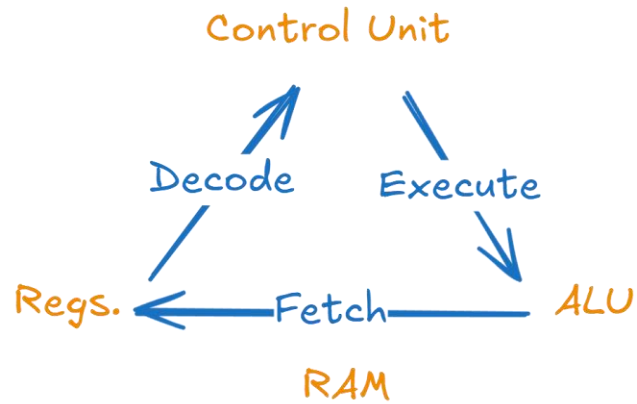
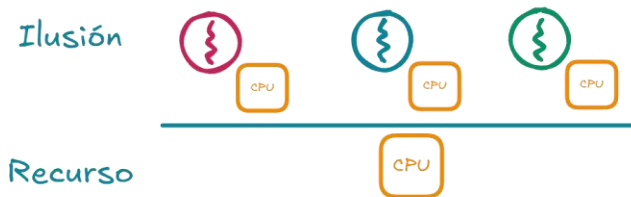


¿Cómo se ejecuta un programa?



Virtualización de la CPU



Time Sharing --> Ilusión de infinitas CPUs

Código compilado de un software, es decir, un archivo ejecutable

Instancia de un programa en ejecución, incluyendo el estado de ejecución



Sistema Operativo

- Administrar justa y eficientemente los recursos
- Controlar la ejecución de programas
 - Con el acceso adecuado al sys
- Brinda una representación virtual (abstracción) de los recursos para los programas, para facilitar el uso.

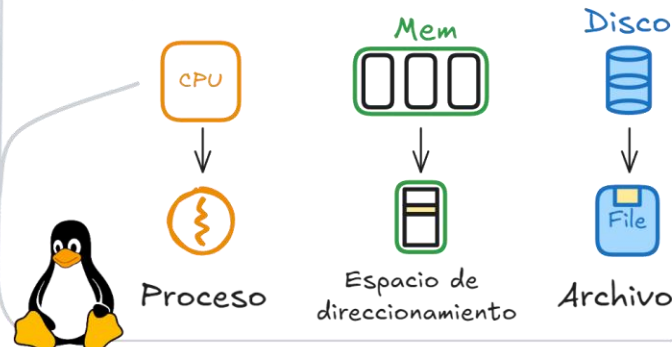
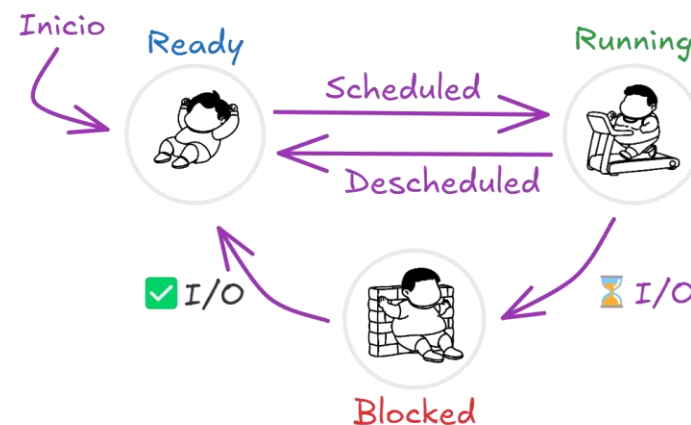


Diagrama de tres estados

El manejo de procesos concurrentes se lleva a cabo a través de estados. Ejemplo simple:



LDE (Ejecución Directa Limitada)

El SO hace de intermediario en las operaciones que comprometen partes importantes del sistema, a través de la interfaz de llamados al sistema.

Es un mecanismo esencial en todos los SO, debido a que permite la ejecución limitada de servicios de sistema por parte del usuario

Modos del procesador

Brinda seguridad en el SO y otros procesos

Modo Usuario

No privilegiado -> Aplicaciones

Modo Kernel

Privilegiado -> kernel del SO
Acceso directo a los recursos HW

Estructuras de datos del SO

- Process list

Listado de procesos en cada estado

- Register context

Contenido de los registros del programa, PC, S..., T...

- PCB (Process Control Block)

Información acerca de cada proceso (estructura en C)

Planificación de procesos (Scheduler)

Es un mecanismo que implementa el SO para determinar cuál es el siguiente proceso que debe ejecutarse



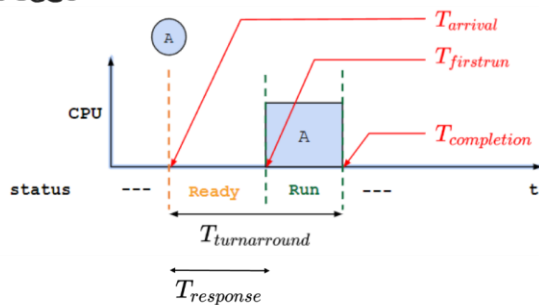
--> Planificador de procesos

work load --> Procesos en ejecución de un sistema

Métricas de desempeño

Turnaround Time (Para completar):

Métrica de desempeño para saber el tiempo entre la llegada y la finalización de un proceso



$$T_{\text{turnaround}} = T_{\text{completion}} - T_{\text{arrival}}$$

Response Time (Para responder):

Métrica de desempeño para saber el tiempo entre la llegada y la primera respuesta ante un proceso

$$T_{\text{response}} = T_{\text{firstrun}} - T_{\text{arrival}}$$

FIFO - FCFS

Esta política consiste en ir resolviendo procesos a medida que van entrando al sistema, es útil cuando los procesos tienen un tiempo de procesamiento similar o equivalente.

- ✓ Response Time
- ✗ Turnaround Time

SJF (Shortest Job First)

Esta política consiste en ir resolviendo primero los procesos que tardan menos tiempo en procesarse.

- ✗ Response Time
- ✓ Turnaround Time

No Preemptive

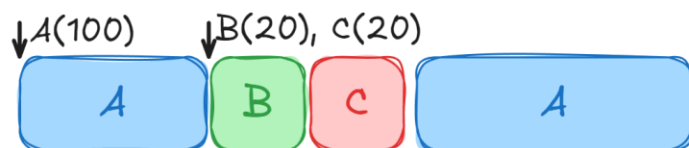
Posee problemas de inanición cuando hay muchos trabajos ligeros y algún proceso lento

STCF (Shortest Time-to-completion First)

Esta política es similar a la anterior, pero agrega la posibilidad de detener un proceso para realizar otro que le quede menos tiempo para completarse

- ✗ Response Time
- ✓ Turnaround Time

Preemptive



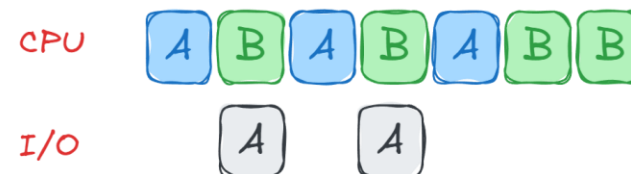
Round Robin

Esta política consiste en ejecutar procesos en porciones de tiempo llamadas **quantum** e ir intercalando entre procesos hasta que el proceso queda completado.

- ✓ Response Time
- ✗ Turnaround Time

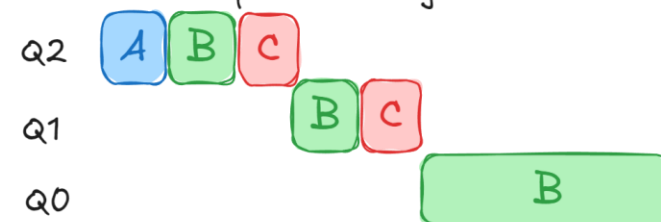


Puede adaptarse para incluir I/O



Multi-level feedback queue

1. Si prioridad(A) > prioridad(B), se ejecuta A
2. Si prioridad(A) = prioridad(B), RR para A y B
3. Nuevo proceso -> Cola más alta
5. Después de cierto tiempo, todos los procesos se ponen en la más alta prioridad
4. Cuando un proceso termina el control, su prioridad baja



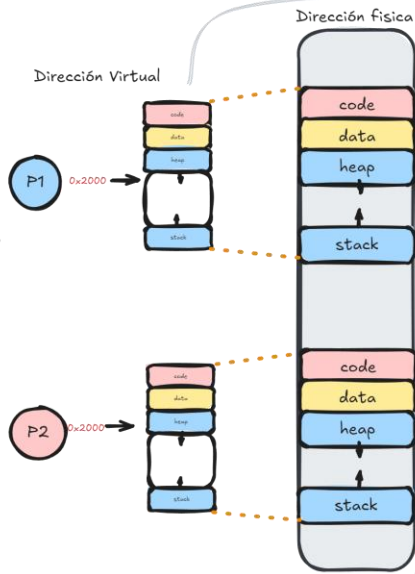
Virtualización de memoria

Crea un espacio de direcciones propio para cada proceso si un proceso falla, no va a fallar todo el sistema.

Beneficios

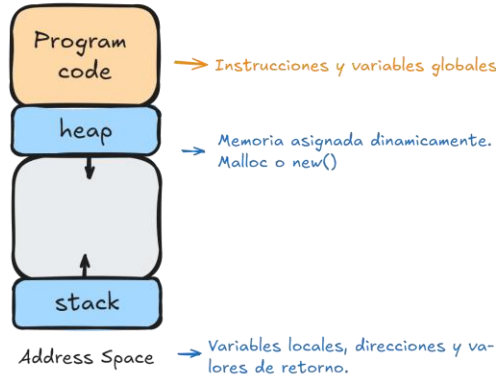
- Facilidad de uso para el programador
- Eficiencia de memoria en términos de tiempo y espacio
- Garantía de aislamiento entre procesos
- Cargar varios procesos a la memoria

Nota:
Todas las direcciones creadas por un programa son virtuales o lógicas. El proceso nunca va a poder ver la dirección física



Address Space

Es la abstracción de la memoria física creada por el S.O y contiene todos los datos de un proceso en ejecución

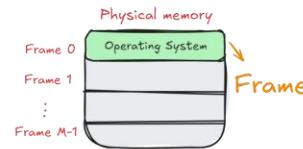


Paginación

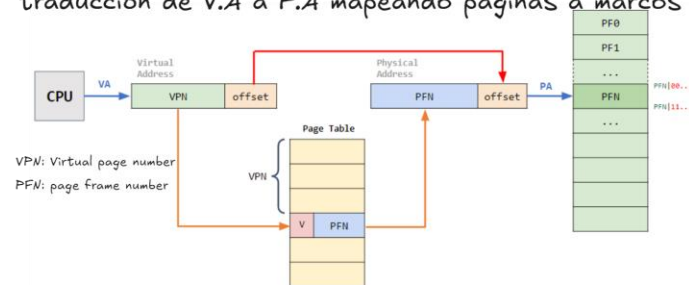
1. Divide el address space en unidades de tamaño fijo -> (pages)



2. La memoria física también se divide en unidades del mismo tamaño -> (pages frames)



3. Se necesita una page table para el proceso de traducción de V.A a P.A mapeando paginas a marcos



Beneficios: Sin fragmentación externa

Desventajas: Ocupa mucho espacio, peor desempeño

Traducción de direcciones

Se necesita un soporte en hardware (MMU)

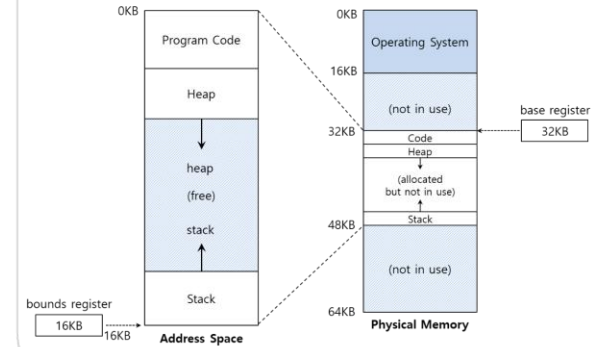
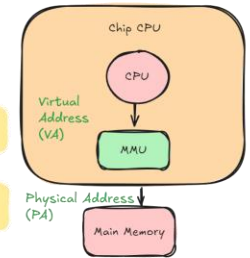
MMU (Memory Management Unit) es el hardware encargado de la traducción de VA a PA

Fijar un valor para el registro base

$$P.A = V.A + base$$

Verificar que las VA no sean mayores al valor del registro bound ni negativas

$$0 \leq V.A < bounds$$



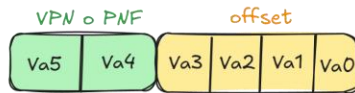
Traducción de direcciones - paginación

Las direcciones virtuales y físicas están compuestas de 2 partes:

VPN (virtual page number): Índice asociado a una página en la dirección virtual

PNF (Page frame number): Frame dentro del que se encuentra la dirección física.

offset: Desplazamiento dentro de la página seleccionada y son los bits menos significativos de la V.A



Para obtener el número de páginas o marcos

$$\#pages, frames = size(address\ space) / size(page)$$

Para calcular los bits necesarios de la dirección virtual

$$\log_2 \langle posiciones\ de\ memoria \rangle = m$$

Para calcular los bits necesarios del offset:

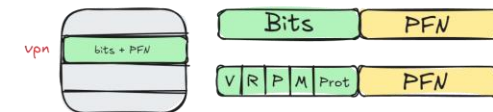
$$\log_2 \langle bytes\ de\ las\ paginas \rangle = n$$

Para calcular los bits necesarios del número de página (VPN):

$$V.A - offset = m - n$$

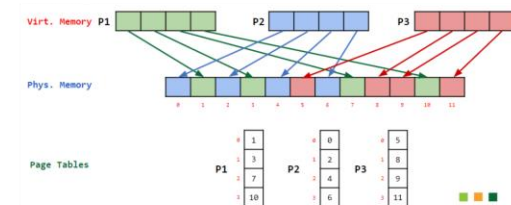
Tabla de página

Estructura de una PTE (Page Table Entry)



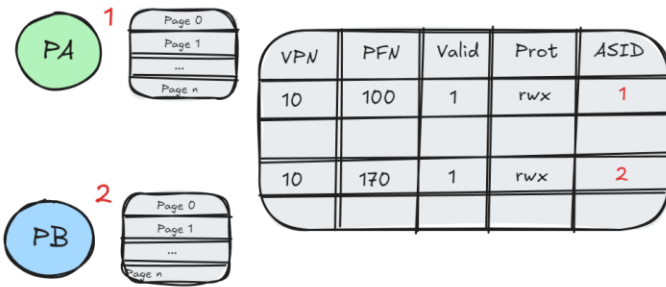
- Bit de validez (V): indica si la traducción es válida
- Bit de protección (prot): read, write, execute
- Bit de presencia (P): indica si la pag está en P.A
- Bit de sucio (M): indica si la pag ha sido modificada
- Bit de referencia (R): indica si la pag ha sido accedida

Cuando hay varios procesos se crea una tabla separada para cada proceso



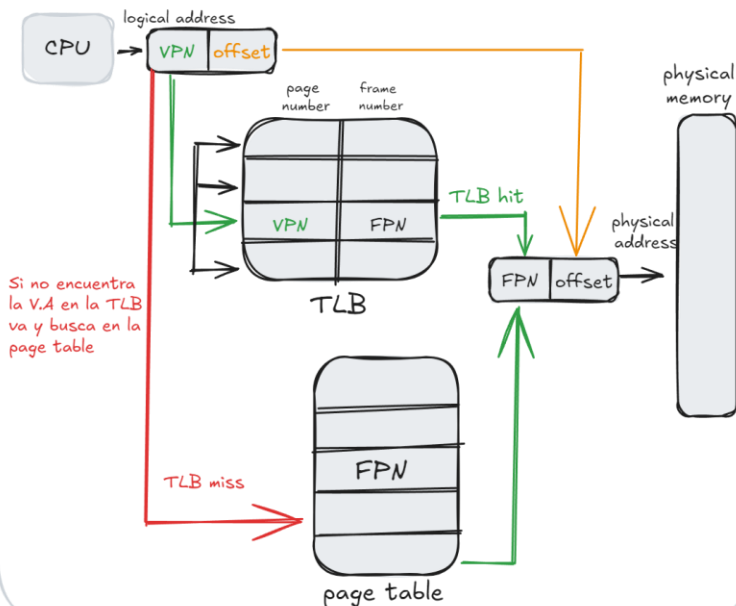
Cambio de contexto - solucion

Asignarle la columna ASID (Address space identifier) a la tabla de TLB



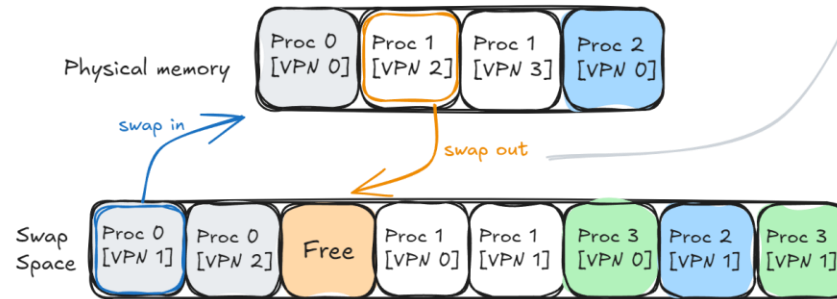
TLB

Es una memoria cache para acelerar el proceso de traducción de direcciones y hace parte de la MMU. Se mejora el rendimiento reduciendo los accesos directos a la page table



Espacio Swap

Es un espacio que nos permite ampliar las capacidades de nuestra memoria principal haciendo uso del disco duro. y el S.O divide el espacio swap en paginas.



Se necesita un bit adicional, Bit de presencia:

1: La pagina esta en la memoria fisica

0: La pagina no esta en la memoria fisica, pero si en el disco

Nota: Pueden ocurrir fallos de pagina cuando un proceso se encuentra en el espacio swap y tiene que ser llevada a la memoria principal.

Algoritmos de reemplazo

FIFO:

Reemplaza la pagina que ha entrado primero a la memoria, no puede determinar la importancia de las paginas. Presenta anomalia de Belady

Aleatoria:

Reemplaza de forma aleatoria la pagina a reemplazar lo que hace que su desempeño sea aleatorio

LRU (least recently used):

Reemplaza la pagina menos recientemente usada

Políticas de reemplazo

Definen los criterios para seleccionar cuál pagina debe salir de la memoria y el objetivo es minimizar los fallos (misses) en la cache.

AMAT(average memory access time) =

$$(P_{hit} * T_m) + (P_{miss} * T_d)$$

T_m = Tiempo de acceso a memoria (100ns)

T_d = Tiempo de acceso a disco (50ms)

Politica de reemplazo optima

→ Asegura el menor número de misses

→ Solo sirve de comparacion, no se puede implementar

Para realizar las politicas de reemplazo se usa una traza de memoria para identificar los hits y misses de la respectiva politica a usar

Anexo: Paginación

