

Sistemas Operativos

Cursada 2022

Comisión S21 y S22

Memoria Virtual

La **memoria virtual** es la técnica por la cual la ejecución de un proceso, **NO** requiere que el mismo este **totalmente en memoria**.

La **ventaja** mas visible es que ahora no importa el **tamaño del proceso**, el programador se desentiende de este problema.

La **desventaja** principal de esto es que tiene que estar bien **administrada** para que la performance no se vea alterada en demasía.

Memoria Virtual

- La **memoria virtual** como concepto no es nuevo, sino que se empezó a hablar en la **década del '70**.
- Ya en la década del **'80** los equipos empezaron a tener cada vez mas **memoria**, sobre todo los **mainframe**.
- Nace **UNIX** con una tecnología de **32bits**.
- Ya en los **'90** el desarrollo del software crece drásticamente en todos los ámbitos.

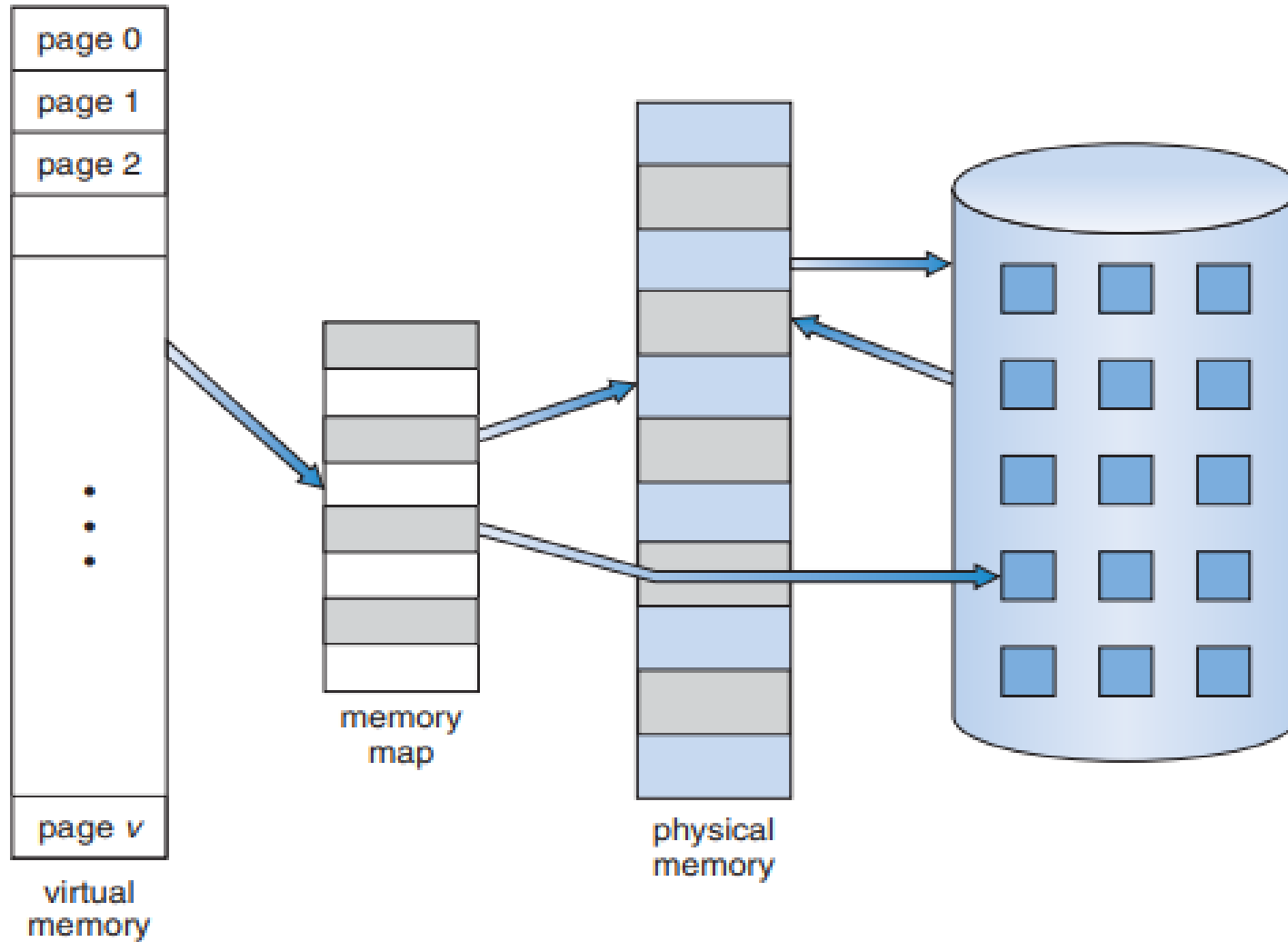
Memoria Virtual

- Unix en esa época explota mucho los procesos cooperativos, había poca cantidad de memoria.
- Década del '80 y cerca de los '90 los sistemas eran un conjunto de programas aislados.
- Muchas veces bajo un '**Main principal**' que en realidad era un **Menú**.
- Esta forma de trabajo tenía algunas consecuencias que eran de mantenimiento del sistema.

Memoria Virtual

- Como ya había mucho mas memoria se pensó en otra forma de trabajo.
- Aparecieron los paquetes de software
- El sistema era un solo **Programa Ejecutable** que contenían todas las rutinas.
- Todo ese programa se cargada en la memoria
- Ahora todo este gran programa se usaba por completo?
- De aquí que **surgió ?**

Memoria Virtual



Memoria Virtual

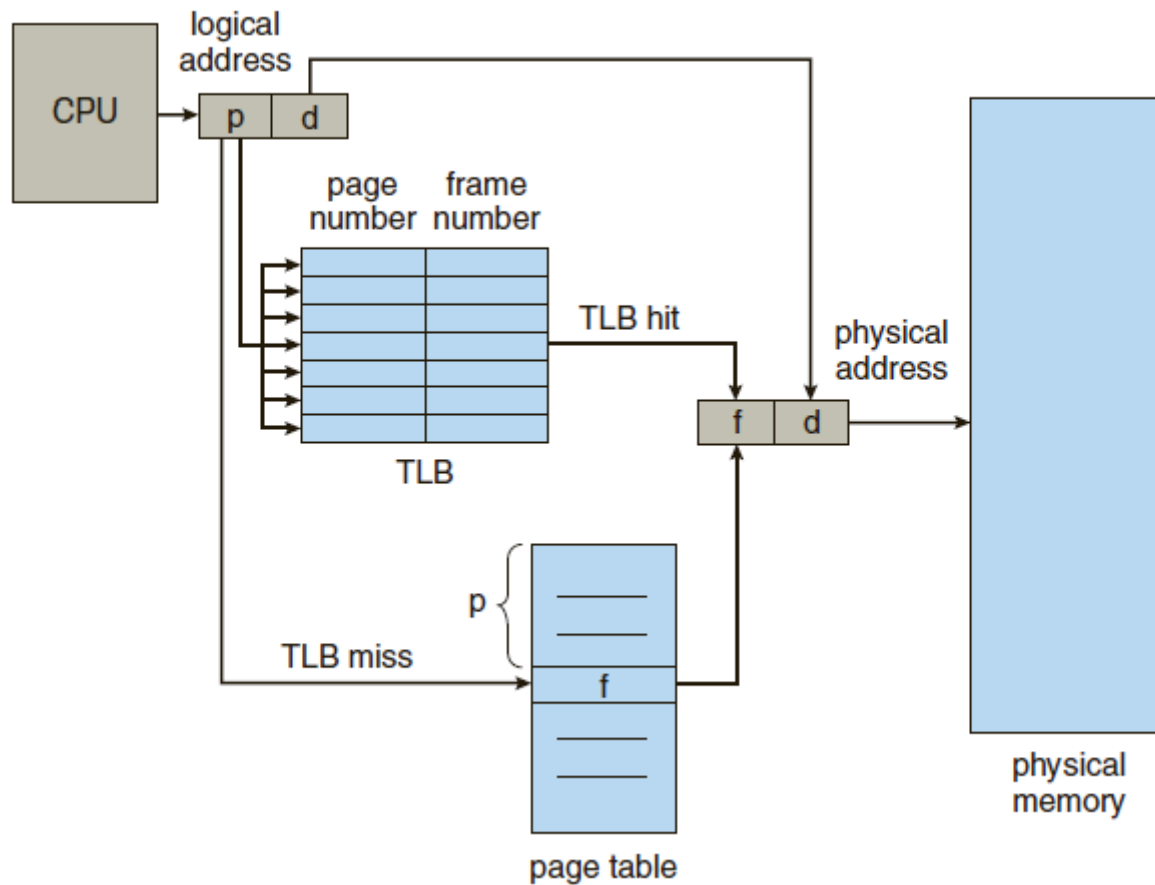
Esto tiene enormes ventajas:

- Me permite tener muchos procesos en ejecución.
- Además me permite ejecutar un proceso rápidamente.
- Esto porque cargo solo lo que necesito.
- El resto esta en la memoria virtual (Disco)
- A medida que va requiriendo el sistema se piden al disco, esto se llama:

Paginación bajo Demanda

Memoria Virtual

Vamos a usar como ejemplo este grafico ya visto



Memoria Virtual

Ahora como sabemos si la pagina esta o no en memoria:

Nro de Pagina	Dirección Marco	Bit de valides
P0	2048	1
P1	1024	1
P2	4096	0
P3	3072	0
P4	6144	0
P5	9216	1
P6	15360	0
P7	8192	1

Memoria Virtual

Si el **BV** esta en **0** se produce lo que se llama:

Fallo de Pagina

Por lo tanto se activa la rutina que atiende los fallos de pagina.

Esta deberá buscar en la pila de marcos libres cuales están disponibles

**La Dirección lógica
ahora**

Dirección Virtual

M4
M45
M20
M7

Memoria Virtual

Que pasa si esta Pila esta vacía:



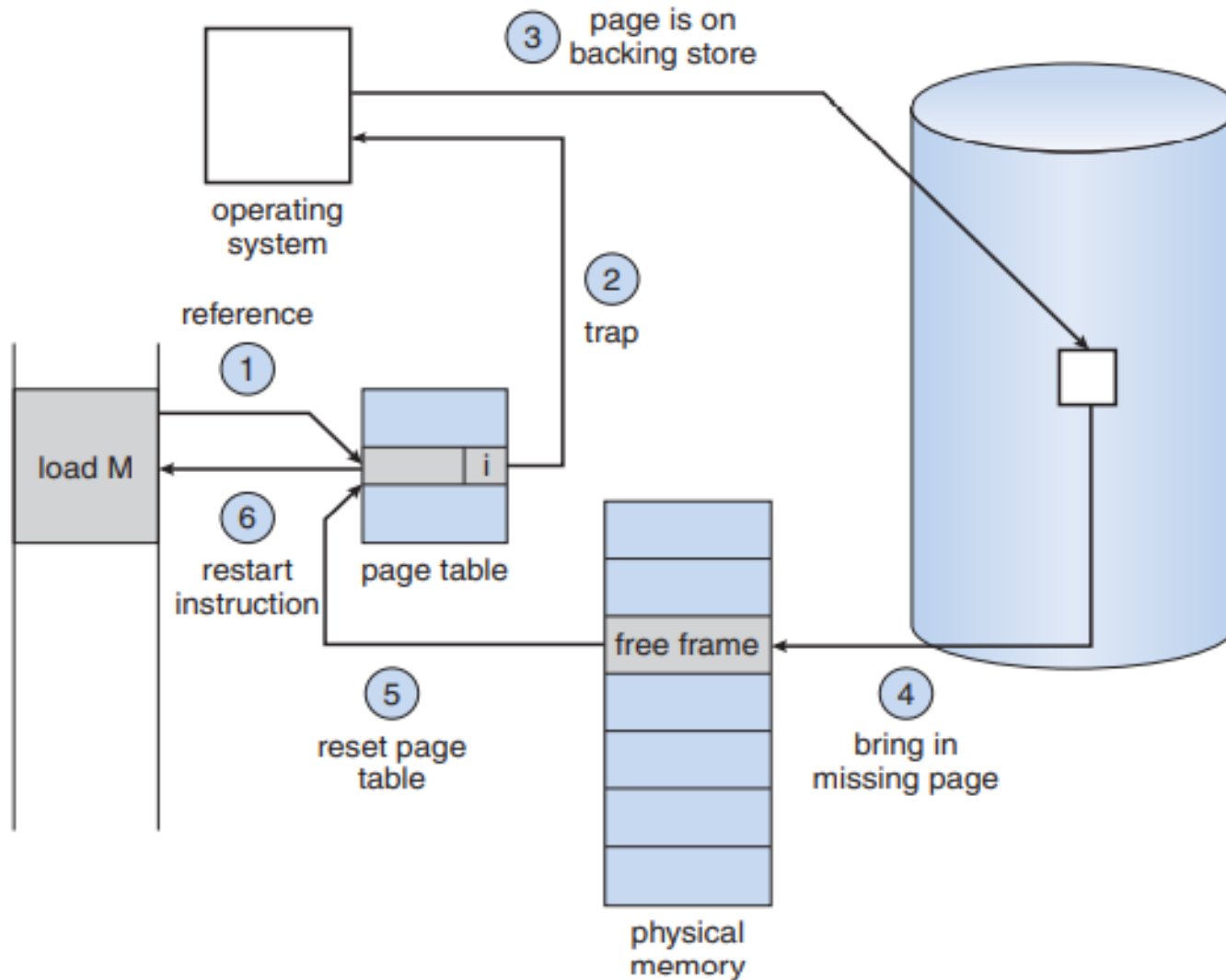
No se pueden cargar mas procesos, por lo tanto deberíamos sacrificar una pagina de las que están en memoria, a esta pagina se le llama:

Pagina Victima

La idea es tener la menor cantidad de fallos de pagina. Para esto los SO toman las medidas necesaria para que los fallos no sean recurrentes. La programación estructurada y modelo de objetos bien resueltos evitan tener muchos fallos de pagina.

Memoria Virtual

Concepto básico del intercambiador de paginas



Memoria Virtual

En M.V. cuando nace un proceso carga pocas paginas en la memoria real. En realidad habíamos visto el concepto de **Paginación bajo Demanda** (Cargaría la inicial), pero este no es el adecuado. (9.2.2 libro Rendimiento de paginas bajo demanda)

Copia Durante la Escritura

- **Habíamos dicho que podíamos cargar solo una**
- **Que pasa cuando se ejecuta un fork(), el padre crea un hijo**

Memoria Virtual

Algoritmos de selección de Pagina Victima

FIFO

Optimo

LRU

Segunda Chance o Reloj

Fin del Tema