

# Administración de Memoria

## Sistemas Operativos

Departamento de Computación  
Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires

20 de abril de 2023

# Memoria Virtual: Repaso

- ▶ Memoria Virtual: Hacerle creer al proceso que dispone de más memoria de la que realmente tiene en cada momento.
- ▶ Memoria física < Memoria Virtual.
- ▶ La memoria virtual puede implementarse con diferentes técnicas, incluyendo paginación, segmentación, o una combinación de ambos.
- ▶ El tamaño de la memoria virtual depende de la capacidad de direccionamiento (cantidad de bits dispuesto para esto).
- ▶ La memoria virtual requiere soporte especial del hardware y del sistema operativo.

# Repasemos definiciones

- ▶ **Dirección virtual:** dirección lógica que usa el proceso.
- ▶ **Dirección física:** dirección real en memoria física.
- ▶ **Marcos de página:** “pedazos” de tamaño fijo en que se divide la memoria física.
- ▶ **Páginas:** “pedazos” en los que se divide la memoria virtual, del mismo tamaño que los marcos de página.
- ▶ **Fragmentación:** memoria que se vuelve inutilizable.
- ▶ **Page fault:** evento que ocurre cuando una página solicitada no está en memoria y debe ser copiada desde el disco.

# Ejercicios:

1. Tengo direcciones de 16 bits. ¿Cuánta memoria virtual puedo tener direccionando cada byte? ¿Cuánta física?

16 bits  $\rightarrow 2^{16}$  direcciones diferentes.

Queda la misma cantidad de memoria física que virtual.

2. Tengo 65536 ( $2^{16}$ ) bytes de memoria física dividida en unidades de direccionamiento de 16 bits. ¿Cuántos bits necesito para direccionar?

16 bits = 2 bytes

$$\frac{tam\_mem\_en\_bytes}{tam\_bloque\_en\_bytes} = \frac{2^{16}}{2} = 2^{15} \text{ bloques}$$

Necesito 15 bits

# Paginación y Segmentación: Repaso

Paginación	Segmentación
<ul style="list-style-type: none"><li>• <i>Pedazos</i> del mismo tamaño.</li></ul>	<ul style="list-style-type: none"><li>• <i>Pedazos</i> de tamaño variable.</li></ul>
<ul style="list-style-type: none"><li>• El usuario ni se entera del particionamiento.</li></ul>	<ul style="list-style-type: none"><li>• El usuario tiene que saber de los tamaños y límites de los segmentos.</li></ul>
<ul style="list-style-type: none"><li>• Favorece fragmentación interna.</li></ul>	<ul style="list-style-type: none"><li>• Favorece fragmentación externa.</li></ul>

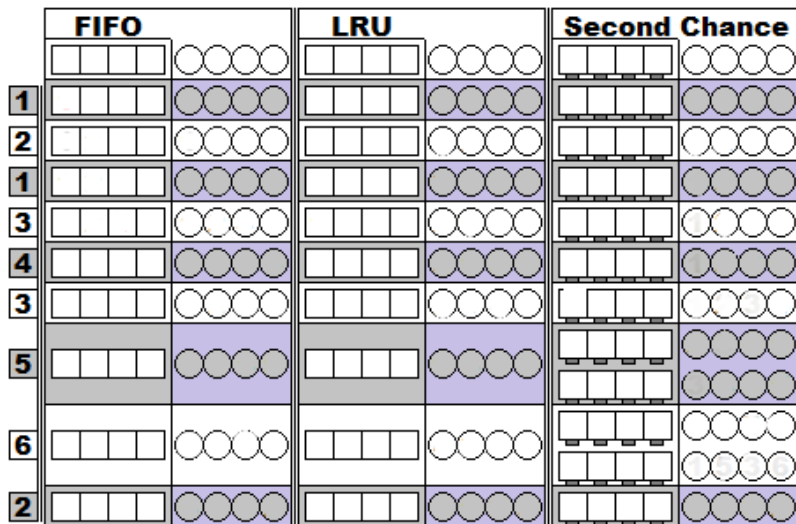
- ▶ En un enfoque combinado, el espacio de direcciones virtuales se divide en segmentos de tamaño variable, y los segmentos se dividen en páginas de tamaño fijo.

- ▶ ¿Qué pasa si no hay lugar en la memoria? (por lo general,  $\text{MEM\_VIRTUAL} > \text{MEM\_FISICA}$ ).
- ▶ Algoritmos de remoción:
  - ▶ **FIFO:** La clásica de siempre.
  - ▶ **LRU:** Desalojo la página que hace más tiempo que no se usa.
  - ▶ **Segunda oportunidad:** Si fue referenciada, le doy otra oportunidad.
  - ▶ **Not Recently Used:** Primero desalojo las que no fueron referenciadas ni modificadas. Después, las solamente referenciadas y por último las modificadas.

# Ejercicio:

- ▶ Tengo un sistema con 6 páginas y sólo 4 marcos de página. La memoria comienza vacía.
- ▶ Llegan los siguientes pedidos de memoria (número de página) en ese orden:
- ▶ 1, 2, 1, 3, 4, 3, 5, 6, 2
- ▶ Indique qué página se desaloja tras cada pedido utilizando los algoritmos FIFO, LRU y Second Chance y calcule el *hit-rate* en cada caso.
- ▶  $Hit-Rate = \frac{\text{Páginas que pedí y ya estaban cargadas en memoria}}{\text{páginas totales pedidas}}$

# Solución





# Solución

	<b>FIFO</b>		<b>LRU</b>		<b>Second Chance</b>	
	<div><div></div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div><div></div></div>
1	<div><div>1</div><div></div><div></div><div></div></div>	<div><div>1</div><div></div><div></div><div></div></div>	<div><div>1</div><div></div><div></div><div></div></div>	<div><div>1</div><div></div><div></div><div></div></div>	<div><div>1</div><div></div><div></div><div></div></div>	<div><div>1</div><div></div><div></div><div></div></div>
2	<div><div>1</div><div>2</div><div></div><div></div></div>	<div><div>1</div><div>2</div><div></div><div></div></div>	<div><div>1</div><div>2</div><div></div><div></div></div>	<div><div>1</div><div>2</div><div></div><div></div></div>	<div><div>1</div><div>2</div><div></div><div></div></div>	<div><div>1</div><div>2</div><div></div><div></div></div>
1	<div><div>1</div><div>2</div><div></div><div></div></div>	<div><div>1</div><div>2</div><div></div><div></div></div>	<div><div>1</div><div>2</div><div></div><div></div></div>	<div><div>2</div><div>1</div><div></div><div></div></div>	<div><div>1</div><div>2</div><div></div><div></div></div>	<div><div>1</div><div>2</div><div></div><div></div></div>
3	<div><div>1</div><div>2</div><div>3</div><div></div></div>	<div><div>1</div><div>2</div><div>3</div><div></div></div>	<div><div>1</div><div>2</div><div>3</div><div></div></div>	<div><div>2</div><div>1</div><div>3</div><div></div></div>	<div><div>1</div><div>2</div><div>3</div><div></div></div>	<div><div>1</div><div>2</div><div>3</div><div></div></div>
4	<div><div>1</div><div>2</div><div>3</div><div>4</div></div>	<div><div>1</div><div>2</div><div>3</div><div>4</div></div>	<div><div>1</div><div>2</div><div>3</div><div>4</div></div>	<div><div>2</div><div>1</div><div>3</div><div>4</div></div>	<div><div>1</div><div>2</div><div>3</div><div>4</div></div>	<div><div>1</div><div>2</div><div>3</div><div>4</div></div>
3	<div><div>1</div><div>2</div><div>3</div><div>4</div></div>	<div><div>1</div><div>2</div><div>3</div><div>4</div></div>	<div><div>1</div><div>2</div><div>3</div><div>4</div></div>	<div><div>2</div><div>1</div><div>4</div><div>3</div></div>	<div><div>1</div><div>2</div><div>3</div><div>4</div></div>	<div><div>1</div><div>2</div><div>3</div><div>4</div></div>
5	<div><div>5</div><div>2</div><div>3</div><div>4</div></div>	<div><div>2</div><div>3</div><div>4</div><div>5</div></div>	<div><div>1</div><div>5</div><div>3</div><div>4</div></div>	<div><div>1</div><div>4</div><div>3</div><div>5</div></div>	<div><div>1</div><div>2</div><div>3</div><div>4</div></div>	<div><div>2</div><div>3</div><div>4</div><div>1</div></div>
	<div><div></div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div><div></div></div>	<div><div>1</div><div>5</div><div>3</div><div>4</div></div>	<div><div>3</div><div>4</div><div>1</div><div>5</div></div>
6	<div><div>5</div><div>6</div><div>3</div><div>4</div></div>	<div><div>3</div><div>4</div><div>5</div><div>6</div></div>	<div><div>6</div><div>5</div><div>3</div><div>4</div></div>	<div><div>4</div><div>3</div><div>5</div><div>6</div></div>	<div><div>1</div><div>5</div><div>3</div><div>4</div></div>	<div><div>4</div><div>1</div><div>5</div><div>3</div></div>
	<div><div></div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div><div></div></div>	<div><div>1</div><div>5</div><div>3</div><div>6</div></div>	<div><div>1</div><div>5</div><div>3</div><div>6</div></div>
2	<div><div>5</div><div>6</div><div>2</div><div>4</div></div>	<div><div>4</div><div>5</div><div>6</div><div>2</div></div>	<div><div>6</div><div>5</div><div>3</div><div>2</div></div>	<div><div>3</div><div>5</div><div>6</div><div>2</div></div>	<div><div>2</div><div>5</div><div>3</div><div>6</div></div>	<div><div>5</div><div>3</div><div>6</div><div>2</div></div>

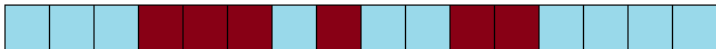
- ▶ Hit-Rate (FIFO) = 2 / 9
- ▶ Hit-Rate (LRU) = 2 / 9
- ▶ Hit-Rate (SC) = 2 / 9

# Asignación de memoria

- ▶ ¿Qué porción de memoria conviene asignar?
- ▶ Algoritmos de elección de bloque libre:
  - ▶ **First fit:** La primera sección de memoria contigua del tamaño necesario.
  - ▶ **Best fit:** De todas las secciones de tamaño mayor o igual al tamaño necesario, tomo la más chica.
  - ▶ **Worst fit:** Mejor tomo la más grande.
  - ▶ **Quick fit:** Se usan listas de bloques de determinados tamaños, para accederlos más rápido.

# Ejercicio:

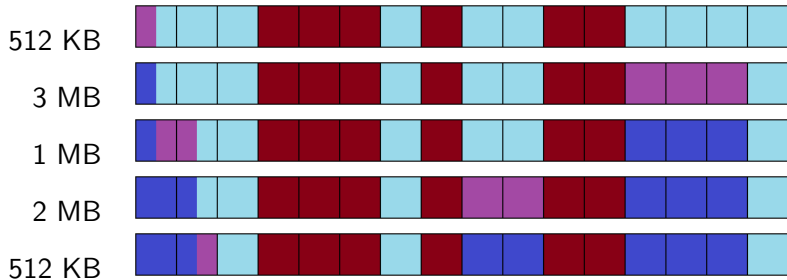
- ▶ Tengo un sistema con 16 MB de memoria **sin particionar** que direcciona a byte. El estado actual de la memoria es el siguiente (cuadrado= 1MB):



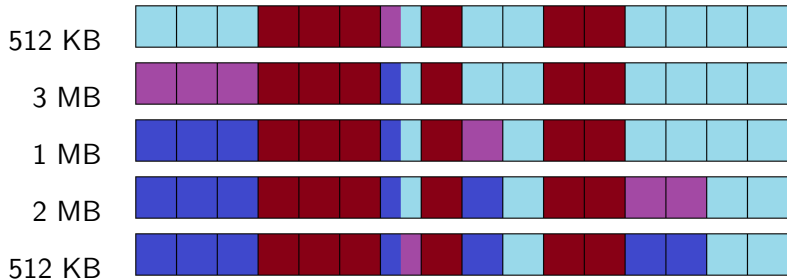
Llegan los siguientes pedidos de memoria en ese orden:  
512 KB, 3 MB, 1 MB, 2MB, 512 KB.

- ▶ Indique qué bloques se asignan para cada pedido utilizando first-fit, best-fit y worst-fit.

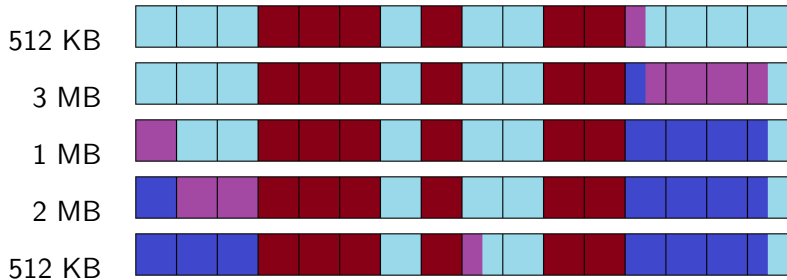
# Solución First-Fit



# Solución Best-Fit



# Solución Worst-Fit



# Entonces, ¿Cuál es mejor?

