

Informe

SISTEMAS OPERATIVOS FCFS/SJF-Págination-FIFO-NC-R



UNIVERSIDAD DE BURGOS

Jose Maria Santos Romero

Índice

Índice	2
Explicación de los algoritmos	3
Ejercicio hecho a mano	4
Ejecución y volcado	13
Modificaciones realizadas	21
Modificaciones sugeridas	22

Explicación de los algoritmos

Para realizar la práctica de control he elegido el siguiente algoritmo:

- Gestión de procesos: FCFS o SJF
- Paginación
- Gestión de memoria: FIFO
- Memoria Continua: NO
- Memoria Reubicable: SI

Algoritmo FCFS:

Es un algoritmo de programación en sistemas operativos, que se encarga de ejecutar automáticamente los procesos según su orden de llegada (siempre), es un algoritmo no apropiativo, lo que quiere decir que una vez que un proceso entra en ejecución no saldrá de la CPU hasta que este haya finalizado. El algoritmo FCFS (First Come First Served) es sencillo y fácil de implementar, aunque su rendimiento es relativamente bajo y su tiempo de espera alto.

Algoritmo SJF:

Es un algoritmo de programación en sistemas operativos, que se encarga de ejecutar los procesos según sus respectivos tiempos de ejecución, donde el proceso más corto se ejecutará primero y el más largo el último, es un algoritmo no apropiativo, de igual manera que el FCFS, un proceso no comenzará a ejecutarse hasta que el anterior haya finalizado. El algoritmo SJF (Shortest Job First) se considera como un algoritmo óptimo, porque da un tiempo de espera promedio mínimo; además este algoritmo en caso de que dos tiempos de ejecución sean iguales, tendrá en cuenta el tiempo de llegada de cada proceso.

Algoritmo FIFO:

El algoritmo FIFO (First Input First Output), utilizado para la gestión de memoria, este es el más sencillo de implementar, trata de que la primera página en llegar será la primera en salir. Para su implementación en el script, se utiliza un apuntador que indica en qué marco entra la siguiente página.

Memoria No Continua y Reubicable:

La reubicación consiste en trasladar procesos activos en la memoria, para así maximizar la utilización de la CPU y la propia memoria. Para su implementación se utiliza una variable en la que se almacena el mínimo número de marcos para realizar la reubicación, es decir, en el caso de que exista en memoria el número contiguo de marcos indicado o menor a ese, se produce la reubicación.

La asignación de memoria no continua puede asignar los bloques de memoria (marcos) separados en el espacio de memoria de forma no contigua.

Ejercicio hecho a mano

Datos iniciales del ejemplo:

Número de direcciones: 1000

Tamaño de página: 100

Número de marcos: 10

Mínimo para realizar reubicación: 2

Ref.	T. Lleg	T.Ejec	Nº Marcos	Págs- Dirs.
P01	10	7	5	123-1 423-4 543-5 222-2 124-1 226-2 783-7
P02	2	5	3	875-8 234-2 124-1 423-4 1235-12
P03	4	5	2	123-1 542-5 483-4 234-2 432-4
P04	5	8	6	234-2 764-7 732-7 92-0 123-1 987-9 665-6 234-2

El número de marcos (10) es el tamaño de la memoria o número de direcciones (1000) entre el tamaño de cada página (100).

Podemos observar en los datos iniciales que los procesos no están ordenados por el tiempo de llegada, hacerlo puede facilitarnos el trabajo a la hora de utilizar el algoritmo FCFS.

El tiempo de ejecución coincide con el número de direcciones de cada proceso.

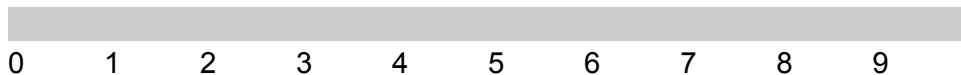
El número de marcos o mínimo estructural es el espacio que va a ocupar dicho proceso en memoria.

T=0

En el tiempo 0 todavía no ha llegado ningún proceso, el primer proceso en llegar es el proceso "P02" en el tiempo 2.

Banda de memoria:

VACÍO



Banda temporal:

VACÍO



T=2

El tiempo de llegada del proceso "P02" es 2, por lo que entra en memoria ya que tiene espacio disponible, ocupando 3 marcos de página (0, 1, 2), la página 8 entra en el marco 0 que está vacío y se produce el primer fallo de página.

Banda de memoria:

P02



Banda temporal:

P02



T=3

En el tiempo 3 el proceso "P02" sigue en ejecución, ocupando 3 marcos de página (0, 1, 2), la página 2 entra en el marco 1 que está vacío y se produce el segundo fallo de página. Vemos en la banda temporal como la página 8 ha sido ejecutada y acaba de entrar la página 2 en el tiempo 3.

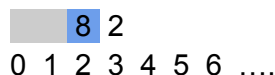
Banda de memoria:

P02



Banda temporal:

P02



T=4

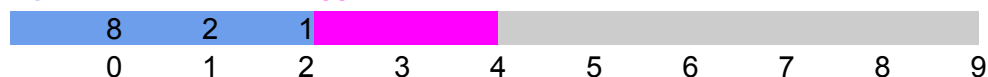
En el tiempo 4 el proceso "P02" sigue en ejecución, ocupando 3 marcos de página (0, 1, 2), la página 1 entra en el marco 2 que está vacío y se produce el tercer fallo de página. Vemos en la banda temporal como la página 2 ha sido ejecutada y acaba de entrar la página 1 en el tiempo 4.

El proceso "P03" entra en memoria porque hay espacio suficiente pero no comienza a ejecutarse porque el proceso "P02" está en ejecución y estamos utilizando un algoritmo no apropiativo

Banda de memoria:

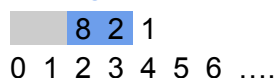
P02

P03



Banda temporal:

P02

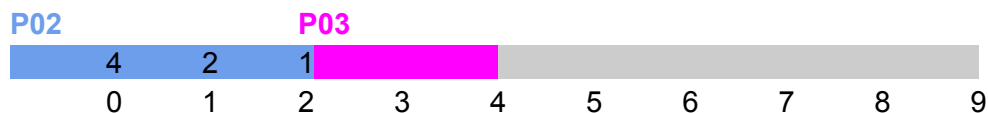


T=5

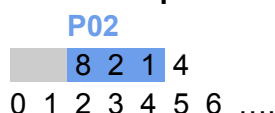
En el tiempo 5 llega el proceso "P04" pero este no entra a memoria porque no hay espacio suficiente.

Entra la página 4 en el marco 0 y sale la página 8 ya que fue la primera en entrar, por lo tanto será la primera en salir.

Banda de memoria:



Banda temporal:



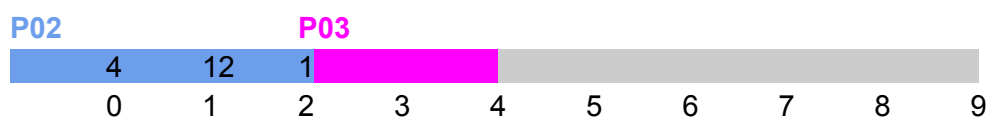
T=6

En el tiempo 6 llega el proceso "P04" pero este no entra a memoria porque no hay espacio suficiente.

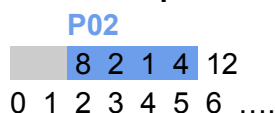
Entra la página 12 en el marco 1 y sale la página 2 ya que fue la segunda en entrar, por lo tanto será la segunda en salir.

En el tiempo del 6 al 7 se ejecutará la página 12.

Banda de memoria:



Banda temporal:



T=7

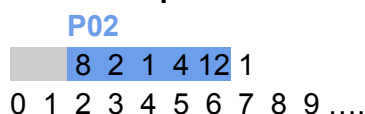
En el tiempo 7 finaliza el proceso "P02" y deja el espacio que estaba ocupando en memoria libre, por lo que el proceso "P04" entra en memoria ocupando 6 marcos de página.

También, al finalizar el proceso "P02", el proceso "P03" comienza a ejecutarse, entrando la página 1 en el marco 3.

Banda de memoria:



Banda temporal:



Resumen de los fallos de página del proceso "P02":

Páginas	8	2	1	4	12
Marco 0	8	8	8	4	4
Marco 1		2	2	2	12
Marco 2			1	1	1
Fallos	X	X	X	X	X
Tiempo	2	3	4	5	6

- *Fallos de Página:* 5 (Número total de fallos de página durante la ejecución)
- *Tiempo de Retorno:* 5 (Tiempo desde que es cargado hasta que finaliza)
- *Tiempo de Espera:* 0 (Tiempo que estuvo en cola)

T=8

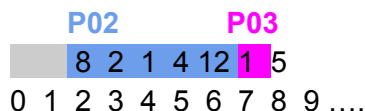
En el tiempo 8 el proceso "P03" sigue en ejecución.

La página 5 entra en el marco 4 ya que este está vacío, mientras tanto el proceso "P04" sigue en memoria pero no comienza a ejecutarse hasta que el proceso "P03" finalice.

Banda de memoria:



Banda temporal:



T=9

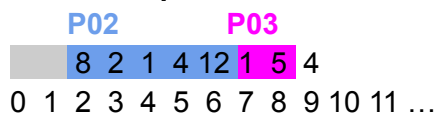
En el tiempo 9 el proceso "P03" sigue en ejecución.

La página 4 entra en el marco 3 y sale la página 1 que fue la primera en entrar, mientras tanto el proceso "P04" sigue en memoria pero no comienza a ejecutarse hasta que el proceso "P03" finalice.

Banda de memoria:



Banda temporal:

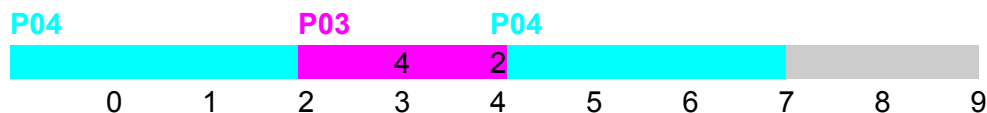


T=10

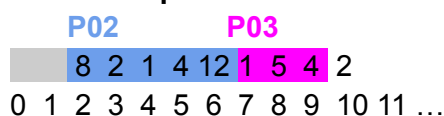
En el tiempo 10 llega el proceso "P01" pero este se queda en espera porque no hay espacio suficiente disponible en memoria, mientras el proceso "P03" sigue en ejecución.

La página 2 entra en el marco 4 y sale la página 5 que fue la segunda en entrar, produciéndose otro fallo de página, mientras tanto el proceso "P04" sigue en memoria pero no comienza a ejecutarse hasta que el proceso "P03" finalice.

Banda de memoria:



Banda temporal:



T=11

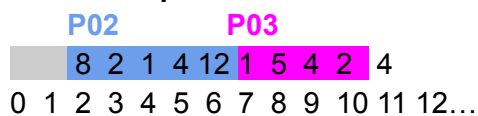
En el tiempo 11 llega el proceso el proceso "P03" sigue en ejecución.

La página 4 ya se encuentra en memoria por lo que no se produce ningún fallo de página, mientras tanto el proceso "P04" sigue en memoria pero no comienza a ejecutarse hasta que el proceso "P03" finalice.

Banda de memoria:



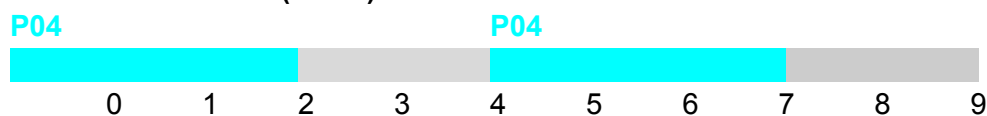
Banda temporal:



T=12

En el tiempo 12 el proceso el proceso "P03" finaliza la ejecución, dejando un espacio de dos marcos disponibles en memoria, quedando la banda de memoria de la siguiente manera:

Banda de memoria (antes):



Como el mínimo número de marcos para que se produzca reubicación es 2, se realiza la reubicación de la siguiente manera:

Banda de memoria (después):



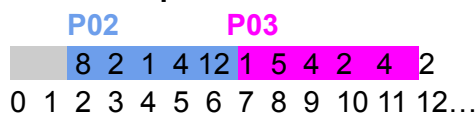
Tras realizarse la reubicación el proceso "P04" comienza a ejecutarse, entrando la página 2 en el primer marco disponible, el marco 0. El proceso "P01" sigue en espera porque no hay espacio disponible en memoria.

Banda de memoria:

P04



Banda temporal:



Resumen de los fallos de página del proceso "P03":

Páginas	1	5	4	2	4
Marco 3	1	1	4	4	4
Marco 4		5	5	2	2
Fallos	X	X	X	X	-
Tiempo	7	8	9	10	11

- *Fallos de Página:* 4 (Número total de fallos de página durante la ejecución)
- *Tiempo de Retorno:* 8 (Tiempo desde que es cargado hasta que finaliza)
- *Tiempo de Espera:* 3 (Tiempo que estuvo en cola)

T=13

En el tiempo 13 el proceso "P04" sigue en ejecución.

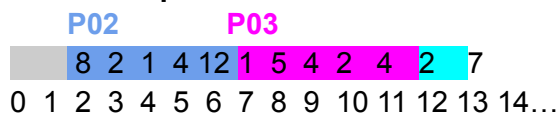
La página 7 entra en el marco 1 que está disponible, produciéndose también otro fallo de página, mientras tanto el proceso "P01" sigue en espera hasta que haya espacio disponible en memoria..

Banda de memoria:

P04



Banda temporal:



El proceso "P04" ocupa la memoria hasta el tiempo 20, produciendo una serie de fallos de página.

En el tiempo 20 el proceso “P04” finaliza la ejecución, dejando toda la memoria disponible, por lo que entra a memoria el proceso “P01” ocupando 5 marcos de página (0,1,2,3,4) y comienza a ejecutarse.

Banda de memoria:

A number line from 0 to 9. A yellow bar is placed above the line, starting at 0 and ending at 4. The number 1 is written inside the yellow bar.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
P02			8	2	1	4	12	P03	1	5	4	2	4	P04	2	7	7	0	1	9	6	2	1

<i>Páginas</i>	2	7	7	0	1	9	6	2
<i>Marco 0</i>	2	2	2	2	2	2	2	2
<i>Marco 1</i>		7	7	7	7	7	7	7
<i>Marco 2</i>				0	0	0	0	0
<i>Marco 3</i>					1	1	1	1
<i>Marco 4</i>						9	9	9
<i>Marco 5</i>							6	6
<i>Fallos</i>	X	X	-	X	X	X	X	-
<i>Tiempo</i>	12	13	14	15	16	17	18	19

- *Fallos de Página:* 6 (Número total de fallos de página durante la ejecución)
- *Tiempo de Retorno:* 15 (Tiempo desde que es cargado hasta que finaliza)
- *Tiempo de Espera:* 7 (Tiempo que estuvo en cola)

T=21

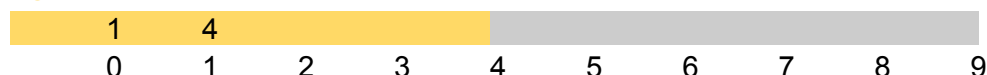
En el tiempo 21 el proceso "P01" sigue en ejecución.

La página 4 entra en el marco 1 que está disponible, produciéndose también otro fallo de página.

En cola no se encuentra ningún proceso más, ya que el proceso "P01" es el último en llegar, por lo que el último en ejecutarse, tal y como indica el algoritmo FCFS (First Come First Served).

Banda de memoria:

P01



Banda temporal:



T=22

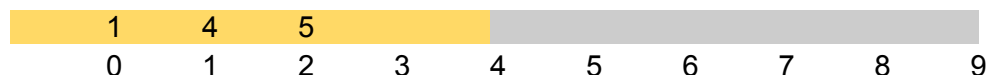
En el tiempo 22 el proceso "P01" sigue en ejecución.

La página 5 entra en el marco 2 que está disponible, produciéndose el tercer fallo de página durante la ejecución del proceso "P01".

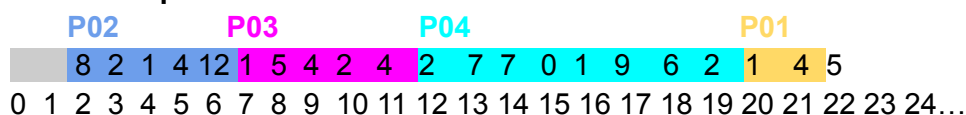
En cola no se encuentra ningún proceso más, ya que el proceso "P01" es el último en llegar, por lo que el último en ejecutarse, tal y como indica el algoritmo FCFS (First Come First Served).

Banda de memoria:

P01



Banda temporal:



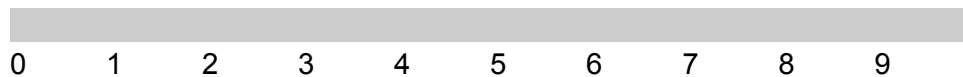
El proceso "P01" ocupa la memoria hasta el tiempo 27, produciendo una serie de fallos de página.

T=27

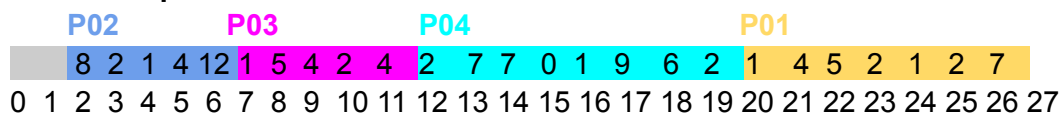
En el tiempo 27 el proceso "P01" finaliza la ejecución, dejando toda la memoria disponible y concluyendo así la ejecución de todos los procesos existentes.

Banda de memoria:

VACÍO



Banda temporal:



Resumen de los fallos de página del proceso "P01":

Páginas	1	4	5	2	1	2	7
Marco 0	1	1	1	1	1	1	1
Marco 1		4	4	4	4	4	4
Marco 2			5	5	5	5	5
Marco 3				2	2	2	2
Marco 4							7
Fallos	X	X	X	X	-	-	X
Tiempo	20	21	22	23	24	25	26

- *Fallos de Página:* 5 (Número total de fallos de página durante la ejecución)
- *Tiempo de Retorno:* 17 (Tiempo desde que es cargado hasta que finaliza)
- *Tiempo de Espera:* 10 (Tiempo que estuvo en cola)

FINALIZA EL PROGRAMA

Todos los procesos existentes han sido ejecutados y finalizados, podemos calcular el tiempo medio de espera y de retorno del programa para así tener una evaluación del rendimiento del algoritmo utilizado.

Tiempo medio de Retorno: $(5 + 8 + 15 + 17) / 4 = 11'25$

Tiempo medio de Espera: $(0 + 3 + 7 + 10) / 4 = 5$

Ejecución y volcado

La ejecución del script está desarrollada en video con una explicación detallada de cada procedimiento y tiempo de la ejecución.

1º Nos muestra varias cabeceras con información acerca del algoritmo utilizado.

```
Algoritmo de procesos : FCFS/SJF
Tipo de algoritmo      : PAGINACIÓN
Algoritmo de memoria  : FIFO
Memoria continua      : NO
Memoria reubicable    : SÍ

Autor: Santos Romero, Jose Maria

Autores anteriores:
RR-Pag-NRU-C-FI: Diego García Muñoz
PriMayor-SN-NC-R: Iván Cortés
R-R-Pag-Reloj-C-FI: Ismael Franco He
FCFS-SJF-Pag-NFU-NC-R: Cacuci Catali

Asignatura: Sistemas Operativos
Profesor: Jose Manuel Saiz Diez

Este script se creó usando la versión
5.0.17 de Bash si no se ejecuta con esta
versión pueden surgir problemas.

© Creative Commons
BY - Atribución (BY)
NC - No uso Comercial (NC)
SA - Compartir Igual (SA)

Pulsa ENTER para continuar.
```

AVISO

Para visualizar correctamente la información es necesario poner la ventana del terminal en pantalla completa. Si no, hay elementos que no se van a ver correctamente.

También es recomendable tener la terminal con un tema oscuro.

Pulsa ENTER para continuar.

2º Nos pregunta si queremos modificar el nombre predeterminado de los informes (blanco y negro o en color) que se generarán tras finalizar la ejecución del script.

```
FCFS/SJF - Pag - FIFO - NC - R
Selección de informes

Los informes por defecto son informeBW.txt y informeCOLOR.txt.
¿Quieres cambiarlos?

[1] <- Sí
[2] <- No

Selección: 2|
```

3º Seleccionamos la opción que queremos ejecutar, en este caso, ejecutar el programa..

```
FCFS/SJF - Pag - FIFO - NC - R
Menu

¿Qué quieres hacer?

[1] <- Ejecutar el programa
[2] <- Ver la ayuda

Selección: |
```

4º Seleccionamos el tipo de algoritmo que queremos utilizar para la ejecución, en este caso FCFS, que obtiene el mismo resultado que SJF en este ejemplo.

```
FCFS/SJF - Pag - FIFO - NC - R
Selección de algoritmo

¿Qué algoritmo quieres usar?

[1] <- FCFS
[2] <- SJF

Selección: 1|
```

5º Introducimos los datos del ejercicio, podemos introducir los datos seleccionando un fichero con los datos guardados, introduciendolos manualmente o de manera aleatoria. En este caso utilizaremos la opción 2, por el fichero de datos de última ejecución.

```
FCFS - Pag - FIFO - NC - R
Método de introducción de datos

¿Cómo quieres introducir los datos?

[1] <- Por teclado
[2] <- Por fichero de datos de última ejecución (datos.txt)
[3] <- Por otro fichero de datos
[4] <- Aleatoriamente Manual
[5] <- Por fichero de rangos de última ejecución (datosrangos.txt)
[6] <- Por otro fichero de rangos

Selección: 2|
```

6º Muestra un breve resumen de los datos que se van a utilizar durante la ejecución.
El script muestra los procesos ordenados por orden de llegada, al haber seleccionado el algoritmo FCFS.

```
Tamaño memoria : 1000
Tamaño página : 100
Número marcos : 10
mNUR : 2

Pulsa ENTER para continuar.
```

```
Ref TlL Tej nMar Dirección - Página
P02 2 5 3 875-8 234-2 124-1 423-4 1235-12
P03 4 5 2 123-1 542-5 483-4 234-2 432-4
P04 5 8 6 234-2 764-7 732-7 92-0 123-1 987-9 665-6 234-2
P01 10 7 5 123-1 423-4 543-5 222-2 124-1 226-2 783-7

Pulsa ENTER para continuar.
```

7º Muestra una pantalla donde indicaremos cómo queremos ejecutar el algoritmo, en este caso mostrando los eventos interesantes.

```
FCFS - Pag - FIFO - NC - R
Método de ejecución

¿Cómo quieres ejecutar el algoritmo?

[1] <- Mostrar los eventos interesantes
[2] <- Ejecución automática
[3] <- Ejecución completa
[4] <- Mostrar solo el resumen final

Selección: 1|
```

8º Comienza la ejecución de todos los procesos, mostrará por pantalla los eventos interesantes que ocurren durante la ejecución.

```
FCFS-Paginación-FIFO-NC-R
T: 0      N° Dirs: 1000    Tam Pág: 100    N° Marc: 10    mNUR: 2
Ref Tll Tej nMar Tesp Tret Trej Mini Mfin Estado Dirección - Página
P02 2 5 3 - - - - - Fuera de sist. 875-8 234-2 124-1 423-4 1235-12
P03 4 5 2 - - - - - Fuera de sist. 123-1 542-5 483-4 234-2 432-4
P04 5 8 6 - - - - - Fuera de sist. 234-2 764-7 732-7 92-0 123-1 987-9 665-6 234-2

P01 10 7 5 - - - - - Fuera de sist. 123-1 423-4 543-5 222-2 124-1 226-2 783-7
TespM: -      TretM: -

Proceso: [NADA]
N° Marco: [ 0][ 1][ 2][ 3][ 4][ 5][ 6][ 7][ 8][ 9]
Página: [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
Apuntador: [ -][ -][ -][ -][ -][ -][ -][ -][ -][ -]

BM: |-----| M:10
    | 0 |
BT: | | T:0
    | 0 |
Pulsa ENTER para continuar.
```

En el tiempo T:2 entra a memoria el proceso "P02"

```
FCFS-Paginación-FIFO-NC-R
T: 2      N° Dirs: 1000    Tam Pág: 100    N° Marc: 10    mNUR: 2
Ha llegado el proceso P02.
El proceso P02 ha entrado a memoria a partir de la posición 0.
El proceso P02 ha iniciado su ejecución.
Ref Tll Tej nMar Tesp Tret Trej Mini Mfin Estado Dirección - Página
P02 2 5 3 0 0 5 0 2 En ejecución 875-8 234-2 124-1 423-4 1235-12
P03 4 5 2 - - - - - Fuera de sist. 123-1 542-5 483-4 234-2 432-4
P04 5 8 6 - - - - - Fuera de sist. 234-2 764-7 732-7 92-0 123-1 987-9 665-6 234-2
P01 10 7 5 - - - - - Fuera de sist. 123-1 423-4 543-5 222-2 124-1 226-2 783-7
TespM: 0      TretM: 0

Proceso: [P02]
N° Marco: [ 0][ 1][ 2][ 3][ 4][ 5][ 6][ 7][ 8][ 9]
Página: [ 8][ ][ ][ ][ ][ ][ ][ ][ ][ ]
Apuntador: [ -][ 1][ -][ -][ -][ -][ -][ -][ -][ -]

BM: |P02 8 - - - - -| M:10
    | 0 | 3 |
BT: |P02 8| T:2
    | 0 | 2 |
Pulsa ENTER para continuar.
```


En el tiempo T:4 entra en memoria el proceso "P03" pero no comienza a ejecutarse porque el algoritmo FCFS es no apropiativo.

```
FCFS-Paginación-FIFO-NC-R
T: 4      Nº Dirs: 1000    Tam Pág: 100    Nº Marc: 10    mNUR: 2
Ha llegado el proceso P03.
El proceso P03 ha entrado a memoria a partir de la posición 3.
Cola(Orden ejecución): P03
Ref Tll Tej nMar Tesp Tret Trej Mini Mfin Estado Dirección - Página
P02 2 5 3 0 2 3 0 2 En ejecución 875-8 234-2 124-1 423-4 1235-12
P03 4 5 2 0 0 5 - - En memoria 123-1 542-5 483-4 234-2 432-4
P04 5 8 6 - - - - - Fuera de sist. 234-2 764-7 732-7 92-0 123-1 987-9 665-6 234-2
P01 10 7 5 - - - - - Fuera de sist. 123-1 423-4 543-5 222-2 124-1 226-2 783-7
TespM: 0      TretM: 1.00

Proceso: [P02] [P03] [NADA]
Nº Marco: ( 0) ( 1) ( 2) ( 3) ( 4) ( 5) ( 6) ( 7) ( 8) ( 9)
Página: ( 8) ( 2) ( 1) ( ) ( ) ( ) ( ) ( ) ( ) ( )
Apuntador: ( 1) ( -) ( -) ( -) ( -) ( -) ( -) ( -) ( -) ( -)

BM: | P02 P03 |
    | 8 2 1 - - | M:10
    | 0 3 5 |
BT: | P02 |
    | 8 2 1 | T:4
    | 0 2 |
Pulsa ENTER para continuar.
```

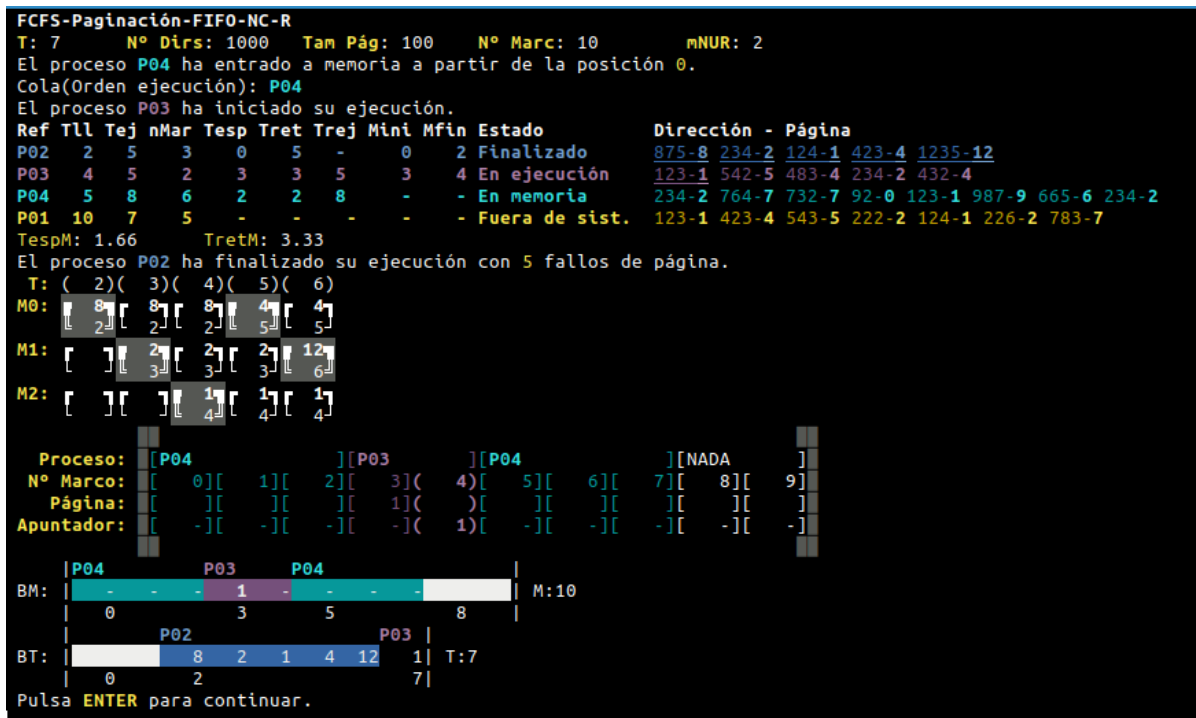
En el tiempo T:5 llega el proceso "P04" pero no entra en memoria porque no hay espacio suficiente en memoria (marcos disponibles).

```
FCFS-Paginación-FIFO-NC-R
T: 5      Nº Dirs: 1000    Tam Pág: 100    Nº Marc: 10    mNUR: 2
Ha llegado el proceso P04.
Cola(Orden ejecución): P03
Ref Tll Tej nMar Tesp Tret Trej Mini Mfin Estado Dirección - Página
P02 2 5 3 0 3 2 0 2 En ejecución 875-8 234-2 124-1 423-4 1235-12
P03 4 5 2 1 1 5 - - En memoria 123-1 542-5 483-4 234-2 432-4
P04 5 8 6 0 0 - - En espera 234-2 764-7 732-7 92-0 123-1 987-9 665-6 234-2
P01 10 7 5 - - - - - Fuera de sist. 123-1 423-4 543-5 222-2 124-1 226-2 783-7
TespM: .33    TretM: 1.33

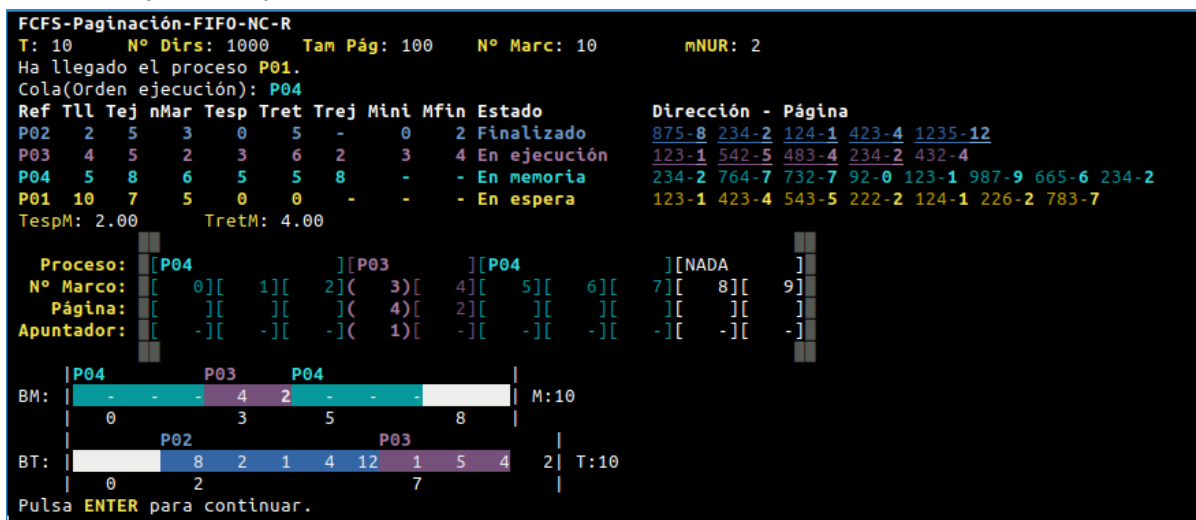
Proceso: [P02] [P03] [NADA]
Nº Marco: ( 0) ( 1) ( 2) ( 3) ( 4) ( 5) ( 6) ( 7) ( 8) ( 9)
Página: ( 4) ( 2) ( 1) ( ) ( ) ( ) ( ) ( ) ( ) ( )
Apuntador: ( -) ( 1) ( -) ( -) ( -) ( -) ( -) ( -) ( -) ( -)

BM: | P02 P03 |
    | 4 2 1 - - | M:10
    | 0 3 5 |
BT: | P02 |
    | 8 2 1 | T:5
    | 0 2 |
Pulsa ENTER para continuar.
```

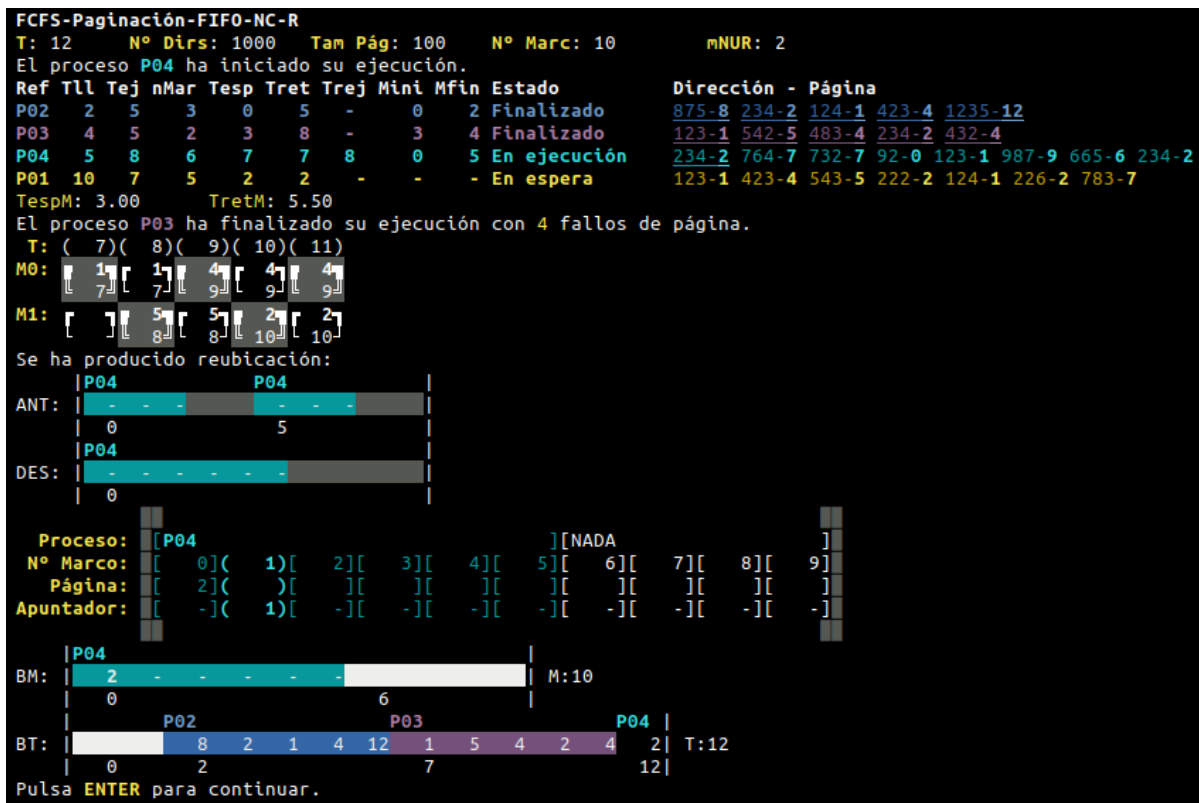
En el tiempo T:7 entra el proceso “P04” a memoria porque el proceso “P02” ha finalizado y deja espacio disponible para que este entre, como la memoria es no continua se pueden asignar los marcos con esta distribución



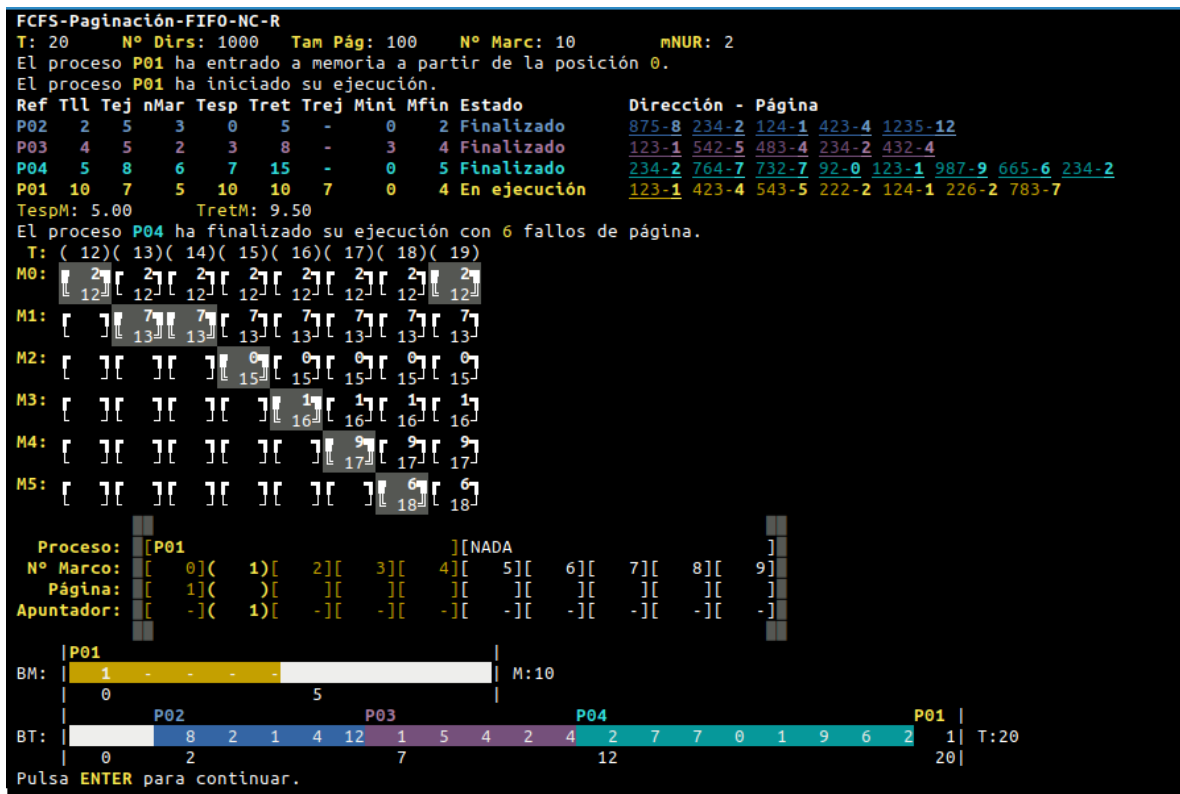
En el tiempo T:10 el proceso “P03” sigue en ejecución y el proceso “P01” llega, pero no entra en memoria porque no hay espacio suficiente en memoria, el mínimo estructural de “P01” es 5 y solo hay 2 marcos disponibles..



En el tiempo T:12 el proceso "P03" finaliza y se produce reubicación porque quedan dos marcos vacíos y nuestro mínimo para la reubicación es 2, el proceso "P04" comienza a ejecutarse.



En el tiempo T:20 el proceso "P04" finaliza y deja la memoria disponible para que el proceso "P01" comience a ejecutarse.



```

FCFS-Paginación-FIFO-NC-R
T: 27  Nº Dirs: 1000  Tam Pág: 100  Nº Marc: 10  mNUR: 2
Ref Tll  NeJ nMar Tesp Tret Trej Mini Mfin Estado  Dirección - Página
P02  2  5  3  0  5  -  0  2 Finalizado  875-8 234-2 124-1 423-4 1235-12
P03  4  5  2  3  8  -  3  4 Finalizado  123-1 542-5 483-4 234-2 432-4
P04  5  8  6  7  15  -  0  5 Finalizado  234-2 764-7 732-7 92-0 123-1 987-9 665-6 234-2
P01  10  7  5  10  17  -  0  4 Finalizado  123-1 423-4 543-5 222-2 124-1 226-2 783-7

TespM: 5.00  TretM: 11.25
El proceso P01 ha finalizado su ejecución con 5 fallos de página.
T: ( 20)( 21)( 22)( 23)( 24)( 25)( 26)
M0: [ 1 ][ 1 ][ 1 ][ 1 ][ 1 ][ 1 ][ 1 ][ 1 ]
    [ 20 ][ 20 ][ 20 ][ 20 ][ 20 ][ 20 ][ 20 ][ 20 ]
M1: [  ][ 4 ][ 4 ][ 4 ][ 4 ][ 4 ][ 4 ][ 4 ]
    [  ][ 21 ][ 21 ][ 21 ][ 21 ][ 21 ][ 21 ][ 21 ]
M2: [  ][  ][ 5 ][ 5 ][ 5 ][ 5 ][ 5 ][ 5 ]
    [  ][  ][ 22 ][ 22 ][ 22 ][ 22 ][ 22 ][ 22 ]
M3: [  ][  ][  ][ 2 ][ 2 ][ 2 ][ 2 ][ 2 ]
    [  ][  ][  ][ 23 ][ 23 ][ 23 ][ 23 ][ 23 ]
M4: [  ][  ][  ][  ][  ][  ][ 7 ]
    [  ][  ][  ][  ][  ][  ][ 26 ]

Proceso: [ NADA ]
Nº Marco: [  ][ 0 ][ 1 ][ 2 ][ 3 ][ 4 ][ 5 ][ 6 ][ 7 ][ 8 ][ 9 ]
Página: [  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ]
Apuntador: [ - ][ - ][ - ][ - ][ - ][ - ][ - ][ - ][ - ][ - ][ - ]

BM: [ 0 ] M:10
BT: [ 0 2 1 4 12 1 5 4 2 4 2 7 7 0 1 9 6 2 1 4 5 2 1 2 7 ] T:27
Pulsa ENTER para continuar.

```

```
FCFS - Pag - FIFO - NC - R
Resumen Final
```

RefTll	Tej	Tesp	Tret	Ini	Fin	Fallos	
P02	2	5	0	5	2	7	5
P03	4	5	3	8	7	12	4
P04	5	8	7	15	12	20	6
P01	10	7	10	17	20	27	5

Tiempo de espera medio: 5.00
Tiempo de retorno medio: 11.25

Pulsa ENTER para continuar.

Modificaciones realizadas

- Eliminación del contador de usos de una página llamado “memoriaNFU” que es utilizado exclusivamente para el algoritmo de gestión NFU (Not Frequently Used), ya que no es necesario para utilizar el algoritmo FIFO.
- Modificación del nombre de los ficheros utilizados para los informes, son: informeBW.txt e informeCOLOR.txt.
- Dar solución a la salida inesperada del script tras visualizar la ayuda, volviendo a utilizar la función “opciones_menu”.
- Creación de nuevas funciones para la introducción de datos, estas funciones son: datos_random, datos_archivo_ultima_ejecucion, datos_archivo_ultima_ejecucion_random y datos_archivo_rangos.
- Guardar en un nuevo directorio llamado “rangos” los ficheros que contengan los rangos utilizados, además de almacenar los datos asignados en un fichero de datos de última ejecución y los rangos de última ejecución (datos.txt y datosrangos.txt).
- Modificación de la introducción de datos de manera aleatoria, añadiendo que los valores de la memoria también se asignen aleatoriamente, como el número de marcos de página o el tamaño de marco de página.
- Mostrar los valores asignados aleatoriamente a la vez que se van introduciendo los rangos e incorporar todo a ambos informes.
- Eliminar la asignación de colores claros a los procesos, modificando la variable global “init_colores”.
- Modificar la función “ej” añadiendo una herramienta case que permite ejecutar el algoritmo de manera automática, estableciendo un tiempo entre pantallas o una ejecución completa.
- Modificación de las funciones “ej_ejecutar_memoria_pagina” y “ej_calcular_marco_siguiete”.
- Crear un diccionario con la clave \$proceso, en el que se almacena el marco inicial y el marco final de cada proceso, para ello se han creado varias funciones nuevas.
- Mostrar los marcos iniciales y finales en la tabla durante la ejecución.
- Creación de la variable “memoriaFIFO” que se encarga de almacenar el tiempo de entrada de cada página en su respectivo marco y así mostrarlo en el resumen de fallos de página.
- Corrección al subrayar el uso de una página, ya que no es necesario para el algoritmo FIFO, dejando indicado los fallos de página de cada proceso.
- Utilización de variables globales para facilitar el manejo del código y posterior modificación si fuera necesaria.
- Añadir el apuntador que señala al siguiente marco.
- Reajustar el ancho de las páginas a uno más grande, añadiendo un espacio a la izquierda a la hora de mostrar las páginas en la banda de memoria y de tiempo.
- Añadir en las bandas de memoria y de tiempo los marcos totales que tiene la memoria y el tiempo actual.
- Se han comentado las nuevas modificaciones realizadas y eliminación de comentarios o líneas redundantes.

Modificaciones sugeridas

- Modificar el resumen de la tabla de fallos, indicando cada fallo en color rojo.
- En la tabla de resumen de fallos indicar cual es el siguiente marco.
- Una nueva opción que cree datos aleatoriamente sin tener que introducir ningún rango.
- Mantener los márgenes una vez la banda de tiempo sobrepase el ancho de la pantalla, de igual manera que la banda de memoria.
- Mostrar también los marcos iniciales y finales de cada proceso en el resumen final.
- Cambiar los colores oscuros al cambiar del tema oscuro al tema claro de la terminal.