

PRÁCTICA DE CONTROL

FCFS/SJF-Pag-Segunda Oportunidad-C-R

*Ingeniería Informática
Sistemas Operativos*

Alumno: Rocío Esteban Valverde
rev1002@alu.ubu.es

Profesor: José Manuel Saiz Díez

8/1/2023



UNIVERSIDAD DE BURGOS

ÍNDICE

Introducción.....	1
Algoritmos.....	1
Ejercicio resuelto a mano	3
Ejercicio 1	3
FCFS	3
SJF	13
Ejercicio resuelto con el Script	22
Muestra de la ejecución del script.....	22
Ejercicio 1	26
FCFS	26
SJF	35
Modificaciones y mejoras realizadas.....	45
Mejoras por realizar:	46
Estructura del Script	47
Conclusiones	48
Bibliografía.....	48

Introducción

Práctica de control que consiste en la programación de un script en bash sobre la resolución de algoritmos de gestión de procesos y memoria con el fin de conocer y estudiar los contenidos teóricos y prácticos de la asignatura de sistemas operativos.

En este informe se pretende explicar todo lo relacionado a la realización de esta práctica. Este documento forma parte del entregable que incluye un script con el sistema de directorios y archivos necesarios para su ejecución, un video donde se muestra el funcionamiento del script y este informe.

La práctica realizada es FCFS/SJF-Pag-Segunda Oportunidad-C-R.

Algoritmos

A continuación, se explican los algoritmos y conceptos que definen la práctica.

Los algoritmos de planificación de procesos se utilizan para determinar el orden en que deben ejecutarse los procesos en un sistema operativo.

Un algoritmo es apropiativo cuando puede quitar un proceso que ya está en la CPU, para cambiarlo por otro y por tanto no apropiativo es que cuando entra un proceso a la CPU no sale hasta que haya terminado.

FCFS (del inglés First-Come, First-Served) o también llamado **FIFO** (del inglés First In, First Out) consiste en que el primer proceso que llega es el primero en ser enviado a la CPU y, hasta que no termine con este, no entra el siguiente (salvo los procesos de gestión del propio sistema). Por tanto, el proceso no es apropiativo. Es un método de planificación sencillo y justo, pero puede llevar a tiempos de espera muy largos para algunos procesos si hay procesos más largos que bloquean el acceso a la CPU.

SJF (del inglés Shortest Job First) prioriza los procesos según su tiempo de ejecución estimado. seleccionando siempre el proceso más corto. Es no apropiativo. Es un método de planificación eficiente ya que permite atender a los procesos más cortos de manera rápida, pero requiere conocer de antemano el tiempo de ejecución de cada proceso y puede ser injusto para los procesos más largos.

La partición por PAGINACIÓN es una técnica de memoria virtual por la cual el espacio se divide en secciones físicas de igual tamaño, denominadas marcos de página. Cada proceso está dividido también en pequeños trozos de tamaño fijo e igual a los de memoria, conocidos como páginas. El SO mantiene una tabla de páginas para cada proceso. Cada entrada contiene el número del marco de memoria que alberga la página correspondiente, incluyendo también una lista de marcos libres de la memoria. Las páginas se meten y sacan de la memoria según son necesitadas lo cual permite ejecutar procesos mucho mayores que el tamaño de la memoria principal.

Los algoritmos de reemplazo de páginas sirven para gestionar qué páginas de la memoria virtual deben ser cargadas en la memoria física y cuáles deben ser reemplazadas por otras cuando se necesita espacio en la memoria física.

SEGUNDA OPORTUNIDAD es un algoritmo de reemplazo de páginas. Es una variante del algoritmo de reemplazo de páginas FIFO (primero en entrar, primero en salir), con la diferencia de que le da a cada página una segunda oportunidad antes de ser reemplazada.

Esto se consigue marcando las páginas con un bit de referencia que indica el uso que han tenido. El funcionamiento es el siguiente:

Cuando una página debe de ser sacada para la entrada de una nueva lo primero que se hace es comprobar su bit de referencia:

- Si el bit está a 1: Se cambia a 0 y se coloca al final de la cola dándole una segunda oportunidad.
- Si el bit está a 0: La página se saca de la memoria y se introduce la nueva.

Cada vez que se referencia a una página que ya se encuentre en memoria se actualiza el bit de referencia de esa página a 1.

Esta implementación es realmente ineficiente ya que tiene que estar desplazando constantemente las páginas en una lista.

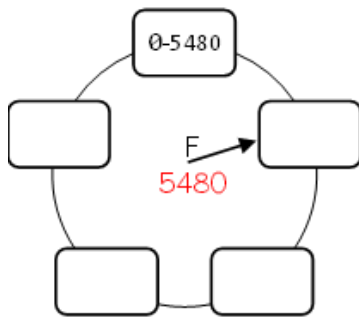
Para mejorar la eficiencia se utiliza una variación en la implementación del algoritmo con la que se obtiene el mismo resultado. Se trabaja con una cola circular llamada 'reloj' y un índice que indica la página siguiente a reemplazar.

Esta variación es la forma en la que veremos que trabaja nuestro script.

MEMORIA CONTINUA Y REUBICABLE: Los procesos deben meterse en un único hueco en memoria, es decir, que no pueden dividirse (Continuo). Cuando hay huecos de memoria separados entre sí que sumados generan suficiente espacio para meter otro proceso, se modifica la posición de los procesos dentro de memoria para que pueda entrar el siguiente (Reubicable).

Ejercicio resuelto a mano

Para la representación del reloj en los ejemplos se utilizará un diagrama como el siguiente.



El reloj tendrá tantas “horas” como marcos ocupe el proceso en ejecución.

Si un marco tiene una página pondrá el número de esta dentro, si está vacío, el cuadrado estará vacío. Antes del número de página se indica el bit se parando los datos con un guion.

En el centro del reloj está la página que se ha ejecutado en ese momento. Si encima de este número hay una F significa que ha habido un fallo de página.

El reloj tiene una aguja que representa el índice que indica que página será sustituida la próxima vez que ocurra un fallo de página. La aguja avanza en el sentido de las agujas del reloj, de marco en marco. Siguiendo el algoritmo de segunda oportunidad si encuentra un marco con bit 0 apuntará a este y si encuentra un bit 1 lo cambia a 0 y avanza a al marco siguiente, para que se vea claro se resaltará el bit que se ha cambiado de esta forma ya que en los diagramas se verá directamente el marco al que se apunta al final.

Ejercicio 1

Datos:

Datos	
Memoria del Sistema:	1200
Tamaño de Página:	100
Marcos totales de la memoria:	12

Ref	TII	nMar	Dirección-Página
P01	2	5	548057-5480 548057-5480 548057-5480 548057-5480 548057-5480 548057-5480 548057-5480
P02	3	5	424666-4246 591789-5917 475005-4750 703990-7039 424666-4246 475005-4750 324430-3244 19268-192 69279-692 637416-6374
P03	4	3	6869-68 441318-4413 258922-2589 6869-68 441318-4413
P04	10	3	111-1 222-2 333-3 222-2 111-1 555-5 666-6 777-7 888-8 999-9 888-8 777-7 444-4 555-5 66-0
P05	42	6	2021-20 200-2 100-1 2022-20 2222-22 200-2 0-0 2323-23

Los marcos en los que se divide la memoria dependen del cociente entre el tamaño de la memoria principal y el tamaño de las páginas.

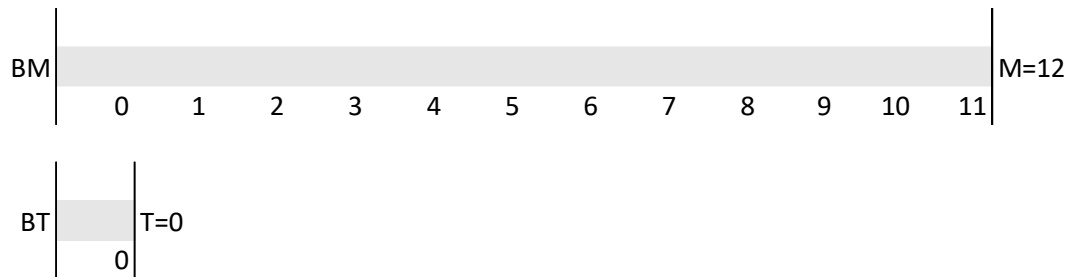
La página en la que se encuentra cada dirección se calcula dividiendo el numero el numero de la dirección entre el tamaño de la página.

El tiempo de ejecución en nuestro caso coincidirá con el número de direcciones que tenga el proceso suponiendo que se ejecuta una dirección por unidad de tiempo.

FCFS

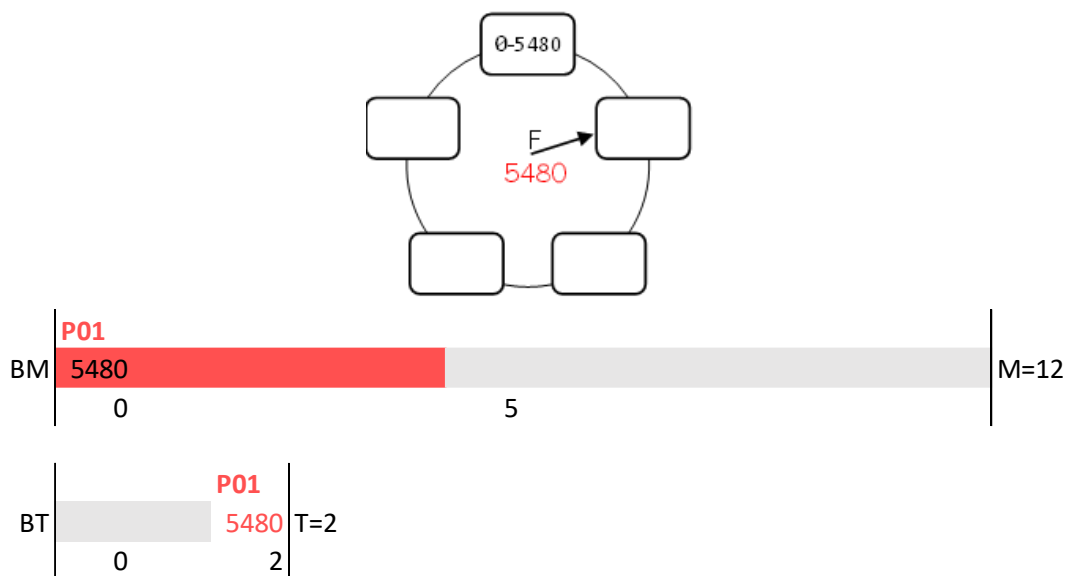
T=0

No ha llegado ningún proceso, por lo que la memoria está vacía.



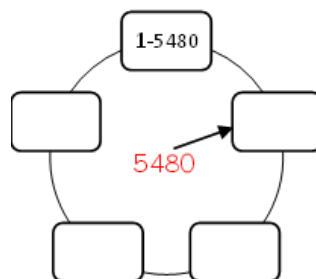
$T=2$

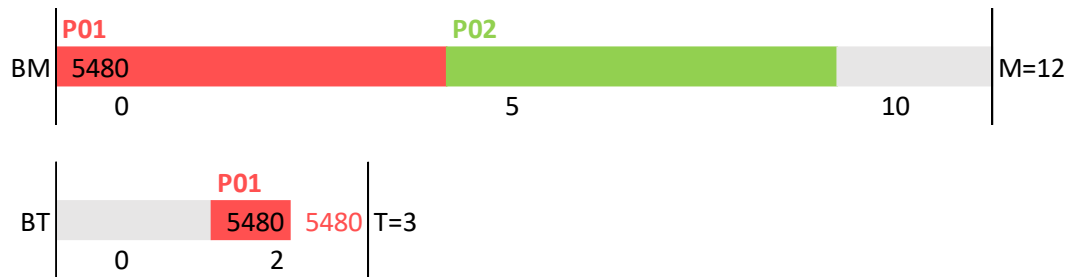
Llega **P01**. Entra en memoria ocupando 5 marcos y como la CPU está libre comienza a ejecutarse. El proceso introduce su primera página, produciéndose un fallo de página. El bit del marco 0 se pone a 0. El reloj apunta al siguiente marco.



$T=3$

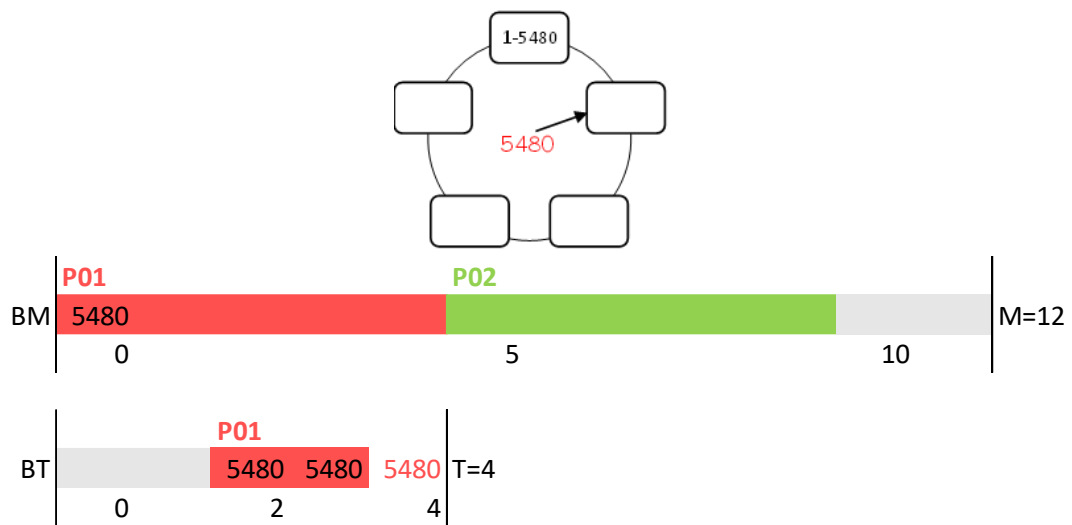
Llega **P02** y entra en memoria ocupando 5 marcos, quedan 2 marcos libres. No se ejecuta porque la CPU está ocupada. Se ejecuta otra página de **P01**, no se produce fallo de página porque ya está en memoria. El bit del marco 0 se pone a 1.





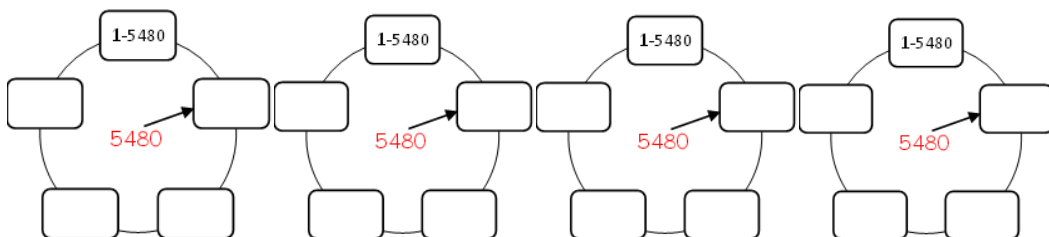
T=4

Llega **P03**, pero como no cabe en memoria se queda en espera. Se ejecuta otra página de **P01**, no se produce fallo de página porque ya está en memoria. El bit se mantiene igual y el puntero también.



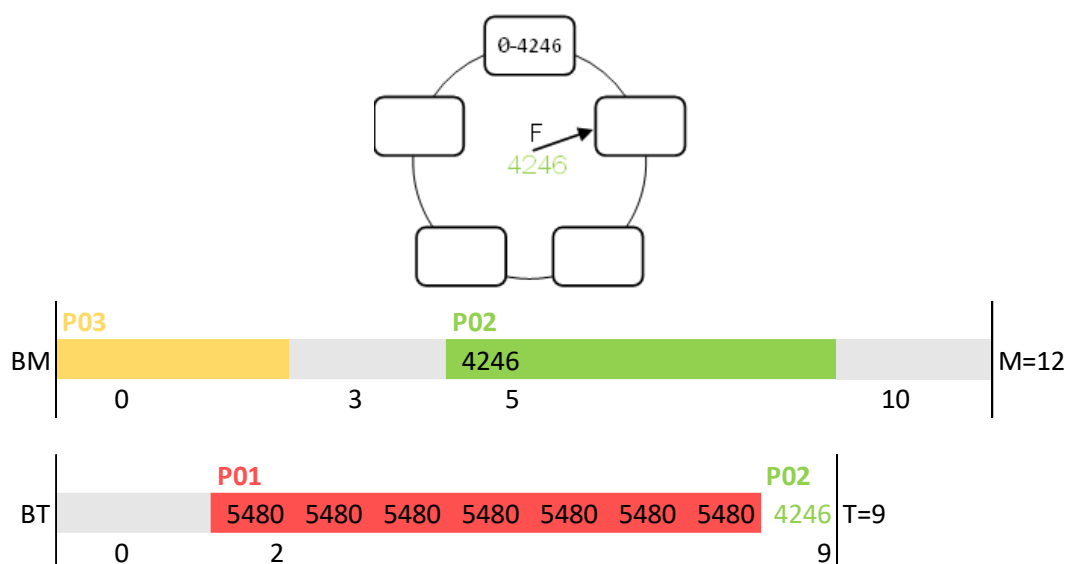
T=9

P01 ha ejecutado 4 páginas más, no se han producido fallos de página ya que todas eran la misma:



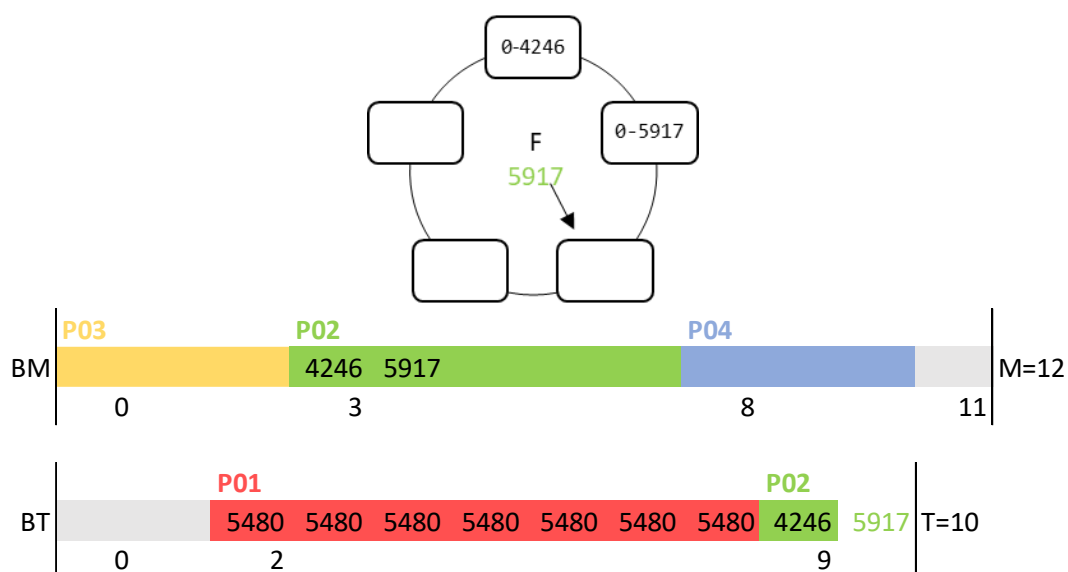
Ha finalizado su ejecución y se liberan los marcos de la memoria que estaba ocupando.

Entra en memoria el proceso **P03** que ocupa 3 marcos de página. La CPU ha quedado libre así que entra un nuevo proceso. Como es FCFS, entra en ejecución **P02**, que es el que ha llegado antes entre los procesos que hay en memoria. Se carga en memoria su primera página, produciéndose un fallo de página. El bit del marco 5 se pone a 0. El reloj apunta al marco siguiente.



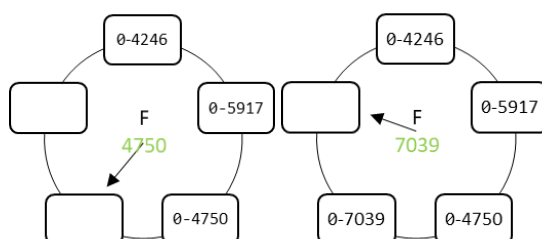
T=10

Llega el proceso **P04** que necesita 3 marcos de página, hay 4 marcos libres, pero no de forma continua por lo tanto se hace REUBICACIÓN, moviendo el proceso **P02**. Tras la reubicación entra **P04** en memoria. El proceso **P02** sigue ejecutándose y carga su siguiente página, produciéndose un fallo de página. El bit del marco 4 se pone a 0. El reloj apunta al marco siguiente.

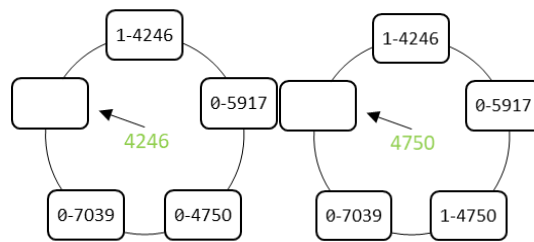


T=19

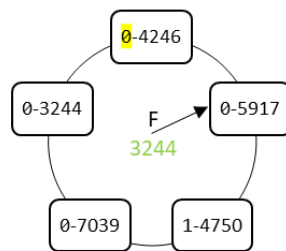
P02 ha ejecutado 8 páginas más, se han producido 6 fallos de página.



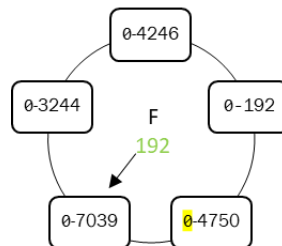
Van entrando páginas ,produciéndose fallos de página, y avanza el índice del reloj.



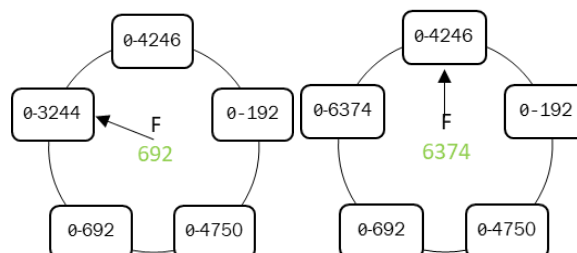
Estas páginas ya estaban en memoria por lo que no se ha producido fallo de página y el bit de los marcos donde están se cambia a 1, el puntero se ha mantenido en su posición.



Ha entrado la nueva página en el marco donde apuntaba la flecha antes, fallo de página. La flecha avanza una posición, pero ese marco tenía el bit a 1 así que lo cambia a 0 y avanza al siguiente, como el bit está a 0 esa página será la siguiente en ser reemplazada.

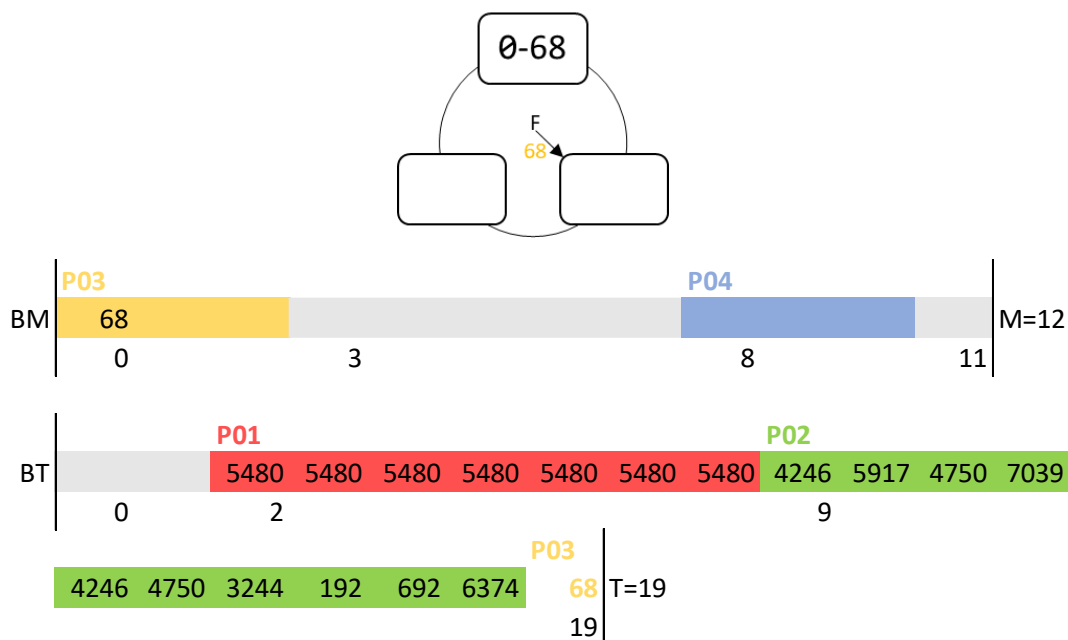


Ocurre lo mismo que en el instante anterior. Ha entrado una página y la flecha avanza dos marcos.



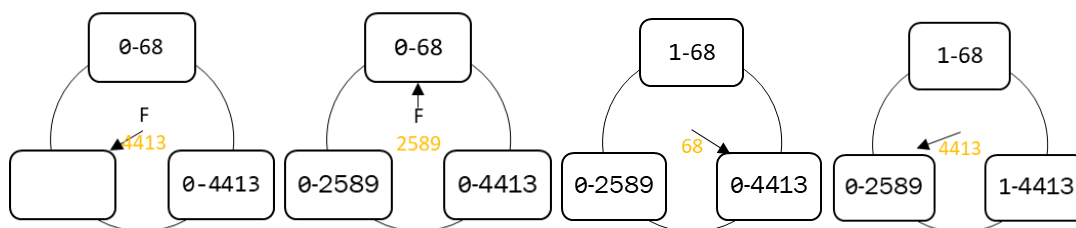
Entran dos páginas más y finaliza la ejecución de **P02**. Se liberan los marcos que ocupaba.

La CPU ha quedado libre así que entra un nuevo proceso. Como es FCFS, entra en ejecución **P03**, que es el que ha llegado antes entre los procesos que hay en memoria. Se carga en memoria su primera página, produciéndose un fallo de página. El bit del marco 0 se pone a 0. El reloj apunta al marco siguiente.



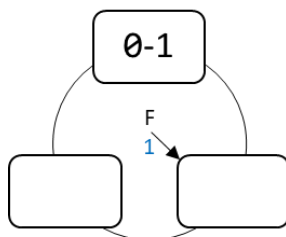
$T=24$

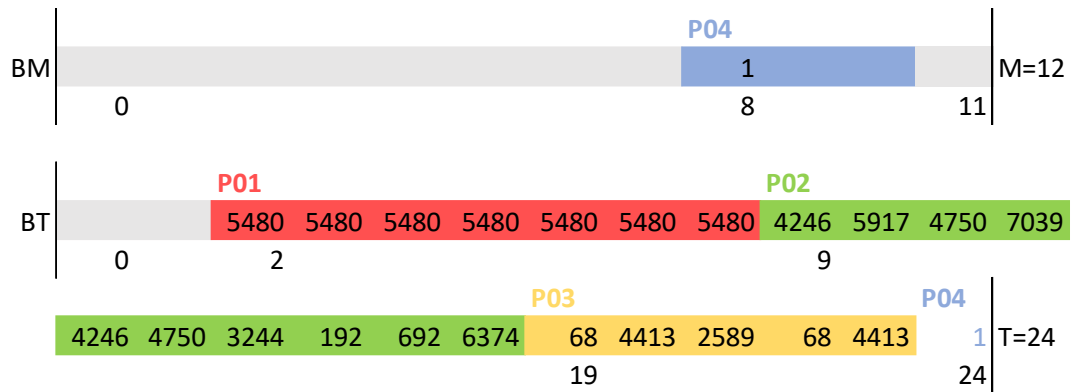
P03 ha ejecutado 4 páginas más, se han producido 2 fallos de página.



Finaliza su ejecución y libera los marcos que ocupaba.

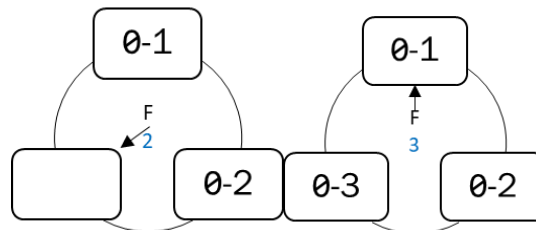
La CPU ha quedado libre así que entra **P04**, que es el único que queda en memoria. Se carga en memoria su primera página, produciéndose un fallo de página. El bit del marco 8 se pone a 0. El reloj apunta al marco siguiente.



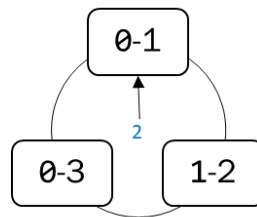


T=39

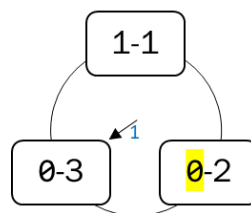
P04 ha ejecutado 14 páginas más, se han producido 10 fallos de página.



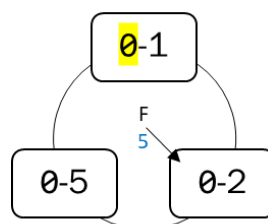
Van entrando páginas, se producen fallos de página.



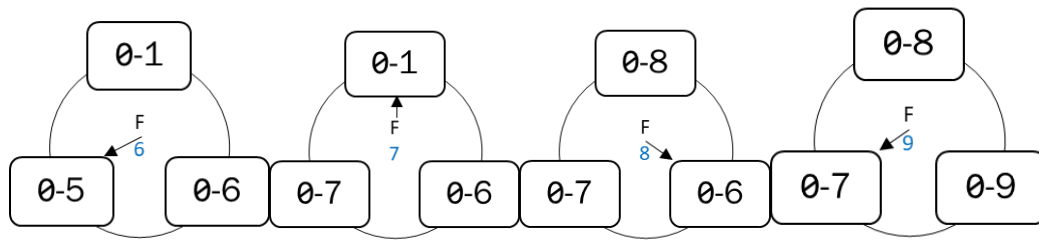
La página 2 ya estaba en un marco, no se produce fallo de página, se cambia el bit a 1.



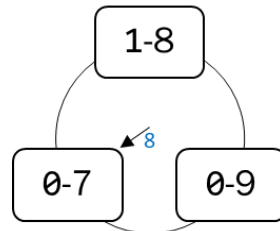
Se ha ejecutado la página 1, como ya estaba en memoria se pone el bit del marco a 1. Avanza la flecha, el bit del marco siguiente estaba a 1 por lo que el bit se ha cambiado a 0 y el reloj apunta al marco siguiente, este sí que está a cero por lo tanto cuando ocurra un fallo de página se reemplazará la página de ese marco.



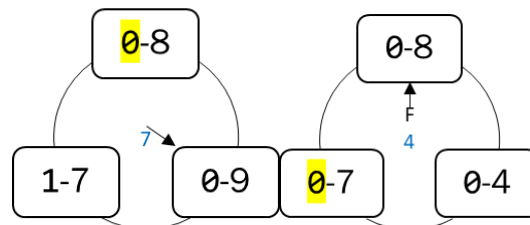
De nuevo, lo mismo que en el anterior.



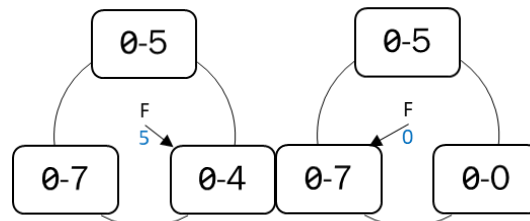
Continúan entrando y sustituyéndose páginas.



La página 8 ya está en memoria, no se produce fallo de página, bit a 1.

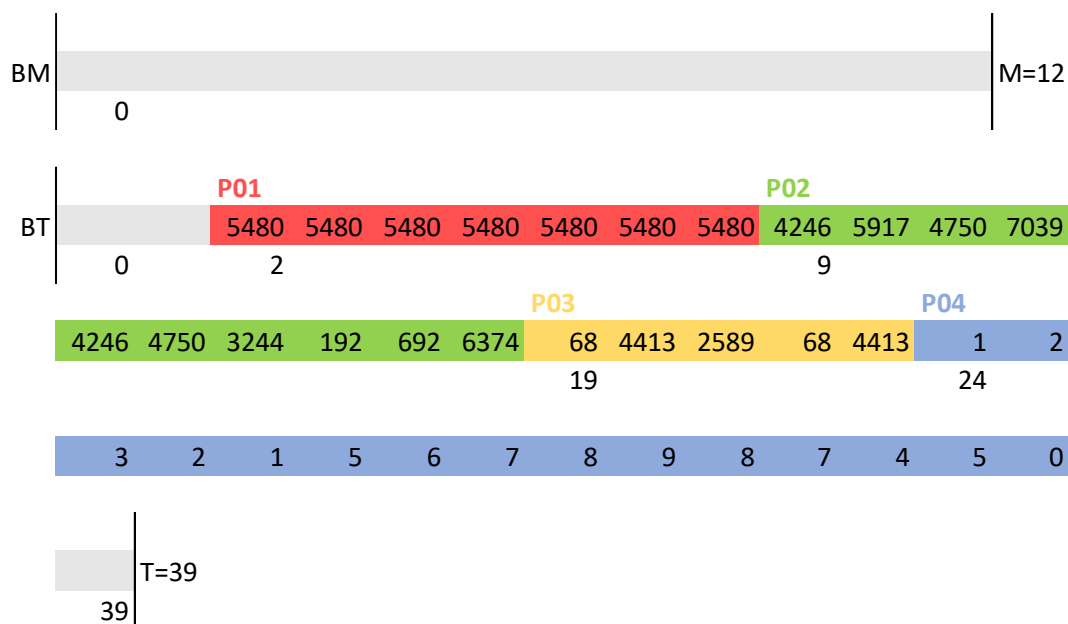


Entran más páginas y la flecha al avanzar se encuentra con marcos con bit a 1 que han sido cambiados a 0.



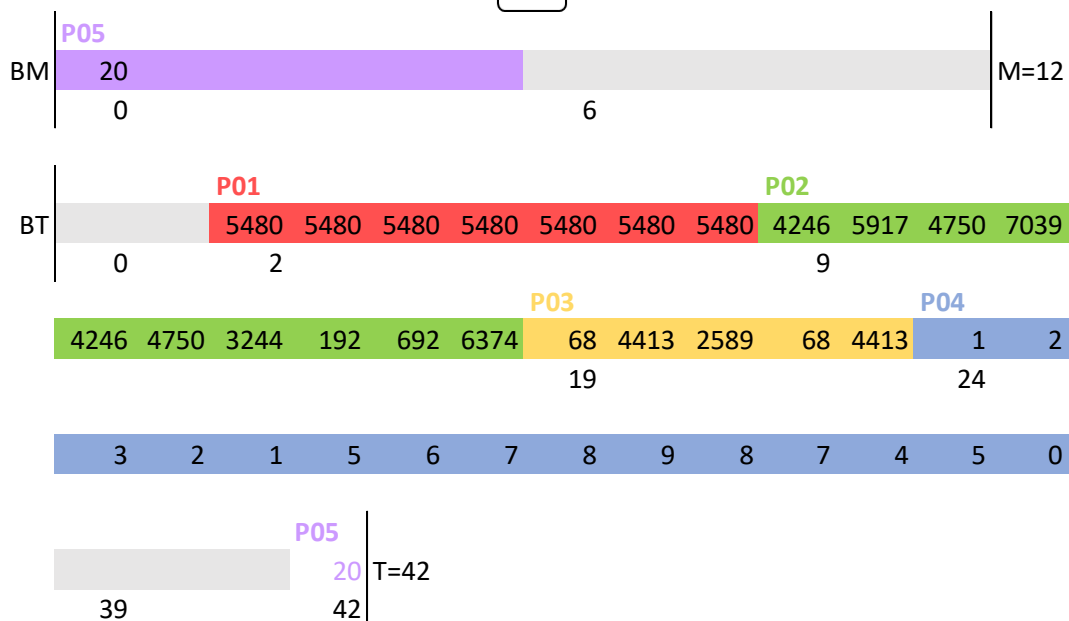
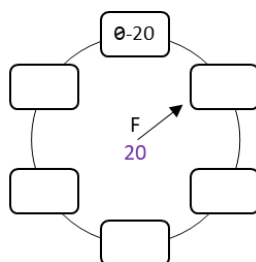
Últimas páginas. Finaliza la ejecución de P04 y se liberan los marcos que ocupaba.

La memoria queda vacía ya que no hay procesos en espera. La CPU también está vacía.



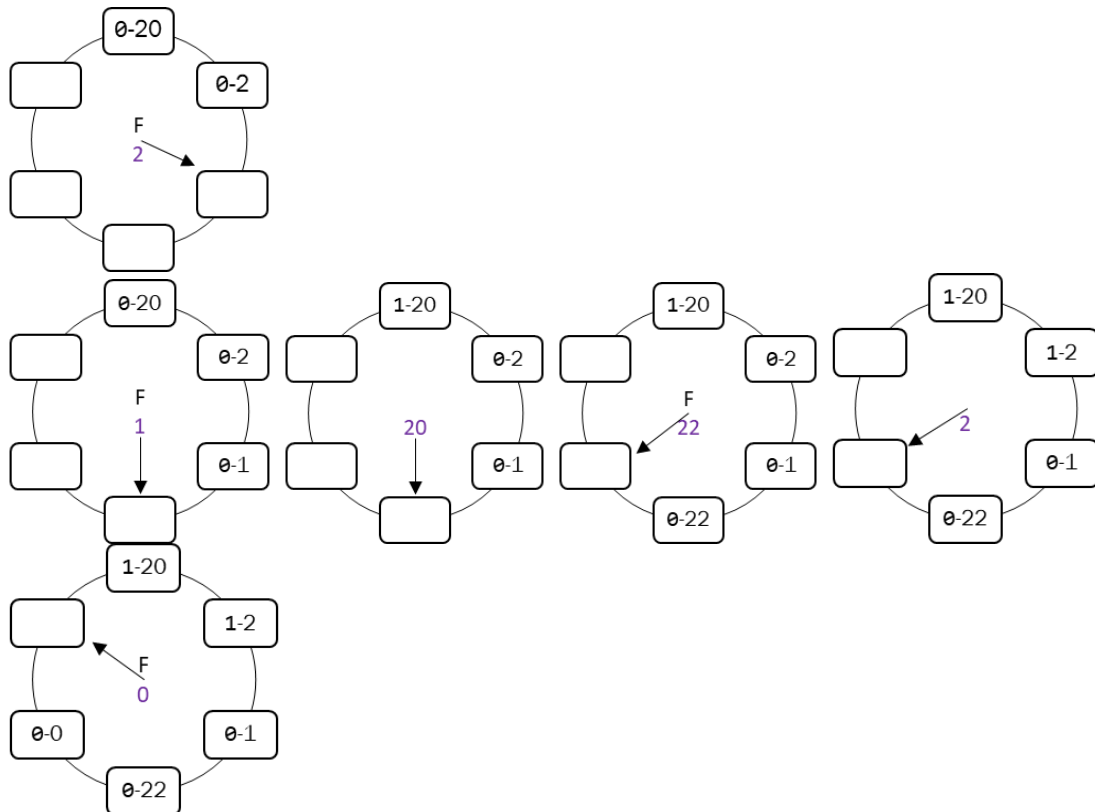
T= 42

Llega **P05**, entra en memoria, ocupando 6 marcos, entra en la CPU. Se carga en memoria su primera página, produciéndose un fallo de página. El bit del marco 0 se pone a 0. El reloj apunta al marco siguiente.

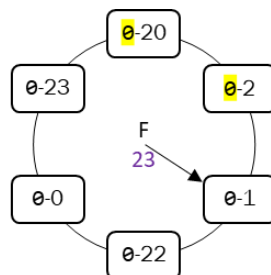


T=50

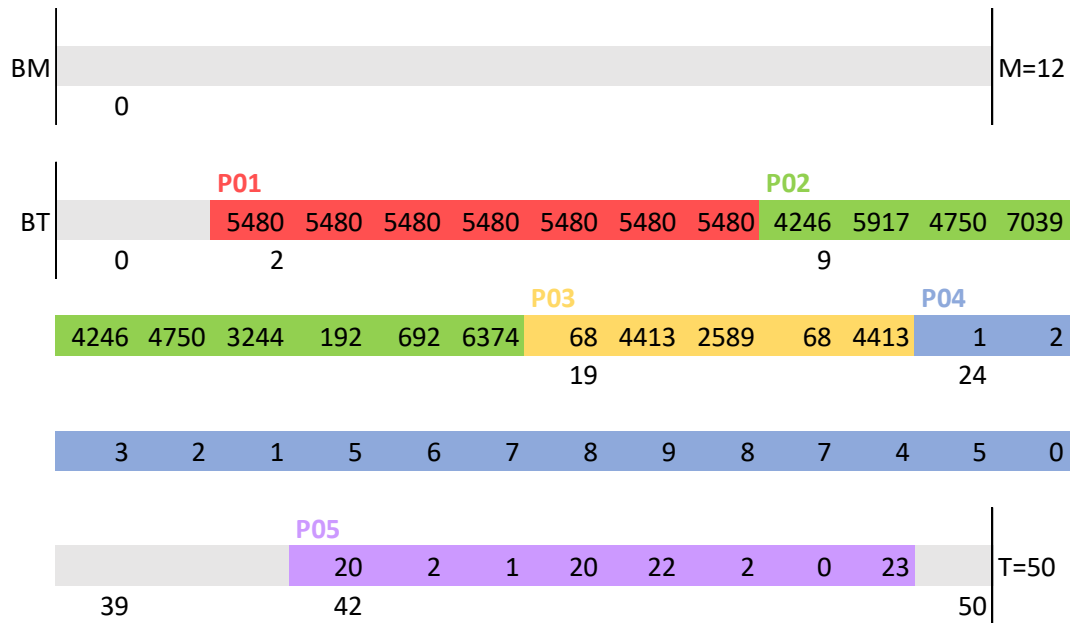
P05 ha ejecutado 7 páginas más, se han producido 5 fallos de página.

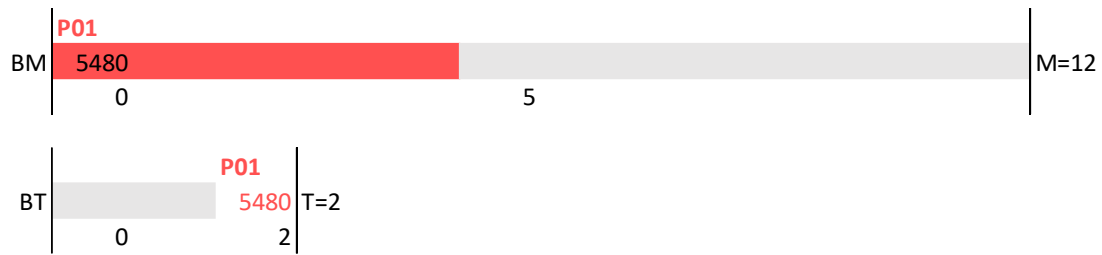


Van entrando páginas, produciendo fallos de página. Hasta este momento hay dos páginas que se han repetido, no han producido fallo de página y los bits de sus marcos están a 1.



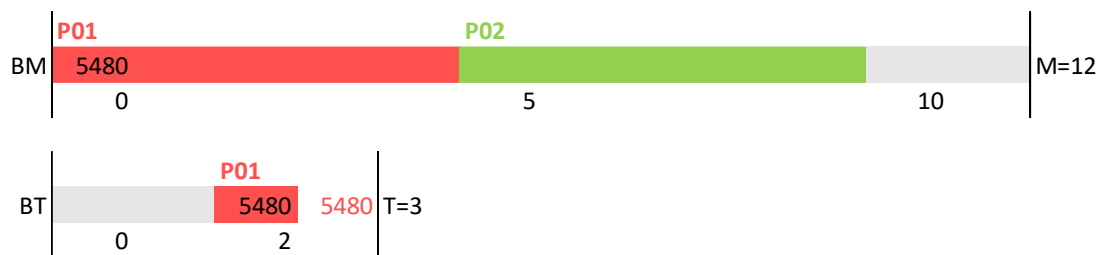
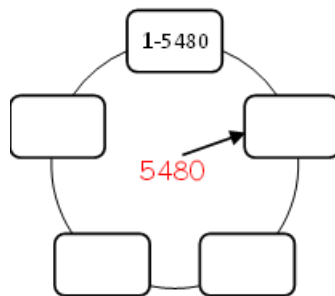
Entra una nueva página donde apuntaba la flecha antes. Se produce fallo de página, la flecha avanza una posición, pero el bit estaba a 1, lo cambia por 0 y lo mismo con el siguiente marco. Al final encuentra un marco con bit 0 y queda apuntando ahí. Si hubiera más páginas y se produjese un fallo de página, se sustituiría la página que hay en ese marco, pero esta era la última página y por tanto finaliza la ejecución de **P05**, liberándose la CPU y los marcos que ocupaba en memoria.





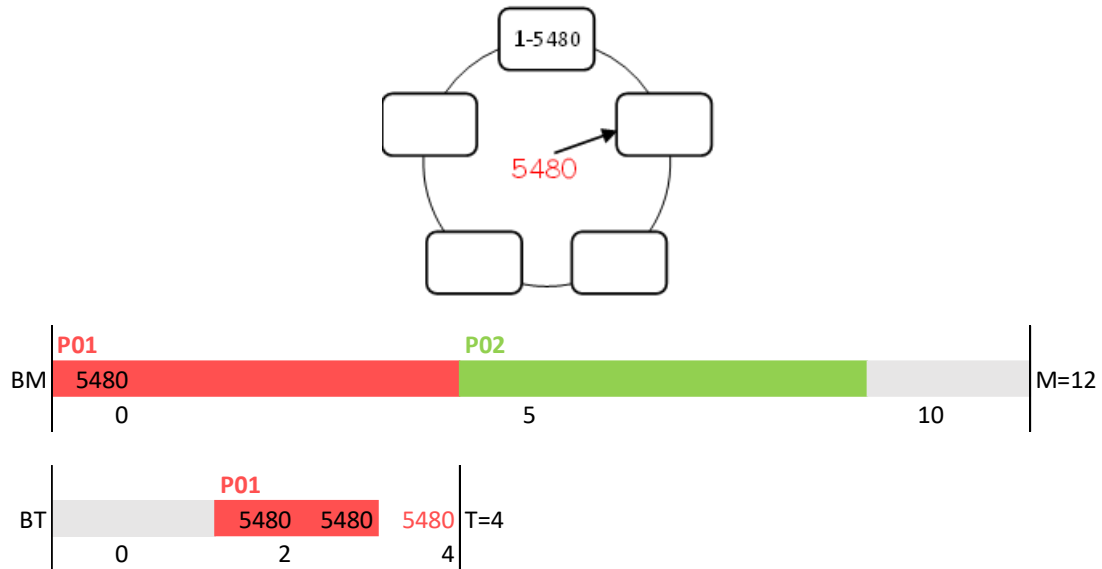
T=3

Llega **P02** y entra en memoria ocupando 5 marcos, quedan 2 marcos libres. No se ejecuta porque la CPU está ocupada. Se ejecuta otra página de **P01**, no se produce fallo de página porque ya está en memoria. El bit del marco 0 se pone a 1.



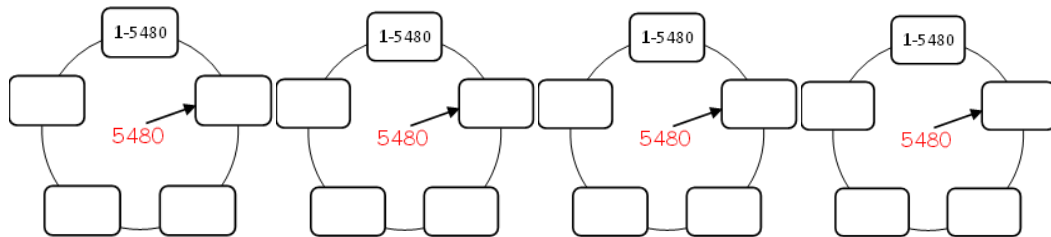
T=4

Llega **P03**, pero como no cabe en memoria se queda en espera. Se ejecuta otra página de **P01**, no se produce fallo de página porque ya está en memoria. El bit se mantiene igual y el puntero también.



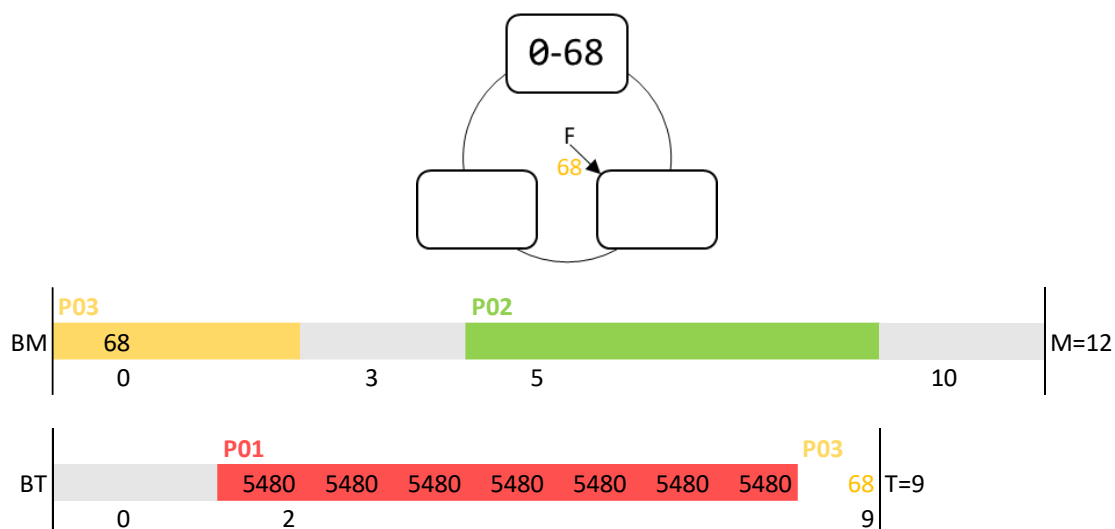
T=9

P01 ha ejecutado 4 páginas más, no se han producido fallos de página ya que todas eran la misma:



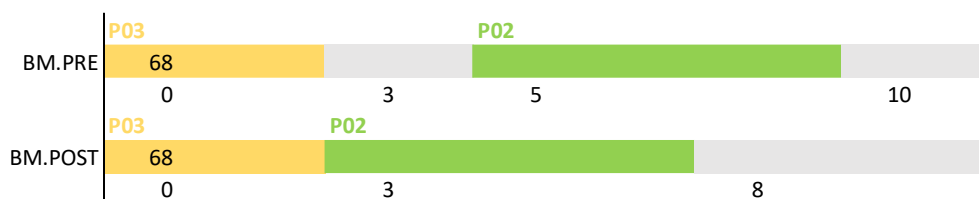
Ha finalizado su ejecución y se liberan los marcos de la memoria que estaba ocupando.

Entra en memoria el proceso **P03** que ocupa 3 marcos de página. La CPU ha quedado libre así que entra un nuevo proceso. En este caso, como el algoritmo es SJF entra en ejecución **P03**, que es el que tiene el menor tiempo de ejecución entre los procesos que hay en memoria. Se carga en memoria su primera página, produciéndose un fallo de página. El bit del marco 0 se pone a 0. El reloj apunta al marco siguiente.

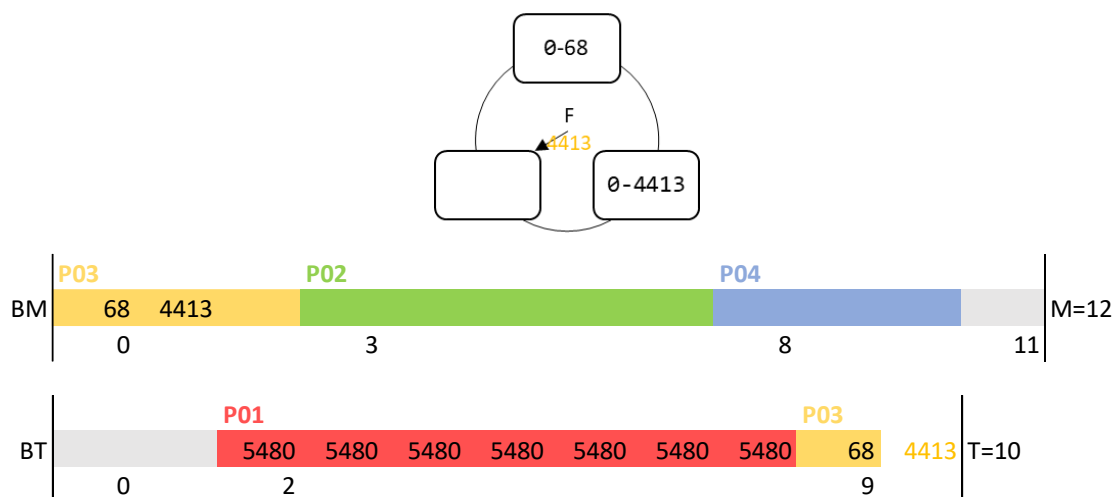


T=10

Llega el proceso **P04** que necesita 3 marcos de página, hay 4 marcos libres, pero no de forma continua por lo tanto se hace REUBICACIÓN, moviendo el proceso **P02**.

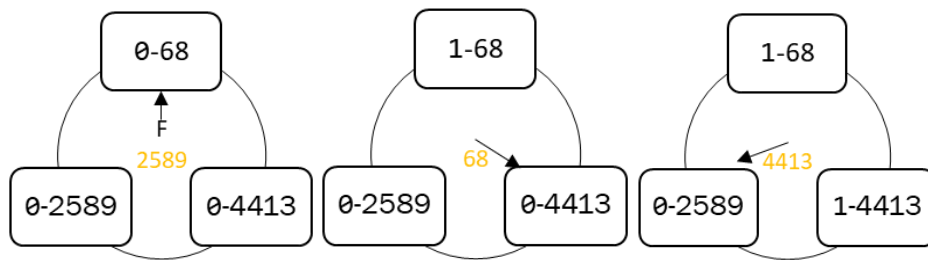


Tras la reubicación entra **P04** en memoria. El proceso **P03** sigue ejecutándose y carga su siguiente página, produciéndose un fallo de página. El bit del marco 1 se pone a 0. El reloj apunta al marco siguiente.



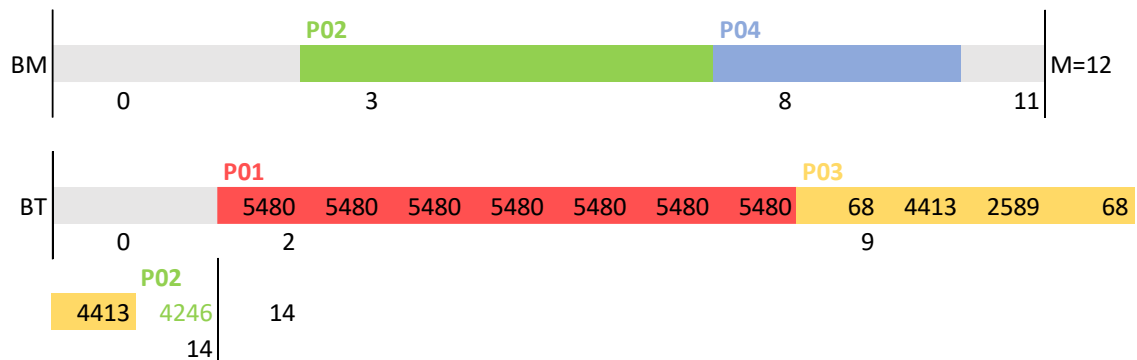
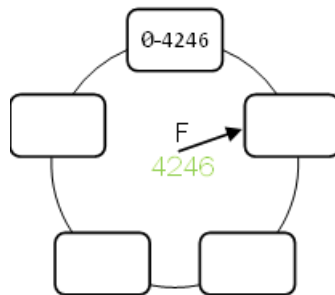
T=14

P03 ha ejecutado 3 páginas más, se ha producido 1 fallo de página.



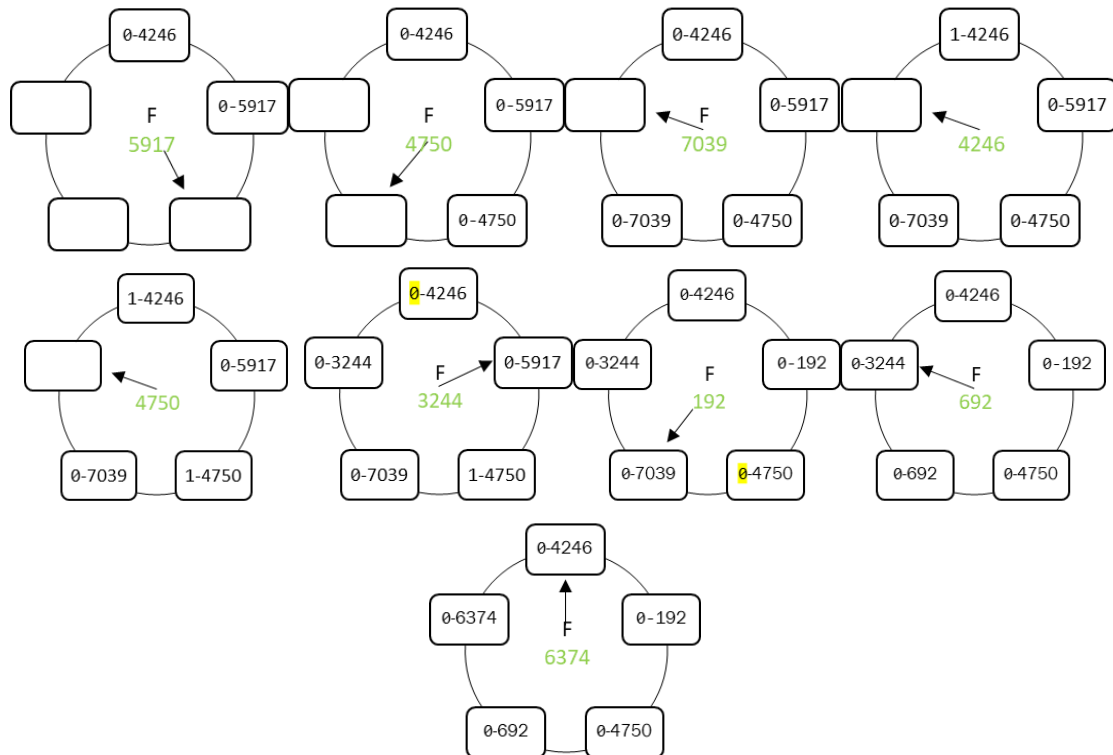
Finaliza su ejecución y libera los marcos que ocupaba y la CPU.

La CPU ha quedado libre así que entra un nuevo proceso. Como es SJF, entra en ejecución **P02**, que tiene el menor tiempo de ejecución de los procesos que hay en memoria, $10 < 15$. Se carga en memoria su primera página, produciéndose un fallo de página. El bit del marco 3 se pone a 0. El reloj apunta al marco siguiente.



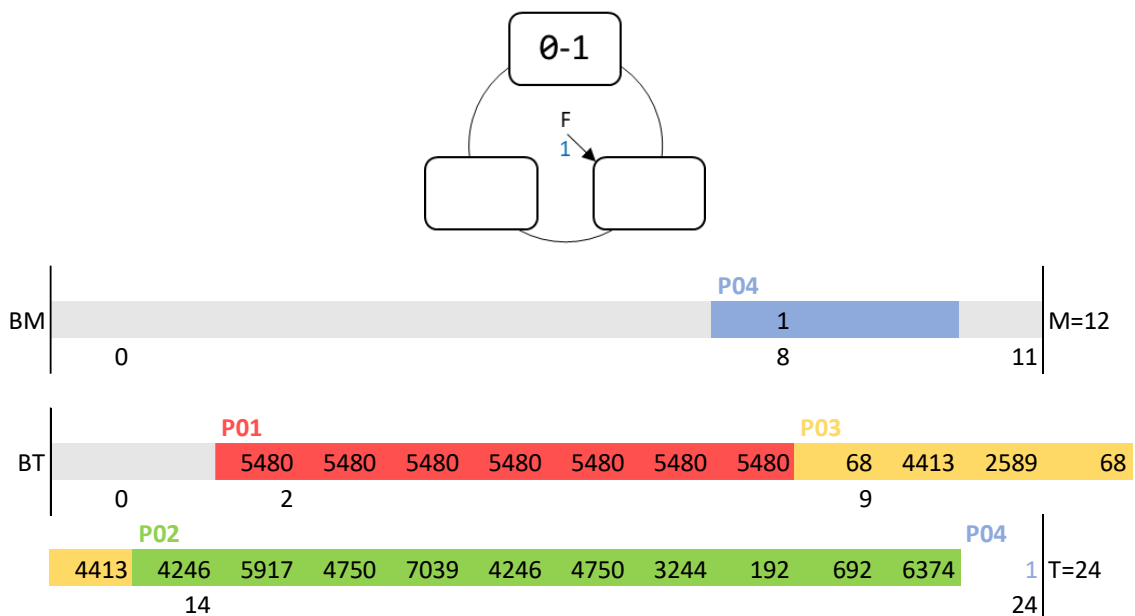
T=24

P02 ha ejecutado 9 páginas más, se han producido 6 fallos de página. La ejecución es la misma que con el algoritmo anterior.



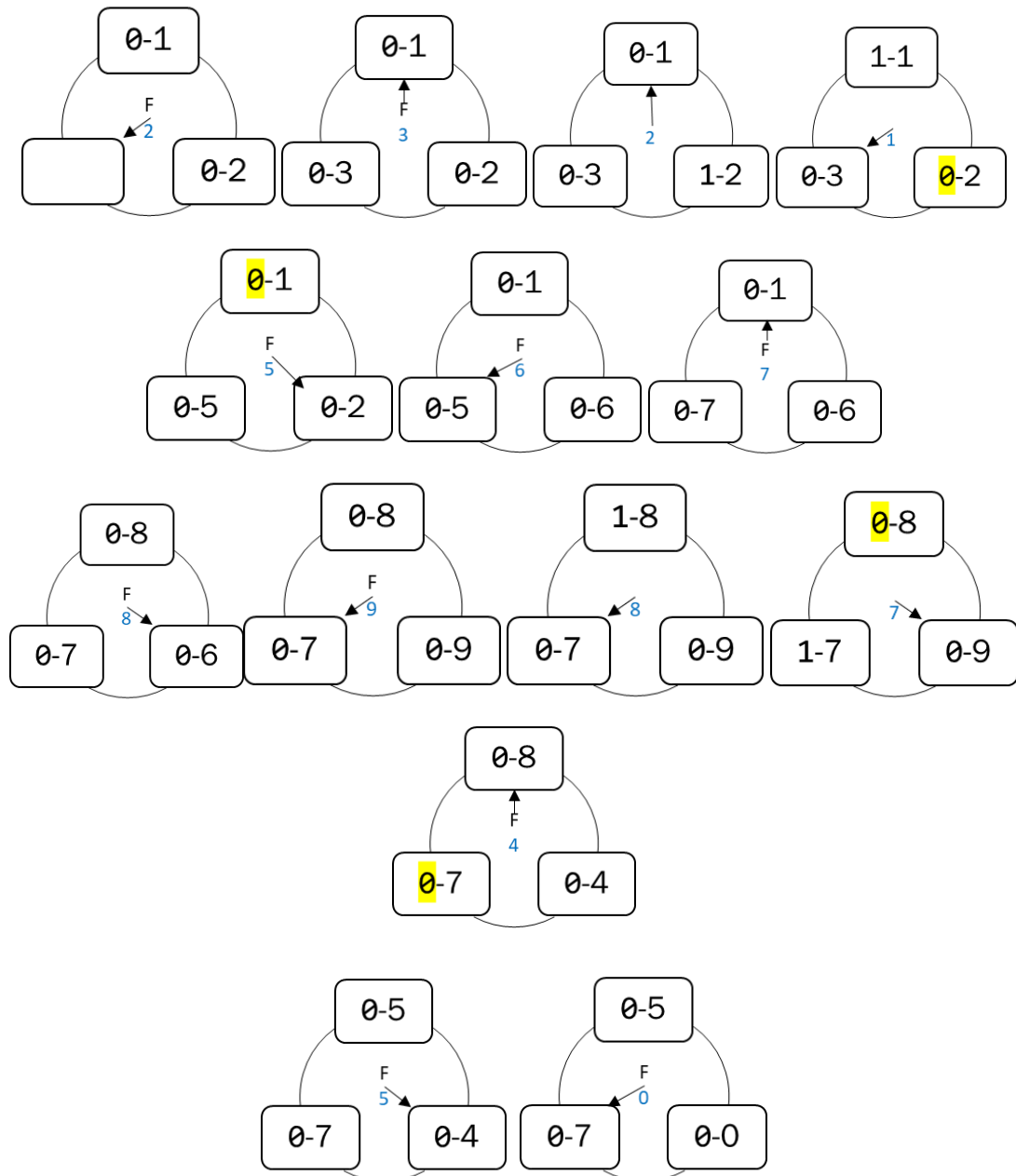
Finaliza la ejecución de **P02**. Se liberan los marcos que ocupaba y la CPU.

La CPU ha quedado libre así que entra **P04**, que es el único que queda en memoria. Se carga en memoria su primera página, produciéndose un fallo de página. El bit del marco 8 se pone a 0. El reloj apunta al marco siguiente.



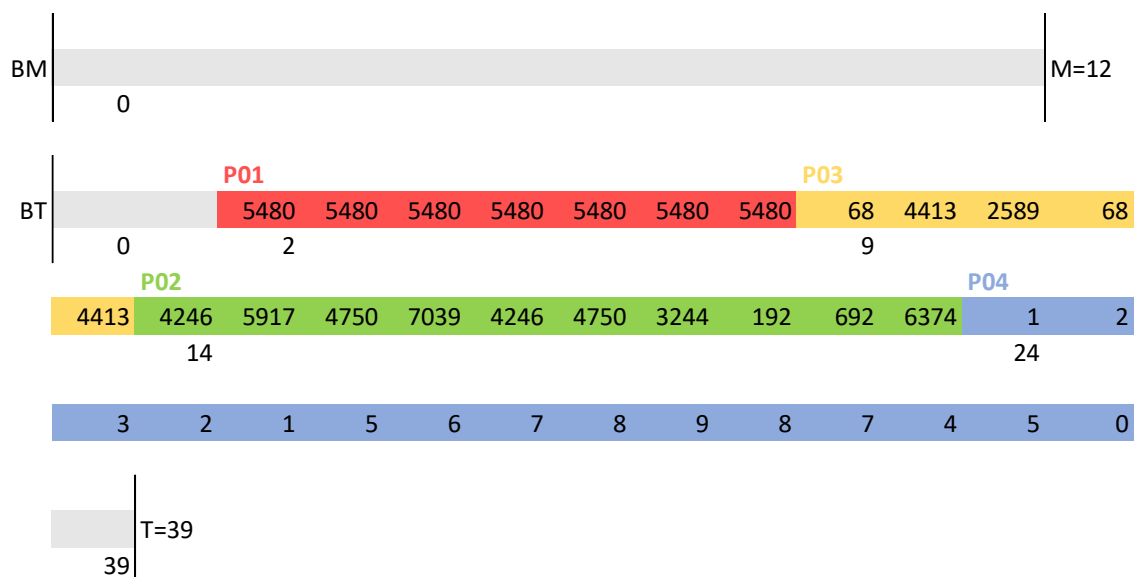
T=39

P04 ha ejecutado 14 páginas más, se han producido 10 fallos de página. La ejecución es la misma que con el algoritmo anterior.



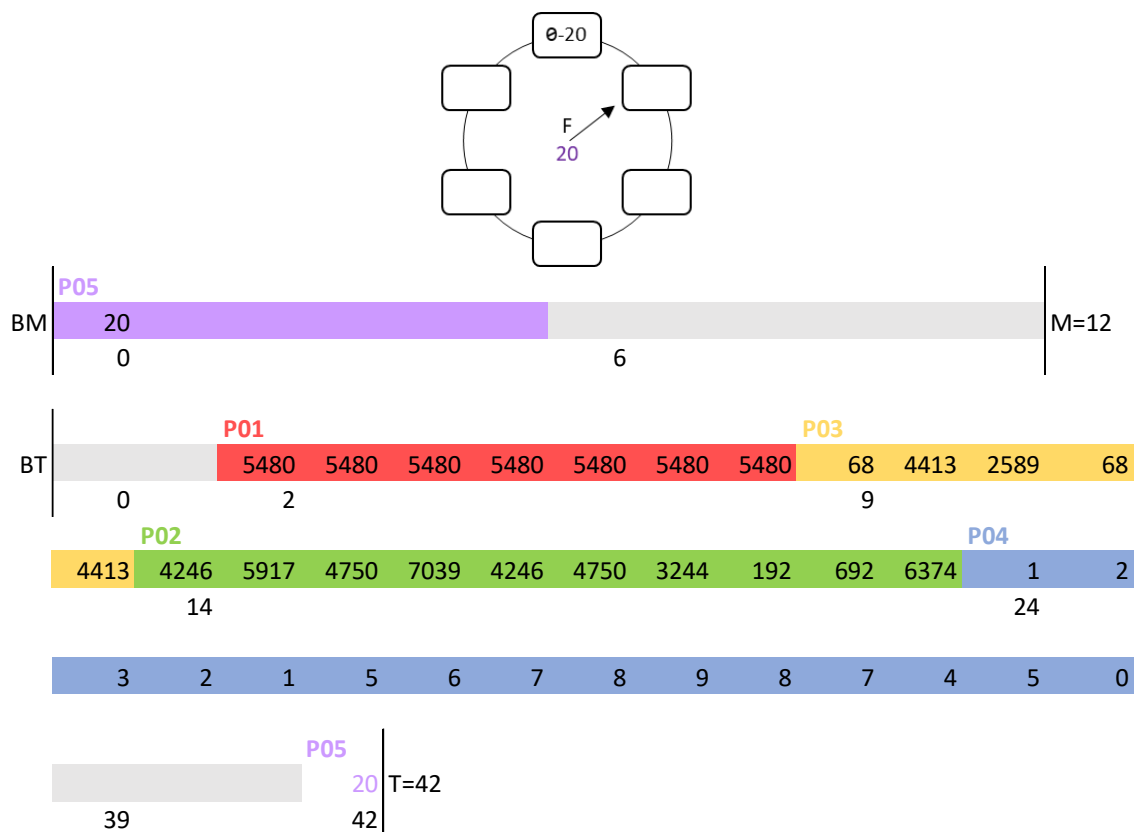
Finaliza la ejecución de P04 y se liberan los marcos que ocupaba.

La memoria queda vacía ya que no hay procesos en espera. La CPU también está vacía.



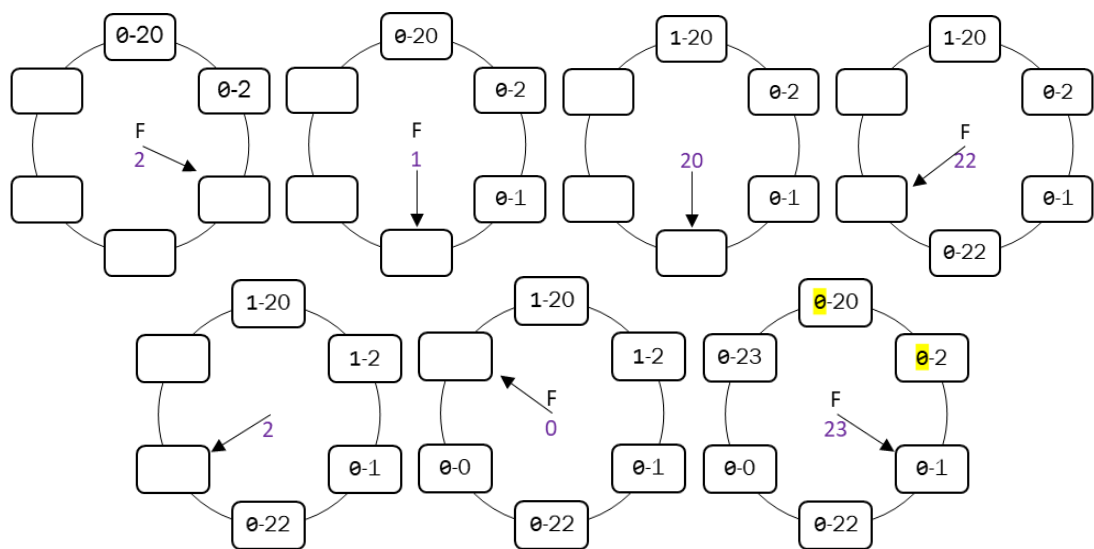
T=42

Llega **P05**, entra en memoria, ocupando 6 marcos, entra en la CPU. Se carga en memoria su primera página, produciéndose un fallo de página. El bit del marco 0 se pone a 0. El reloj apunta al marco siguiente.

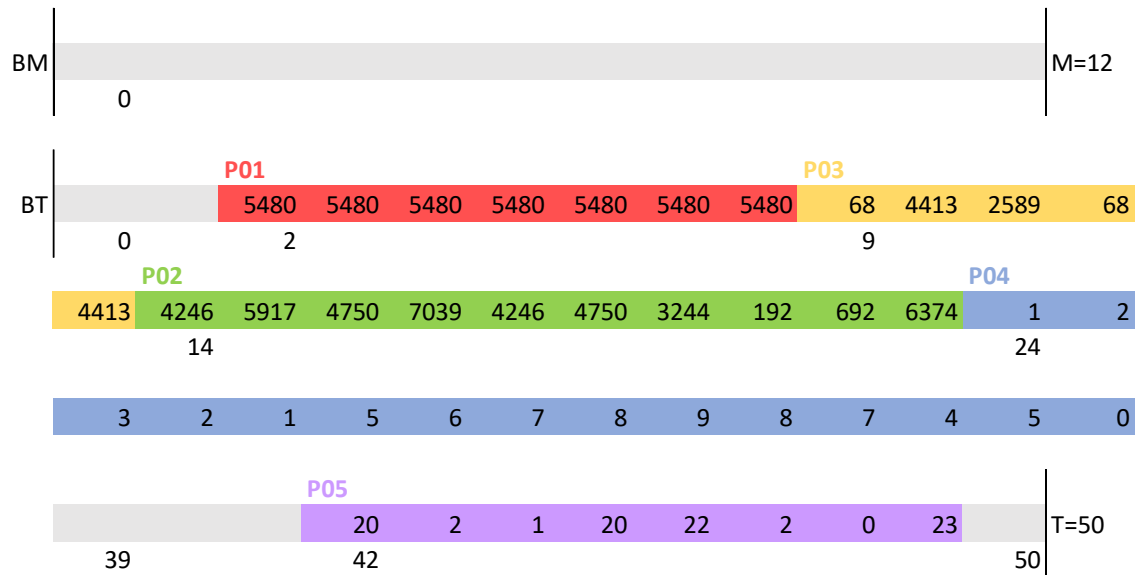


T=50

P05 ha ejecutado 7 páginas más, se han producido 5 fallos de página.



Finaliza la ejecución de P05, liberándose la CPU y los marcos que ocupaba en memoria.



FIN

Ejercicio resuelto con el Script

Muestra de la ejecución del script

Script se ejecuta desde un terminal en Linux con bash.

```
#####
#                               #
#          © Creative Commons   #
#          BY - Atribución (BY) #
#          NC - No uso Comercial (NC) #
#          SA - Compartir Igual (SA) #
#                               #
#          Créditos:           #
#          LRU 1º: Ruben Uruñuela, Alejandro caballero #
#          LRU 2º: Daniel Delgado, Ruben Marcos #
#          LRU 3º: Daniel Mellado, Noelia Ubierna #
#          LRU 4º: Fernando Antón Ortega & Daniel Beato de la Torre #
#          Reloj: Ismael Franco Hernando #
#          Reloj: Luis Miguel Agüero Hernando, Alberto Diez Busto #
#          NFU: Catalin Andrei, Cacuci #
#                               #
#          Algoritmo de gestión de procesos: FCFS/SJF #
#          Gestión de memoria: PAGINACIÓN #
#          Algoritmo de reemplazo de páginas: SEGUNDA OPORTUNIDAD #
#          Memoria continua: SÍ #
#          Memoria reubicable: SÍ #
#                               #
#          Autor: Rocío Esteban Valverde #
#          Sistemas Operativos 2º Semestre #
#          Grado en ingeniería informática (2020-2021) #
#                               #
#          Tutor: José Manuel Saiz Diez #
#                               #
#####
Pulsa INTRO para continuar. 
```

Pantalla de inicio, con los datos del script, copyright, créditos a antiguos creadores, la información sobre los algoritmos, y sobre la asignatura.

Después de pulsar intro, se le pide al usuario que escoja algunas opciones para ejecutar los ejercicios.

```
#####
#                               #
#          ALGORITMO           #
#          GESTIÓN DE MEMORIA VIRTUAL #
#          FCFS/SJF + Paginación + Segunda Oportunidad: #
#          _____ #
#                               #
#####
¿Desea leer el fichero de ayuda o ejecutar el algoritmo?

1- Ejecutar el algoritmo
2- Visualizar la ayuda
3- Salir

Seleccione una opción: 
```

Primero se ofrece la posibilidad de ver el fichero de ayuda antes de ejecutar el algoritmo, una vez vista la ayuda volvemos a tener las mismas opciones.


```
#####
#
#          ALGORITMO
#      GESTIÓN DE MEMORIA VIRTUAL
#  FCFS/SJF + Paginación + Segunda Oportunidad:
#
#####

¿Dónde quieres guardar los informes?

1- En el directorio estandar (datosScript/informes)
2- En otro directorio

Seleccione una opción: ☐
```

Durante la ejecución del script se crearán unos informes que se guardan en una carpeta por defecto, pero se da la opción de guardarlos en otro directorio. Si se introduce un directorio que ya tiene informes se da la opción de sobrescribirlos o elegir otro directorio.

```
#####
#
#          ALGORITMO
#      GESTIÓN DE MEMORIA VIRTUAL
#  FCFS/SJF + Paginación + Segunda Oportunidad:
#
#####

Elige un método de ordenación de procesos:

1.Algoritmo FCFS.
2.Algoritmo SJF

Seleccione una opción: ☐
```

Lo siguiente que se debe configurar es el algoritmo de planificación de procesos.

```
#####
#
#          ALGORITMO
#      GESTIÓN DE MEMORIA VIRTUAL
#  FCFS + Paginación + Segunda Oportunidad:
#
#####

¿Cómo desea introducir los datos?

1- Por teclado
2- Fichero con los datos de la última ejecución (datos.txt)
3- Otro fichero de datos
Generar aleatoriamente:
4- Introducir rangos por teclado
5- Fichero con los rangos de la última ejecución (datosrangos.txt)
6- Otro fichero de rangos

Seleccione una opción: ☐
```

Hay 6 formas de introducir los datos al programa. Se debe elegir uno. Están las opciones de meter datos directamente ya sea manualmente o mediante ficheros y las de crear datos aleatorios. En ambos modos se dan las opciones de introducir datos o rangos a mano, desde el fichero de última ejecución o desde otro fichero, esta última da una lista de los ficheros disponibles en el directorio de datos o rangos respectivamente.

```
#####
#
#          ALGORITMO
#      GESTIÓN DE MEMORIA VIRTUAL
#      FCFS + Paginación + Segunda Oportunidad:
#
#####

[Algoritmo usado: FCFS
|Memoria del Sistema: 1200
|Tamaño de Página: 100
|Marcos totales de la memoria: 12
Ref Tll Tej nMar Dirección-Página
P01 2 7 5 548057-5480 548057-5480 548057-5480 548057-5480 548057-5480
548057-5480
P02 3 10 5 424666-4246 591789-5917 475005-4750 703990-7039 424666-4246 475005-4750
324430-3244 19268-192 69279-692 637416-6374
P03 4 5 3 6869-68 441318-4413 258922-2589 6869-68 441318-4413
P04 10 15 3 111-1 222-2 333-3 222-2 111-1 555-5 666-6 777-7 888-8 999-9 888-8
777-7 444-4 555-5 66-0
P05 42 8 6 2021-20 200-2 100-1 2022-20 2222-22 200-2 0-0 2323-23

Esto es un resumen de los datos obtenidos. Pulsa INTRO para continuar ☐
```

Una vez obtenidos los datos se muestra un resumen de estos con los datos del sistema y los procesos ordenados según el tiempo de llegada. Estos son los datos del ejemplo resuelto a mano y que ejecutaremos luego con el script.

```
#####
#
#          ALGORITMO
#      GESTIÓN DE MEMORIA VIRTUAL
#      FCFS + Paginación + Segunda Oportunidad:
#
#####

Modo de visualización de la ejecución:

1- Por eventos (Pulsar Enter en cada evento)
2- Automático (Esperar unos segundos entre cada evento)
3- Completo (Directo al resumen con datos de la ejecución y fallos de página)

Seleccione una opción: ☐
```

Por último, se dan tres opciones de visualización de la ejecución. Al final de la ejecución siempre se muestra un resumen de la ejecución de cada proceso.

```
FCFS/SJF+Paginación+Seg.Op+M. Continúa y Reubicable
T=3 | Algoritmo usado: FCFS | Memoria del Sistema: 1200 | Tamaño de página: 100 | Marcos totales de la memoria: 12
Llega el proceso P02
Entra a memoria el proceso P02
Se sigue ejecutando el proceso P01

Ref Tll Tej nMar Tresp Tret Trej Mini Mfin Estado Dirección-Página
P01 2 7 5 0 1 6 0 4 En ejecución 548057-5480 548057-5480 548057-5480 548057-5480
P02 3 10 5 0 0 10 5 9 En memoria 424666-4246 591789-5917 475005-4750 703990-7039 424666-424
19268-192 69279-692 637416-6374
P03 4 5 3 - - - - - Fuera de sistema 6869-68 441318-4413 258922-2589 6869-68 441318-4413
P04 10 15 3 - - - - - Fuera de sistema 111-1 222-2 333-3 222-2 111-1 555-5 666-6 777-7 888-8 999-
555-5 66-0
P05 42 8 6 - - - - - Fuera de sistema 2021-20 200-2 100-1 2022-20 2222-22 200-2 0-0 2323-23

T. MEDIO ESPERA: 0 T. MEDIO RETORNO: 0.500
Cola de ejecución: P02
Marcos de página
P01 P02
M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 | M=12
5480 - - - - - - - - - - - -
1 - - - - - - - - - - - -

BM | P01 P02 | M=12
| 5480 - - - - - 5 - - - - - |
| 0 - - - - - 10 - - - - - |

BT | P01 | T=3
| 0 2 |
```

Pulsa **INTRO** para continuar ☐

En cada pantalla de evento vemos una pantalla como esta.

Dónde nos tenemos que fijar, como nos indica la imagen son (empezando de izquierda a derecha de arriba abajo):

- T=_ indica el instante de la ejecución que se está mostrando. Estos momentos son en los que el script ha detectado que ha habido un evento importante.
- Lista de los eventos importantes por los que se está parando para mostrar lo que sucede en ese instante.
- Tesp, Tret, Trej, Mini, Mfin: que son el tiempo de espera de cada proceso, el tiempo de retorno, tiempo restante de ejecución y las posiciones de la banda de memoria de inicio y fin.
- El estado, que indica el estado de cada proceso, es decir si está en espera, en memoria, si está ejecutándose...
- Direcciones, cada dirección junto con la página en negrilla. Las que están subrayadas han sido ejecutadas.
- Tiempo medio de espera y tiempo medio de retorno, nos muestra el promedio del tiempo de espera y de retorno en cada evento.
- Cola de ejecución, muestra los procesos que esperan a ejecutarse y en qué orden pasarán a la CPU.
- Resumen de marcos, destaca que se puede ver la información del bit de referencia de cada marco del proceso que esté en ejecución.
- Banda de memoria, que nos indica los colores de los procesos en memoria, con las proporciones correctas. En la banda de color se muestra la página que hay en cada marco o si está vacío.
- Banda de tiempo, que nos indica los colores de los procesos con el tiempo en ejecución de cada uno y la página que se ha ejecutado en cada unidad de tiempo.

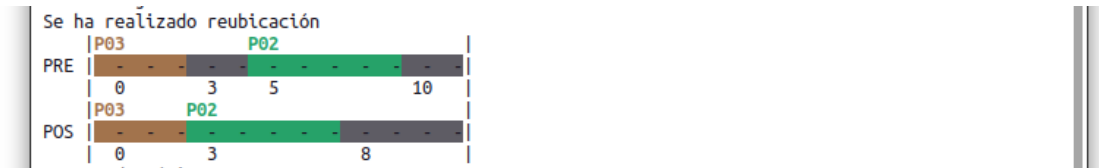
Otros diagramas que pueden aparecer son:

Se han producido 11 fallos de página en la ejecución del proceso P04

	1	2	3	2	1	5	6	7	8	9	8	7	4	5	0
M08	0=1	0-1	0-1		1=1	0-1	0-1	0-1	0=8	0-8	1=8	0-8	0-8	0=5	0-5
M09		0=2	0-2	1=2	0-2	0-2	0=6	0-6	0-6	0=9			0=4	0-4	0=0
M10			0=3			0=5	0-5	0=7	0-7	0-7		1=7	0-7	0-7	0-7
	F	F	F			F	F	F	F			F	F	F	F

- Resumen de fallos de página, que se mostrará cuando un proceso finalice su ejecución. Consiste en una tabla con los marcos del proceso como filas y una columna por cada página que se ha ejecutado. Cada celda contiene el estado del marco cuando se ha ejecutado esa página, vemos 'bit de referencia-página'. En cada columna la información en la que hay que fijarse es:
 - La celda que no está en negrilla y separa el bit de referencia y la página con un = en vez de un -, indica que esa es la página que se ha ejecutado.
 - Celda subrayada marca a qué marco apunta el indicador del reloj. Este será el marco en el que entrará la siguiente página cuando se produzca fallo de página. Se muestra directamente ya cambiados los bits por los que este índice ha pasado hasta llegar a su posición definitiva. Vemos en el ejemplo la columna marcada en amarillo, la página 5 ha entrado donde apuntaba el indicador en la columna anterior. El bit del primer marco en la columna anterior estaba a 1, ahora está a 0 y el indicador está en el siguiente marco.
 - Si hay una F abajo o no. Esto indica si ha habido fallo de página o no. Si ha habido fallo de página se mostrará el contenido de todos los marcos, si no ha habido fallo de página, solo se muestra la página que se ha ejecutado y en caso de que el indicador haya cambiado el bit de algún

marco, esa celda también se mostrará para que quede claro lo que ha sucedido. Esto se ve en el ejemplo en la columna marcada de naranja.



- Reubicación, muestra dos barras de memoria con el estado previo 'PRE' y posterior 'POS' a la reubicación. En este proceso se mueve el proceso con el contenido de cada marco que ocupaba, lo que incluye las páginas y los bits de referencia.

A continuación, se mostrará la ejecución de un ejemplo.

Ejercicio 1

Ejecutaremos el mismo ejercicio que a mano, ya que es muy completo y muestra una variedad de situaciones. Datos:

```

Algoritmo usado: FCFS
Memoria del Sistema: 1200
Tamaño de Página: 100
Marcos totales de la memoria: 12
Ref Tll Tej nMar Dirección-Página
P01 2 7 5 548057-5480 548057-5480 548057-5480 548057-5480 548057-5480 548057-5480
548057-5480
P02 3 10 5 424666-4246 591789-5917 475005-4750 703990-7039 424666-4246 475005-4750
324430-3244 19268-192 69279-692 637416-6374
P03 4 5 3 6869-68 441318-4413 258922-2589 6869-68 441318-4413
P04 10 15 3 111-1 222-2 333-3 222-2 111-1 555-5 666-6 777-7 888-8 999-9 888-8
777-7 444-4 555-5 66-6
P05 42 8 6 2021-20 200-2 100-1 2022-20 2222-22 200-2 0-0 2323-23
  
```

Las direcciones que no caben en la pantalla porque tienen muchos dígitos y se truncarían, saltan a la siguiente línea

Se ejecutarán estos datos con el algoritmo FCFS y después con SJF.

FCFS

T=0

```

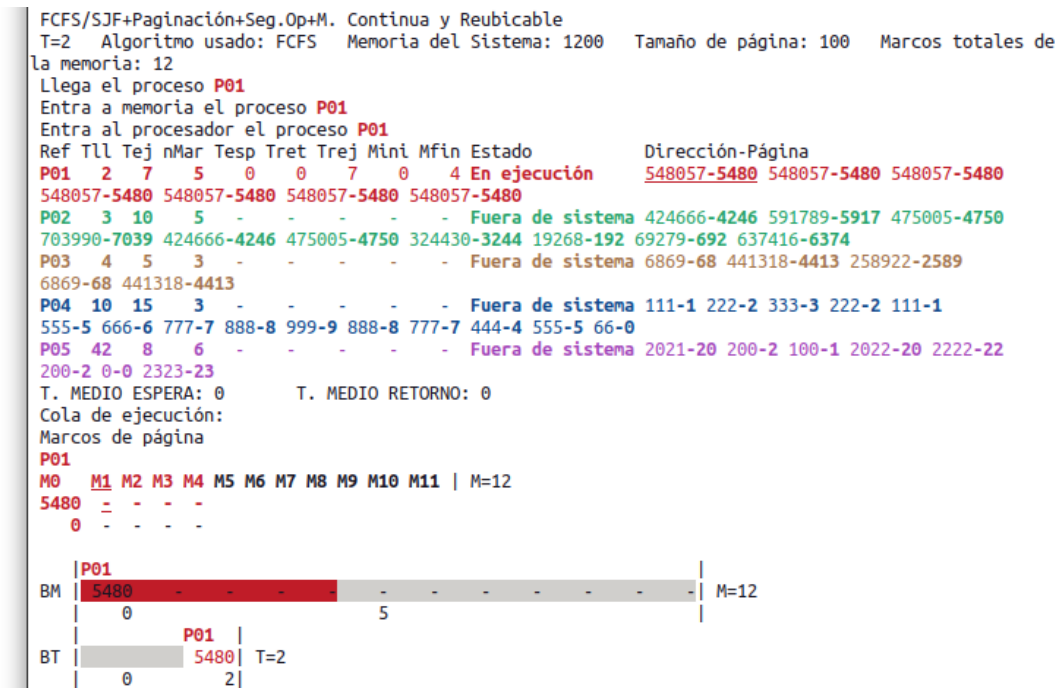
FCFS/SJF+Paginación+Seg.Op+M. Continua y Reubicable
T=0 Algoritmo usado: FCFS Memoria del Sistema: 1200 Tamaño de página: 100 Marcos totales de
la memoria: 12
Ref Tll Tej nMar Tresp Tret Trej Mini Mfin Estado Dirección-Página
P01 2 7 5 - - - - - Fuera de sistema 548057-5480 548057-5480 548057-5480
548057-5480 548057-5480 548057-5480 548057-5480
P02 3 10 5 - - - - - Fuera de sistema 424666-4246 591789-5917 475005-4750
703990-7039 424666-4246 475005-4750 324430-3244 19268-192 69279-692 637416-6374
P03 4 5 3 - - - - - Fuera de sistema 6869-68 441318-4413 258922-2589
6869-68 441318-4413
P04 10 15 3 - - - - - Fuera de sistema 111-1 222-2 333-3 222-2 111-1
555-5 666-6 777-7 888-8 999-9 888-8 777-7 444-4 555-5 66-6
P05 42 8 6 - - - - - Fuera de sistema 2021-20 200-2 100-1 2022-20 2222-22
200-2 0-0 2323-23
T. MEDIO ESPERA: 0 T. MEDIO RETORNO: 0
Cola de ejecución:
Marcos de página

M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 | M=12

BM | - - - - - - - - - - - - | M=12
  | 0 |
BT | - - - - - - - - - - - - |
  | 0 |
  
```

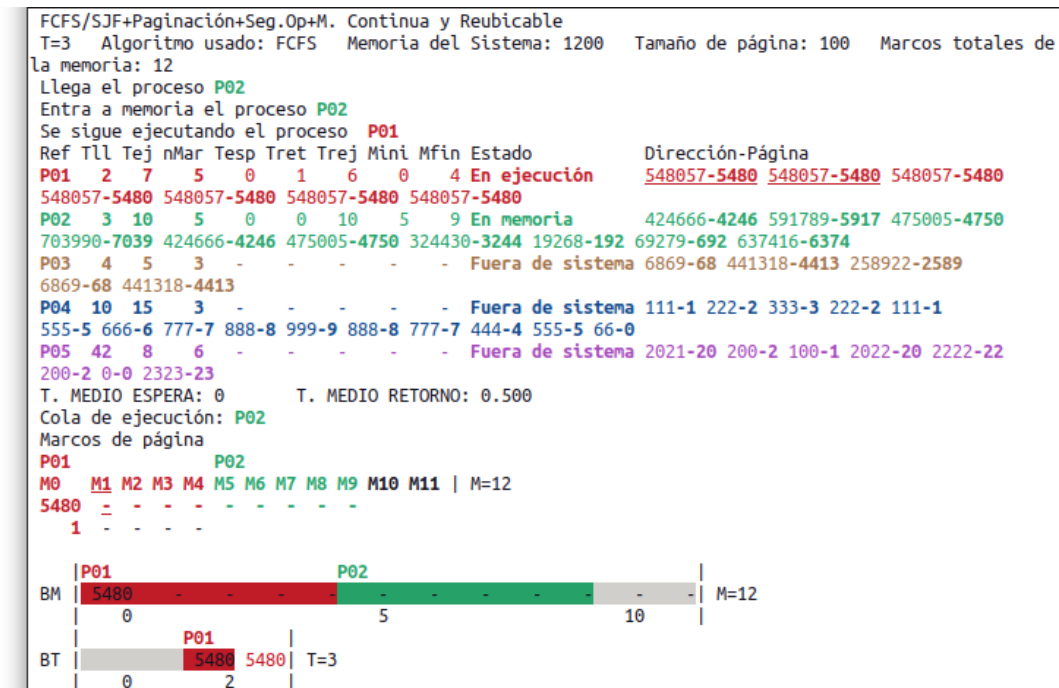
No ha llegado ningún proceso, el estado de todos los procesos es 'Fuera de sistema', no hay nada en la cola de ejecución y no hay nada en memoria.

T=2



Llega **P01** . Entra en memoria ocupando 5 marcos y como la CPU está libre comienza a ejecutarse, introduciendo su primera página, que produce un fallo de página. En el resumen de los marcos se puede ver que el bit del marco 0 se pone a 0. El indicador apunta al siguiente marco M1.

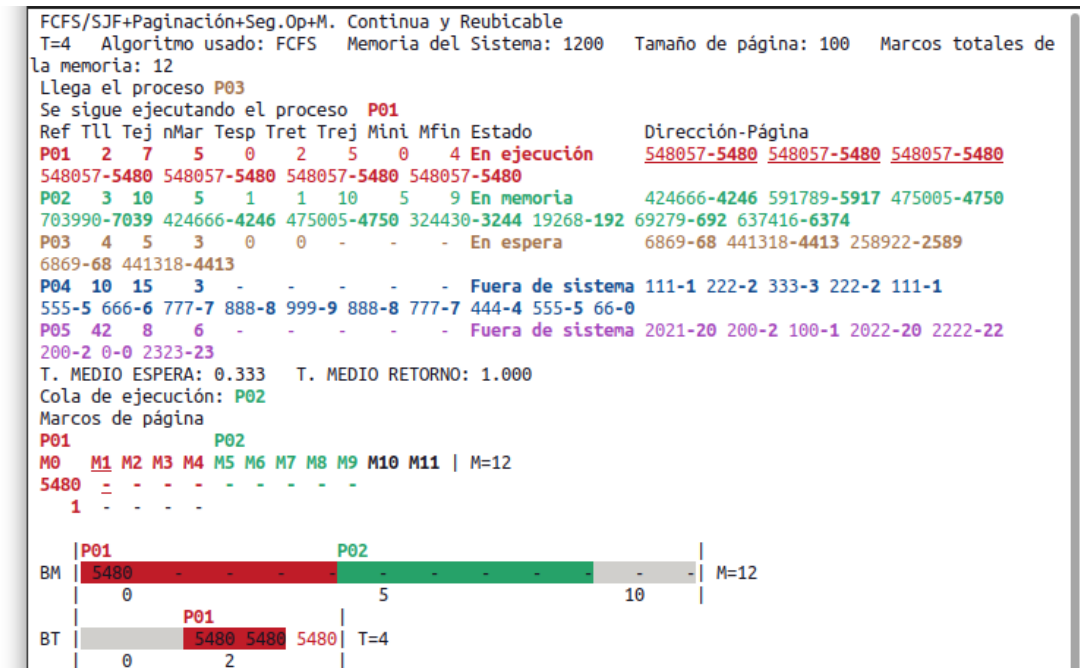
T=3



Llega **P02**, pasa a estar en memoria, ocupando 5 marcos, del 5 al 9, quedan 2 marcos libres. Se queda en la cola de ejecución. A **P01** le quedan seis páginas por ejecutar, la que

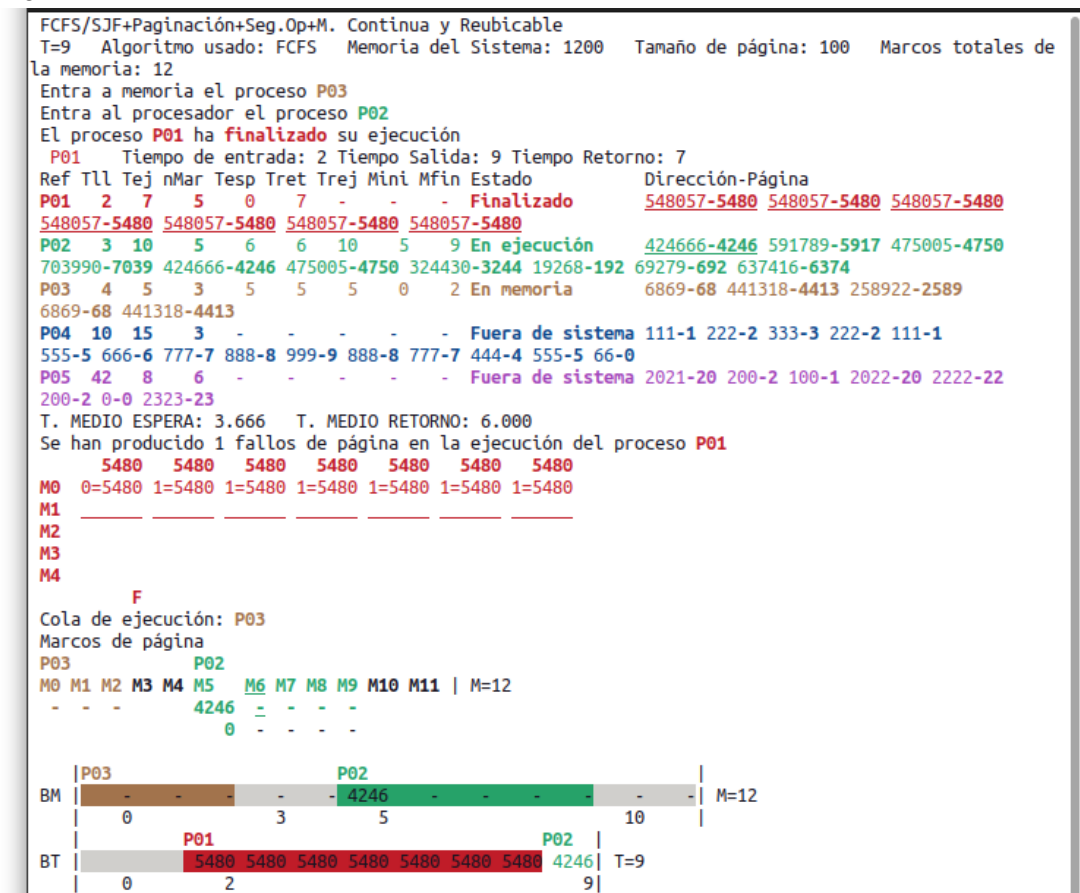
se está ejecutando y cinco más, en ninguna se producirá fallo de página ya que todas son la misma que está en memoria. El bit del M0 está a 1.

T=4



Llega P03, pero como ocupa tres marcos y solo hay dos libres, se queda en espera.

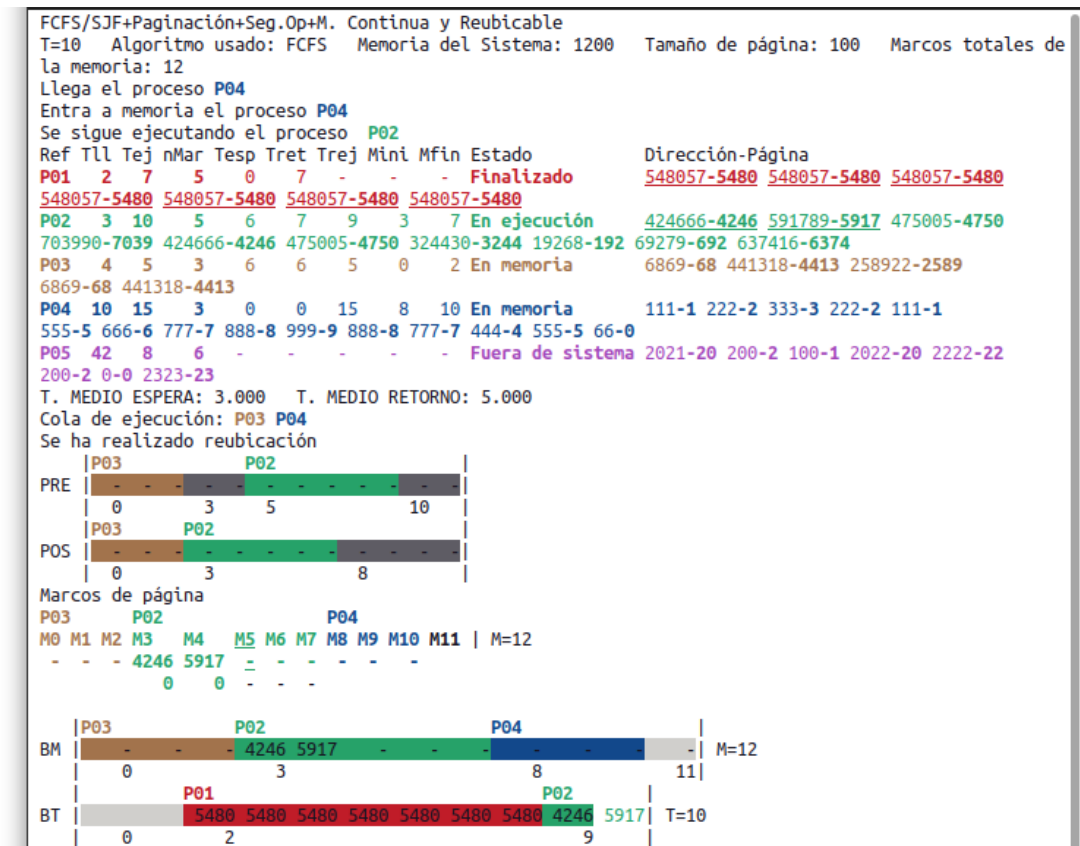
T=9



P01 ha finalizado su ejecución con un total de 1 fallo de página y se liberan los 5 marcos de la memoria que estaba ocupando. En ese hueco cabe **P03** que ahora sí entra en memoria, ocupa 3 marcos de página. Quedan dos huecos de dos marcos libres cada uno.

La CPU ha quedado libre, en la cola de ejecución estarían ahora **P02** y **P03**. Por FCFS, como **P02** ha llegado antes, es el que entra en ejecución. Mete a ejecutar su primera página produciéndose un fallo de página. El bit del marco 5 se pone a 0. El reloj apunta al marco siguiente.

T=10



Llega **P04**, necesita 3 marcos para entrar en memoria, hay 4 marcos libres, pero están separados por lo que se hace REUBICACIÓN. Los marcos de **P02** pasan de [5, 9] a [3, 7].

Tras la reubicación **P04** entra en memoria, ocupando los marcos [8, 10].

Ahora en la cola están **P03** y **P04**, en ese orden porque así han llegado.

T=19



P02 finaliza su ejecución con un total de 8 fallos de página. Se liberan los marcos que ocupaba.

En esta pantalla se muestra el resumen de fallos, y hay algo que quizá llame la atención. La primera página empieza en el marco 5, y el puntero en el 6, pero en la siguiente página están en el marco 3 y 4 y la siguiente página ha entrado en el marco 6. Esto se debe a la reubicación. Antes de la reubicación se ha registrado que la página estaba en ese marco, pero con la reubicación se ha desplazado el proceso junto con todas sus páginas y los datos de los marcos en los que estaba.

La CPU ha quedado libre así que entra un nuevo proceso. Como es FCFS, entra en ejecución **P03**, que es el que ha llegado antes entre los procesos que hay en memoria. Se carga en memoria su primera página, produciéndose un fallo de página. El bit del marco 0 se pone a 0. El reloj apunta al marco siguiente.

T=24

```

FCFS/SJF+Paginación+Seg.op+M. Continua y Reubicable
T=24 Algoritmo usado: FCFS Memoria del Sistema: 1200 Tamaño de página: 100 Marcos totales de
la memoria: 12
Entra al procesador el proceso P04
El proceso P03 ha finalizado su ejecución
P03 Tiempo de entrada: 19 Tiempo Salida: 24 Tiempo Retorno: 20
Ref Tll Tej nMar Tesp Tret Trej Mini Mfin Estado Dirección-Página
P01 2 7 5 0 7 - - - Finalizado 548057-5480 548057-5480 548057-5480
548057-5480 548057-5480 548057-5480 548057-5480
P02 3 10 5 6 16 - - - Finalizado 424666-4246 591789-5917 475005-4750
703990-7039 424666-4246 475005-4750 324430-3244 19268-192 69279-692 637416-6374
P03 4 5 3 15 20 - - - Finalizado 6869-68 441318-4413 258922-2589
6869-68 441318-4413
P04 10 15 3 14 14 15 8 10 En ejecución 111-1 222-2 333-3 222-2 111-1
555-5 666-6 777-7 888-8 999-9 888-8 777-7 444-4 555-5 66-0
P05 42 8 6 - - - - Fuera de sistema 2021-20 200-2 100-1 2022-20 2222-22
200-2 0-0 2323-23
T. MEDIO ESPERA: 8.750 T. MEDIO RETORNO: 14.250
Se han producido 3 fallos de página en la ejecución del proceso P03
68 4413 2589 68 4413
M0 0=68 0-0068 0-0068 1=0068
M1 0=4413 0-4413 1=4413
M2 0=2589
F F F
Cola de ejecución:
Marcos de página
P04
M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 | M=12
1 - -
0 - -
BM | - - - - - - - - - - P04 |
| 0 - - - - - - - - 8 - - - 11 | M=12
| P01 P02
BT | 5480 5480 5480 5480 5480 5480 5480 4246 5917 4750 7039 4246 4750 3244 192 692 6374
| 0 2 9
P03 P04 |
68 4413 2589 68 4413 1 | T=24
19 24

```

P03 finaliza su ejecución con 3 fallos de página y libera los marcos que ocupaba.

La CPU ha quedado libre así que entra **P04**, que es el único que queda en memoria. Se carga en memoria su primera página, produciéndose un fallo de página. El bit del marco 8 se pone a 0. El reloj apunta al marco siguiente.

T=39

```

FCFS/SJF+Paginación+Seg.Op+M. Continua y Reubicable
T=39 Algoritmo usado: FCFS Memoria del Sistema: 1200 Tamaño de página: 100 Marcos totales de
la memoria: 12
El proceso P04 ha finalizado su ejecución
P04 Tiempo de entrada: 24 Tiempo Salida: 39 Tiempo Retorno: 29
Ref Tll Tej nMar Tesp Tret Trej Mini Mfin Estado Dirección-Página
P01 2 7 5 0 7 - - - Finalizado 548057-5480 548057-5480 548057-5480
548057-5480 548057-5480 548057-5480 548057-5480
P02 3 10 5 6 16 - - - Finalizado 424666-4246 591789-5917 475005-4750
703990-7039 424666-4246 475005-4750 324430-3244 19268-192 69279-692 637416-6374
P03 4 5 3 15 20 - - - Finalizado 6869-68 441318-4413 258922-2589
6869-68 441318-4413
P04 10 15 3 14 29 - - - Finalizado 111-1 222-2 333-3 222-2 111-1
555-5 666-6 777-7 888-8 999-9 888-8 777-7 444-4 555-5 66-0
P05 42 8 6 - - - - Fuera de sistema 2021-20 200-2 100-1 2022-20 2222-22
200-2 0-0 2323-23
T. MEDIO ESPERA: 8.750 T. MEDIO RETORNO: 18.000
Se han producido 11 fallos de página en la ejecución del proceso P04
1 2 3 2 1 5 6 7 8 9 8 7 4 5 0
M08 0=1 0-1 0-1 1=1 0-1 0-1 0-1 0=8 0-8 1=8 0-8 0-8 0=5 0-5
M09 0=2 0-2 1=2 0-2 0-2 0=6 0-6 0-6 0=9 0=4 0-4 0=0
M10 0=3 0=5 0-5 0=7 0-7 0-7 1=7 0-7 0-7 0-7
F F F F F F F F F F F
Cola de ejecución:
Marcos de página
M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 | M=12

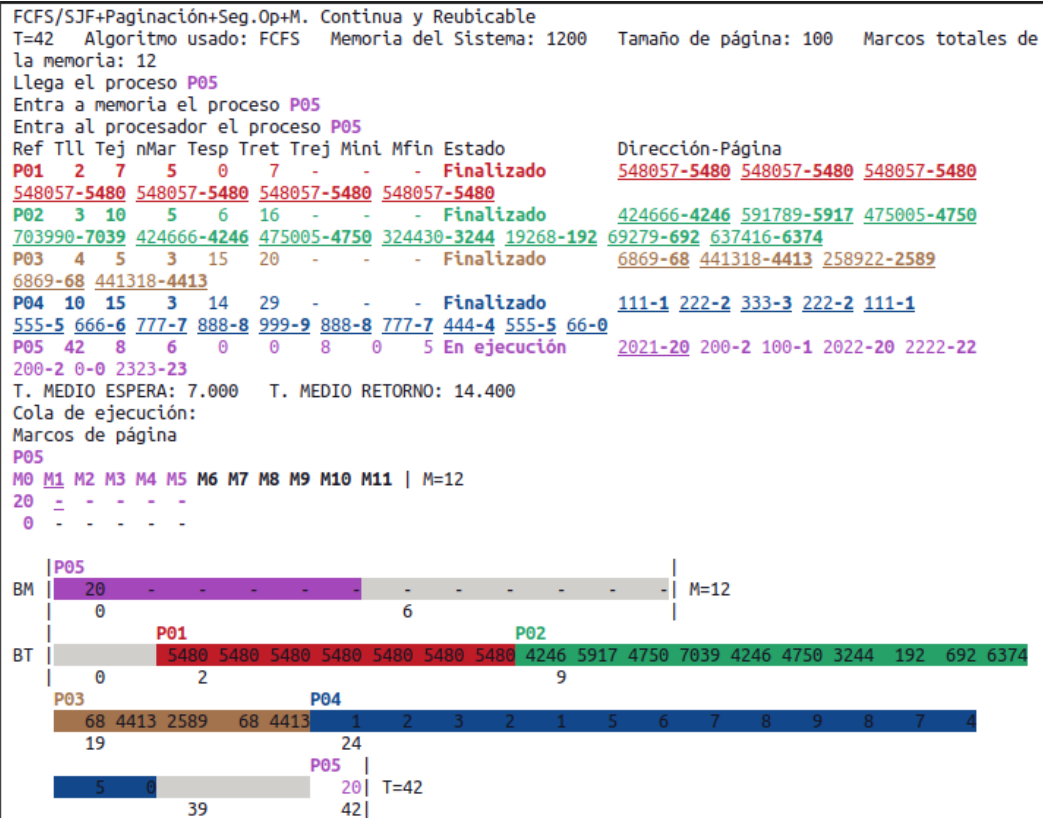
BM | - - - - - - - - - - - - - - | M=12
0
BT | P01 5480 5480 5480 5480 5480 5480 5480 4246 5917 4750 7039 4246 4750 3244 192 692 6374
0 2 9
P03 68 4413 2589 68 4413 P04 1 2 3 2 1 5 6 7 8 9 8 7 4
19 24
5 0 T=39
39

```

P04 finaliza su ejecución con un total de 11 fallos de página y se liberan los marcos que ocupaba.

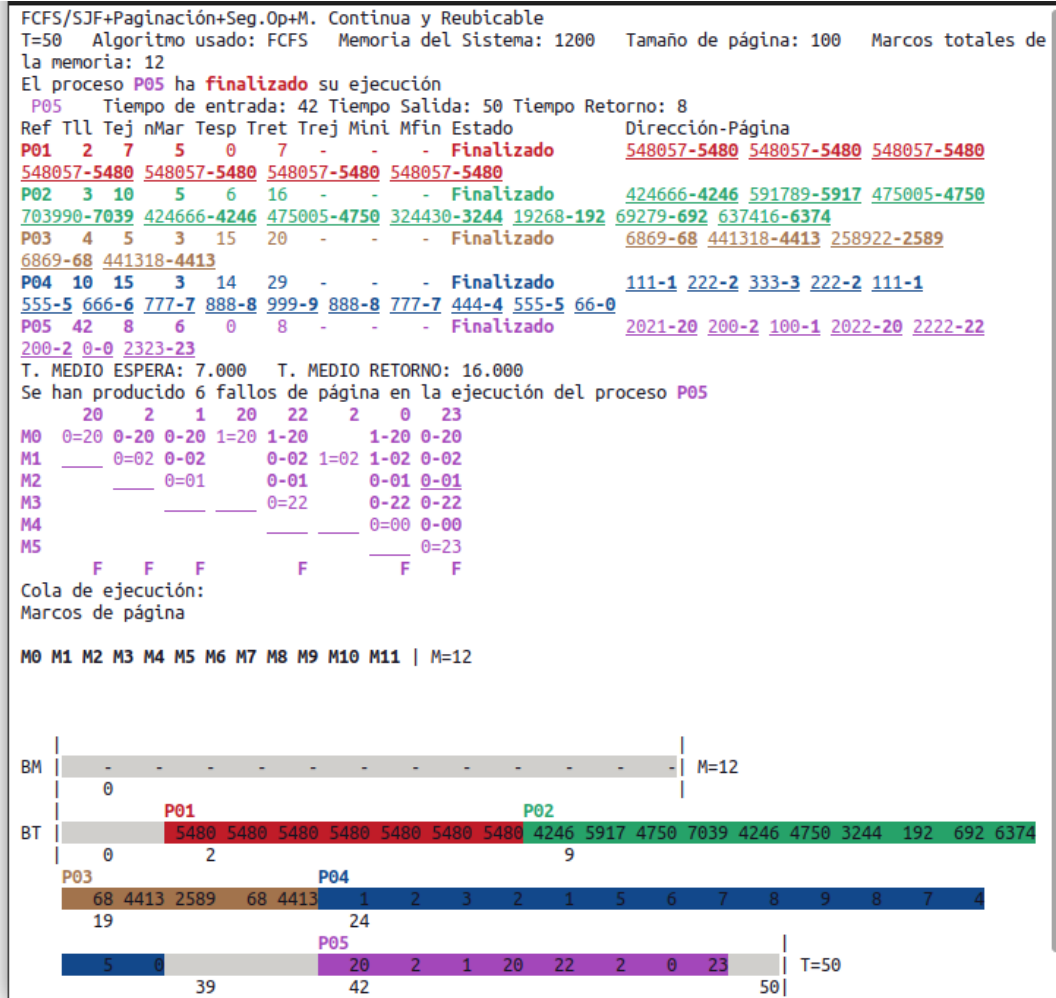
La memoria queda vacía ya que no hay procesos en espera y la CPU también está vacía, pero queda un proceso por llegar.

T=42



El tiempo ha avanzado hasta la llegada de P05, que entra en memoria, ocupando 6 marcos y entra en la CPU. Se carga en memoria su primera página, produciéndose un fallo de página. El bit del marco 0 se pone a 0. El reloj apunta al marco siguiente.

T=50



Finaliza la ejecución de **P05** con un total de 6 fallos de página, se libera la CPU y los marcos que ocupaba en memoria.

Era el último proceso por lo que a continuación se muestra el resumen final.

T.ESPERA: Tiempo que el proceso no ha estado ejecutándose en la CPU desde que entra en memoria hasta que sale
ESPERA TOTAL: Tiempo que el proceso no ha estado ejecutándose en la CPU desde T=0 (Inicio del sistema) hasta que sale
INICIO/FIN: Tiempo de llegada al gestor de memoria del proceso y tiempo de salida del proceso
T.RETORNO: Tiempo total de ejecución del proceso, incluyendo tiempos de espera, desde la señal de entrada hasta la salida

RESUMEN FINAL con tiempos de ejecución y fallos de página de cada proceso

Proceso	T.Espera	Inicio/Fin	T.Retorno	Fallos de página
P01	0	2/9	7	1
P02	6	9/19	16	8
P03	15	19/24	20	3
P04	14	24/39	29	11
P05	0	42/50	8	6

Tiempo total transcurrido en ejecutar todos los procesos: 50
Media tiempo espera de todos los procesos: 7.000
Media tiempo retorno de todos los procesos: 16.000

Fin de la ejecución.

Vemos que los procesos 3 y 4 han tenido tiempos de espera mayores eso es porque han sido de los últimos en llegar. El proceso 5 es diferente porque, aunque ha llegado muy

tarde no había llegado ningún otro proceso recientemente y para cuando ha llegado, los que estaban antes en ejecución ya habían terminado.

SJF

T=0

```
FCFS/SJF+Paginación+Seg.Op+M. Continua y Reubicable
T=0 Algoritmo usado: SJF Memoria del Sistema: 1200 Tamaño de página: 100 Marcos totales de l
a memoria: 12
Ref Tll Tej nMar Tesp Tret Trej Mini Mfin Estado Dirección-Página
P01 2 7 5 - - - - - Fuera de sistema 548057-5480 548057-5480 548057-5480
548057-5480 548057-5480 548057-5480 548057-5480
P02 3 10 5 - - - - - Fuera de sistema 424666-4246 591789-5917 475005-4750
703990-7039 424666-4246 475005-4750 324430-3244 19268-192 69279-692 637416-6374
P03 4 5 3 - - - - - Fuera de sistema 6869-68 441318-4413 258922-2589
6869-68 441318-4413
P04 10 15 3 - - - - - Fuera de sistema 111-1 222-2 333-3 222-2 111-1
555-5 666-6 777-7 888-8 999-9 888-8 777-7 444-4 555-5 66-0
P05 42 8 6 - - - - - Fuera de sistema 2021-20 200-2 100-1 2022-20 2222-22
200-2 0-0 2323-23
T. MEDIO ESPERA: 0 T. MEDIO RETORNO: 0
Cola de ejecución:
Marcos de página

M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 | M=12

BM | - - - - - - - - - - - - - - | M=12
| 0 |
BT | - | T=0
| 0 |

Pulsa INTRO para continuar
```

No ha llegado ningún proceso, el estado de todos los procesos es 'Fuera de sistema', no hay nada en la cola de ejecución y no hay nada en memoria.

T=2

```
FCFS/SJF+Paginación+Seg.Op+M. Continua y Reubicable
T=2 Algoritmo usado: SJF Memoria del Sistema: 1200 Tamaño de página: 100 Marcos totales de l
a memoria: 12
Llega el proceso P01
Entra a memoria el proceso P01
Entra al procesador el proceso P01
Ref Tll Tej nMar Tesp Tret Trej Mini Mfin Estado Dirección-Página
P01 2 7 5 0 0 7 0 4 En ejecución 548057-5480 548057-5480 548057-5480
548057-5480 548057-5480 548057-5480 548057-5480
P02 3 10 5 - - - - - Fuera de sistema 424666-4246 591789-5917 475005-4750
703990-7039 424666-4246 475005-4750 324430-3244 19268-192 69279-692 637416-6374
P03 4 5 3 - - - - - Fuera de sistema 6869-68 441318-4413 258922-2589
6869-68 441318-4413
P04 10 15 3 - - - - - Fuera de sistema 111-1 222-2 333-3 222-2 111-1
555-5 666-6 777-7 888-8 999-9 888-8 777-7 444-4 555-5 66-0
P05 42 8 6 - - - - - Fuera de sistema 2021-20 200-2 100-1 2022-20 2222-22
200-2 0-0 2323-23
T. MEDIO ESPERA: 0 T. MEDIO RETORNO: 0
Cola de ejecución:
Marcos de página
P01
M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 | M=12
5480 - - - - - - - - - - - - - -
0 - - - - -

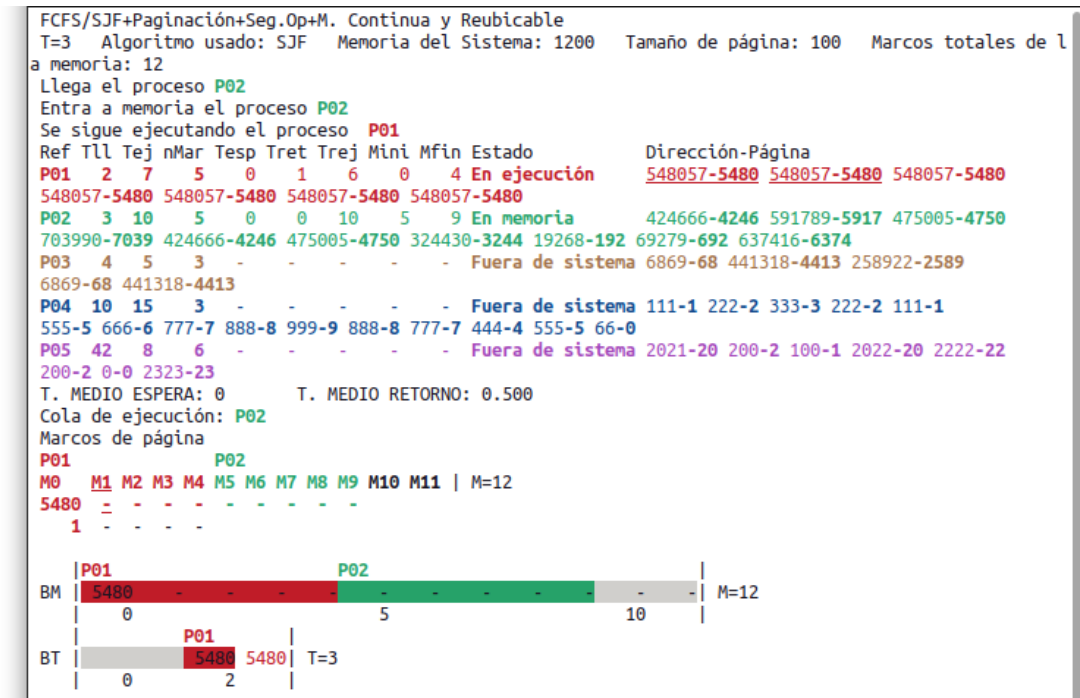
BM | P01 |
| 5480 - - - - - - - - - - - - - - | M=12
| 0 | 5 |
BT | P01 |
| 5480 - - - - - - - - - - - - - - | T=2
| 0 | 2 |

Pulsa INTRO para continuar
```

Llega **P01**. Entra en memoria ocupando 5 marcos y como la CPU está libre comienza a ejecutarse, introduciendo su primera página, que produce un fallo de página. En el

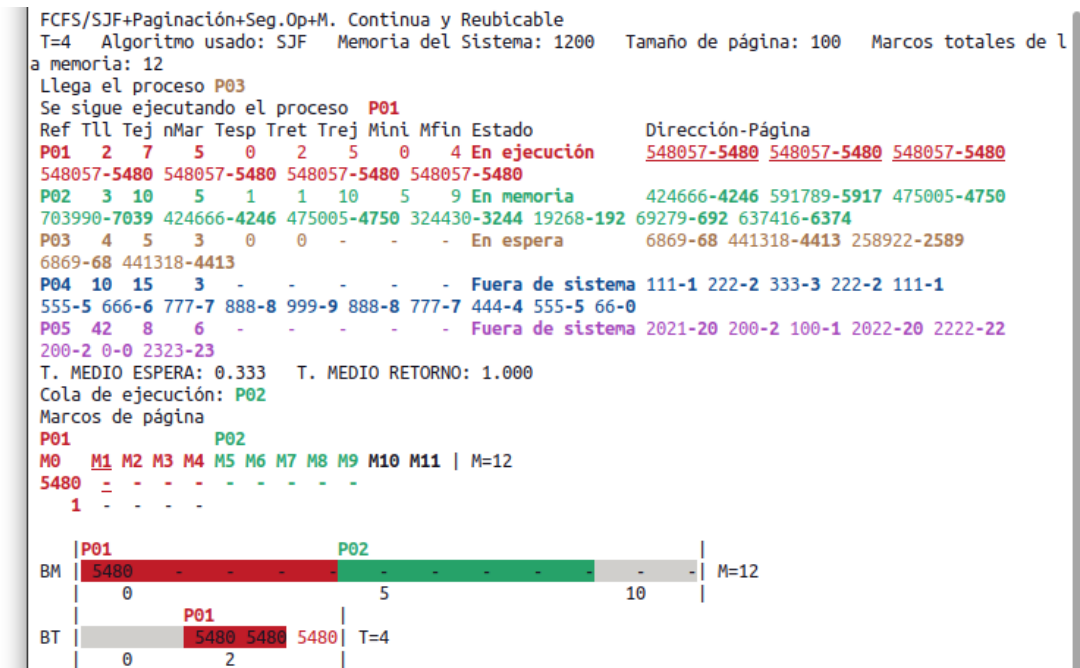
resumen de los marcos se puede ver que el bit del marco 0 se pone a 0. El indicador apunta al siguiente marco M1.

T=3



Llega **P02**, pasa a estar en memoria, ocupando 5 marcos, del 5 al 9, quedan 2 marcos libres. Se queda en la cola de ejecución. A **P01** le quedan seis páginas por ejecutar, la que se está ejecutando y cinco más, en ninguna se producirá fallo de página ya que todas son la misma que está en memoria. El bit del M0 está a 1.

T=4



Llega **P03**, pero como ocupa tres marcos y solo hay dos libres, se queda en espera.

T=9

```

FCFS/SJF+Paginación+Seg.Op+M. Continua y Reubicable
T=9 Algoritmo usado: SJF Memoria del Sistema: 1200 Tamaño de página: 100 Marcos totales de 1
a memoria: 12
Entra a memoria el proceso P03
Entra al procesador el proceso P03
El proceso P01 ha finalizado su ejecución
P01 Tiempo de entrada: 2 Tiempo Salida: 9 Tiempo Retorno: 7
Ref Tll Tej nMar Tesp Tret Trej Mini Mfin Estado Dirección-Página
P01 2 7 5 0 7 - - - Finalizado 548057-5480 548057-5480 548057-5480
548057-5480 548057-5480 548057-5480 548057-5480
P02 3 10 5 6 6 10 5 9 En memoria 424666-4246 591789-5917 475005-4750
703990-7039 424666-4246 475005-4750 324430-3244 19268-192 69279-692 637416-6374
P03 4 5 3 5 5 5 0 2 En ejecución 6869-68 441318-4413 258922-2589
6869-68 441318-4413
P04 10 15 3 - - - - Fuera de sistema 111-1 222-2 333-3 222-2 111-1
555-5 666-6 777-7 888-8 999-9 888-8 777-7 444-4 555-5 66-0
P05 42 8 6 - - - - Fuera de sistema 2021-20 200-2 100-1 2022-20 2222-22
200-2 0-0 2323-23
T. MEDIO ESPERA: 3.666 T. MEDIO RETORNO: 6.000
Se han producido 1 fallos de página en la ejecución del proceso P01
5480 5480 5480 5480 5480 5480 5480
M0 0=5480 1=5480 1=5480 1=5480 1=5480 1=5480 1=5480
M1
M2
M3
M4
F
Cola de ejecución: P02
Marcos de página
P03 P02
M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 | M=12
68 - - - - - - - - - - -
0 - - - - - - - - - - -

```

```

BM | P03 | P02 | | | | | | | | | | M=12
   | 68 | - | - | - | - | - | - | - | - | - | - |
   | 0 | 3 | 5 | 10 | | | | | | | | |
BT | P01 | P03 | | | | | | | | | | T=9
   | 5480 5480 5480 5480 5480 5480 5480 | 68 |
   | 0 | 2 | 9 | | | | | | | | |

```

P01 ha finalizado su ejecución con un total de 1 fallo de página y se liberan los 5 marcos de la memoria que estaba ocupando. En ese hueco cabe **P03** que ahora sí entra en memoria, ocupa 3 marcos de página. Quedan dos huecos de dos marcos libres cada uno.

La CPU ha quedado libre, en la cola de ejecución estarían ahora **P02** y **P03**. Por SJF, como el tiempo de ejecución de **P03** es menor, es el que entra en ejecución. Mete a ejecutar su primera página produciéndose un fallo de página. El bit del marco 0 se pone a 0. El reloj apunta al marco siguiente.

T=10

```

FCFS/SJF+Paginación+Seg.Op+M. Continua y Reubicable
T=10 Algoritmo usado: SJF Memoria del Sistema: 1200 Tamaño de página: 100 Marcos totales de
la memoria: 12
Llega el proceso P04
Entra a memoria el proceso P04
Se sigue ejecutando el proceso P03
Ref Tll Tej nMar Tesp Tret Trej Mini Mfin Estado Dirección-Página
P01 2 7 5 0 7 - - - Finalizado 548057-5480 548057-5480 548057-5480
548057-5480 548057-5480 548057-5480 548057-5480
P02 3 10 5 7 7 10 3 7 En memoria 424666-4246 591789-5917 475005-4750
703990-7039 424666-4246 475005-4750 324430-3244 19268-192 69279-692 637416-6374
P03 4 5 3 5 6 4 0 2 En ejecución 6869-68 441318-4413 258922-2589
6869-68 441318-4413
P04 10 15 3 0 0 15 8 10 En memoria 111-1 222-2 333-3 222-2 111-1
555-5 666-6 777-7 888-8 999-9 888-8 777-7 444-4 555-5 66-0
P05 42 8 6 - - - - - Fuera de sistema 2021-20 200-2 100-1 2022-20 2222-22
200-2 0-0 2323-23
T. MEDIO ESPERA: 3.000 T. MEDIO RETORNO: 5.000
Cola de ejecución: P02 P04
Se ha realizado reubicación
PRE | P03 | P02 | | |
| 0 | 3 | 5 | | 10 |
| P03 | P02 | | | |
POS | 0 | 3 | | | 8 |
Marcos de página
P03 P02 P04
M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 | M=12
68 4413 - - - - - - - - - -
0 0 -
BM | P03 | P02 | P04 | | M=12
| 68 4413 | 3 | 8 | 11 |
| 0 | | | | |
BT | P01 | P03 | | T=10
| 5480 5480 5480 5480 5480 5480 5480 | 68 4413 |
| 0 | 2 | 9 |

```

Llega **P04**, necesita 3 marcos para entrar en memoria, hay 4 marcos libres, pero están separados por lo que se hace REUBICACIÓN. Los marcos de **P02** pasan de [5, 9] a [3, 7].

Tras la reubicación **P04** entra en memoria, ocupando los marcos [8, 10].

Ahora en la cola están **P02** y **P04**, en ese orden porque así han llegado.

T=14

```

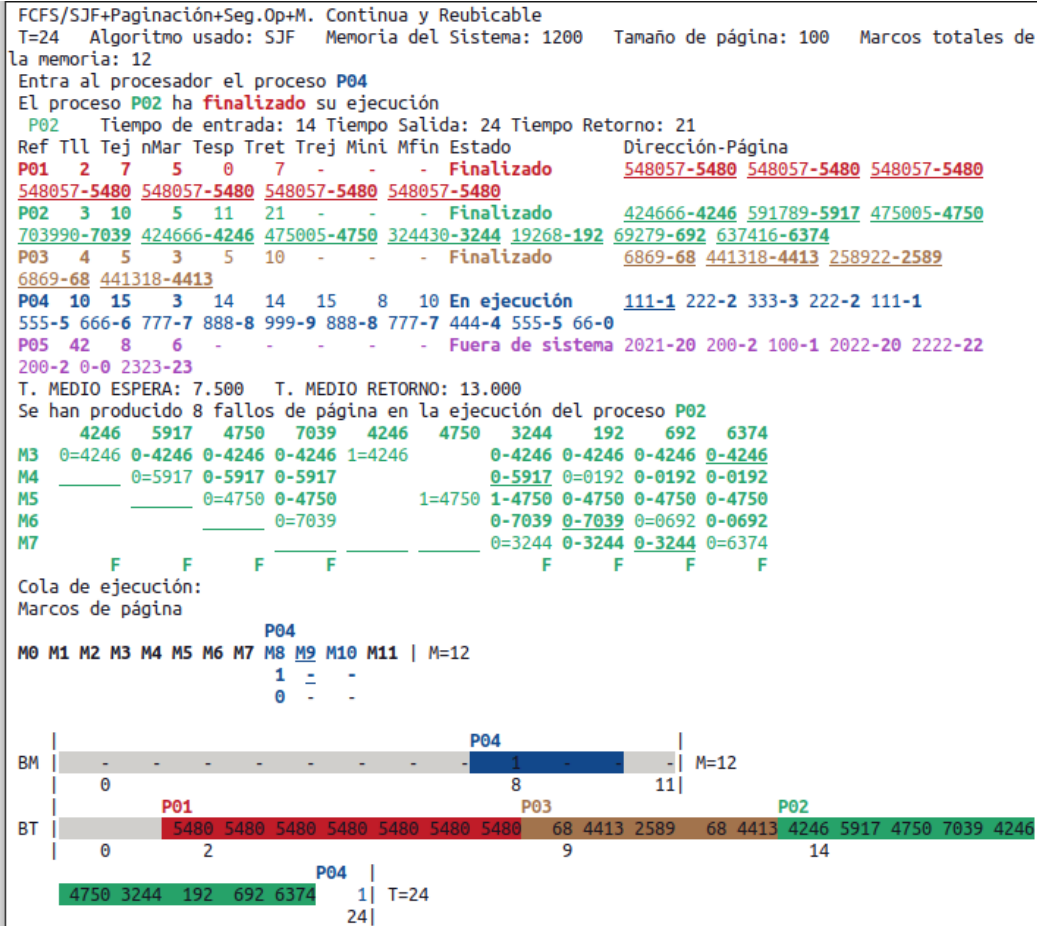
FCFS/SJF+Paginación+Seg.Op+M. Continua y Reubicable
T=14 Algoritmo usado: SJF Memoria del Sistema: 1200 Tamaño de página: 100 Marcos totales de
la memoria: 12
Entra al procesador el proceso P02
El proceso P03 ha finalizado su ejecución
P03 Tiempo de entrada: 9 Tiempo Salida: 14 Tiempo Retorno: 10
Ref Tll Tej nMar Tesp Tret Trej Mini Mfin Estado Dirección-Página
P01 2 7 5 0 7 - - - Finalizado 548057-5480 548057-5480 548057-5480
548057-5480 548057-5480 548057-5480 548057-5480
P02 3 10 5 11 11 10 3 7 En ejecución 424666-4246 591789-5917 475005-4750
703990-7039 424666-4246 475005-4750 324430-3244 19268-192 69279-692 637416-6374
P03 4 5 3 5 10 - - - Finalizado 6869-68 441318-4413 258922-2589
6869-68 441318-4413
P04 10 15 3 4 4 15 8 10 En memoria 111-1 222-2 333-3 222-2 111-1
555-5 666-6 777-7 888-8 999-9 888-8 777-7 444-4 555-5 66-0
P05 42 8 6 - - - - Fuera de sistema 2021-20 200-2 100-1 2022-20 2222-22
200-2 0-0 2323-23
T. MEDIO ESPERA: 5.000 T. MEDIO RETORNO: 8.000
Se han producido 3 fallos de página en la ejecución del proceso P03
68 4413 2589 68 4413
M0 0=68 0=0068 0=0068 1=0068
M1 0=4413 0=4413 1=4413
M2 0=2589
F F F
Cola de ejecución: P04
Marcos de página
P02 P04
M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 | M=12
4246 - - - - - - - - - - -
0 - - - - - - - - - - -
BM | - - - 4246 - - - - - | M=12
| 0 3 8 11 |
BT | - - - P01 P03 P02 |
| 0 2 9 68 4413 2589 68 4413 4246 | T=14
| 0 2 9 68 4413 2589 68 4413 4246 |

```

P03 finaliza su ejecución con un total de 3 fallos de página. Se liberan los marcos que ocupaba.

La CPU ha quedado libre así que entra un nuevo proceso. Como es SJF, entra en ejecución **P02**, que es el que tiene menor tiempo de ejecución entre los procesos que hay en memoria. Se carga en memoria su primera página, produciéndose un fallo de página. El bit del marco 3 se pone a 0. El reloj apunta al marco siguiente.

T=24



P02 finaliza su ejecución con 8 fallos de página y libera los marcos que ocupaba.

La CPU ha quedado libre así que entra **P04**, que es el único que queda en memoria. Se carga en memoria su primera página, produciéndose un fallo de página. El bit del marco 8 se pone a 0. El reloj apunta al marco siguiente.

T=39



P04 finaliza su ejecución con un total de 11 fallos de página y se liberan los marcos que ocupaba.

La memoria queda vacía ya que no hay procesos en espera y la CPU también está vacía, pero queda un proceso por llegar.

En la barra de tiempo se ve que **P04** ha terminado de ejecutar la última página y en el actual instante no ha entrado ninguna página más.

T=42

```

FCFS/SJF+Paginación+Seg.Op+M. Continua y Reubicable
T=42 Algoritmo usado: SJF Memoria del Sistema: 1200 Tamaño de página: 100 Marcos totales de
la memoria: 12
Llega el proceso P05
Entra a memoria el proceso P05
Entra al procesador el proceso P05
Ref Tll Tej nMar Tesp Tret Trej Mini Mfin Estado Dirección-Página
P01 2 7 5 0 7 - - - Finalizado 548057-5480 548057-5480 548057-5480
548057-5480 548057-5480 548057-5480 548057-5480
P02 3 10 5 11 21 - - - Finalizado 424666-4246 591789-5917 475005-4750
703990-7039 424666-4246 475005-4750 324430-3244 19268-192 69279-692 637416-6374
P03 4 5 3 5 10 - - - Finalizado 6869-68 441318-4413 258922-2589
6869-68 441318-4413
P04 10 15 3 14 29 - - - Finalizado 111-1 222-2 333-3 222-2 111-1
555-5 666-6 777-7 888-8 999-9 888-8 777-7 444-4 555-5 66-0
P05 42 8 6 0 0 8 0 5 En ejecución 2021-20 200-2 100-1 2022-20 2222-22
200-2 0-0 2323-23
T. MEDIO ESPERA: 6.000 T. MEDIO RETORNO: 13.400
Cola de ejecución:
Marcos de página
P05
M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 | M=12
20 - - - - -
0 - - - - -

BM | P05
| 20 - - - - - | M=12
| 0 - - - - - |
| 6 - - - - - |

BT | P01 P03 P02
| 5480 5480 5480 5480 5480 5480 5480 68 4413 2589 68 4413 4246 5917 4750 7039 4246
| 0 2 9 14
| 4750 3244 192 692 6374 1 2 3 2 1 5 6 7 8 9 8 7 4
| 24
| P05
| 5 8 20 | T=42
| 39 42 |

```

El tiempo ha avanzado hasta la llegada de **P05**, que entra en memoria, ocupando 6 marcos y entra en la CPU. Se carga en memoria su primera página, produciéndose un fallo de página. El bit del marco 0 se pone a 0. El reloj apunta al marco siguiente.

T=50

```

FCFS/SJF+Paginación+Seg.Op+M. Continua y Reubicable
T=50 Algoritmo usado: SJF Memoria del Sistema: 1200 Tamaño de página: 100 Marcos totales de
la memoria: 12
El proceso P05 ha finalizado su ejecución
P05 Tiempo de entrada: 42 Tiempo Salida: 50 Tiempo Retorno: 8
Ref Tll Tej nMar Tesp Tret Trej Mini Mfin Estado Dirección-Página
P01 2 7 5 0 7 - - - Finalizado 548057-5480 548057-5480 548057-5480
548057-5480 548057-5480 548057-5480 548057-5480
P02 3 10 5 11 21 - - - Finalizado 424666-4246 591789-5917 475005-4750
703990-7039 424666-4246 475005-4750 324430-3244 19268-192 69279-692 637416-6374
P03 4 5 3 5 10 - - - Finalizado 6869-68 441318-4413 258922-2589
6869-68 441318-4413
P04 10 15 3 14 29 - - - Finalizado 111-1 222-2 333-3 222-2 111-1
555-5 666-6 777-7 888-8 999-9 888-8 777-7 444-4 555-5 66-0
P05 42 8 6 0 8 - - - Finalizado 2021-20 200-2 100-1 2022-20 2222-22
200-2 0-0 2323-23
T. MEDIO ESPERA: 6.000 T. MEDIO RETORNO: 15.000
Se han producido 6 fallos de página en la ejecución del proceso P05
20 2 1 20 22 2 0 23
M0 0=20 0=20 0=20 1=20 1=20 1=20 0=20
M1 0=02 0=02 0=02 1=02 1=02 0=02
M2 0=01 0=01 0=01 0=01 0=01
M3 0=22 0=22 0=22
M4 0=00 0=00
M5 0=23
F F F F F F
Cola de ejecución:
Marcos de página
M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 | M=12

BM | - - - - - - - - - - - - - - | M=12
BT | 0 |
P01 | 5480 5480 5480 5480 5480 5480 5480 |
P03 | 68 4413 2589 |
P02 | 68 4413 4246 5917 4750 7039 4246 |
0 | 2 | 9 | 14 |
P04 | 4750 3244 192 692 6374 |
1 | 2 | 3 | 2 | 1 | 5 | 6 | 7 | 8 | 9 | 8 | 7 | 4 |
24 |
P05 | 20 2 1 20 22 2 0 23 |
5 | 8 | 39 | 42 | 50 | T=50

```

Finaliza la ejecución de P05 con un total de 6 fallos de página, se libera la CPU y los marcos que ocupaba en memoria.

Era el último proceso por lo que a continuación se muestra el resumen final.

```

T.ESPERA: Tiempo que el proceso no ha estado ejecutándose en la CPU desde que entra en memoria hasta que sale
ESPERA TOTAL: Tiempo que el proceso no ha estado ejecutándose en la CPU desde T=0 (Inicio del sistema) hasta que sale
INICIO/FIN: Tiempo de llegada al gestor de memoria del proceso y tiempo de salida del proceso
T.RETORNO: Tiempo total de ejecución del proceso, incluyendo tiempos de espera, desde la señal de entrada hasta la salida

RESUMEN FINAL con tiempos de ejecución y fallos de página de cada proceso
Proceso T.Espera Inicio/Fin T.Retorno Fallos de página
P01 0 2/9 7 1
P02 11 14/24 21 8
P03 5 9/14 10 3
P04 14 24/39 29 11
P05 0 42/50 8 6

Tiempo total transcurrido en ejecutar todos los procesos: 50
Media tiempo espera de todos los procesos: 6.000
Media tiempo retorno de todos los procesos: 15.000

Fin de la ejecución.
Pulsa INTRO para continuar.

```

Si comparamos este resultado con la ejecución FCFS, vemos que SJF ha sido más eficiente porque el tiempo medio de espera de los procesos es menor y por tanto también lo es el de retorno.

Modificaciones y mejoras realizadas

Código

- Se han fusionado dos prácticas. La razón principal :
La práctica del reloj funcionaba, pero el script de NFU es muy superior en cuanto a modularidad, documentación interna del código y en general ausencia de código ofuscado, lo cual facilita inmensamente la comprensión y edición del código.
A pesar de esta fusión se ha conseguido que funcionara todo lo que ya funcionaba en el script de Reloj.
- Todo el código está indentado.
- Se han añadido comentarios, algunos pueden parecer excesivos, pero es para que el que esté aprendiendo bash se entere mejor.
- Se han eliminado o sustituido las variables que no servían o no se podía averiguar cuál era su función.
- Se han creado funciones que permiten reutilizar código y que las distintas partes del script sean fácilmente editables o sustituibles.
- Con todo lo mencionado, se han reducido notablemente el número de líneas del script. La práctica de Reloj tenía unas 7000 líneas, ahora obteniendo el mismo funcionamiento y nuevas opciones añadidas tiene casi la mitad de líneas. Además, de estas líneas muchas son comentarios sobre el código.

Menús

- Implementada las opciones de introducción de datos manuales, a través de un fichero y a través del fichero correspondiente a la última ejecución.
- En la introducción a mano ahora se pregunta si se quiere guardar en la versión estándar del fichero (orientado a la última ejecución) o en una nueva versión de fichero (orientado al almacenamiento de diferentes ficheros de datos iniciales).
- Implementado un MODELO ALEATORIO de creación de datos con las opciones de introducción manual de los datos de los rangos, a través de un fichero y a través del fichero correspondiente a la última ejecución. En la introducción a mano también se pregunta si se quiere guardar en la versión estándar del fichero o en una nueva versión de fichero.
- Ya no es necesario tener que calcular la división del número de direcciones entre las direcciones por página y obligar a que fuera exacta ya que se calcula el número de direcciones multiplicando las direcciones por página y el número de marcos de la memoria, dos datos que se pide al usuario.
- Avisa al usuario sobre el desperdicio de memoria en caso de tener más marcos que direcciones y permite su cambio.
- A la hora de elegir el tipo de ejecución si se elige el modo Automático, se pide al usuario que especifique el tiempo de espera entre eventos
- Añadidas las opciones de ejecutar de nuevo el programa o salir de este en lugar de que se termine abruptamente después de ver los informes.

Volcado en pantalla

- En la impresión por pantalla de las direcciones de la tabla de datos inicial y del volcado por eventos se calcula el ancho de la pantalla y se determina hasta dónde puede llegar la impresión en la misma línea.

- Cálculo del tamaño adecuado de las unidades en las barras de tiempo y de memoria ajustándolo al del dato que más ocupe. Siendo en ambas bandas el mismo tamaño para que queden proporcionadas.
- Cálculo del tamaño necesario en cada columna del Resumen de marcos para añadir espacios donde sea necesario para que todos los datos se mantengan alineados
- Añadidos ceros delante del número de página en el resumen de fallos de página cuando en la misma columna hay un valor con más dígitos para los coeficientes y todo quede alineado.
- Se remarca cuándo se produce un fallo de paginación añadiendo una fila al final de la tabla de fallos que lo indica con una F para que quede claro aun cuando ya se marca a quién le toca en el siguiente fallo y cuál es la página nueva.
- Añadidas las tres líneas verticales al final de las barras de memoria y tiempo y tras estas, la información sobre la memoria y el tiempo respectivamente.
- Alineados los recuadros de # de las cabeceras.
- En los menús se limpia la pantalla entre cada uno, pero se mantiene la cabecera.

Ficheros

- Nuevo modo de registrar la salida por pantalla en los ficheros de informes.
- Los datos iniciales se guardan mientras se están introduciendo.
- En el informe BN se sustituyen los colores de las bandas por guiones bajos y las unidades no utilizadas por asteriscos, para diferenciar la parte ocupada de la parte libre en cada partición

Mejoras por realizar:

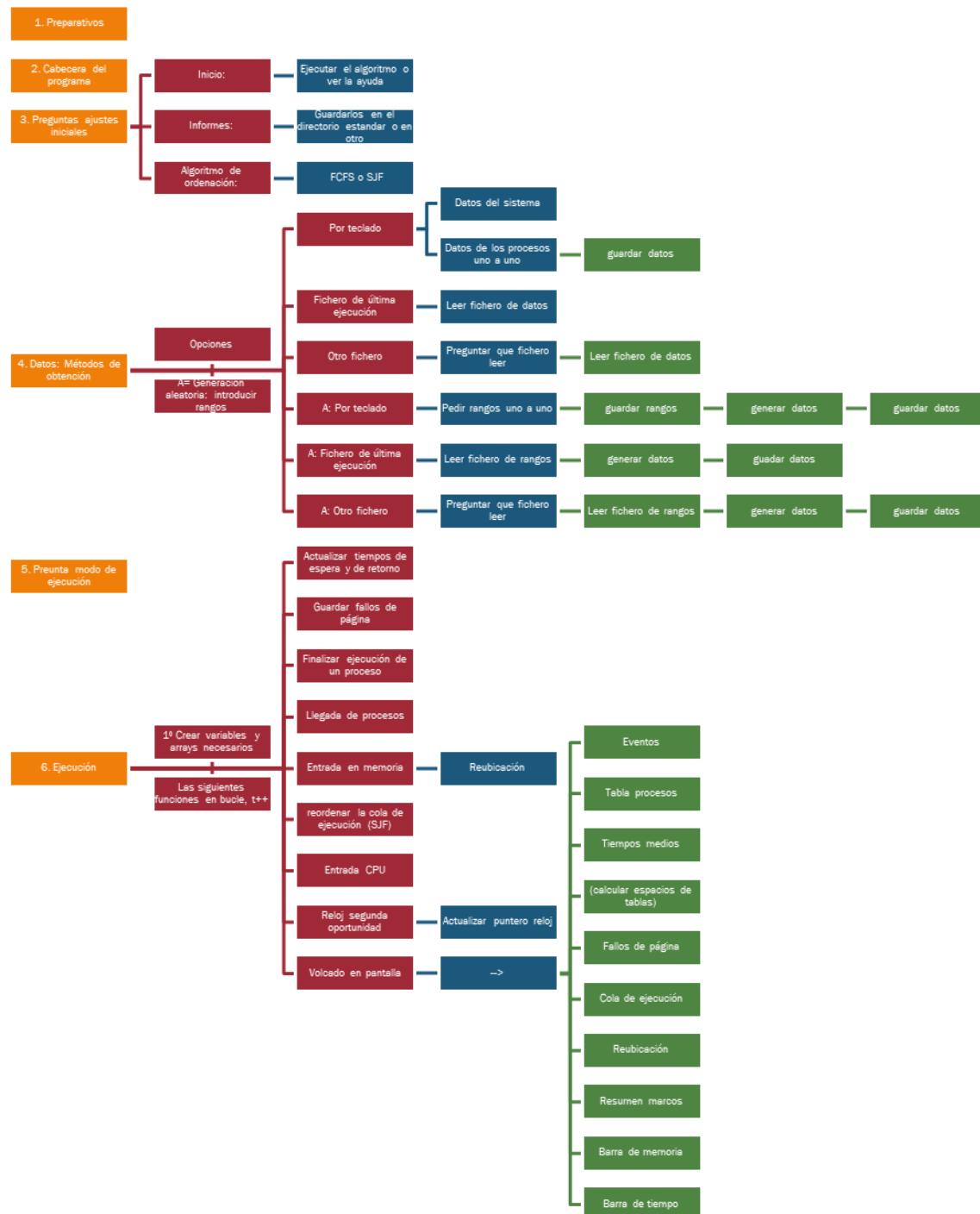
Las mejoras sugeridas a continuación no se han llevado a cabo por motivos de prioridad, tiempo, complejidad.

- Cabeceras y menús del programa en función del ancho de pantalla.
- Resumen final de la ejecución en función del ancho de pantalla.
- Formato uniforme en la interfaz, colores del texto, indentaciones, saltos de línea entre mensajes, avisos, mensajes de error...
- Calcular los espacios que van a ocupar las columnas del resumen de procesos para ajustar su ancho en función de los datos y que nunca se desalinee esa tabla.
- Optimización general del uso de funciones y variables globales y locales.
- Al pedir datos al usuario se muestra un resumen de los datos que se van introduciendo, en el resumen de introducción manual de rangos se podría poner en negrilla o resaltar de algún modo el dato que toca introducir aparte del mensaje pidiendo el dato.
- Validar la correcta lectura de los datos desde los ficheros, si falta algún dato o es inválido lanzar un mensaje de fichero de datos inválido y dar a elegir otro.
- Conseguir que el fichero de ayuda se lea desde el lector del terminal (que se abra desde el principio) y no como si se abriera con cat.
- Añadir la opción de volver atrás o de salir del programa en algunos menús.
- Mejorar la documentación interna del código, la primera parte en especial

Estructura del Script

El script tiene dos partes, la parte de recogida de datos y la de ejecución de los algoritmos. La ejecución está dividida a su vez en la gestión de los procesos y el volcado en pantalla.

Las funciones en general están escritas en orden de ejecución y si dentro de una función se llama a otras, esas estarán debajo. Esto no es como debería pensarse bash, de hecho, debería organizarse justo en el orden opuesto, pero es posible gracias a llamar a una función 'main' al final del script. A continuación, un esquema de las principales funciones del programa.



Conclusiones

La práctica de control me ha parecido una tarea muy útil para familiarizarnos con la programación de scripts en bash y con el entorno de Linux. También para entender mejor el temario de la asignatura referente a algoritmos de gestión de procesos, memoria virtual y algoritmos de reemplazo de páginas en especial los que implementados en este trabajo.

Lo más difícil de la práctica ha sido aprender a trabajar en un código de esa extensión, en mi caso eran unas 7000 líneas, que además estaba hecho por otros alumnos con su propia forma de programar y en algunos casos se nota que con algo de prisa también.

Creo que si la práctica se hiciera un cuatrimestre más tarde esto de trabajar sobre prácticas de otros años iría mucho mejor ya que después de haber cursado programación, leer los scripts es menos difícil y los alumnos ya sabrían un poco más de programación lo cual mejoraría a la vez la calidad del código y su legibilidad para el siguiente año.

Como conclusión, esta práctica no es fácil pero sí que es útil para aprender bash y los contenidos de la asignatura.

Bibliografía

En la realización de la práctica se han consultado principalmente las siguientes fuentes:

- Saiz Díez, José Manuel. Apuntes de Sistemas Operativos. Burgos: Universidad de Burgos, 2022.
- DevHints. Bash Cheatsheet. <https://devhints.io/bash> . Accedido en 2022.
- Atareao. Tutorial de Terminal. <https://atareao.es/tutorial/terminal/> . Accedido en 2022.

Para la realización del script se han tomado como base las siguientes prácticas :

- 002 - Agüero Hernando, Luis Miguel-Diez Busto, Alberto - P - 00282 – FCFS-SJF-Pag-Reloj-C-R-SN
- 016 - Cacuci , Catalin Andrei - P - 01690 – FCFS-SJF-Pag-NFU-NC-R