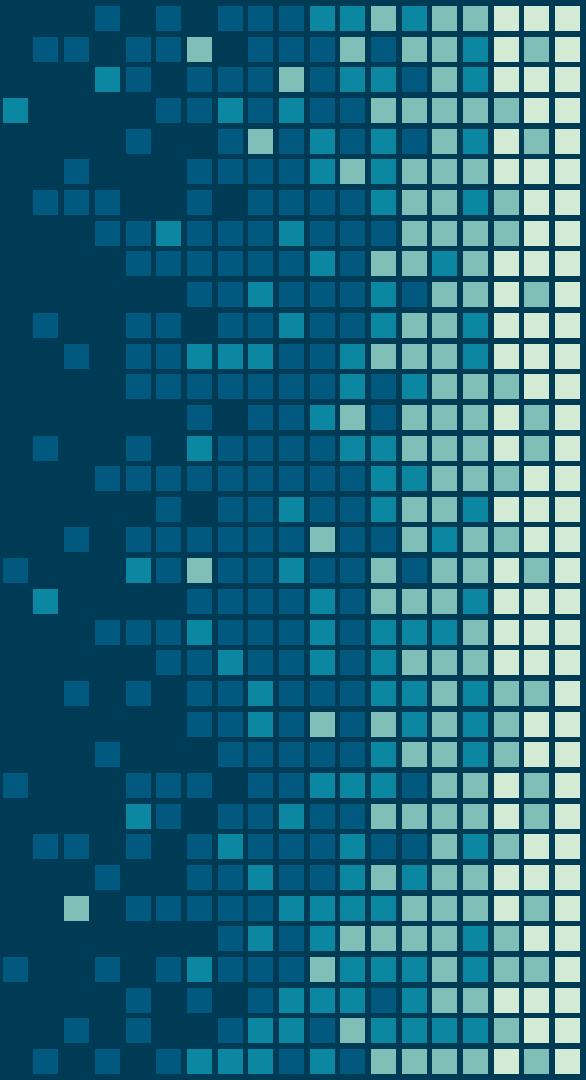
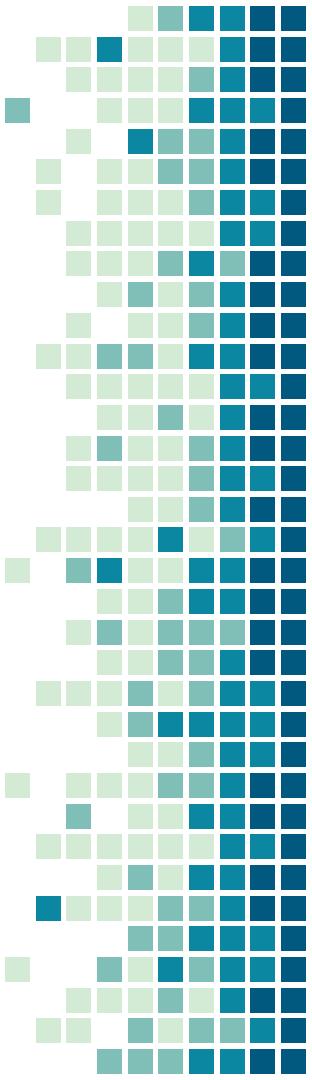


Administración de memoria en los SO

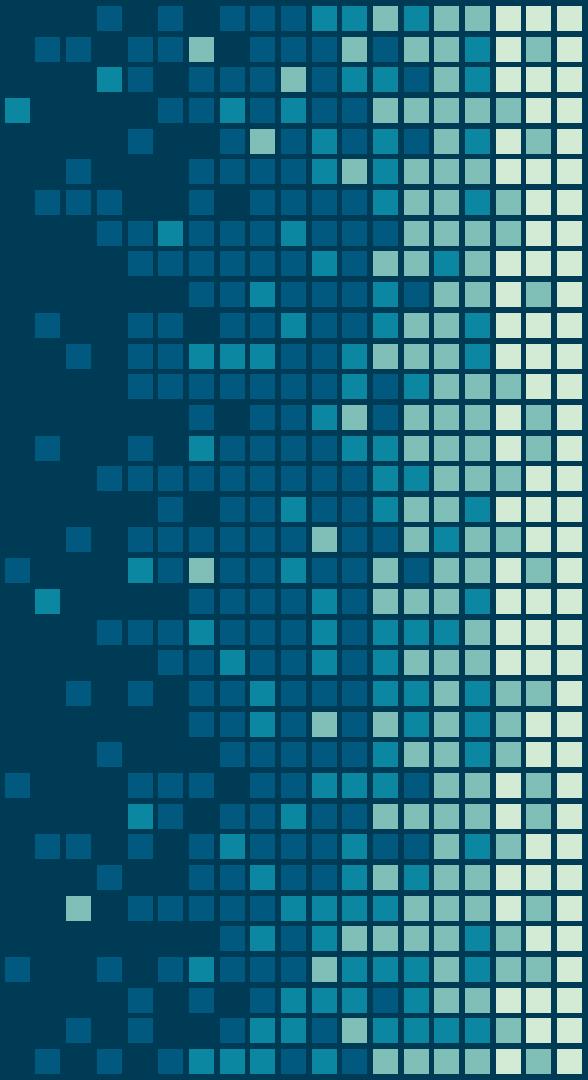


Agenda

- **Introducción**
- **Administración de memoria**
- **Sin abstracción de memoria**
- **Espacio de direccionamiento**
- **Requisitos de la abstracción de memoria**
- **Particionamiento**



Introducción





- ¿Cuál sería la memoria ideal para un Ingenier@ en computadores?
- Es imposible ¿Qué es lo que más se asemeja?
- ¿Se necesita una abstracción? ¿Cuál?
- ¿Entonces será necesario un administrador de memoria?

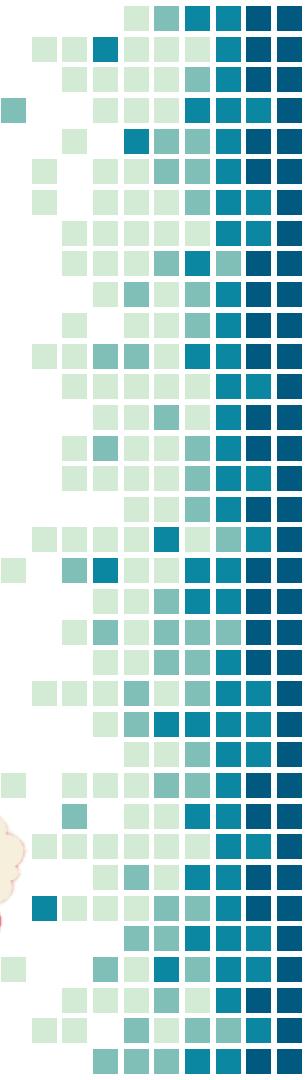
¿Por qué no se puede memorias más rápidas al igual que procesadores?

Efecto de pared de memoria.

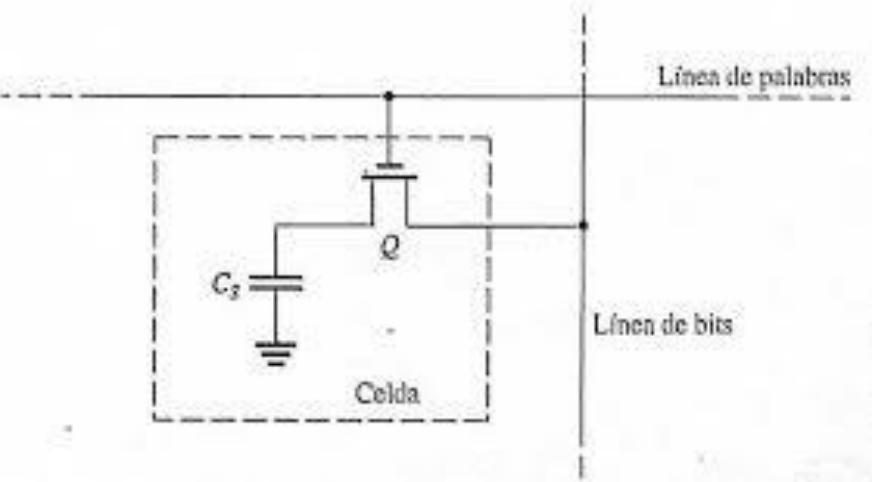
¿Cuál es la limitante?

El condensador no se puede hacer más pequeño porque si no se rompe el dieléctrico.

Por condiciones físicas no se puede igualar la rapidez del procesador.

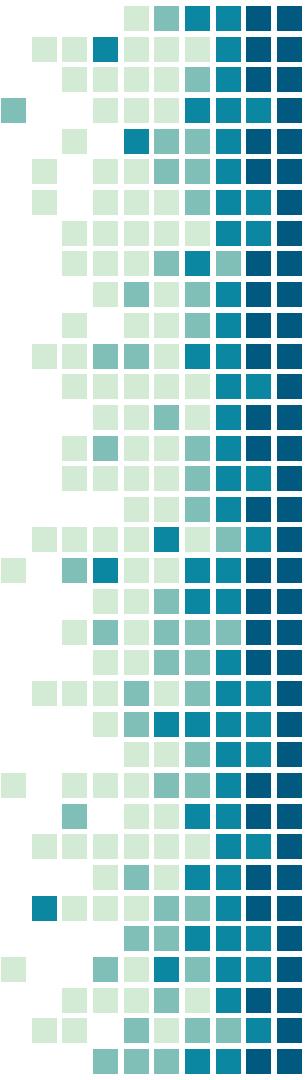


¿Por qué no se pueden memorias más rápidas al igual que procesadores?

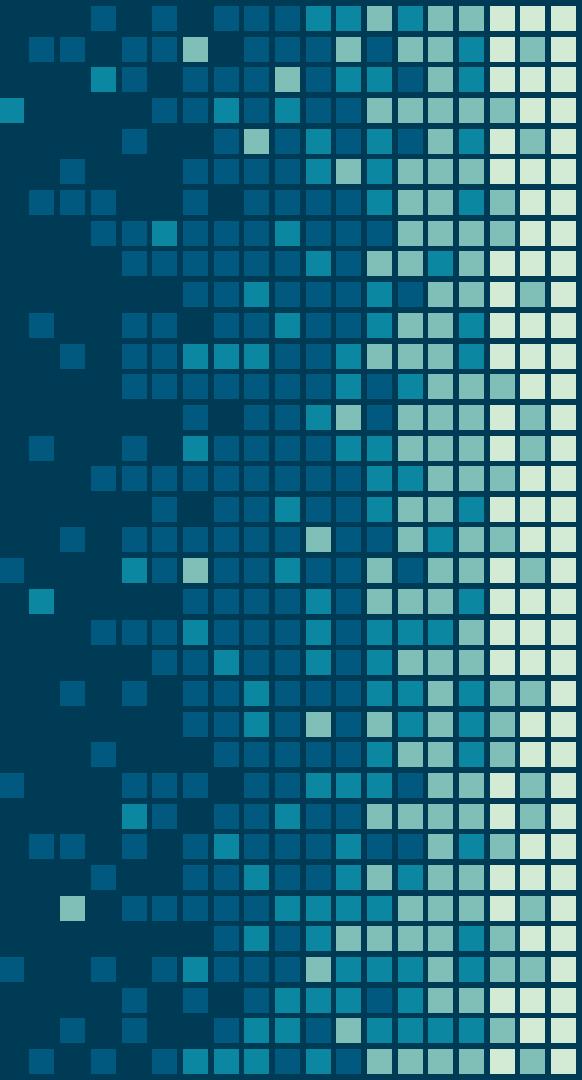


Administrador de memoria.

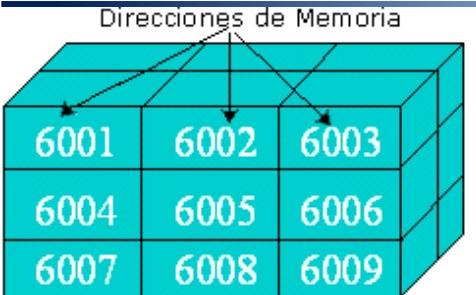
Su trabajo es administrar la memoria con eficiencia, es decir, llevar el registro de cuáles partes de la memoria están en uso, asignar memoria en los procesos cuando la necesiten y desasignarla cuando terminen.



Sin abstracción de memoria

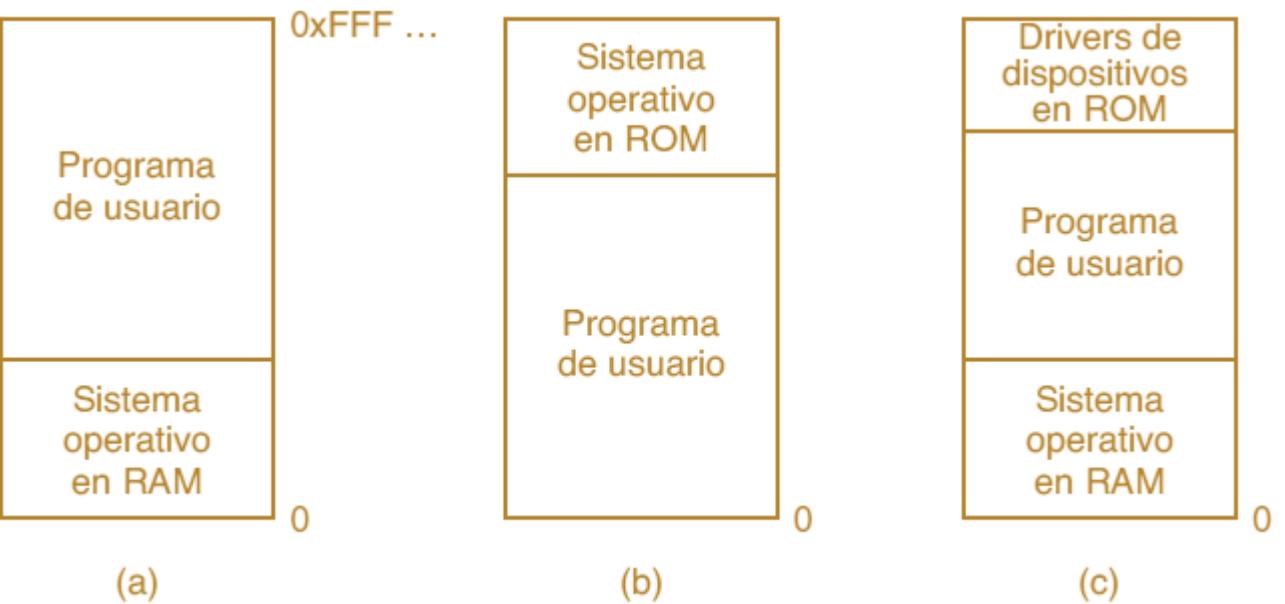


Sin abstracción de memoria



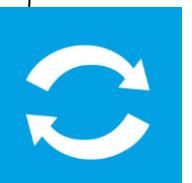
- Las primeras computadoras veían la memoria física tal cual.
- Por ejemplo mov R1,100: la computadora movía lo que estaba en la posición 100 a R1.
- El modelo que tenía el programador era simple. Un conjunto de direcciones.

Opciones de organizar la memoria



¿Cómo ejecutar múltiples programas sin abstracción de memoria?

El sistema operativo debe guardar todo el contenido de la memoria en un archivo en disco, cargar y ejecutar el siguiente programa.



Este concepto de intercambio da origen a las técnicas actuales de manejo de memoria.

Ejecución de múltiples programas sin abstracción.



Sistemas embebidos.

- En sistemas embebidos como los de las lavadoras, microondas y refrigeradoras se aplica esta técnica por simplicidad, pero es porque se dispone de la información sobre los programas que se ejecutarán.



Abstracción de memoria: Espacios de direcccionamiento



Espacios de direcciones

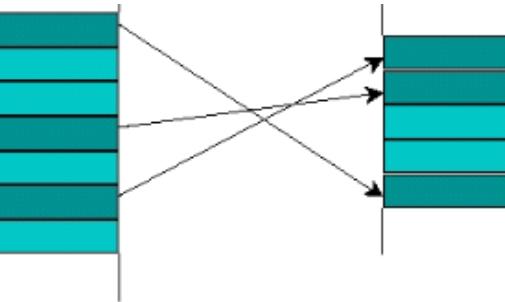
En general no es bueno exponer la memoria física a los programas.



💡 Si los programas pueden acceder por Byte pueden estropear fácilmente al SO (si hay hardware especial como claves no pasa).

💡 Es costoso computacionalmente mantener más de un programa en ejecución y tomando turnos por CPU.

Espacio de direccionamiento



- ❖ Es el conjunto de direcciones que puede utilizar un proceso para direccionar la memoria.
- ❖ Cada proceso tiene su propio espacio de direcciones, independiente de los que pertenecen a otros procesos.
- ❖ La idea fundamental es que sirva como una abstracción de la memoria física.

Espacio de direccionamiento

La abstracción de memoria debe contar con ciertos criterios para garantizar el buen funcionamiento de la memoria y con ello el del sistema:



Reubicación.



Protección.



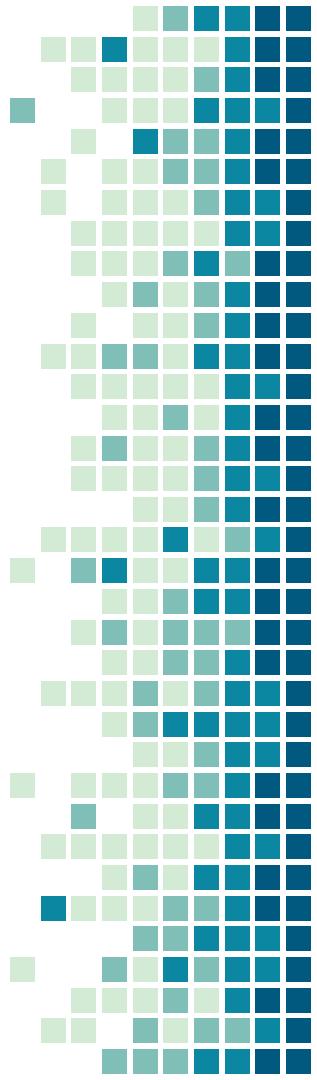
Compartición.



Organización lógica.



Organización física.



Reubicación



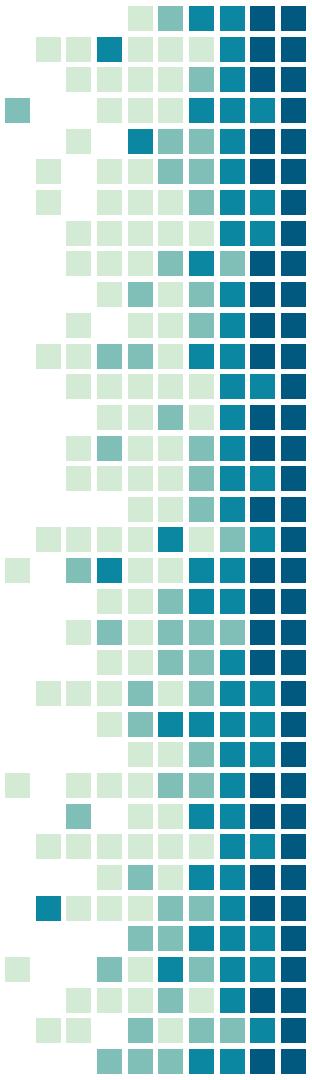
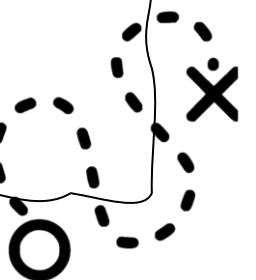
Reubicación

- ✓ En un sistema de multiprogramación, la memoria principal se comparte entre varios procesos.
- ✓ Cuando un proceso se almacena en disco, sería limitante tener que volver a cargarlo donde estaba anteriormente.
- ✓ Es necesario **reubicar** el proceso en un espacio diferente con el fin de no generar overhead realizando otras acciones como copiar o almacenar datos.



Reubicación

- El sistema operativo necesita saber sobre los límites del programa, es decir, el rango que direcciones tomadas.
- Las instrucciones de salto contienen direcciones referenciadas.
- Esto implica que SO operativo debe de asegurarse que las direcciones referenciadas estén bien mapeadas a las físicas.



Protección



¿Cómo atacar la condición de no apropiativa?



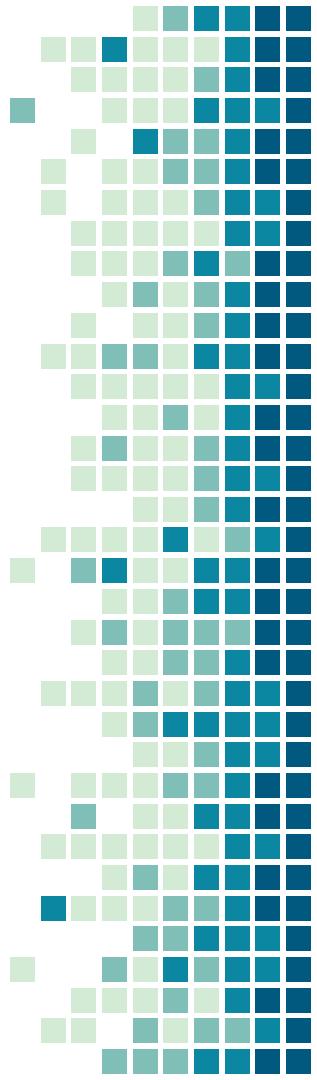
Cada proceso debe protegerse contra interferencias no deseadas por parte de otros procesos, sean accidentales o intencionadas.

Lograr los requisitos de reubicación aumenta la dificultad de satisfacer los requisitos de protección.

Todas las referencias generadas por un proceso deben verificarse en tiempo de ejecución.



+

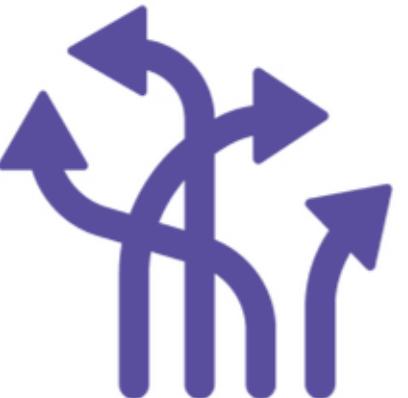


Compartición



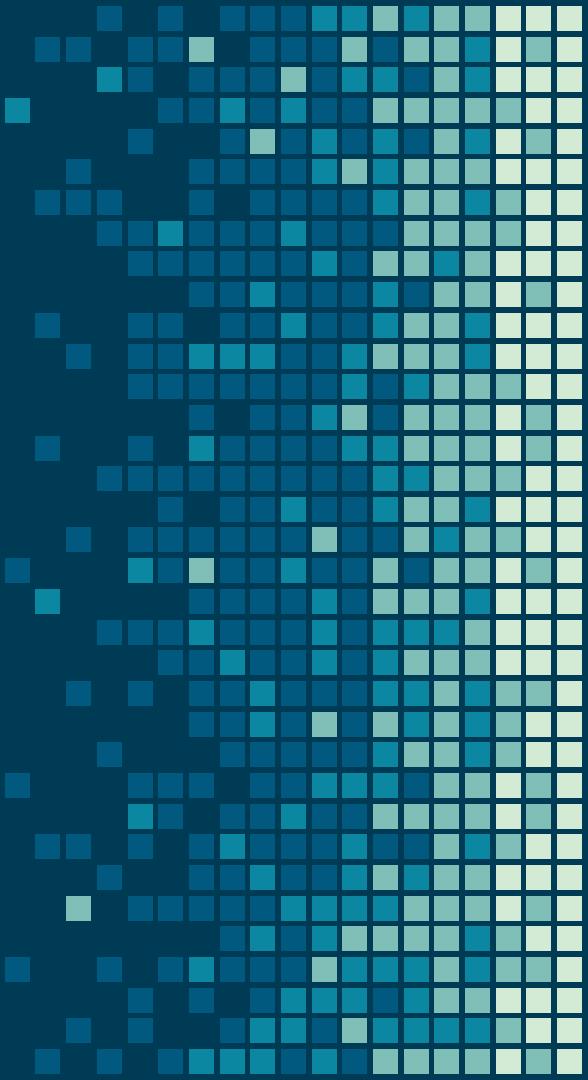
Compartición

Se debe contar con la suficiente flexibilidad para permitir a varios procesos acceder a la misma porción de memoria principal.



-
Por ejemplo si dos procesos ejecutan el mismo código.

Organización lógica



Organización lógica

La organización de una memoria consiste en un espacio de almacenamiento lineal, compuesto por una secuencia de Bytes o palabras.

La memoria secundaria está organizada de manera similar, sin embargo, los programas se organizan en módulos.

El sistema operativo lidia con esto al hacer las referencias correspondientes desde los diferentes módulos.

Organización Física



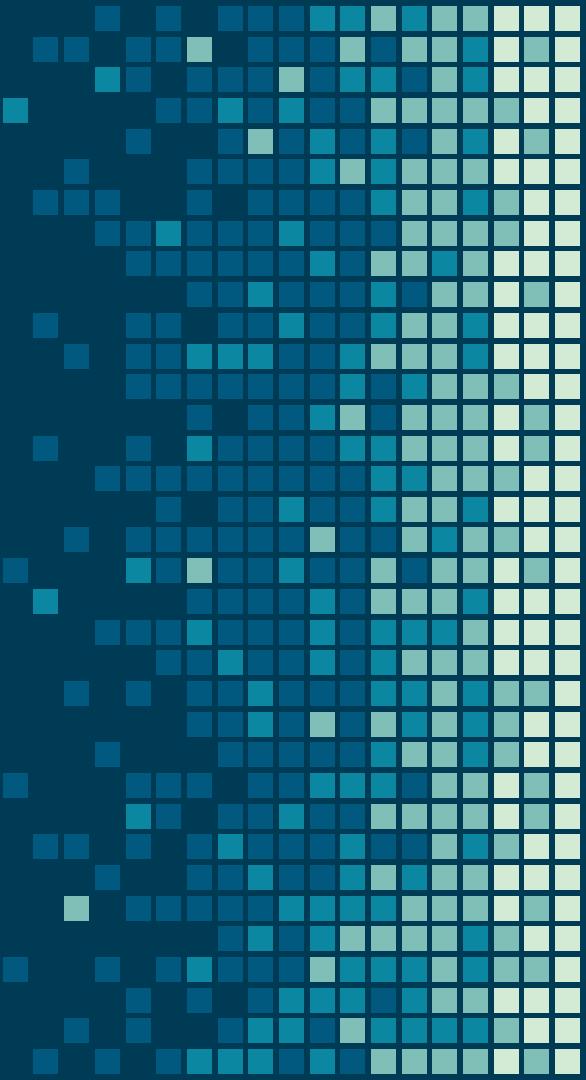
Organización física.

Memoria principal: Es volátil, es de acceso más rápida que la secundaria.



Memoria secundaria: Más lenta, y barata, de mayor capacidad.

Particionamiento

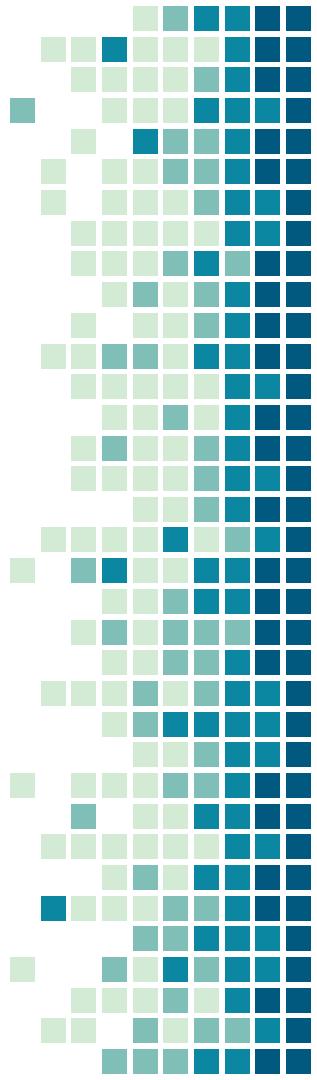


Particionamiento de la memoria



La operación principal de la gestión de la memoria es traer los procesos a la memoria principal para ser ejecutados. En los sistemas de multiprogramación modernos es inevitable comentar sobre este concepto, ya que muchos más se basan en el mismo.

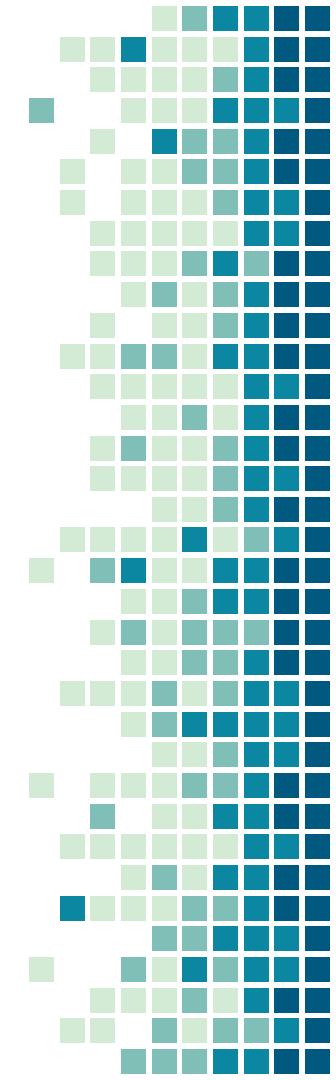
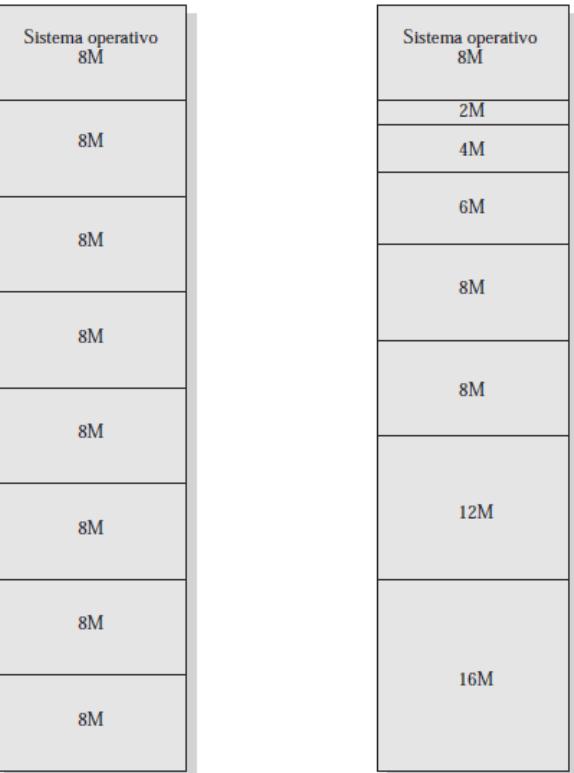
Particionar la memoria significa dividir la memoria en secciones con algún fin en específico.



Particionamiento de tamaño fijo

- El sistema operativo utiliza alguna porción de memoria.
- Todo el resto de la memoria está disponible para la ejecución de programas.
- La idea fundamental es dividir la memoria disponible en regiones para que los programas puedan ejecutarse en algunas de ellas.

Dos enfoques de particionamiento fijo



Conflictos

- Un programa podría ser más grande que las particiones y no se podría ejecutar. ¿Qué se podría hacer en este caso?
- El uso de la memoria es ineficiente. ¿Qué pasa si hay un programa con que ocupa 2Mb?



Algoritmo de ubicación para particionamiento fijo

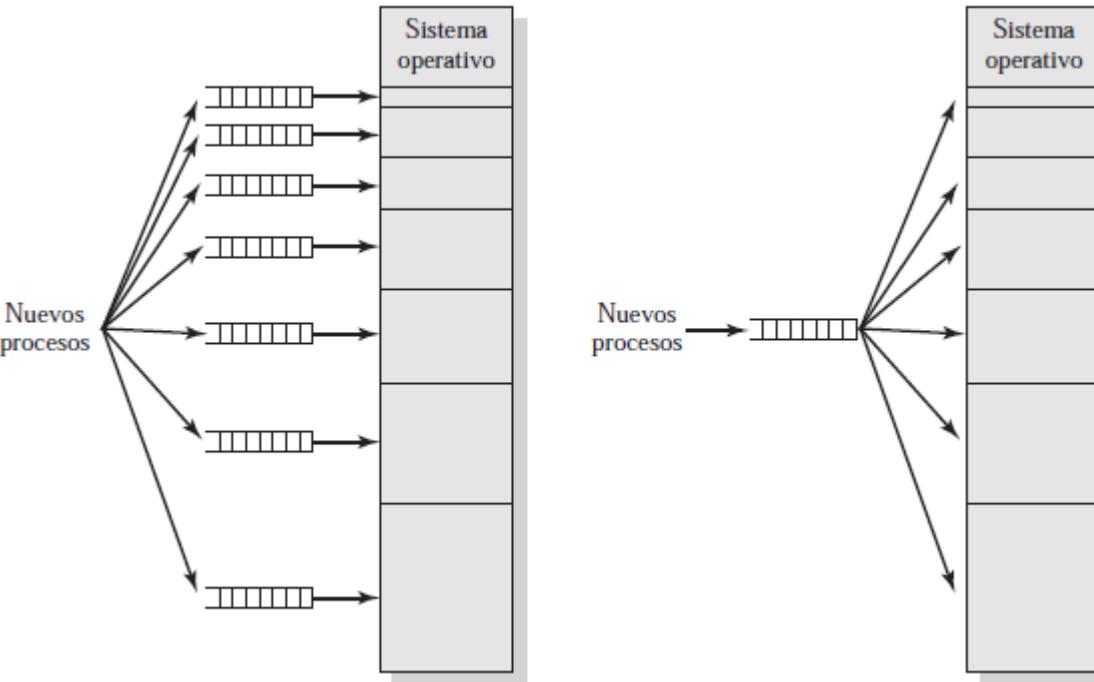
Con particiones del mismo tamaño, la ubicación es trivial.

En caso de que sea de diferente tamaño hay dos maneras:

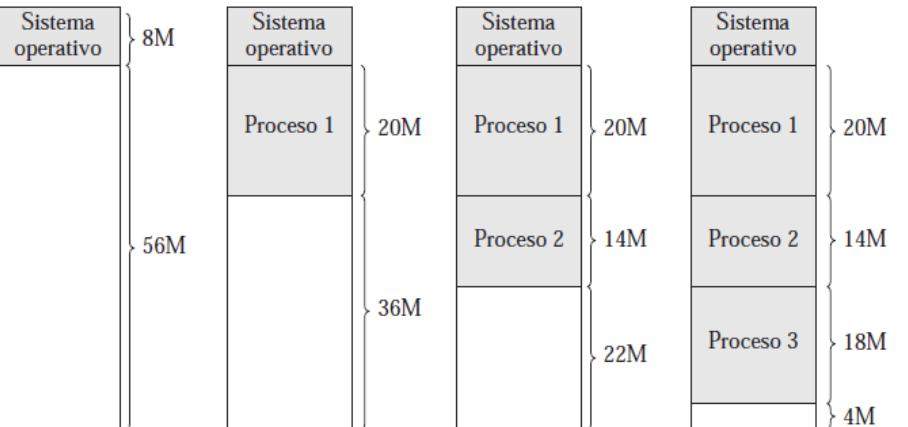


- Asignar al proceso la partición más pequeña para ser ejecutado utilizando una cola para cada partición.
- Utilizando una cola general

Asignación de memoria con particionamiento fijo.



Particionamiento dinámico



Las particiones son de longitud y número variable.
Cuando un proceso es cargado a la memoria, se le asigna justamente la cantidad de memoria que necesaria.

Fragmentación de memoria

En ambos enfoques ocurre fragmentación

- **Fragmentación interna:** Ocurre cuando el particionamiento es estático y se refiere al mal uso del espacio de una fragmentación (espacio libre en una fragmentación).

- **Fragmentación externa:** Ocurre en particionamiento dinámico y se refiere al mal uso en la memoria en general. Huecos de direccionamiento en la memoria. ¿Cómo solucionarlo?

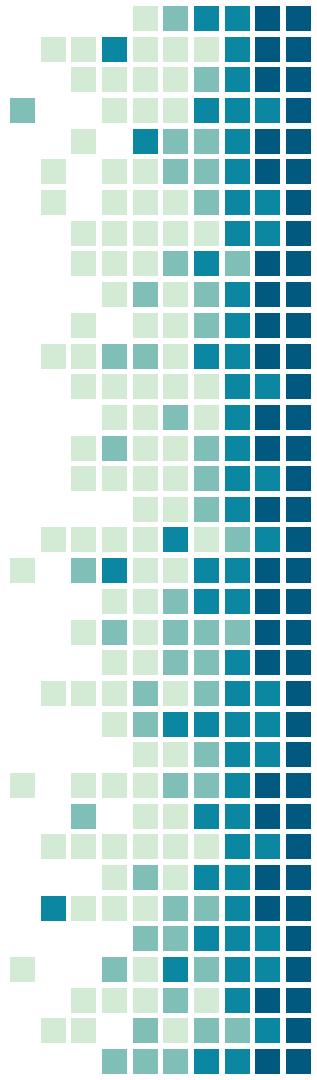
Maneras administrativas de asignar particiones

- 👉 First Fit: Primer espacio donde se pueda ejecutar.
- 👉 Best Fit: El mejor espacio, donde se desperdicie.
- 👉 Worst Fit: En la partición más grande.
- 👉 *Buddy System: particiones de tamaño en potencias de 2.

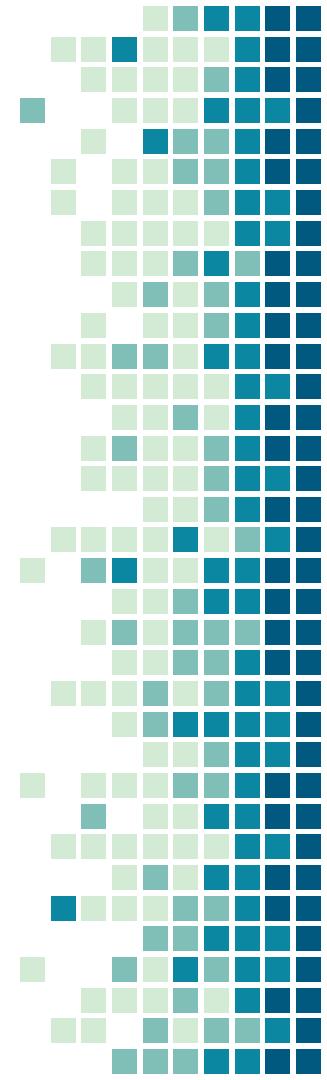
¿Cómo solucionamos la fragmentación?

- Compactación: mover los espacios libres, con el fin de que los procesos estén contiguos.

Relocalización: Reajustar todas las direcciones del programa que sean sensibles a la ubicación. Para eso se utiliza el mapa de relocalización, creado por el compilador.



¿Cuál es, realmente, el problema de la fragmentación?



Memoria virtual



Memoria Virtual

Un disco SATA transfiere datos a una tasa de 100MB/s, lo que implica que un programa de 1GB duraría 10s en escribirlo en disco y otros 10 al cargarlo.

¿Qué ocurre cuando la memoria está llena y se necesita ejecutar más programas?

¿Es el swapping la solución más conveniente para el problema del bloatware?



¿A qué se refiere Memoria Virtual?



La idea fundamental es que cada programa contiene su propio espacio de direccionamiento, el cual está dividido en trozos llamados páginas.

Las páginas son un conjunto de instrucciones contiguas pertenecientes a un determinado programa.

No todas las páginas que conforman un programa deben estar en la memoria

¿Qué ocurre si una página se necesita y no está en memoria?

- El sistema operativo recibe una alerta para buscar la parte faltante con el fin de ejecutar la instrucción que corresponde.
- Se debe de controlar el rango de instrucciones que corresponde a la página para delimitar la misma.



Paginación

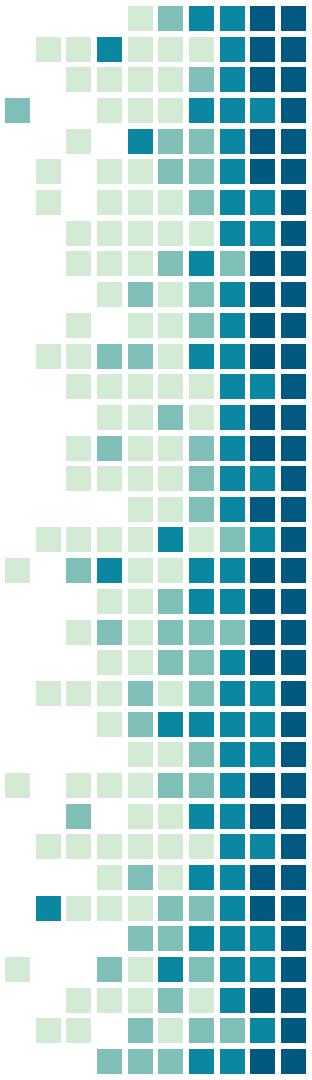
- La memoria se divide en bloques de tamaño fijo relativamente pequeños, y cada proceso también se divide en porciones del mismo tamaño fijo.



¿Qué ocurre con las fragmentaciones?

...

El sistema operativo mantiene una tabla de páginas por proceso, con el fin de controlar cuales páginas se necesitan y cuales no, muestra el bloque por cada página.



Paginación

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14

0	A.0
1	A.1
2	A.2
3	A.3
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

0	A.0
1	A.1
2	A.2
3	A.3
4	B.0
5	B.1
6	B.2
7	
8	
9	
10	
11	
12	
13	
14	

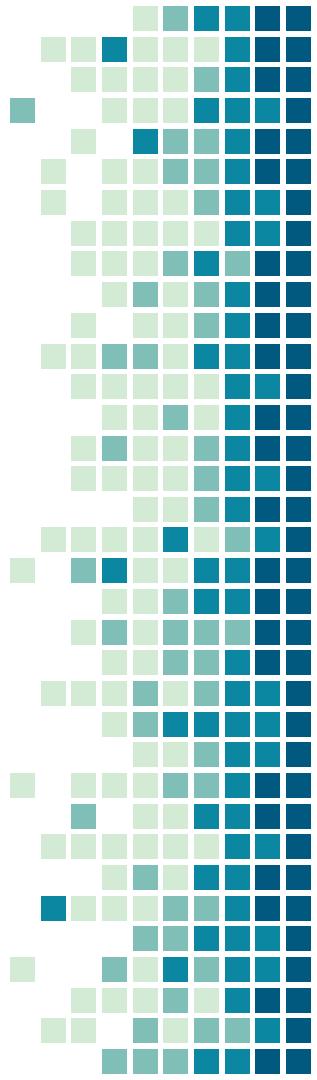
- Cada dirección física está compuesta por la ubicación del bloque y un desplazamiento.

Paginación

0	A.0
1	A.1
2	A.2
3	A.3
4	B.0
5	B.1
6	B.2
7	C.0
8	C.1
9	C.2
10	C.3
11	
12	
13	
14	

0	A.0
1	A.1
2	A.2
3	A.3
4	
5	
6	
7	C.0
8	C.1
9	C.2
10	C.3
11	
12	
13	
14	

0	A.0
1	A.1
2	A.2
3	A.3
4	D.0
5	D.1
6	D.2
7	C.0
8	C.1
9	C.2
10	C.3
11	D.3
12	D.4
13	
14	



Paginación

0	0
1	1
2	2
3	3

Tabla de páginas del proceso A

0	—
1	—
2	—

Tabla de páginas del proceso B

0	7
1	8
2	9
3	10

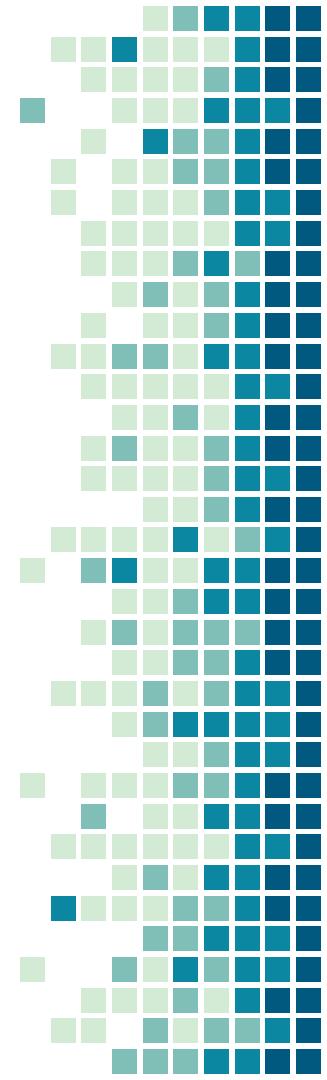
Tabla de páginas del proceso C

0	4
1	5
2	6
3	11
4	12

Tabla de páginas del proceso D

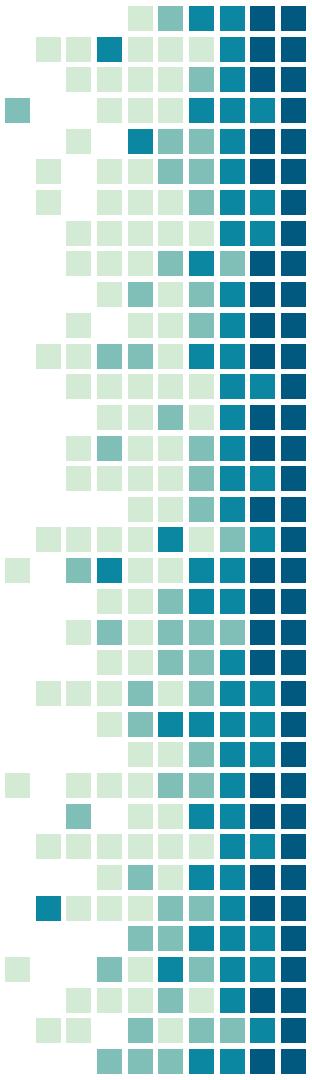
13
14

Lista de marcos libre

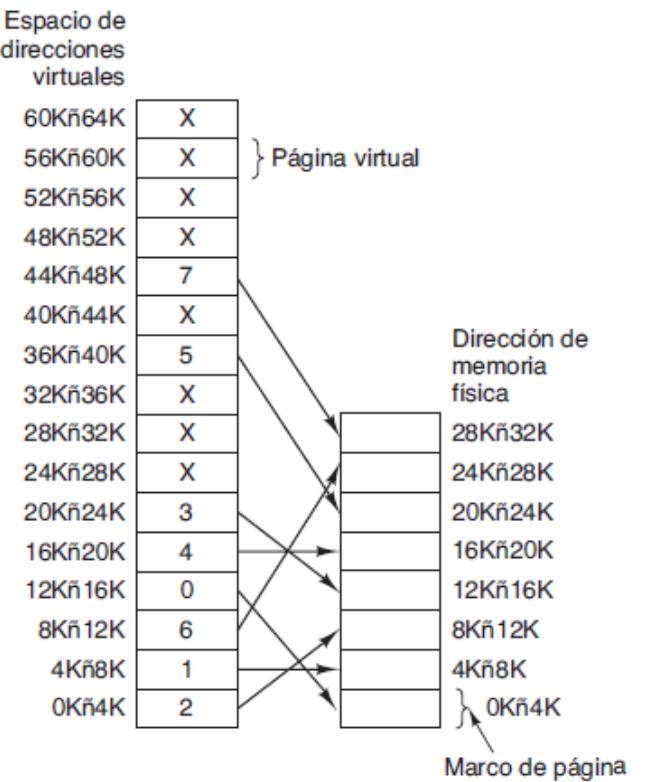


Ejemplo

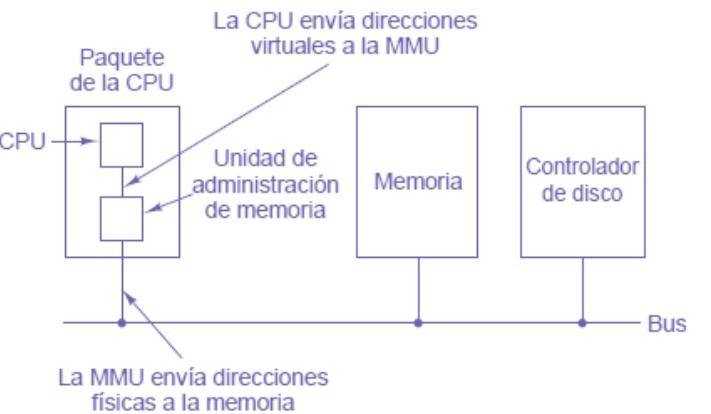
- ¿Cómo sería la dirección 1502 con un tamaño de página 1K y bus de direcciones de 16 bits?



Direcciones virtuales VRS Direcciones físicas



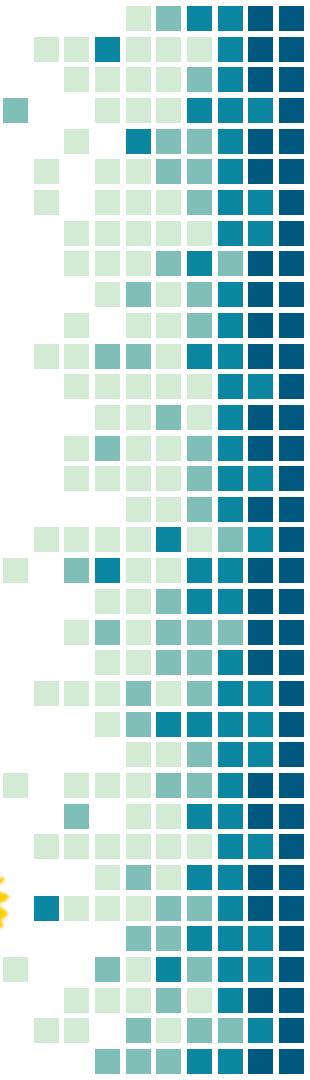
¿Cómo ocurre la transformación de direcciones lógicas a las físicas?



- MMU (Memory Management Unit): asocia las direcciones virtuales con las físicas. Se encarga de la administración de las tablas de las páginas.

Tabla de páginas.

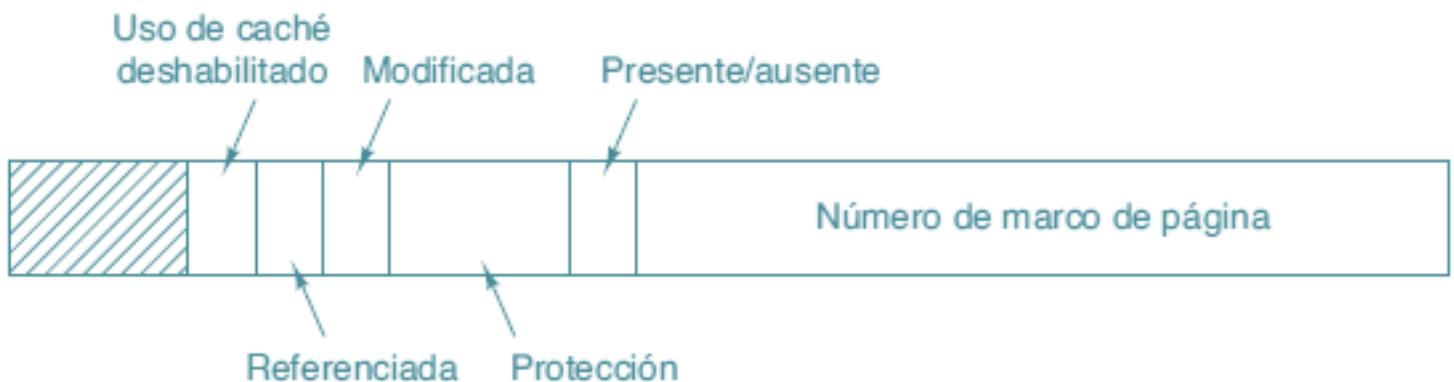
- ✓ En sentido matemático, la tabla de páginas es una función donde el número de página virtual es un argumento y el número de marco físico es un resultado.
- ✓ El número de página virtual se utiliza como índice en la tabla de páginas para buscar el número de marco en la memoria.
- ✓ El número de la pagina se reemplaza por el número del marco.



Estructura de una entrada de la tabla de páginas

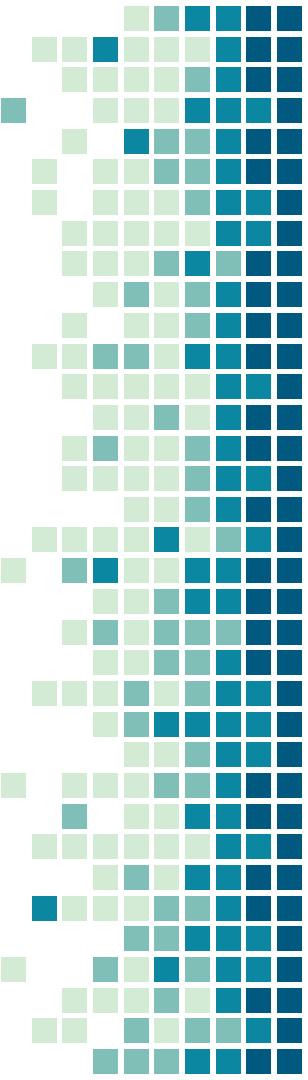


El tamaño va a variar entre cada sistema y la arquitectura del mismo, sin embargo, los valores comunes son 32 y 64 bits.



Algoritmos de reemplazo de página





Administración de recursos

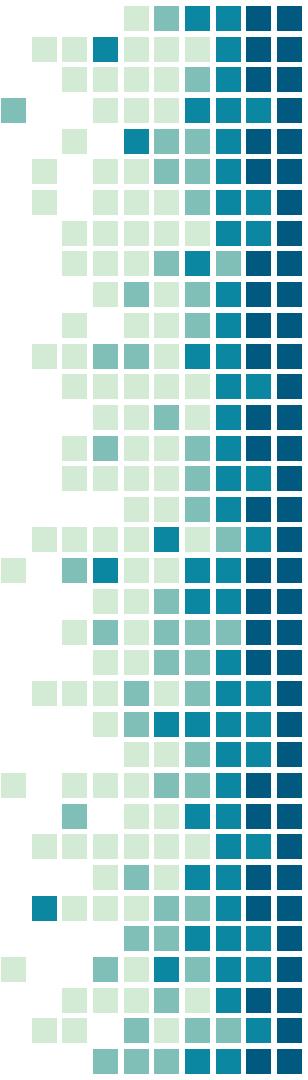
¿Qué ocurre cuando se produce un fallo de página?

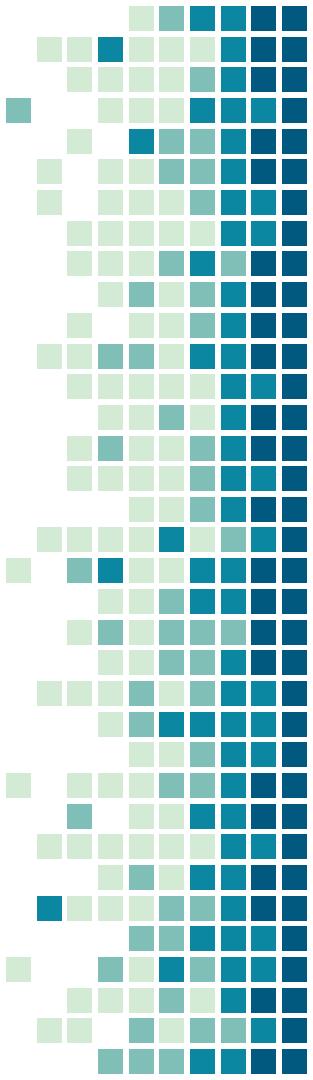


- El SO tiene que elegir una página para desalojarla y hacer espacio para la página entrante.
- Si la página se modificó, se debe de volver a escribir de lo contrario no es necesario.

Algoritmos de reemplazo: No usadas recientemente

- Se realiza por medio de estadísticas sobre el uso de páginas.
- Para esto se asocian 2 bits a cada entrada de la tabla R, M.
- R se establece cada vez que se hace referencia a la página.
- M se establece cuando se escribe en la página.
- La idea del algoritmo es evaluar las cuatro combinaciones de R y M.





No usadas recientemente

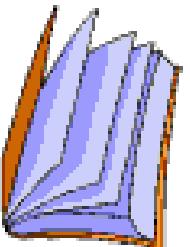
Entonces el algoritmo elimina una al azar de la menor clase posible.



- 0: No ha sido referenciada, no ha sido modificada.
- 1: No ha sido referenciada, ha sido modificada.
- 2: Ha sido referenciada, no ha sido modificada.
- 3: ha sido referenciada, ha sido modificada.

Algoritmos de reemplazo: FIFO

El SO mantiene una lista de todas las páginas actualmente en memoria, en donde la llegada más reciente está en la parte final y la menos reciente en la parte frontal.



En un fallo de página se elimina la que está la parte frontal y la nueva se agrega al final.

Algoritmos de reemplazo:

Segunda oportunidad

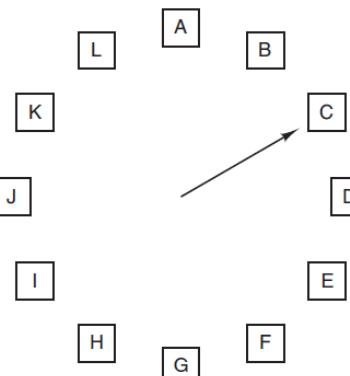
- >Es una modificación simple al algoritmo FIFO que evita descartar una página de uso frecuente.
- >Inspecciona el bit R de la página más antigua. Si es 0, la página no se ha utilizado y se desecha.
- >Si el bit es 1, entonces se borra y la página es puesta como si fuera entrando a la lista.

¿Qué ocurre si todos tienen el bit en 1?



Reloj

- ✓ Mantiene todos los marcos de página en una lista circular en forma de reloj. La manecilla apunta a la más antigua.
- ✓ Cuando ocurre un fallo, se inspecciona el bit R, si es 0 se desaloja, de lo contrario se borra y avanza la manecilla.



Algoritmos de reemplazo: menos usadas recientemente



- ✓ Se basa en la observación de que las páginas que se hayan utilizado con frecuencia probablemente se vuelvan a utilizar.
- ✓ La idea es que cuando ocurra un fallo se debe descartar la página que no se haya utilizado durante mayor tiempo.
- ✓ Se puede hacer por hardware y por software, pero por software tiene un alto costo computacional.

Algoritmos de reemplazo: menos usadas recientemente

		Página			
		0	1	2	3
0	0	0	1	1	1
	1	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0

(a)

		Página			
		0	1	2	3
0	0	0	0	1	1
	1	0	1	1	1
2	0	0	0	0	0
3	0	0	0	0	0

(b)

		Página			
		0	1	2	3
0	0	0	0	0	1
	1	0	0	1	1
2	1	1	0	1	0
3	0	0	0	0	0

(c)

		Página			
		0	1	2	3
0	0	0	0	0	0
	1	0	0	0	0
2	1	1	0	0	0
3	1	1	1	0	0

(d)

		Página			
		0	1	2	3
0	0	0	0	0	0
	1	0	0	0	0
2	1	1	0	0	0
3	1	1	1	0	0

(e)

		Página			
		0	1	2	3
0	0	0	0	0	0
	1	0	1	1	1
2	1	0	0	1	1
3	1	0	0	0	0

(f)

		Página			
		0	1	2	3
0	0	0	1	1	1
	1	0	0	1	1
2	0	0	0	1	0
3	0	0	0	0	0

(g)

		Página			
		0	1	2	3
0	0	0	1	1	0
	1	0	0	1	0
2	0	0	0	0	0
3	1	1	1	0	0

(h)

		Página			
		0	1	2	3
0	0	1	0	0	0
	1	1	1	0	1
2	1	1	1	0	0
3	1	1	1	0	0

(i)

		Página			
		0	1	2	3
0	0	1	0	0	0
	1	1	1	0	0
2	1	1	1	0	0
3	1	1	1	0	0

(j)

Se tienen cuatro páginas referenciadas de la siguiente manera: 0 1 2 3 2 1 0 3 2 3. ¿Cuál se remplazaría?

Algoritmos de reemplazo: conjunto de trabajo

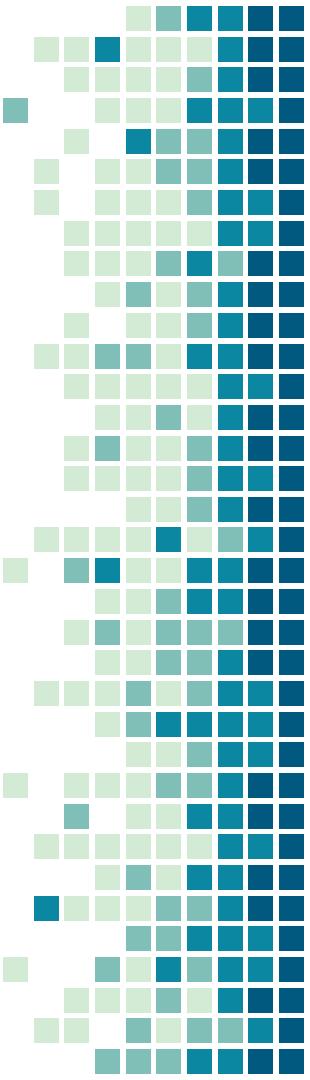


- La idea es que un conjunto de páginas estén en memoria antes de ejecutar el programa.
- Es necesario que el SO lleve el control de cuales páginas están en el conjunto de trabajo.
- El conjunto de trabajo es variable y se considera actualmente como el conjunto de páginas que se han utilizado en un tiempo menor a T. Generalmente 100 ms

Algoritmos de reemplazo: conjunto de trabajo

- La idea fundamental es remplazar una página que no esté en el conjunto de trabajo.
- También toma en cuenta el bit R.
- El caso donde se elimina la página es cuando no ha sido referenciada y el tiempo es mayor a T.

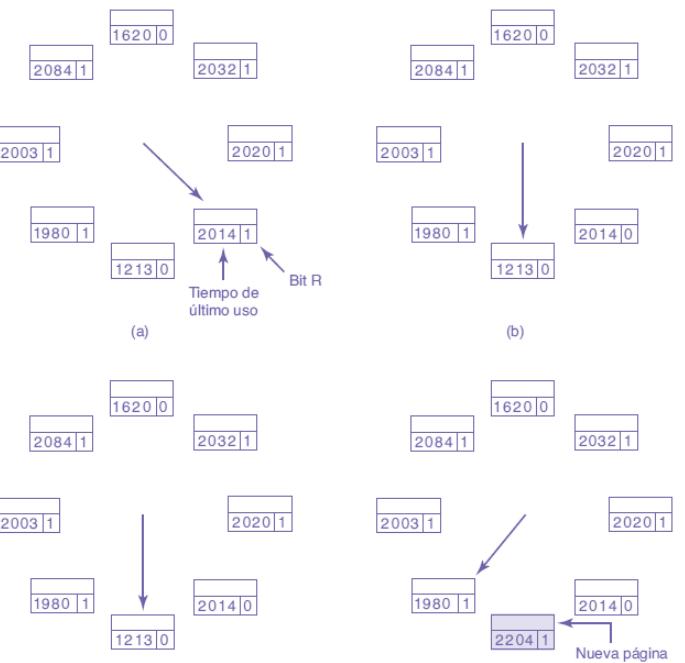




Algoritmos de reemplazo: WSClock

- Se implementa con una lista circular de marcos de páginas, similar al algoritmo de reloj.
- Se analiza el bit R y el bit M.
- En el momento del remplazo se tiene que verificar si el tiempo es mayor a T, si ha sido referenciada y si ha sido modificada.
- El caso en que se remplaza es cuando el tiempo es mayor, no ha sido referenciada y no ha sido modificada.

Algoritmos de reemplazo: WSClock



Referencias

- Tanenbaum, A. S. (2015).
Sistemas operativos modernos. Pearson Educación
- Stallings, W. (2003).
Sistemas operativos. Martin Iturbide.
- Calvaresi, D., Marinoni, M., Sturm, A., Schumacher, M., & Buttazzo, G. (2017, August).
The challenge of real-time multi-agent systems for enabling IoT and CPS.
In *Proceedings of the international conference on web intelligence* (pp. 356-364). ACM.

¿Preguntas?

Realizado por: Jason Leiton Jiménez.

Tecnológico de Costa Rica
Ingeniería en Computadores
2024

TEC