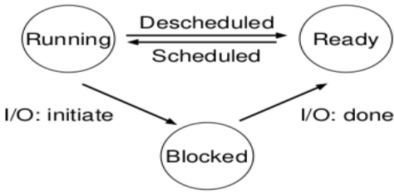


Estado de los procesos

**Running:** El proceso se encuentra ejecutando y haciendo uso de la cpu.

**Ready:** El proceso se encuentra listo para comenzar a ejecutarse en cuanto el sistema lo seleccione.

**Blocked:** El proceso se interrumpió por una llamada al sistema o un proceso I/O.



Principales componentes de un sistema de cómputo: Dispositivos I/O, Memoria Principal y secundaria (RAM y Disco Duro), CPU

Llamados al sistema

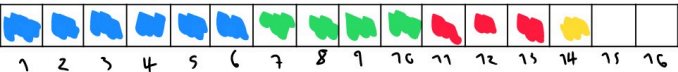
- Fork():** Crea un nuevo proceso, tiene su propia copia de address space y registros
- Exec():** Ejecuta un programa diferente al padre
- Wait():** Espera a que el programa hijo finalice retorna pid del hijo.
- Kill():** Terminar la ejecución de un proceso
- Algoritmo preemptive:** Es capaz de apropiarse de la CPU (la liberan periódicamente), espera por llamados al SO (RR-STCR-MLFQ)
- Algoritmo no preemptive:** Procesos completan ejecución antes de ser reemplazados, el SO toma el control, Hay timer interrupt, pueden generar inanición "starvation" (FIFO-SJF)

Políticas de Scheduling:

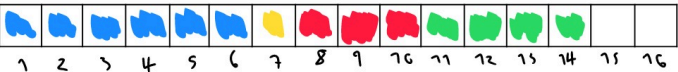
- FIFO:** Se ejecuta el proceso que haya llegado primero.
- SJF:** Se ejecuta el proceso que tenga menor tiempo de ejecución al terminar otro proceso.
- STCF:** Se ejecuta el proceso que requiera menos tiempo para terminarse en cada momento de tiempo.
- RR:** Se ejecutan todos los procesos por un periodo de tiempo(quantum) de forma cíclica.

Proceso	T. Llegada	T. ejecución
P1	0	6
P2	1	4
P3	5	3
P4	6	1

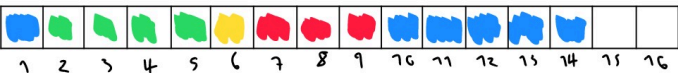
FIFO:



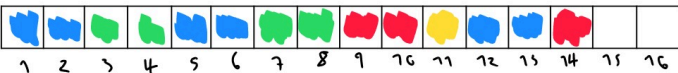
SJF:



STCF:



RR: (Quantum = 2)



TurnAround:Tcompletion-Tarrival

Tresponse:Tfirstrun-Tarrival

- Heap:** Espacio de memoria de un proceso asociado con la memoria dinámica
- Stack:** Memoria automática, Direcciones y valores de retorno, Variables locales y argumentos pasados a rutinas. La interrupción permite detener un proceso en la CPU para otorgarle el control al SO y continuar con otro proceso diferente. (No cooperativo)
- Trap:** Saltar de mod user a mod kernel
- Return-from-trap:** Retorna al programa llamado, restablece la cpu a modo user
- Program Counter:** Se encarga del direccionamiento de los procesos para que se ejecuten en el orden correcto.
- Políticas:** Algoritmos para tomar una decisión dentro del SO
- Mecanismo:** Obtiene un alto performance mientras hace un control de los principales retos de la construcción de un SO

Diferencia mecanismo-políticas

Las políticas dictan las decisiones que se deben tomar, brindan restricciones y criterios a seguir guiando el comportamiento del sistema, los mecanismos implementan estas políticas, tomando las acciones necesarias para seguir estas decisiones basándose en las restricciones y criterios establecidos por la política dada.

- Procesos Batch:** Alto consumo de CPU, poco tiempo en bloqueo, menor interacción del usuario.
- Procesos interactivos:** Bajo consumo de CPU, mucho tiempo en bloqueo(I/O), alta interacción con el usuario.

Multi Level Feedback Queue:

- Sistema de varias colas con distinto nivel de prioridad.
- Un proceso pasa a través de las distintas colas siguiendo una serie de reglas.
- Las colas pueden seguir distintas políticas de scheduling.
- El sistema ejecuta procesos de la cola no vacía con prioridad más alta.

Reglas MLFQ:

- Se ejecuta el proceso con mayor prioridad.
- Si dos o más procesos tienen la misma prioridad, se ejecutan mediante RR.
- Cuando un proceso llega al sistema, se le asigna la prioridad más alta.
- Cuando un trabajo consume su asignación de tiempo(sin importar cuantos quantums se tarda), se reduce su prioridad.
- Cada cierto tiempo, todos los procesos son asignados al mayor nivel de prioridad.

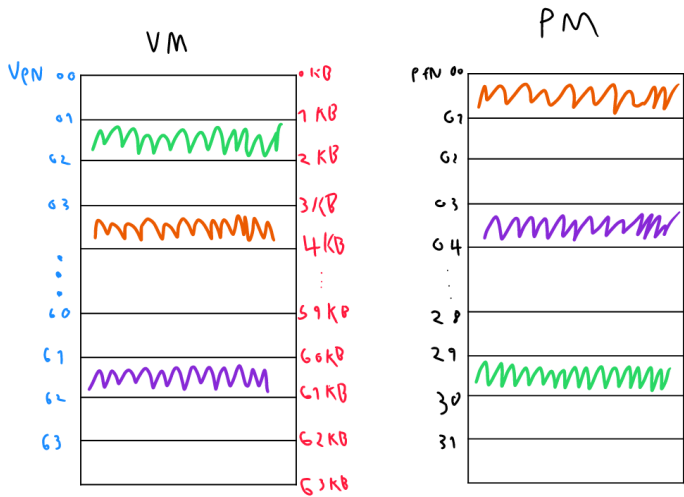
Virtualización de memoria:

- PCB (Process Control Block):** Guarda información acerca de cada proceso.
- Dirección lógica:** Es la dirección que utiliza un proceso o programa para acceder a una ubicación específica en la memoria. Esta dirección es generada por el programa y no representa directamente la ubicación física de la memoria.
- Dirección física:** Es la dirección real de la ubicación en la memoria física donde se almacenan los datos. Esta dirección es utilizada por el hardware para acceder a los datos en la memoria física.

La **fragmentación interna** se refiere al espacio desperdiciado dentro de los bloques de memoria asignados debido a su tamaño excesivo, mientras que la **fragmentación externa** se refiere a la falta de contigüidad en el espacio de memoria disponible que dificulta la asignación de bloques de memoria contiguos para satisfacer las solicitudes de memoria.

Considere un espacio de direccionamiento lógico de:

- 64 páginas
- 1024 Bytes de tamaño de página
- 32 frames de memoria física



por ejemplo, la pagina virtual 1 se encuentra en el marco de pagina 29

VPN (Número de Página Virtual):

El VPN (Virtual Page Number) es el índice que apunta a una página específica en la tabla de páginas. Dado que tenemos 64 páginas, necesitamos 6 bits para representar el VPN (2^6 = 64). Por lo tanto, el VPN puede variar de 0 a 63.

PFN (Número de Marco Físico):

El PFN (Physical Frame Number) es el índice que se utiliza para acceder a un marco específico en la memoria física. Dado que tenemos 32 marcos de página, necesitamos 5 bits para representar el PFN (2^5 = 32). Por lo tanto, el PFN puede variar de 0 a 31.

Desplazamiento dentro de la Página (Offset):

El desplazamiento es la parte restante de la dirección lógica después de considerar el VPN y el PFN. Dado que cada página tiene 1024 bytes (1 KB), necesitamos 10 bits para representar el desplazamiento (2^10 = 1024). Por lo tanto, el desplazamiento puede variar de 0 a 1023. En resumen: VPN: 6 bits (0 a 63) PFN: 5 bits (0 a 31) Desplazamiento: 10 bits (0 a 1023)

El sistema operativo BTV tiene: Virtual Address = 21 bits Page Size: 2KB Physical Address = 16 bits

¿Cuál es el número de entradas en cada uno de los siguientes casos? El tamaño de entrada es siempre 4KB-2^12b Una tabla de página lineal. (se divide el tamaño de la mem virtual sobre el tamaño de pág) 2^10 → 1024 entradas (páginas) Una tabla de página invertida. (se divide el tamaño de la mem física sobre el tamaño de pág) 2^5 → 32 entradas (páginas) (# bits de cada entrada de la tabla de pág invertida = # bits mem virtual + # bits mem física, en este caso 21 + 16 = 37)

Si se tiene una TLB en la cual el 75% de todas las referencias se encuentran efectivamente en la TLB, ¿cuál es el tiempo efectivo de acceso a memoria? (Ausma que encontrar una entrada en la TLB toma 2ns, si dicha entrada

$$T_{AEM} = \text{HitRate}(T_{TLB} + T_{MP}) + \text{MissRate}(T_{TLB} + 2T_{MP})$$
  
Tiempo Acceso Efectivo a Memoria  
Entrada memoria  
Acceso a memoria  
75% 2ns 50ns  
= 0.75(2ns + 50ns) + 0.25(2ns + 2\*50ns)  
= 0.75(52ns) + 0.25(102ns)  
= 39ns + 25,5ns  
= 64,5ns

está presente)

Fórmulas TLB:

Tiempo de acceso promedio = (Hit rate×Tiempo de acceso con TLB)+((1–Hit rate)×Tiempo de acceso sin TLB) Un page fault ocurre cuando un proceso intenta acceder a una página de memoria que no está presente en memoria física, y el Sistema Operativo toma medidas para cargar la página requerida en memoria y reanudar la ejecución del

Se tiene un sistema de cómputo con un tamaño de direccionamiento virtual de 512MB y una memoria física de 1GB. El tamaño de página es de 4KB. ¿Cuál es el tamaño mínimo en Bytes de una TLB de 32 entradas, si adicional a los campos de VPN y PFN se usan 16 bits para almacenar otra información relevante (ASID, protección, validación, etc.)?

proceso. tamaño de la dirección virtual: 512 MB (2^29 bytes, es decir, 512 \* 1024 \* 1024). Tamaño de la página: 4 KB (2^12 bytes). Bits necesarios para la VPN (Virtual Page Number): 17 bits (log2(512 MB / 4 KB)). Bits necesarios para la PFN (Physical Frame Number): 18 bits (log2(1 GB / 4 KB)). Bits adicionales para otra información relevante: 16 bits.Por lo tanto, cada entrada en la TLB (Translation Lookaside Buffer) requerirá un total de: Bits para la VPN: 17 bits. Bits para la PFN: 18 bits. Bits para otra información relevante: 16 bits. Tamaño total de cada entrada en la TLB: 51 bits. Dado que una TLB tiene 32 entradas, el tamaño total mínimo en bits será:

Tamaño total de la TLB = Tamaño de entrada TLB \* Número de entradas = 51 bits/entrada \* 32 entradas = 1632 bits. Para convertir esto a bytes, dividimos por 8 (ya que hay 8 bits en un byte): Tamaño total de la TLB en bytes = 1632 bits / 8 = 204 bytes. Para un sistema con direcciones virtuales y físicas de 12 bits y con páginas de 256 bytes(2^8 bits offset virtual). La lista de páginas libres es 0xD, 0xE, 0xF (siendo D el head de la lista, E la segunda y F la última). Convierta las siguientes direcciones virtuales a su equivalente dirección física. Todos los datos están dados en hexadecimal. Un guión (-) en el campo frame significa que la página no está en memoria 0x9EF → 0x(1001)VPN (1110 1111) OFF → 0x0EF 0x111 → 0x(0001)VPN (0001 0001) OFF → 0x211 0x700 → PFN no válida (página no está en memoria) 0x0FF → PFN no válida (página no está en memoria)

Page	Page Frame
0	–
1	2
2	C
3	A
4	–
5	4
6	3
7	–
8	B
9	0

2^10B = 1KB  
2^20B = 1MB  
2^30B = 1GM  
2^40B = 1TB  
8bits = 1 byte

El trashing suele ocurrir cuando el número de procesos activos en el sistema excede la capacidad de la memoria física disponible, lo que provoca una competencia intensa por los recursos de memoria. Para mitigar el trashing, se pueden tomar medidas como ajustar los algoritmos de planificación de memoria, optimizar el uso de la memoria virtual y limitar el número de procesos activos en el sistema. Considere un sistema de memoria virtual de paginación con 4 entradas en la tabla de página (con valores 0xC, 0x2, 0x8, 0x5, respectivamente). La memoria física es de 128 (2^7 Bytes) bytes con marcos de página de 8 bytes (2^3 Bytes) cada uno. ¿De qué tamaño es la dirección física? (número de bits) log2(2^7) = 7bits ¿De qué tamaño es la dirección virtual?

0xC → 0x1(VPN) 100(OFFSET)  
Tamaño dirección virtual = tamaño VPN + tamaño OFFSET =  
1bit + 3bits = 4bits  
Tamaño dirección virtual = 4bits → 2^4B = 16Bytes

¿Cuál es la dirección física (en hex) que corresponde a la dirección virtual.

0xC → 0x1(VPN) 100(OFF) → PFN = (0001) dir.  
Física = 0x0001 100

0x2 → 0x0(VPN) 010(OFF) → PFN = (0000) dir.  
Física = 0x0000 010

0x8 → 0x1(VPN) 000(OFF) → PFN = (0001) dir.  
Física = 0x0001 000

0x5 → 0x0(VPN) 101(OFF) → PFN = (0000) dir.  
Física = 0x0000 101

En hex:  
0xC → 0x0C  
0x2 → 0x02  
0x8 → 0x08  
0x5 → 0x05

Considere la siguiente cadena de referencias:

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

¿Cuántos fallos de página ocurrirán usando los siguientes algoritmos, asumiendo 1, 2, 3, 4, 5, 6 y 7 marcos de página? (Todos los marcos inician vacíos)

Señalaré con verde cuando sea Hit, con rojo cuando sea Miss (fallo de página) y con azul cuando se encuentra en memoria

**LRU**

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

Vemos que el 6 no se encuentra en los marcos de página, es un Miss, buscamos los menos usados y vemos que son el 3, 4 y 5, los cuales pasan a ser posibles candidatos para ser sacados, en este caso como podemos ver que más adelante hay un 3 vamos a sacar el 4 para darle paso al 6:

1, 2, 3, 6, 2, 1, 5, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

Para el 7 hacemos un proceso similar, el 7 no está y obtenemos un Miss, vemos que los menos utilizados ahora son el 6 y 5, aprovechando que conocemos que el siguiente es un 6, sacaremos el 4 para darle paso al 7:

1, 2, 3, 6, 2, 1, 7, 2, 1, 2, 3, 6, 3, 2, 1, 2, 3, 6.

Finalmente vemos que en la cadena de referencias dada tenemos 7 Misses, 7 fallos de página para el algoritmo LRU, un Hit Rate de 13/20 → 0,65 → 65%

◆ **FIFO**

Esta parte es igual a la LRU

Vemos que el 6 no se encuentra en el arreglo por lo que es un Miss, así que sacamos del mismo al primer elemento que fué ingresado, en este caso el 1 e ingresamos el 6:

3, 4, 2, 1, 5, 6, 7, 2, 1, 2, 3, 6, 3, 2, 1, 2, 3, 6.

Aplicamos lo mismo del caso anterior, el 7 no se encuentra en el arreglo así que es un Miss, sacamos el primer elemento ingresado, en este caso el 2 y damos entrada al 7:

3, 4, 2, 1, 5, 6, 7, 2, 1, 2, 3, 6, 3, 2, 1, 2, 3, 6.

Similar al LRU, resultamos con un total de 7 Misses, es decir 7 fallos de página para el algoritmo FIFO, resultando en un Hit Rate de 13/20 → 0,65 → 65%

🌐 **Optimal**

▶ **Parte inicial igual a las anteriores**

Encontramos en el arreglo que el 6 no está en los marcos de página, por lo que para reemplazarlo buscamos el valor que se accede más lejos en el futuro, en este caso vemos que serían posibles candidatos el 4 y el 5 ya que directamente no se vuelven a ver en el arreglo dado, podemos escoger el que queramos, en este caso escogemos el 4 y lo intercambiamos por el 6:

1, 2, 3, 6, 2, 1, 5, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

Nos sucede lo mismo con el 7 que encontramos ahora, como candidatos tenemos al 4 y al 5, tomamos el que queramos, escogemos arbitrariamente el 5 y lo reemplazamos por el 7:

1, 2, 3, 6, 2, 1, 7, 2, 1, 2, 3, 6, 3, 2, 1, 2, 3, 6.

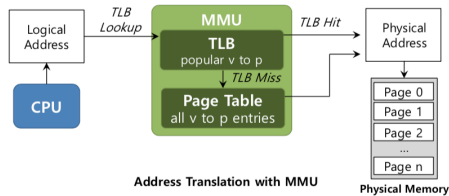
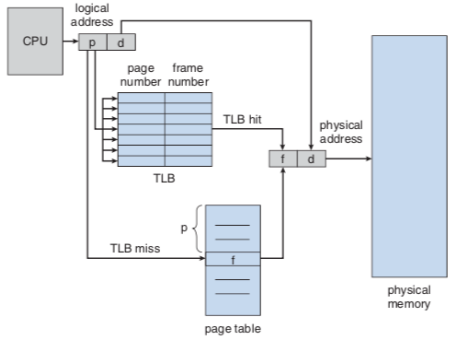
Vemos finalmente que en la cadena de referencias dada tenemos 7 Misses, 7 fallos de página para el algoritmo Optimal, al igual a lo que observamos en los algoritmos anteriores tenemos un Hit Rate de 13/20 → 0,65 → 65%

🔊 **Traducción direcciones:**

Dir.Física=Dir.virtual+base

0<=dir.virtual<bound

Bound:Número límite-1



La dirección de la PTE es:

dir. PTE = base[seg] + (VPN \* sizeof(PTE))

AMAT = T<sub>M</sub> + P<sub>MISS</sub> \* T<sub>D</sub>

T <sub>M</sub>	The cost of accessing memory
T <sub>D</sub>	The cost of accessing disk
P <sub>Hit</sub>	The probability of finding the data item in the cache(a hit)
P <sub>Miss</sub>	The probability of not finding the data in the cache(a miss)