

Practica 1 - Paginación

- La unidad de paginación se encarga principalmente de traducir una dirección virtual en una física.

→ Esta traducción permite separar el espacio de direcciones expuestas al proceso de su ubicación en la memoria física.

→ ¿Por qué
queremos
esto?

Para poder separar el espacio de distintos procesos que ejecutan en el mismo procesador y por ende comparten una memoria principal

- Con esto cada proceso puede manejarse sobre su espacio virtual como si fuera el físico. Luego el sistema operativo puede administrar la memoria física como sea necesario, de forma dinámica y sin afectar la ejecución de cada proceso.

Le unidad va a utilizar dos elementos para la traducción

→ ① La dirección virtual, que es la que conoce el proceso.

→ ② La estructura de paginación (directorio y tablas de páginas) del proceso.

• La ubicación del directorio de páginas se encuentra en CR3 (registro de control)

• Dirección virtual

- 10 bits más altos : page directory index
- 10 bits siguientes : page table index
- 12 bits más bajos : offset desde la base de la página donde está el dato esperado.

CR3

- Desde bit 12 a 31 : dirección page directory] 20 bits + altos
 - Bit 3 : PWT
 - Bit 4 : PCD
 - Bit 0 : Present
- CR3 & FFFFFFF000
consecutivos
20 bits

page directory entry

• Hay 1024 entradas, c/u de 32 bits (4 bytes) el directorio entero ocupa 4KB (una página)

20 bits más altos : dirección de la page table
0: Present

NOTA Bit 1: R/W

2: U/S

pd[31] & 0xFFFFF000

Page Table Entry

Bit 0: Present

1: B/W

2: U/S

20 bits mas altos direc de la pag

$pte_i \& 0xFFFFFFFF000$

$virt = dir(10\text{ bits}) \mid table(10\text{ bits}) \mid offset(12\text{ bits})$

$pd := CR3 \& FFFFFFF000$

$pt_index = (virt \gg 22) \& 0x3FF$

$pt := pd \ll pt_index \& FFFFFFF000$

$pt_index := (virt \gg 12) \& 0x3FF$

$page_addr := pt \ll pt_index \& FFFFFFF000$

$offset := virt \& 0xFFF$

$phys := page_addr \mid offset$

tlbflush cada vez que cambiamos la estructura de ~~pag~~. Intercambia CR3 con un valor temp y lo restaura.

TLB: translation lookaside buffer cuenta con reducciones pre-computadas

Casos de uso

① Swapping

- Cuando varios procesos comparten el uso de memoria y esta se está agotando o un proceso se encuentra suspendido, el sistema operativo puede decidir mover páginas de estos procesos desde y hacia el disco sin afectar la visión del proceso de su espacio de memoria.

↙ Para implementar efectivamente este mecanismo el OS puede registrar internamente

↳ Frecuencia y prioridad de los procesos. Para decidir cuáles van o vuelven del disco.

② Carga bajo demanda

- El OS puede decidir no cargar páginas de un proceso, poner su bit de P en 0, cuando tire page fault, en esa excepción busca la pag en disco y le mapea, marcando el bit P en 1.

↓
Le trae a memoria

③ Bibliotecas Compartidas

- Son utilidades que el OS expone a uno o varios procesos, entonces como varios procesos acceden al mismo código no lo copiamos, mapeamos las mismas páginas (de la biblioteca compartida) en distintas estructuras por proceso.

④ Copy on Write

- Actualmente es común el uso del Fork (crear nuevos procesos a partir de la copia del proceso en ejecución)

↳ mapeamos varios procesos contra el mismo conjunto de páginas físicas que contienen su código.

↳ solo las duplicamos si se escribe en ellas y mas de un proceso las esta accediendo.

- las marcamos como solo lectura, si salta la excepción por querer escribirla se copia y se mapea con escritura para el proceso que generó la interrupción