

Sistemas Operativos

SOLUCIONES DE LOS PROBLEMAS DE MEMORIA REAL

1. Pasos a seguir:

- Se necesitan 20 bits para almacenar direcciones físicas (porque la MP tiene 1MiB).
- Si el tamaño máximo de un proceso es de 512 KiB entonces se necesitan 9+10 bits para almacenar direcciones lógicas (porque 512 necesita 9 bits y 1KiB 10 bits).
- Si la tabla de página tiene como máximo 32 entradas entonces los procesos tienen como máximo 32 páginas (que son 5 bits), por tanto se necesitan 5 bits para almacenar páginas (en las direcciones lógicas).
- Dado que las direcciones lógicas son de 19 bits y 5 son para páginas, los 14 restantes son para desplazamiento. **Por tanto, las direcciones lógicas se dividen en 5 y 14**
- Dado que el tamaño de páginas y marcos coincide entonces el número de bits destinado a desplazamiento en las direcciones lógicas y físicas también coincide (en este caso 14 bits). Además, sabemos que las direcciones físicas son de 20 bits, **por tanto, las direcciones físicas se dividen en 6 y 14**

L:

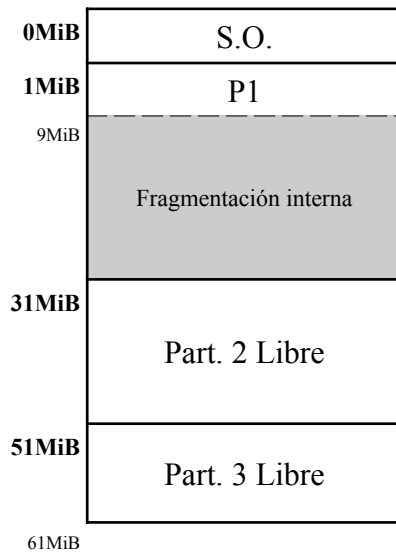
5	14
---	----

 F:

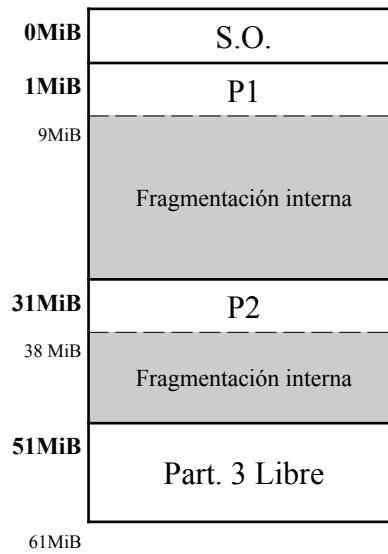
6	14
---	----

2. Se indica a continuación el estado de la MP en los instantes de tiempo donde se producen cambios (el tamaño de las particiones no está a escala):

T=0 (se carga P1)



T=2 (se carga P2)

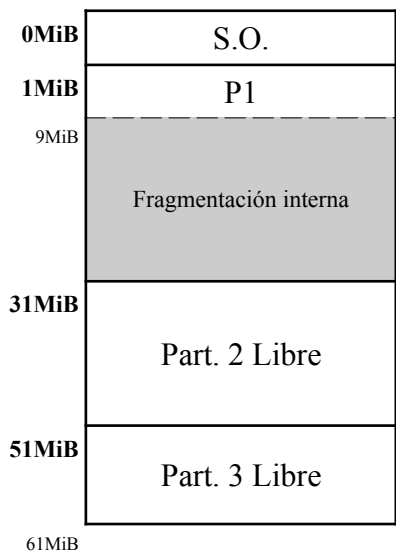


T=4 (Intento carga P3)

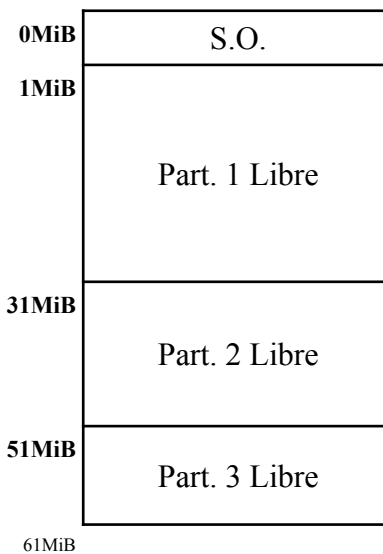
No se puede cargar P3, se espera a que haya sitio

T=7 (finaliza P2)

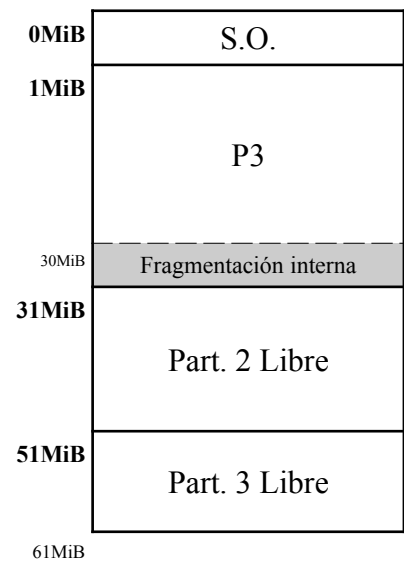
Todavía no se puede cargar P3



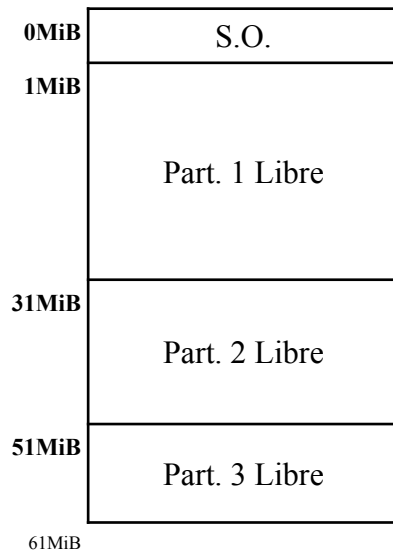
T=10 (finaliza P1)



T=10 (Se carga P3)



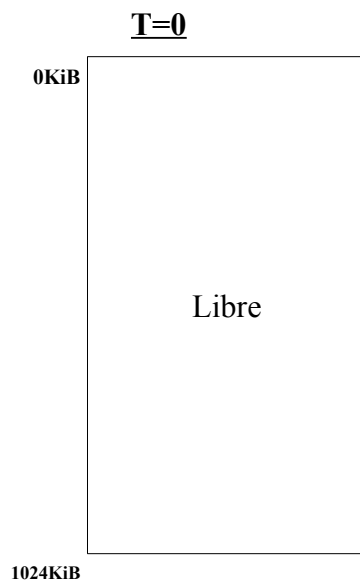
T=18 (finaliza P3)



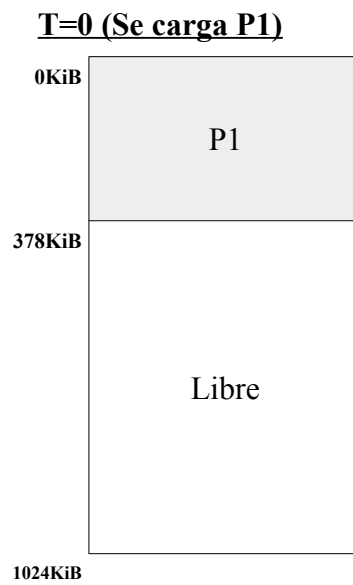
La fragmentación interna en el instante 9 es la misma que en el instante 7 y es:

$$31\text{MiB} - 9\text{ MiB} = 22\text{ MiB} \rightarrow \mathbf{22.528\text{ KiB}}$$

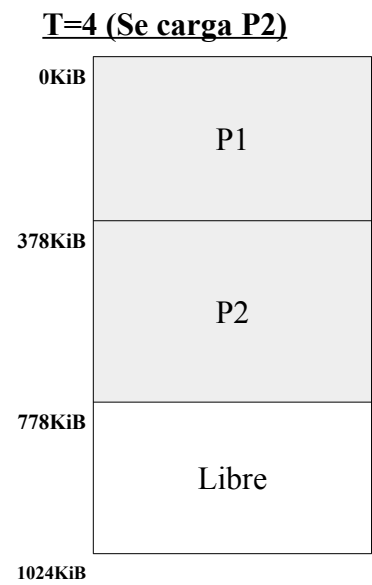
3. Se indica a continuación el estado de la MP en los instantes de tiempo donde se producen cambios (el tamaño de las particiones no está a escala):



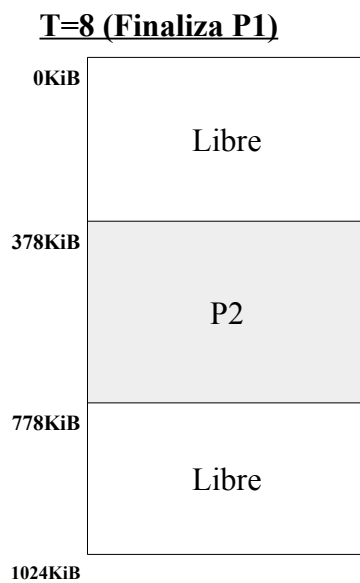
LH → ([0KiB, 1024KiB])
LNH → Vacía



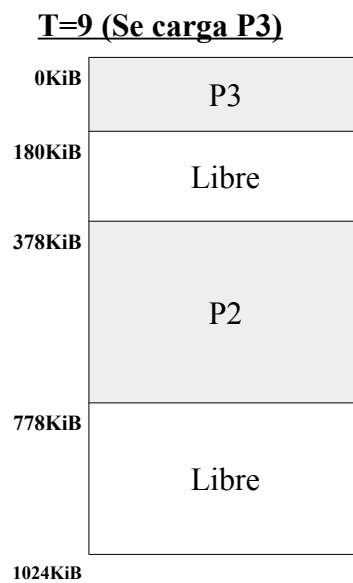
LH → ([378KiB, 646KiB])
LNH → ([0KiB, 378KiB, P1])



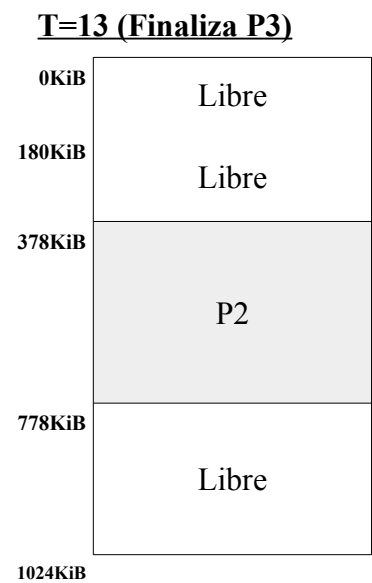
LH → ([778KiB, 246KiB])
LNH → ([0KiB, 378KiB, P1],
[378KiB, 400KiB, P2])



LH → ([0KiB, 378KiB],
[778KiB, 246KiB])
LNH → ([378KiB, 400KiB, P2])

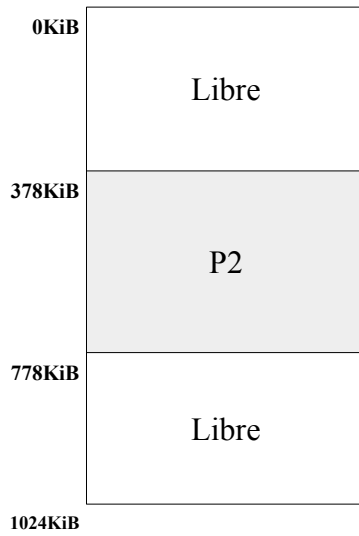


LH → ([778KiB, 246KiB],
[180KiB, 198KiB])
LNH → ([0KiB, 180KiB, P3],
[378KiB, 400KiB, P2])



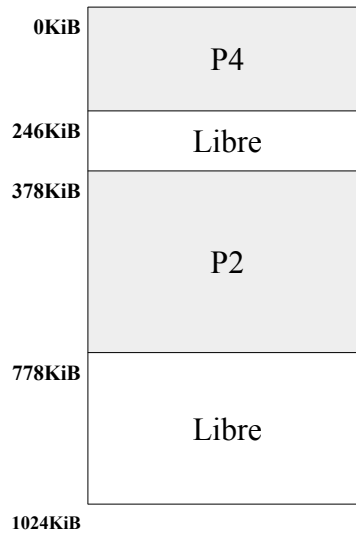
LH → ([778KiB, 246KiB],
[180KiB, 198KiB],
[0KiB, 180KiB])
LNH → ([378KiB, 400KiB, P2])

T=13 (Condensación de huecos)



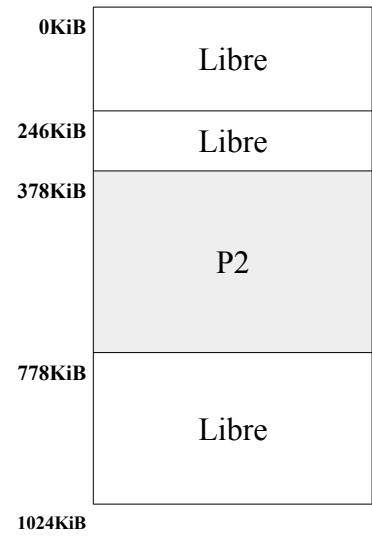
LH → ([0KiB, 378KiB],
[778KiB, 246KiB])
LNH → ([378KiB, 400KiB, P2])

T=16 (Se carga P4)



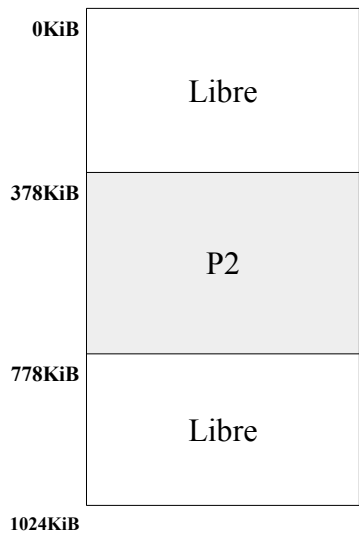
LH → ([778KiB, 246KiB],
[246KiB, 132KiB])
LNH → ([0KiB, 246KiB, P4],
[378KiB, 400KiB, P2])

T=19 (Finaliza P4)



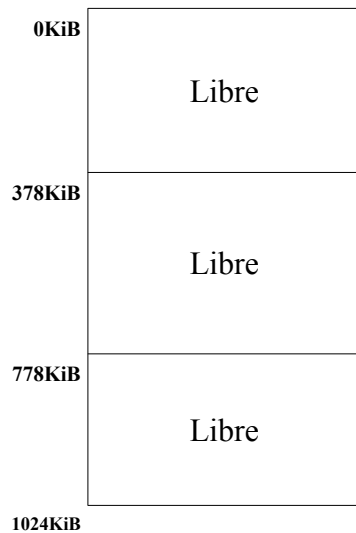
LH → ([778KiB, 246KiB],
[0KiB, 246KiB],
[246KiB, 132KiB])
LNH → ([378KiB, 400KiB, P2])

T=19 (Condensación de huecos)



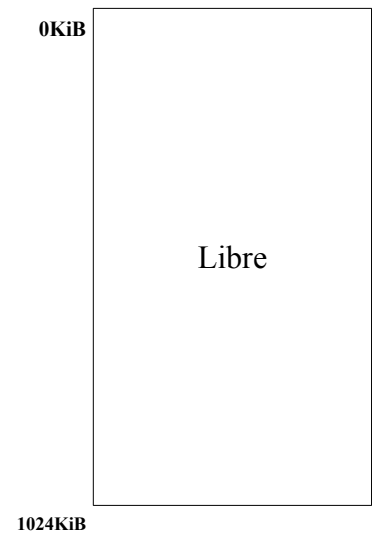
LH → ([0KiB, 378KiB],
[778KiB, 246KiB])
LNH → ([378KiB, 400KiB, P2])

T=20 (Finaliza P2)



LH → ([378KiB, 400KiB],
[0KiB, 378KiB],
[778KiB, 246KiB])
LNH → Vacía

T=20 (Condensación de huecos)



LH → ([0KiB, 1024KiB])
LNH → Vacía

La fragmentación externa máxima será el momento en que haya un mayor número de huecos (**tras haber hecho la condensación**). Por tanto, **la fragmentación externa máxima es 2** y se produce en los intervalos de tiempo: [8, 19]

4. Se indica a continuación el estado de la MP en los instantes de tiempo donde se producen cambios (el tamaño de las particiones no está a escala):

T=0

LH → ([512KiB, 3584KiB])
LNH → Vacía

T=0 (Se carga P1)

LH → ([1912KiB, 2184KiB])
LNH → ([512KiB, 1400KiB, P1])

T=0 (Se carga P2)

LH → ([3412KiB, 684KiB])
LNH → ([512KiB, 1400KiB, P1],
[1912KiB, 1500KiB, P2])

T=5 (Compactación)

QUEDA IGUAL

T=6 (Fin P2)

LH → ([3412KiB, 684KiB],
[1912KiB, 1500 KiB])
LNH → ([512KiB, 1400KiB, P1])

T=6 (Condensación)

LH → ([1912KiB, 2184KiB])
LNH → ([512KiB, 1400KiB, P1])

T=6 (Se carga P3)

LH → ([2412KiB, 1684KiB])
LNH → ([512KiB, 1400KiB, P1],
[1912KiB, 500KiB, P3])

T=10 (Fin P1)

LH → ([512KiB, 1400KiB],
[2412KiB, 1684KiB])
LNH → ([1912KiB, 500KiB, P3])

T=10 (Compactación)

LH → ([1012KiB, 3084KiB])
LNH → ([512KiB, 500KiB, P3])

T=10 (Se carga P4)

LH → ([3112KiB, 984KiB])
LNH → ([512KiB, 500KiB, P3],
[1012KiB, 2100KiB, P4])

T=13 (Fin P3)

LH → ([512KiB, 500KiB],
[3112KiB, 984KiB])
LNH → ([1012KiB, 2100KiB, P4])

T=15 (Compactación)

LH → ([2612KiB, 1484KiB])
LNH → ([512KiB, 2100KiB, P4])

T=20 (Compactación)

QUEDA IGUAL

T=23 (Fin P4)

LH → ([2612KiB, 1484KiB],
[512KiB, 2100KiB])
LNH → Vacía

T=23 (Condensación)

LH → ([512KiB, 3584KiB])
LNH → Vacía

5. Pasos a seguir:

- Si la MP es de 2MiB entonces se necesitan 21 bits ($1 + 20$) para las direcciones físicas.
- Si la MP tiene 1024 marcos entonces se necesitan 10 bits para marcos y, por tanto, habrá 11 para desplazamientos. **Por tanto las direcciones físicas se dividen en 10 y 11**
- Dado que el tamaño de páginas y marcos coincide entonces el número de bits destinado a desplazamiento en las direcciones lógicas y físicas también coincide. Por tanto, las direcciones lógicas tienen 11 bits destinados a desplazamientos.
- Como las direcciones lógicas son de 16 bits y 11 son para desplazamientos, **las direcciones lógicas se dividen en 5 y 11**
- Comparamos la dirección lógica 10240 con el tamaño del proceso para ver si es una dirección válida. ¿Es $10240 < \text{tamaño proceso}$? La respuesta es sí porque el tamaño del proceso es de, al menos, $9 \times 2\text{KiB}$ bytes (el proceso tiene 10 páginas y sabemos que las 9 primeras están completas, ocupando cada una $2\text{KiB} \rightarrow 11$ bits).
- Pasamos la dirección lógica 10240 a binario (101000000000000), la escribimos en 16 bits y lo dividimos en 5 y 11:

L:

0010 1	000 0000 0000
--------	---------------

- La dirección lógica 10240 se corresponde con la página 00101 (5 en decimal) del proceso y desplazamiento cero.
- Vamos a la entrada 5 de la TP del proceso y averiguamos que la página 5 está en el marco 10 (en decimal).
- Pasamos 10 a binario (1010), lo ajustamos a 10 bits (que son los bits destinados a marcos en las direcciones físicas) y añadimos a éstos (a su derecha) los 11 bits del desplazamiento (0 en este caso):

R:

0 0000 0101 0	000 0000 0000
---------------	---------------

6. Pasos a seguir:

- Se nos dice que la máxima dirección de memoria a generar por la CPU es la 131071. Como sabemos que la CPU sólo puede generar direcciones lógicas (al ser un esquema de paginación simple) ya sabemos que los procesos pueden tener un máximo de 131072 bytes.
- Calculamos cuántos bits son necesarios para almacenar 131072 bytes (17 bits) por lo que ya sabemos que las direcciones lógicas son de 17 bits.
- Si el tamaño de los marcos de página es 1KiB entonces se emplean 10 bits para desplazamientos, tanto para las direcciones lógicas como para las físicas. **Por tanto, las direcciones lógicas se dividen en 7 y 10**
- Si un proceso puede ocupar toda la MP disponible es que en la MP hay sitio libre para el máximo tamaño de un proceso (131072 bytes o lo que es lo mismo 131072/1024 marcos, ya que sabemos que los marcos ocupan un KiB). A eso habrá que sumar lo que ocupa el S.O. que son 32KiB, es decir 32KiB/1024 marcos). Por tanto, la memoria principal tiene exactamente $131072/1024 + 32$ marcos ($128+32=160$ marcos).

7. Pasos a seguir:

- Si hay un total de 64 marcos de página entonces se utilizan 6 bits para marcos en las direcciones físicas. **Por tanto, las direcciones físicas se dividen en 6 y 4**
- Dado que el tamaño de páginas y marcos coincide entonces el número de bits destinado a desplazamiento en las direcciones lógicas y físicas también coincide. Por tanto, las direcciones lógicas tienen 4 bits destinados a desplazamientos.
- Comparamos la dirección lógica 1101 0101 (213 en decimal) con el tamaño del proceso para ver si es una dirección válida. ¿Es $213 < 247$? La respuesta es sí.
- Dividimos la dirección lógica 1101 0101 en 4 y 4:

L:

1101	0101
------	------

- La dirección lógica 1101 0101 (213 en decimal) se corresponde con la página 1101 (13 en decimal) del proceso y desplazamiento 0101 (5 en decimal).
- Vamos a la entrada 13 de la tabla de páginas del proceso y averiguamos que la página 13 está en el marco 24 (en decimal).
- Pasamos 24 a binario (11000), lo ajustamos a 6 bits (que son los bits destinados a marcos en las direcciones físicas) y añadimos a éstos (a su derecha) los 4 bits del desplazamiento (0101 en este caso):

R:

011000	0101
--------	------

8. Pasos a seguir:

- Si la MP tiene 256 bytes entonces las direcciones físicas son de 8 bits.
- Si el tamaño máximo de los segmentos es 32 bytes (5 bits) entonces se utilizan 5 bits para desplazamientos de segmento en las direcciones lógicas.
- Dado que estamos en memoria real el espacio de direcciones lógicas no puede superar al espacio de direcciones físicas y, por tanto, las direcciones lógicas serán como máximo de 8 bits. Es decir, se destinarán un máximo de 3 bits para segmentos. Por tanto, los procesos podrán tener como máximo 8 segmentos.
- A partir de la tabla de segmentos deducimos el tamaño total del proceso (que es la suma de todos sus segmentos), es decir 48 bytes.
- La dirección lógica 57 la pasamos a binario (111001) y la dividimos en 3 y 5 bits:

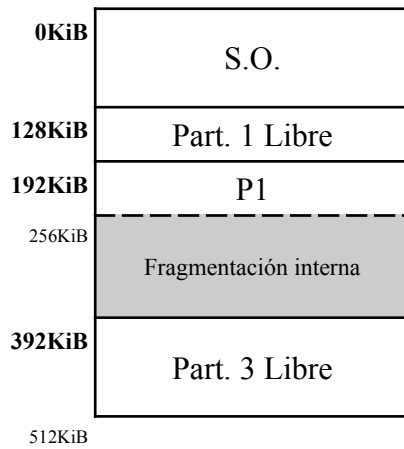
L:

001	11001
-----	-------

- La dirección lógica 57 (111001 en binario) se corresponde con el segmento 1 y desplazamiento 25 (11001) dentro del segmento.
- Comprobamos si la dirección lógica 57 es válida: ¿Tiene el proceso segmento 1? La respuesta es sí. ¿Es $25 < \text{tamaño segmento 1}$? La respuesta es sí. No sería necesario calcular la dirección física para responder a esta pregunta pero aun así se procede a calcularla.
- Vamos a la entrada 1 de la TS del proceso y obtenemos que el segmento 1 comienza en la dirección física 105 (en decimal).
- A la dirección física de comienzo del segmento 1 (105 en decimal) se suma el desplazamiento 25 por lo que la dirección física 130 se corresponde con la dirección lógica 57.

9. Se indica a continuación el estado de la MP en los instantes de tiempo donde se producen cambios (el tamaño de las particiones no está a escala):

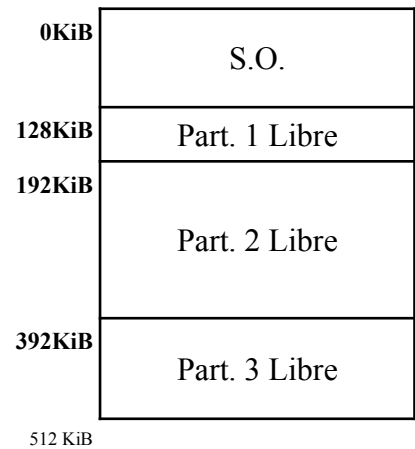
T=0 (Se carga P1)



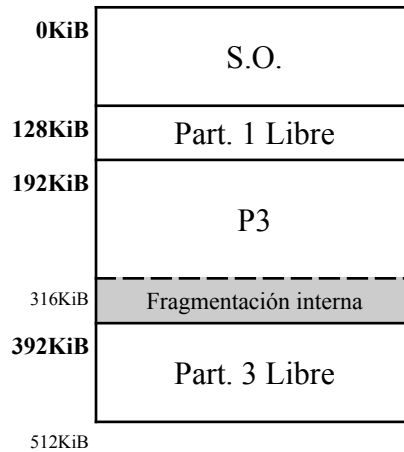
T=2 (Intento carga P2 y P3)

El proceso P2 se descarta porque no cabe en la memoria
El proceso P3 no se puede cargar, se intenta más tarde

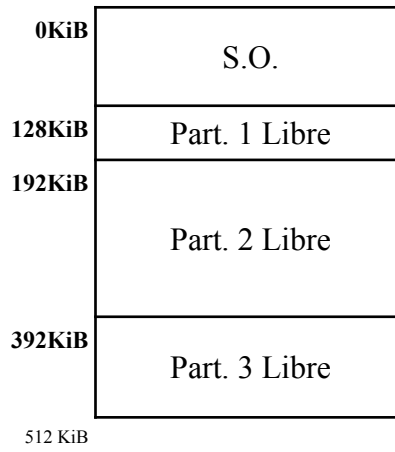
T=17 (Finaliza P1)



T=17 (Se carga P3)



T=32 (Finaliza P3)



La fragmentación interna máxima se da en el instante 0 y es de 136KiB

10. Pasos a seguir:

- Si hay 1MiB de MP entonces las direcciones físicas son de 20 bits.
- Si el tamaño del segmento es de 64 KiB entonces se reservan 6+10 bits para desplazamientos de segmentos (64 necesita 6 bits y 1KiB 10 bits). Como estamos en memoria real las direcciones lógicas son, a los sumo, de tantos bits como las direcciones físicas (20 bits). **Por tanto, las direcciones lógicas se dividen en 4 y 16.**

11. Pasos a seguir:

- Se calcula el estado de la MP después de cargar cada segmento del proceso, para así saber la tabla de segmentos del proceso, que sería el indicado a continuación:
 - A) El primer segmento (de 64KiB de tamaño) se carga en el hueco (760KiB, 190KiB). **Por tanto, el s0 del proceso comienza en la dirección física 760KiB y tiene tamaño 64KiB.** Se recalcula la lista de huecos que pasaría a ser [(38KiB, 128KiB) (824KiB, 126KiB)]
 - B) El segundo segmento (de 28KiB de tamaño) se carga en el hueco (38KiB, 128KiB). **Por tanto, el s1 del proceso comienza en la dirección física 38KiB y tiene tamaño 28KiB.** Se recalcula la lista de huecos que pasaría a ser [(824KiB, 126KiB) (66KiB, 100KiB)]
 - C) El tercer segmento (de 57KiB de tamaño) se carga en el hueco (824KiB, 126KiB). **Por tanto, el s2 del proceso comienza en la dirección física 824KiB y tiene tamaño 57KiB.**
- Pasamos la dirección lógica 131104 a binario (100000000000100000) y se ajusta a 20 bits (bits usados por la direcciones lógicas), lo que daría 00100000000000100000.
- Dividimos la dirección lógica en 4 y 16 bits:

L:

0010	0000 0000 0010 0000
------	---------------------

- La dirección lógica 131104 (en decimal) se corresponde con el segmento 0010 (2 en decimal) del proceso y desplazamiento 0000000000100000 (32 en decimal).
- Comprobamos que la dirección sea válida: ¿el proceso tiene segmento 2? La respuesta es sí. Además, comprobamos que el desplazamiento 32 sea menor que el tamaño del segmento. ¿Es $32 < 57\text{KiB}$? La respuesta es sí.
- Sabemos que el segmento 2 comienza en la dirección física 824KiB que es 843776 en decimal. Le sumamos el desplazamiento (32 en decimal) y **obtenemos la dirección física 843808** que es la que se corresponde con la dirección lógica 131104 del proceso.

12. Pasos a seguir:

- Si hay 2048 marcos de página entonces hay 11 bits reservados para ellos (en las direcciones físicas). **Por tanto, las direcciones físicas se dividen en 11 (m) y 9 (d')**
- Si el número máximo de segmentos es 8 entonces hay 3 bits reservados para segmentos (en las direcciones lógicas). Los 13 bits restantes son desplazamientos dentro del segmento (d) y por tanto para paginación. De esos 13 bits, 9 son para desplazamientos dentro de la página (d'). **Por tanto las direcciones lógicas se dividen en 3 (s), 4 (p) y 9 (d')**
- Pasamos la dirección lógica 24.631 a binario y lo dividimos en las tres secciones de bits calculadas en el paso anterior.

L:

011	0000	0 0011 0111
-----	------	-------------

- Ahora sabemos que la dirección lógica se corresponde con el segmento 3 (tres primeros bits por la izquierda pasados a decimal), desplazamiento 55 dentro del segmento (resto de bits a decimal).
- Primero de todo comprobamos la protección: ¿el proceso tiene segmento 3? Sí. ¿Es el desplazamiento del segmento menor que su tamaño?: $55 < 8900$? Sí, y por tanto no hay excepción (la dirección lógica es válida).
- Obtenemos ahora que se está accediendo a la página 0 (segundo grupo de bits por la izquierda pasados a decimal) de dicho segmento.
- Nos vamos a la tabla de páginas del segmento 3, entrada 0 y obtenemos el marco de página donde está cargada, que es el 1001.
- Pasamos 1001 a binario y lo introducimos en el grupo de bits de la izquierda (m), a la derecha ponemos los mismos bits que los que hay en el tercer grupo de bits calculados en el primer paso (d') y esto nos dará la dirección física en binario.

R:

011 1110 1001	0 0011 0111
---------------	-------------

- Pasamos a decimal la dirección física y **el resultado final es 512.567**