



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto **Tecnológico**
de Aguascalientes

INSTITUTO TECNOLÓGICO DE AGUASCALIENTES

INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES

SISTEMAS OPERATIVOS 1

CLASE 11:00AM-12:00 PM

UNIDAD 3

ALGORITMOS:

EQUIPO 6:

Ángel Omar Negrete Demetrio
David Antonio Rangel García
Armando Reyes Martínez
Humberto Emmanuel Rosales Martínez
Alondra Rubí Valdés Mora

DOCENTE: RUTH MAYELI
PONCE ROSALES

FECHA:
5 DE ABRIL DEL 2024

Considerar un sistema con intercambio, en el que la memoria posee particiones libres de tamaño fijo: 1000kb, 400kb, 1800kb, 700kb, 900kb, 1200 kb y 1500kb. Estos huecos están dispuestos en el orden dado. Se tienen 3 procesos de tamaños 1600kb, 64kb, 1200kb y 900kb. Para los algoritmos: **primer ajuste, mejor ajuste, peor ajuste y siguiente ajuste:**

1. **Particiones Libres:**

- 1000kb, 400kb, 1800kb, 700kb, 900kb, 1200 kb, 1500kb

2. **Procesos a Asignar:**

- Proceso A: 1600kb
- Proceso B: 64kb
- Proceso C: 1200kb
- Proceso D: 900kb

Primer Ajuste:

Particiones	Procesos
1000 kb	B
400 kb	
1800 kb	A
700kb	
900 kb	D
1200 kb	C
1500 kb	

procesos:

Segmento	Fragmentación	Hueco
(A) 1600 kb	1800	200
(B) 64 kb	700	636
(C)1200 kb	1200	0
(D)900 kb	1500	600

Mejor Ajuste:

Particiones	Procesos
1000 kb	
400 kb	B
1800 kb	A
700kb	
900 kb	D
1200 kb	C
1500 kb	

procesos:

Segmento	Fragmentación	Hueco
(A) 1600 kb	1800	200

(B) 64 kb	400	336
(C)1200 kb	1200	0
(D)900 kb	900	0

Peor Ajuste:

Particiones	Procesos
1000 kb	
400 kb	
1800 kb	A
700kb	
900 kb	
1200 kb	D
1500 kb	B y C

procesos:

Segmento	Fragmentación	Hueco
(A) 1600 kb	1800	200
(B) 64 kb	1500	1436
(C)1200 kb	1500	300
(D)900 kb	1200	300

Siguiente Ajuste:

Particiones	Procesos
1000 kb	
400 kb	
1800 kb	A
700kb	B
900 kb	
1200 kb	C
1500 kb	D

procesos:

Segmento	Fragmentación	Hueco
(A) 1600 kb	1800	200
(B) 64 kb	700	636
(C)1200 kb	1200	0
(D)900 kb	1500	600

¿Qué algoritmo aprovecha mejor la memoria? El algoritmo de mejor ajuste es el que aprovecha mejor la memoria.

Considerar un sistema con intercambio, en el que la memoria posee particiones libres de tamaño fijo: 245kb, 600kb, 500kb, 70kb, 64kb, 800 kb y 1200kb. Estos huecos están dispuestos en el orden dado. Se tienen 4 procesos de tamaños 1200kb, 8kb, 264kb y 500kb. Para los algoritmos: **primer ajuste, mejor ajuste, peor ajuste y siguiente ajuste:**

3. **Particiones Libres:**

- 245kb, 600kb, 500kb, 70kb, 64kb, 800 kb, 1200kb

4. **Procesos a Asignar:**

- Proceso A: 1200kb
- Proceso B: 8kb
- Proceso C: 264kb
- Proceso D: 500kb

Primer Ajuste:

Particiones	Procesos
245 kb	B
600kb	C
500 kb	D
70kb	
64 kb	
800 kb	
1200 kb	A

procesos:

Segmento	fragmentacion	Hueco
(A) 1200 kb	1200	0
(B) 8 kb	245	237
(C)264 kb	600	336
(D)500 kb	500	0

Mejor Ajuste:

Particiones	Procesos
245 kb	
600kb	D
500 kb	C
70kb	
64 kb	B
800 kb	
1200 kb	A

procesos:

Segmento	fragmentacion	Hueco
(A) 1200 kb	1200	0

(B) 8 kb	64	56
(C)264 kb	500	236
(D)500 kb	500	0

Peor Ajuste:

Particiones	Procesos
245 kb	
600kb	D
500 kb	
70kb	
64 kb	
800 kb	B y C
1200 kb	A

procesos:

Segmento	fragmentacion	Hueco
(A) 1200 kb	1200	0
(B) 8 kb	800	528
(C)264 kb	800	
(D)500 kb	600	100

Siguiente Ajuste:

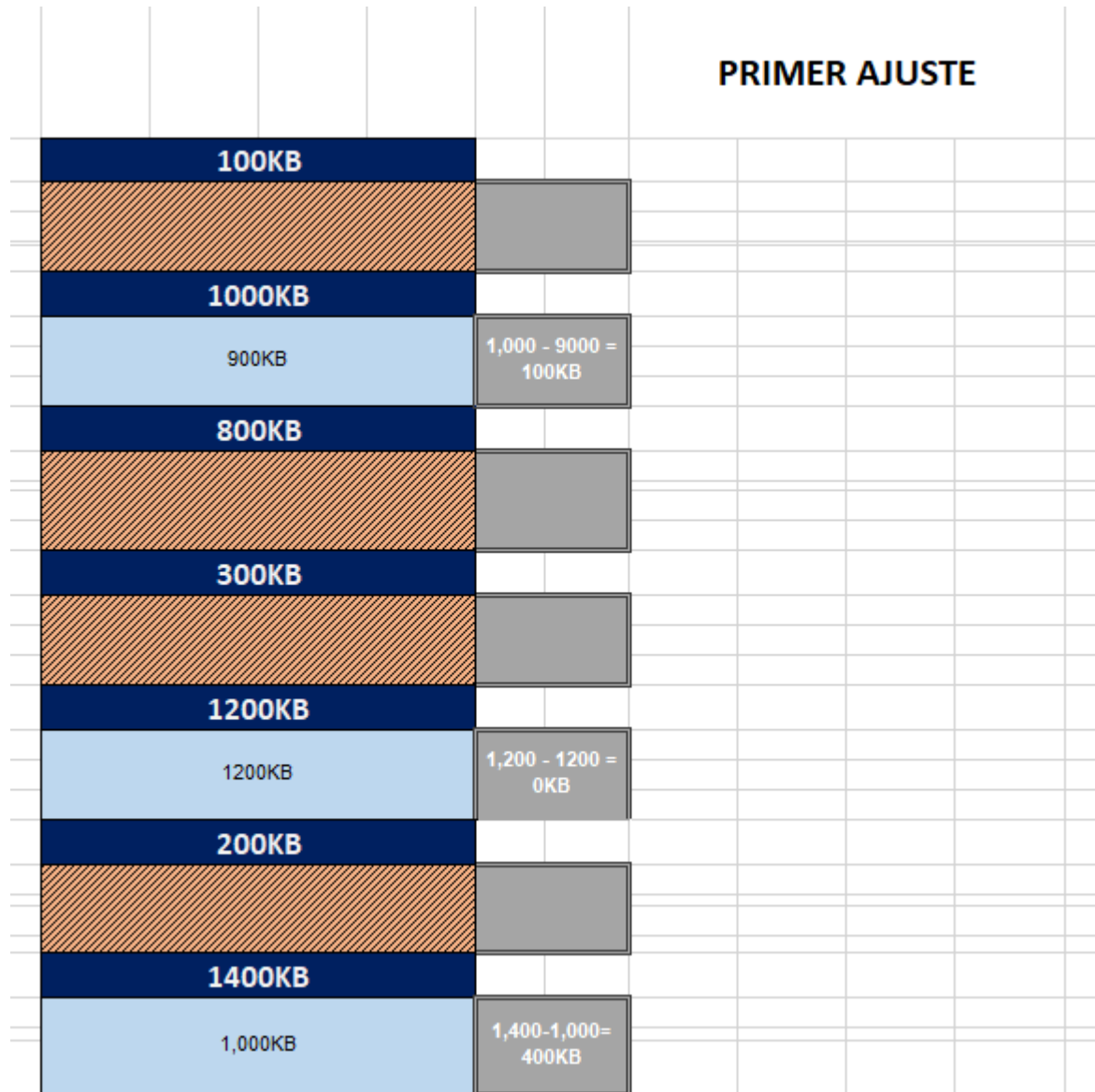
Particiones	Procesos
245 kb	B
600kb	C
500 kb	D
70kb	
64 kb	
800 kb	
1200 kb	A

procesos:

Segmento	fragmentacion	Hueco
(A) 1200 kb	1200	0
(B) 8 kb	245	237
(C)264 kb	600	336
(D)500 kb	500	0

¿Qué algoritmo aprovecha mejor la memoria? Los algoritmos de primer ajuste y próximo ajuste son los que aprovechan mejor la memoria.

Considerar un sistema con intercambio, en el que la memoria posee particiones libres de tamaño fijo: 1400kb, 200kb, 1200kb, 300kb, 800kb, 1000 kb y 100kb. Estos huecos están dispuestos en el orden dado. Se tienen 3 procesos de tamaños 1000kb, 1200kb y 900kb. Para los algoritmos: **primer ajuste, mejor ajuste, peor ajuste y siguiente ajuste:**



MEJOR AJUSTE

100KB	
1000KB	
1000KB	$1,000 - 1000 = 0\text{KB}$
800KB	
300KB	
1200KB	
1200KB	$1,200 - 1200 = 0\text{KB}$
200KB	
1400KB	
900kb	$1,400 - 900 = 500\text{KB}$



SIGUIENTE AJUSTE


$$1,000 - 900 = 100\text{KB}$$
$$1,000 - 900 = 100\text{KB}$$

$$1,200 - 1200 = 0\text{KB}$$
$$1,200 - 1200 = 0\text{KB}$$
$$1,400 - 1,000 = 400\text{KB}$$
$$1,400 - 1,000 = 400\text{KB}$$

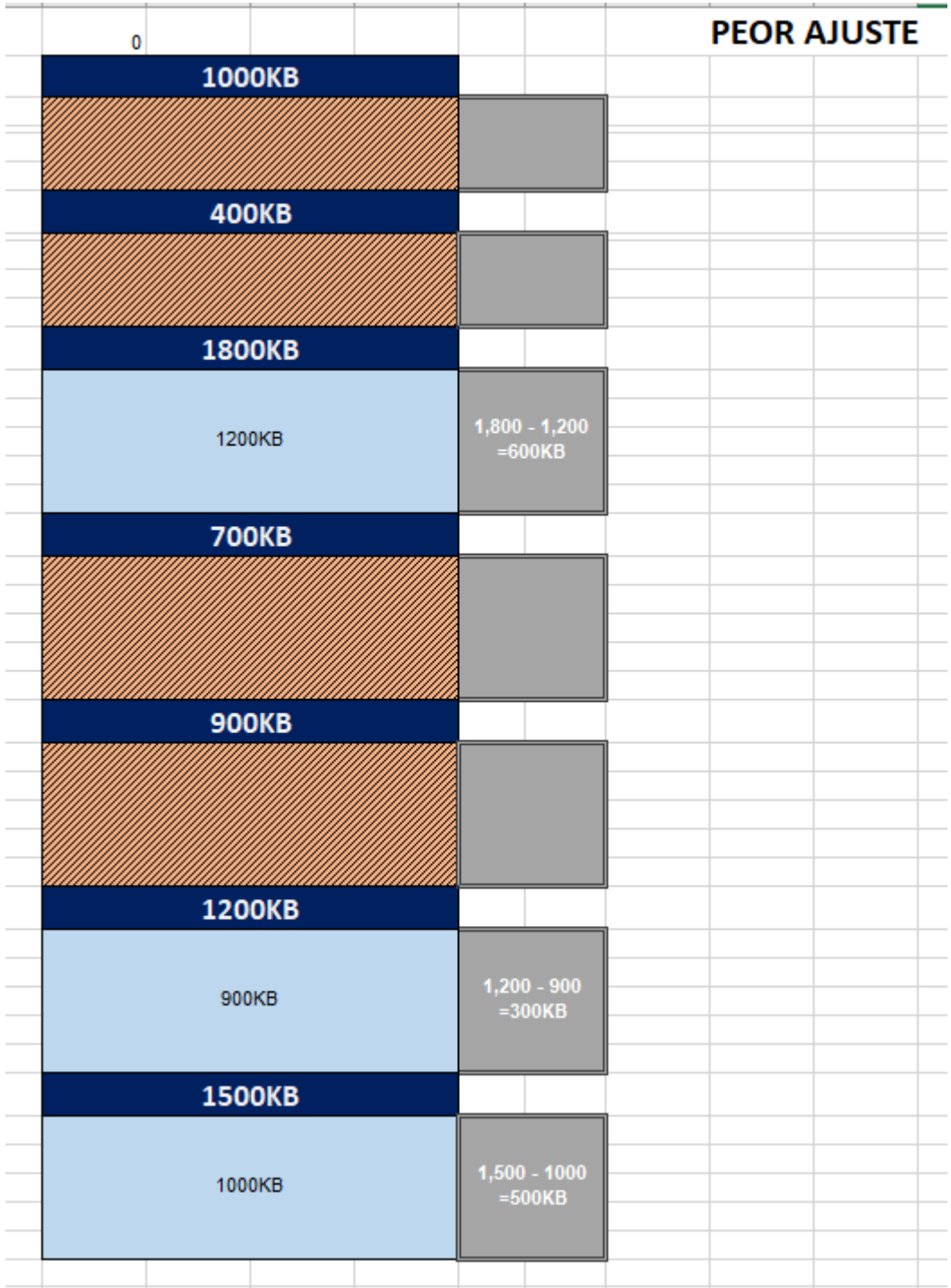
--	--

Considerar un sistema con intercambio, en el que la memoria posee particiones libres de tamaño fijo: 1000kb, 400kb, 1800kb, 700kb, 900kb, 1200 kb y 1500kb. Estos huecos están dispuestos en el orden dado. Se tienen 3 procesos de tamaños 1200kb, 1000kb y 900kb. Para los algoritmos: **primer ajuste, mejor ajuste, peor ajuste y siguiente ajuste.**

PRIMER AJUSTE			
1000KB			
400KB			
1800KB			
700KB			
900KB			
900KB	900-900 = 0KB		
1200KB			
1000KB	1,200 - 1,000 = 200KB		
1500KB			
1200KB	1,500-1,200 = 300		

MEJOR AJUSTE

1000KB	
1000KB	1,000 - 1,000 =0KB
400KB	
1800KB	
700KB	
900KB	
900KB	900 - 900 =0KB
1200KB	
1200KB	1,200 - 1,200 =0KB
1500KB	



SIGUIENTE AJUSTE

1200KB

$$1,500 - 1,200 = 300\text{KB}$$

10

REPORTE DE VIDEO

El video proporciona una exploración profunda de los mecanismos de gestión de memoria en sistemas operativos, examinando específicamente tres enfoques clave: gestión de memoria estática (MFT), gestión de memoria con particiones dinámicas (MBT) y gestión de memoria virtual.

En el caso de la gestión de memoria estática (MFT), se destaca cómo el sistema asigna particiones fijas de memoria a diferentes procesos. Esto implica un diseño predefinido de la memoria en bloques estáticos, donde cada partición se asigna según un algoritmo de ubicación determinado. El texto resalta cómo esta estrategia puede conducir a problemas de fragmentación interna cuando los procesos no llenan completamente las particiones asignadas, dejando espacios no utilizados dentro de las particiones.

Por otro lado, la gestión de memoria con particiones dinámicas (MBT) permite una asignación más flexible de la memoria, adaptándose dinámicamente a las necesidades de los procesos. Este enfoque suele optimizar el uso de la memoria al asignar particiones según la demanda de los procesos, minimizando así la fragmentación interna y aprovechando mejor los recursos disponibles.

El video también aborda la gestión de memoria virtual, un concepto fundamental en sistemas modernos que separa la memoria física de la memoria lógica de los procesos. Esto se logra mediante el uso de paginación u otros métodos, donde las direcciones virtuales de los procesos se traducen a direcciones físicas en la memoria real. Se menciona la importancia de entender cómo se calcula la fragmentación interna en este contexto, así como la determinación de los tamaños de direcciones reales y virtuales para garantizar un funcionamiento eficiente del sistema.

En resumen, el texto profundiza en las complejidades de la gestión de memoria en sistemas operativos, destacando las diferencias entre enfoques estáticos y dinámicos, así como la importancia de la gestión eficiente de la memoria para optimizar el rendimiento del sistema y la capacidad de carga de procesos.