejecución, permitiendo que el programa opere como si estuviera en una dirección lógica fija.

## Carga dinámica

La carga dinámica permite que un programa pueda cargar un módulo durante su ejecución.

**Carga bajo demanda:** Los módulos se cargan cuando se necesitan y se pueden descargar para liberar memoria.

## Montaje dinámico y bibliotecas compartidas

**Bibliotecas compartidas:** Son módulos que pueden ser utilizados por múltiples programas simultáneamente.

**Montaje dinámico:** Es el proceso de vincular las bibliotecas compartidas con el programa durante la ejecución.

## MEMORIA PRINCIPAL

## Reasignación de direcciones

**Reubicación de direcciones:** Las direcciones simbólicas del programa se reasignan a direcciones relativas o absolutas durante la compilación y carga.

**Tiempo de carga:** Si la ubicación en memoria no se conoce en tiempo de compilación, el compilador debe generar código relocable, que se ajusta en el momento de la carga.

**Tiempo de ejecución:** Si el proceso puede moverse durante su ejecución, la reasignación se realiza en tiempo de ejecución utilizando hardware especial.

### Carga de programas en memoria:

**Cola de entrada:** Los procesos en disco que esperan ser cargados en memoria forman una cola de entrada.

**Registros base y límite:** Se utilizan para proteger el espacio de direcciones de memoria y asegurar que los procesos operen dentro de sus límites designados.

**Tiempo de compilación:** Si se conoce la ubicación en memoria del proceso durante la compilación, se puede generar código absoluto. Sin embargo, si el proceso se reubica, se requiere una recompilación del código.

