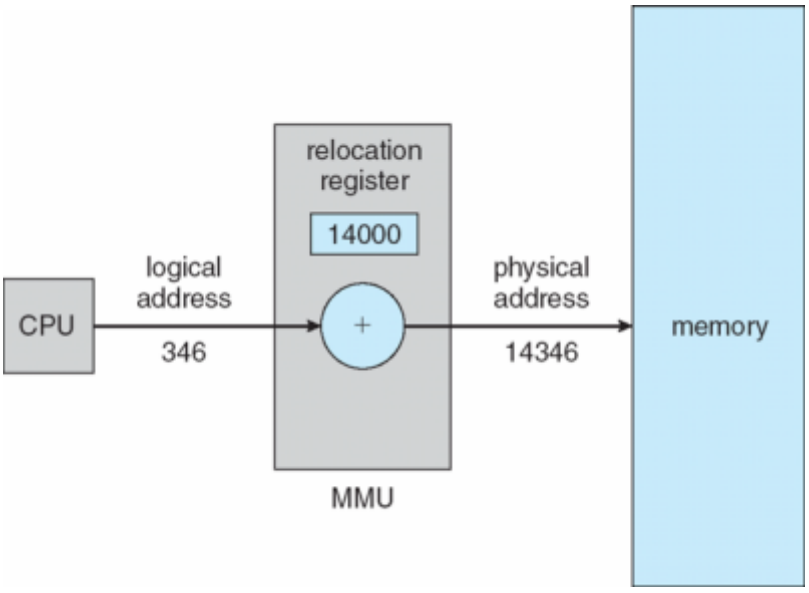
Espacios de direcciones lógico y físico

[1)] Una dirección generada por la CPU (proceso) se denomina comúnmente **dirección lógica**, mientras que una dirección vista por la unidad de memoria (es decir, la que se carga en el registro de direcciones de memoria de la memoria) se denomina comúnmente **dirección física**.

Los métodos de reasignación en tiempo de compilación y en tiempo de carga generan direcciones lógicas y físicas idénticas. Sin embargo, el esquema de reasignación de direcciones en tiempo de ejecución hace que las direcciones lógica y física difieran. En este caso, usualmente decimos que la dirección lógica es una dirección virtual. utilizaremos los términos dirección lógica y dirección virtual de manera intercambiable.

El conjunto de todas las direcciones lógicas generadas por un programa es lo que se denomina un espacio de direcciones lógicas; el conjunto de todas las direcciones físicas correspondientes a estas direcciones lógicas es un espacio de direcciones físicas. Así, en el esquema de reasignación de direcciones en tiempo de ejecución, decimos que los espacios de direcciones lógicas y físicas difieren.

La correspondencia entre direcciones virtuales y físicas en tiempo de ejecución es establecida por un dispositivo hardware que se denomina unidad de gestión de memoria (**MMU, memory-management unit**). Podemos seleccionar entre varios métodos distintos para establecer esta correspondencia. Por el momento, vamos a ilustrar esta operación de asociación mediante un esquema MMU simple, que es una generalización del esquema de registro base. El registro base se denominará ahora registro de reubicación. El valor contenido en el registro de reubicación suma a todas las direcciones generadas por un proceso de usuario en el momento de enviarlas a memoria.



Por ejemplo, si la base se encuentra en la dirección 14000, cualquier intento del usuario de direccionar la posición de memoria cero se reubicará dinámicamente en la dirección 14000; un acceso a la ubicación 346 se convertirá en la ubicación 14346. El sistema operativo MS-DOS que se ejecutaba sobre la familia de procesadores Intel 80×86 utilizaba cuatro registros de reubicación a la hora de cargar y ejecutar procesos.

El programa de usuario nunca ve las direcciones físicas reales. El programa puede crear un puntero a la ubicación 346, almacenarlo en memoria, manipularlo y compararlo con otras direcciones, siempre

como el número 346. Sólo cuando se lo utiliza como dirección de memoria (por ejemplo, en una operación de lectura o escritura indirecta) se producirá la reubicación en relación con el registro base. El programa de usuario maneja direcciones lógicas y el hardware de conversión (mapeo) de memoria convierte esas direcciones lógicas en direcciones físicas. La ubicación final de una dirección de memoria referenciada no se determina hasta que se realiza esa referencia.

Ahora tenemos dos tipos diferentes de direcciones: direcciones lógicas (en el rango comprendido entre 0 y max) y direcciones físicas (en el rango comprendido entre R+0 y R+max para un valor base igual a R). El usuario sólo genera direcciones lógicas y piensa que el proceso se ejecuta en las ubicaciones comprendidas entre 0 y max. El programa de usuario suministra direcciones lógicas y estas direcciones lógicas deben ser convertidas en direcciones físicas antes de utilizarlas.

El concepto de un espacio de direcciones lógicas que se acopla a un espacio de direcciones físicas separado resulta crucial para una adecuada gestión de la memoria. Prueba

Carga dinámica

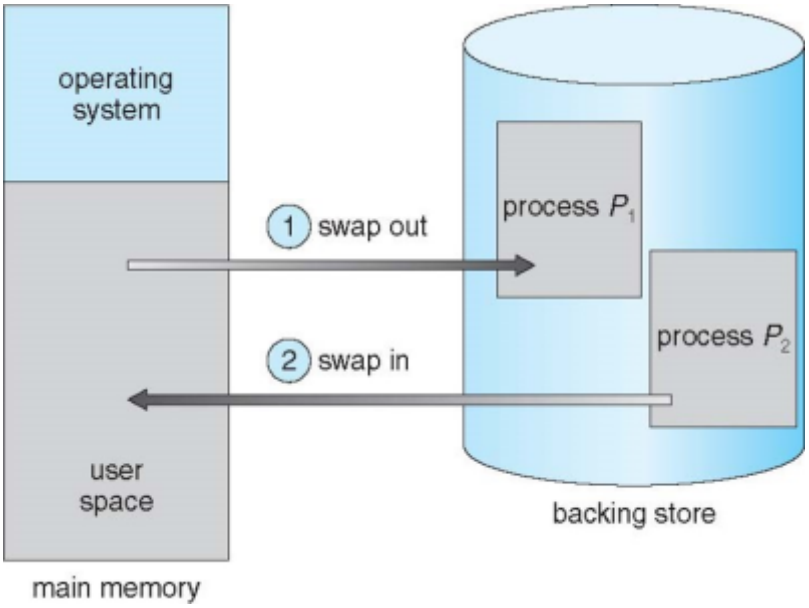
Hasta ahora, todo el programa y todos los datos de un proceso deben encontrase en memoria física para que ese proceso pueda ejecutarse. En consecuen¬cia, el tamaño de un proceso está limitado por el tamaño de la memoria física. Para obtener una mejor utilización del espacio de memoria, podemos utilizar un mecanismo de carga dinámica. Con la carga dinámica, una rutina no se carga hasta que se la invoca; todas las rutinas se mantienen en disco en un formato de carga reubicable.

Según este método, el programa principal se carga en la memoria y se ejecuta. Cuando una rutina necesita llamar a otra rutina, la rutina que realiza la invocación comprueba primero si la otra ya ha sido cargada, si no es así, se invoca el cargador de montaje reubicable para que cargue en memoria la rutina deseada y para que **actualice las tablas de direcciones** del programa con el fin de reflejar este cambio. Después, se pasa el control a la rutina recién cargada.

Swapping

Un proceso debe estar en memoria para ser ejecutado. Sin embargo, los procesos pueden ser intercambiados temporalmente, sacándolos de la memoria y almacenándolos en el backing store y volviéndolos a llevar luego a memoria para continuar su ejecución.

Por ejemplo, supongamos que estamos utilizando un entorno de multiprogramación con un algoritmo de planificación de CPU basado en turnos. Cuando termina un quantum de tiempo, el gestor de memoria comienza a sacar de ésta el proceso que acaba de terminar y a cargar en el espacio de memoria liberado por otro proceso.



Mientras tanto, el planificador de la CPU asignará un quantum de tiempo a algún otro proceso que ya se encuentre en memoria. Cada vez que un proceso termine su quantum asignado, se intercambiará por otro proceso. Idealmente, el gestor de memoria puede intercambiar los procesos con la suficiente rapidez como para que haya siempre algunos procesos en memoria listos para ejecutarse, cuando el planificador de la CPU quiera asignar el procesador a otra tarea. Además, el quantum debe ser lo suficientemente grande como para que pueda realizarse una cantidad razonable de cálculos entre una operación de intercambios y la siguiente.

Los mecanismos de intercambio requieren un almacén de respaldo, que normalmente será un disco suficientemente rápido. El disco debe ser también lo suficientemente grande como para poder albergar copias de todas las imágenes de memoria para todos los usuarios, y debe proporcionar un acceso directo a estas imágenes de memoria.

El sistema mantiene una cola de procesos preparados que consistirá en todos los procesos cuyas imágenes de memoria se encuentren en el backing store o en la memoria y estén listos para ejecutarse. Cada vez que el planificador de la CPU decide ejecutar un proceso, llama al despachador, que mira a ver si el siguiente proceso de la cola se encuentra en memoria. Si no es así, y si no hay ninguna región de memoria libre, el despachador intercambia el proceso deseado por otro proceso que esté actualmente en memoria. A continuación, recarga los registros y transfiere el control al proceso seleccionado.

Volver

(505)

1)

Extraido y resumido de Fundamentos de Sistemas Operativos - Silberschtaz

From:
http://wiki.educabit.ar/ - **Wiki Sistemas**

Permanent link:
http://wiki.educabit.ar/doku.php?id=so_memdirecciones&rev=1758497844

Last update: **2025/09/21 20:37**