

- Los procesos se ponen en memoria
- La imagen contiene datos y memoria



n-1

Tipos de memoria

- ROM → Read Only Memory
- RAM → Random access memory → se activa cualquier posición
- Nel* RAM → Read Write Memory
- nombre
- SRAM → Static RAM

Memoria RAM:

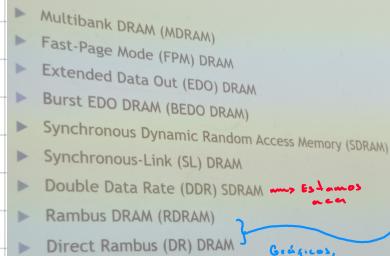
Temporal → Conservan su valor, mientras el pc tiene potencia

- SRAM → D-Flipflops
 - Muy rápidas
 - Bastante costosa
 - Consumo transistores a lo loco
 - Actualmente max (2mb)
- DRAM → Dynamic random access memory
 - Hasta de condensadores que leak electricidad → Descargado: 0 / Cargado: 1 → Si descargan se lo borramos
 - recarga ciertas unidades para mantener datos → Tiene un circuito de refresco → Lee todo el tiempo
 - danza (Can store many bits per chip)
 - Use less power
 - generates less heat

SRAM:

- Barata
- Aynchronous SRAM
- Cache
- Synchronous SRAM
- pipeline burst SRAM

DRAM



GPU → Necesitan memoria para mandar la imagen.

Divide la memoria Video en fragmentos / persistencia visual (Movimiento)

memoria en pequeñas matrices → Va una sola dirección

Algebra lineal → Un solo movimiento

VRAM → Especializada para video/gráfica.

ROM:

- ROM → Biodes (Matriz de biodes) → muy costosa → toca leer como es de memoria los datos
- PROM (Programmable ROM) → Se escriben solo una vez → número alta de 2 → Precio mucho más bajo
- FIRAM → Se queman y programan el chip (se quema el 0)
- EPROM (Erasable PROM)
 - Programables más de una vez → tienen varios capítulos
 - Escribimos en el chip (Ya no cambiamos el chip)
 - Sensible a rayos ultravioleta → Se necesitan lámparas que emiten esos rayos.

EEPROM (Electrically Erasable PROM)

- El computador → Maneja electricidad
- El usuario lo arruga
- El boom del internet → Desarrugar
- Se borra con un campo electromagnético
- Borra porciones del chip, un bit a la vez
- Desventajas: Se van de la tira = perdida (Internodal)
- Muy lento

- Muy usado en celulares
- Flash ROM (EEPROM) → Los queremos más rápidos
- Especialmente EEPROM
- Dada escrita y leída por sectores
- Más rápida que EEPROM

Costo: 2mb → \$20 → $\frac{\text{bit}}{\text{costo}}$

Importante!

Una memoria se daña es por el exceso de refresh, evitando perder datos

Las memorias USB no son para borrar, se dañan.

Desventaja → Tiene tolerancia

Si descargan se lo borramos

refresco → Lee todo el tiempo

Cada compañía tiene el suyo.

Toca reciclar los chips si salen dañados → es DEMASIADO CARO

Producción en masa → Toca quejarse

Estos dos son el problema → no se suben

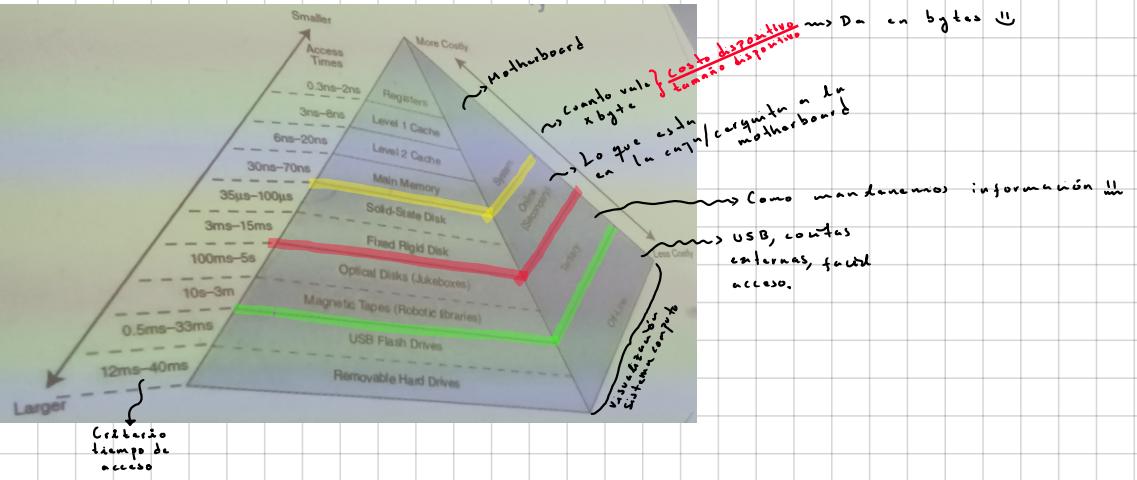
Corrientes se pierden → cuando no suben

se apagan pero es mucho más barato

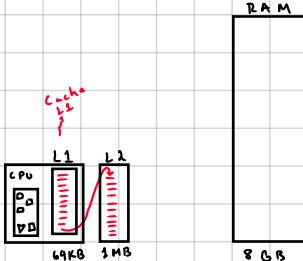
se borra que borrar todo el chip (un poco) para reprogramar

SSD (millones de escrituras) → se pierde acabar → PERO es muy lento que esto sucede → mayor posibilidad de cambiar el pc antes que el disco

Tener sentido con la edad



- System:
 - Registers \rightarrow Memoria utilizada solo para registros (0.3 ns - 2 ns)
 - RAM \rightarrow Principal (30 ns - 30 ns) \rightarrow Solo con estos dos generan un cuello de botella duro !!.
- Buffer: Área de memoria intermedia
 - \rightarrow Memoriza en el medio
 - \rightarrow Tiende a ser SRAM



- + Localidad espacial: Recorrer espacios de la memoria.
 - \rightarrow Se encuentra los datos consecutivos en memoria.
- + Localidad temporal: Accede a mismo lugar, PERO las áreas no son consecutivas (DATOS).
- + Localidad secuencial: Instrucciones consecutivas.

- Nuevas técnicas de programación favorecen a las localidades \rightarrow Orientado a objetos

Locality
temporal

\rightarrow Evolucionamos para mejorar al cache.

- Online: SO puede poner buffers para evitar cuellos de botella.

- Offline: USB, externas ---

Universal serial bus \rightarrow Mucho más lento; Se manda 1 bit al tiempo.

- Tertiary: Backup \rightarrow Copias de respaldo
 - \rightarrow Creado desde el mismo
 - \rightarrow Implementado en guerra fría \rightarrow Datos sobre vienen a una guerra nuclear

Sacar 3 copias:

- 2 \rightarrow En sitio (Caja fuerte...)
- 2 \rightarrow 20 Km/40 Km de distancia (Radio de destrucción de una bomba de 20 kilotonos)
- 2 \rightarrow 200 Km (Bomba nuclear estratégica)

De cualquier ciudad, bases militares...

- 1990 nos cambia la forma de ver los backups.

\rightarrow Nos relajamos (Manos backups...)

- 2002 \rightarrow Toman los backups ravive por las bocas y mas.

\rightarrow El backup no es solo los datos, tambien la infraestructura.
 \rightarrow Backup de la gente !! (Lateralmente desparecieron empresas)

- Hoy: El tamaño

- Jukeboxes \rightarrow Caja con diferentes números de DVD \rightarrow Lleno de unidades de DVD, de cuatro go.

Software

- Maneja la caja
- Toma bronce robótico \rightarrow Encargado de subir cuantos DVD necesitas

\rightarrow DVD lo escoge \rightarrow Otro brazo imprime sticker/DVD, pone a otra torre

En la base de datos queda registrado.

Puede sacar hasta 3 backups.

Cintas magnéticas **mucho** más antiguas/Captan todo (Tambor, ambiente...)

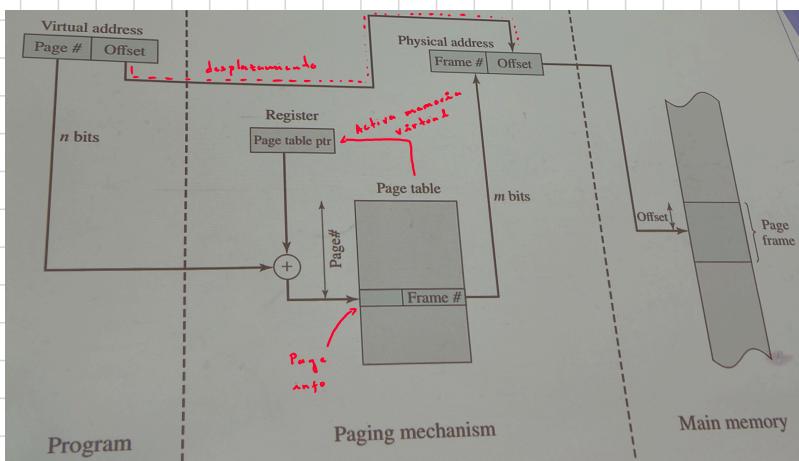
↳ Removible media: Eliminar ruido de fondo

Pueden guardar hasta 1TB

↳ Se comprimen en ZIP los datos
y quedan entre 1,2 - 1,4 TB.

Virtual memory

- Usar el disco como una extensión de la memoria RAM.
 - Múltiples aplicaciones concurrentes
 - Depend. del SO → El SO administra la memoria virtual
 - Virtual address → El CPU genera direcciones virtuales.
 - Physical address → La dirección real en la memoria física
 - Mapping → Dirección virtual → física
 - Page frames → Chunks o bloques en los que se divide (Memoria física)
 - Page → Chunks/Bloques
 - Paging → Marcos de página → página o viceversa (Disco a memoria o viceversa)
 - Fragmentation → Memoria que no se pone usar. (Costoso)
 - Page fault → Página está en disco, no en memoria → Toca traer los datos de disco a memoria
- ↳ VM son buenas si hay pocos saltes de página.



Virtual address translation (1/2)

- Extract the page number from the virtual address.
- Extract the offset from the virtual address.
- Translate the page number into a physical page frame number by accessing the page table.

Virtual address translation (2/2)

Translate the page number into a physical page frame number by accessing the page table.

- Look up the page number in the page table (using the virtual page number as an index).
- Check the valid bit for that page.
 - If the valid bit = 0, the system generates a page fault and the operating system must intervene to
 - Locate the desired page on disk.
 - Find a free page frame (this may necessitate removing a "victim" page from memory and copying it back if memory is full).
 - Copy the desired page into the free page frame in main memory.
 - Update the page table. (The virtual page just brought in must have its frame number and valid bit in the page table modified; if there was a "victim" page, its valid bit must be set to zero.)
 - Resume execution of the process causing the page fault, continuing to Step B.
 - If the valid bit = 1, the page is in memory.
 - Replace the virtual page number with the actual frame number.
 - Access data at offset in physical page frame by adding the offset to the frame number for the given virtual page

Siempre se hace paso 2.
Paso 3, se hace, solamente
si hay un fallo de página.

TLB: Translation look-aside buffer

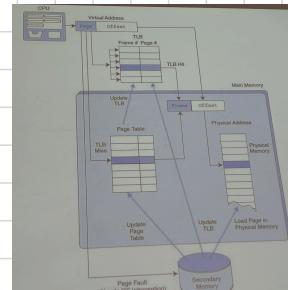
TLB: Translation Look-aside Buffer

Speed up the page table lookup by storing the most recent page lookup values in a page table cache called a translation look-aside buffer (TLB).

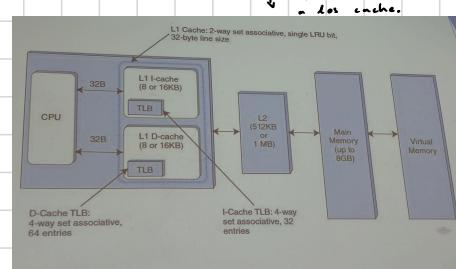
- Extract the page number from the virtual address.
- Extract the offset from the virtual address.
- Search for the virtual page number in the TLB using parallel searching.
- If the (virtual page #, page frame #) pair is found in the TLB, add the offset to the physical frame number and access the memory location.
- If there is a TLB miss, go to the page table to get the necessary frame number. If the page is in memory, use the corresponding frame number and add the offset to yield the physical address.
- If the page is not in main memory, generate a page fault and restart the access when the page fault is complete.

Se guardan copias de las páginas

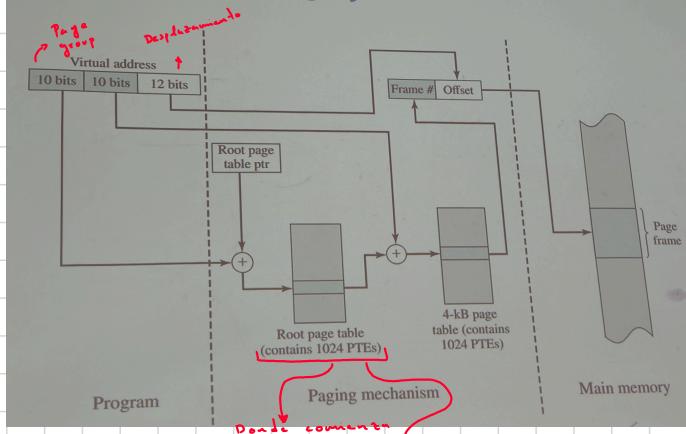
Graficamente



↓ Dibujamos TLB en los cachés.

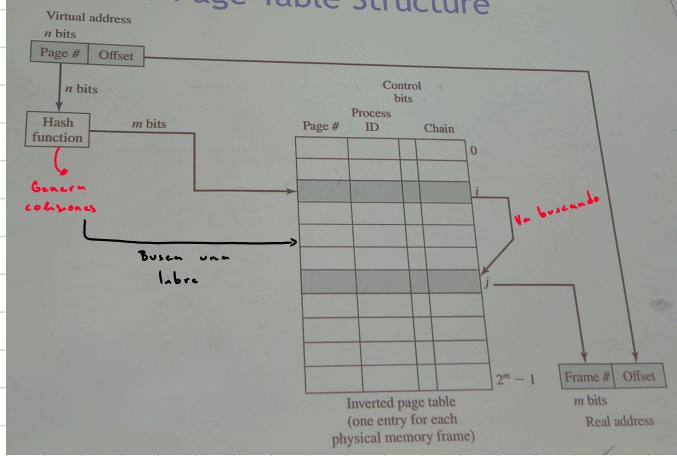


Two Level Paging System



* Subir sub-tabla
* Bajar página visitada

Inverted Page Table Structure



» Tabla de marcos, puede estar en memoria principal

» Mucho más chiquita

» Para cada marco, que página tiene

* Resolución de colisiones por dirección abierta