

PRÁCTICA DE CONTROL

FCFS/SJF-Paginación-Reloj-C-NR



Alumno: Amanda Pérez Olmos

apo1004@alu.ubu.es

Profesor: José Manuel Sáiz
Diez

Sistemas Operativos

Ingeniería Informática

Curso 2022/2023

Índice de contenidos

1. INTRODUCCIÓN	3
2. EXPLICACIÓN DEL ALGORITMO	3
3. CONTENIDO APORTADO EN LA ENTREGA	4
3.1. Explicación contenido directorio FRangosAleTotal	4
4. EJERCICIOS RESUELTOS.....	5
4.1. Ejercicio sencillo	5
4.1.1. Ejemplo a mano.....	6
4.1.2. Ejecución en el script.....	9
4.2. Ejercicio complejo	12
4.2.1. Ejemplo a mano.....	12
4.2.2. Ejecución en el script.....	20
5. MEJORAS REALIZADAS.....	24
5.1. Ejecución	24
5.2. Eventos	25
5.3. Volcados a pantalla/informes.....	25
5.4. Ficheros	26
5.5. Código.....	26
6. MEJORAS POR REALIZAR	27
7. CONCLUSIÓN	27
8. REFERENCIAS	28

1. INTRODUCCIÓN

La práctica entregable consiste en la realización de un script en Bash que permitiera resolver diferentes algoritmos para la gestión de procesos y/o memoria. Este documento recoge los conceptos generales sobre los que se apoya el algoritmo **FCFS/SJF-Pag-Reloj-C-NR**. Se llevarán a cabo ejercicios prácticos tanto de manera manual como utilizando el script, con el fin de observar y comprender el impacto de los distintos escenarios en la ejecución de los procesos. Dichos ejercicios contarán con descripciones detalladas de lo que está sucediendo y análisis de los resultados obtenidos. Finalmente, se indicarán las mejoras realizadas en el código utilizado, así como las mejoras planteadas para optimizar aún más la simulación en futuras prácticas.

2. EXPLICACIÓN DEL ALGORITMO

Como se ha indicado previamente, el algoritmo utilizado para la realización de esta práctica es FCFS/SJF, con memoria virtual paginada. Utiliza Reloj como algoritmo de reemplazo de páginas, memoria continua y no reubicable. A continuación se definirán más a fondo estos conceptos.

FCFS: se trata del algoritmo de planificación de procesos más sencillo. Sus siglas en inglés *"First Come, First Served"* (Primero en llegar, Primero en ser servido), indican que, de los procesos que se encuentran cargados en memoria, el primero que llegó será el primero que entre a la CPU para ser ejecutado. Hasta que no finaliza su ejecución, no deja la CPU libre, por lo que no pueden ejecutarse más procesos. Es por lo tanto un algoritmo no apropiativo. Se trata de un algoritmo simple y predecible, con la desventaja de que los procesos largos hacen esperar a los cortos.

SJF: en inglés *"Shortest Job First"* (El trabajo más corto primero), se trata de otro algoritmo de planificación de procesos. De los trabajos que se encuentren en la cola de entrada, el primer proceso que pasará a la CPU será el que menor tiempo de ejecución requiera (en caso de empate, se aplica el algoritmo FIFO: *First In, First Out*). Esto indica que se deben conocer los tiempos de CPU de antemano. Es también no apropiativo, puesto que una vez que un proceso entra a la CPU, no sale hasta ejecutarse completamente. Los trabajos largos pueden resultar penalizados.

Paginación: se trata de una estrategia de organización de la memoria física que consiste en dividir la memoria principal en trozos de igual tamaño conocidos como marcos de página. Los procesos, a su vez, se dividen en trozos de tamaño igual que los de la memoria, denominados páginas. Es útil puesto que no es necesario que todo el proceso se encuentre cargado por completo en memoria para su ejecución. Esto permite alojar procesos más grandes que la memoria principal. Habrá una tabla de páginas por cada proceso, que indique qué página está albergada en cada marco de memoria en cada instante, o si está libre. Si se quiere cargar una página que no se encuentra en memoria principal, se produce un fallo de página.

Reloj: es un algoritmo de reemplazo de páginas. Cuando ocurre un fallo de página y ya no hay espacios, el SO tiene que elegir una página para reemplazar. El algoritmo de reloj mantiene las páginas en una lista circular y una manecilla apunta a la página más antigua, que es la que será reemplazada. En la práctica actúa como un FIFO pero difiere en la implementación.

Memoria continua: Como su nombre sugiere, la memoria continua indica que un proceso solo puede meterse en un único hueco de memoria. No puede dividirse. Si el siguiente proceso en la cola de entrada a memoria no cabe por completo en ninguno de los huecos libres, se bloquea a él mismo y al resto de procesos que van detrás, puesto que la cola de entrada a memoria es una cola FIFO y hasta que no entre el primero el resto no podrán.

Memoria no reubicable: La memoria no reubicable implica que los espacios en los que se introduzca un proceso serán siempre fijos. Y es que, en algunas situaciones, es posible que los espacios de la MP que se encuentran asignados a procesos se encuentren muy dispersos y podría darse el caso de que, reorganizándolos, cupieran nuevos procesos en memoria que de otra forma no hubiesen podido entrar. Sin embargo, que la memoria sea no reubicable impide que esto mismo pueda pasar, por lo que la única evaluación que se llevará a cabo cuando un proceso sea el siguiente en la cola de memoria será si cabe en alguno de los huecos libres o no, ignorando si pudiera partirse (puesto que es continua) o si se pudieran reubicar los procesos para hacerle hueco.

3. CONTENIDO APORTADO EN LA ENTREGA

La entrega de esta práctica consta de varias partes, entre las que se incluye este informe, el script, el sistema de directorios y ficheros utilizados y dos vídeos explicativos sobre la ejecución del algoritmo y aclaraciones acerca de la organización del código.

Adicionalmente, dentro de los directorios, se incluyen varios ficheros (sobre todo de datos) aparte de los ficheros por defecto, que permiten probar las diferentes opciones y modos de ejecución del algoritmo. A continuación se explicará con más detalle el contenido del directorio FRangosAleTotal, puesto que incluye varios casos de prueba para comprobar su funcionamiento.

3.1. Explicación contenido directorio FRangosAleTotal

Antes que nada, he de mencionar que la opción de 'Aleatorio Total' difiere un poco de lo que se pedía debido a que lo entendí mal en un principio. En mi código, se calculan los mínimos en un primer instante y, si salen por debajo del valor mínimo real (por ejemplo, menos de un marco de memoria principal), se vuelven a recalcular, pero esta vez entre el valor mínimo real y el máximo (siguiendo el ejemplo anterior, se recalcularía el mínimo entre 1 y el valor máximo del fichero). Resulta que, con corregirlos al valor mínimo directamente, ya bastaba y no hacía falta recalcularlos aleatoriamente. Esto fue hablado en su momento con el profesor y me dijo que lo dejase como estaba, que simplemente notificase entre qué valores se iban a recalcular los mínimos y listo.

He mencionado esto puesto que varios de los ficheros que voy a describir tienen en cuenta casos que solo tendrían lugar de la forma que yo lo planteé y no de la otra. En cualquier caso, si se quisiera modificar para que directamente asigne los valores mínimos reales en vez de recalcularlos, es completamente viable y muy sencillo.

Cabe mencionar que el programa solo lee los datos alojados en DatosRangosAleatorioTotal.txt, es decir, si se quisieran probar diferentes situaciones, habría que copiar y pegar su contenido en

este fichero. Lo marcado con ‘->’ hace referencia a los resultados esperados al ejecutar cada caso. Los ficheros de prueba son los siguientes:

caso1.txt: Todos los min/max son positivos y el nº marcos de los procesos siempre será menor que el de memoria -> En el primer volcado, debería notificar que no hay error y los valores se mantienen.

caso2.txt: Todos los min/max son negativos -> Todos los rangos deberían ser el mínimo valor, tanto mínimos como máximos y deberían aparecer los 7 errores. Por ejemplo, marcos de la memoria (min=1, max=1), tiempo de llegada (min=0, max=0).

caso3.txt: Todos los min/max son positivos SALVO el de nº marcos y el nº marcos de los procesos (en el fichero) es mayor que el de memoria -> Debería situar el nº marcos como límite superior. Se informa de que era negativo, no se informa explícitamente que se ha cambiado el límite superior (el cálculo del valor máximo también debe tener esto en cuenta).

>> Para poder probarlo, inicialmente se pone como mínimo un número negativo grande (para que haya muchas más posibilidades de que sea negativo) y el máximo debe ser notablemente más grande que el nº marcos para que luego se note que se está usando realmente el nº marcos como límite.

caso4.txt: Todos los min/max son positivos y el mínimo de nº marcos generado excede el nº marcos en memoria -> Informa de este error y recalcula el mínimo entre (1-nºmarcos). Al máximo le pasa lo del ejemplo anterior (si nºmar mem<maxmarfichero entonces el límite será el nºmar mem y si no se deja como estaba). Este es uno de los casos que no pasaría si se hubiese puesto directamente el límite inferior en vez de volverlo a calcular.

casoNormal.txt: Es el usado en el vídeo explicativo y contiene unos rangos con límites inferiores negativos y límites superiores positivos, es decir, algunos parámetros se generarán correctamente y otros deberán ser corregidos.

4. EJERCICIOS RESUELTOS

A continuación se realizarán dos ejercicios para mostrar la ejecución paso a paso del algoritmo. Dichos ejercicios se encontrarán disponibles en el directorio ‘FDatos’ con los nombres ‘ej1_FCFS.txt’ y ‘ej2_SJF.txt’. El primer ejercicio se resolverá por FCFS y será un ejemplo sencillo, para ir familiarizándose con el planteamiento de la ejecución y cómo se van a ver representados los datos. El segundo ejercicio se resolverá por SJF y será más complejo, para abarcar un mayor número de excepciones y demostrar las posibilidades del programa. Ambos serán resueltos primero a mano y finalmente con el script.

4.1. Ejercicio sencillo

Ref	Til	Tej	nMar	Dirección-Página
P01	2	5	3	98-0 6325-63 23-0 452-4 314-3
P02	4	5	2	523-5 66-0 120-1 38-0 785-7
P03	4	3	2	4444-44 10-0 241-2

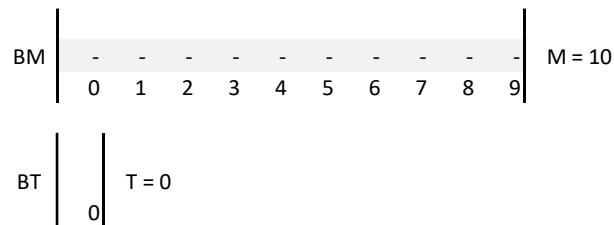
Algoritmo: FCFS
Memoria sistema = 1000
Tamaño página = 100
Nº marcos memoria = 10

Nótese que número de marcos de la memoria principal es igual al tamaño total de la memoria dividido entre el tamaño de página.

4.1.1. Ejemplo a mano

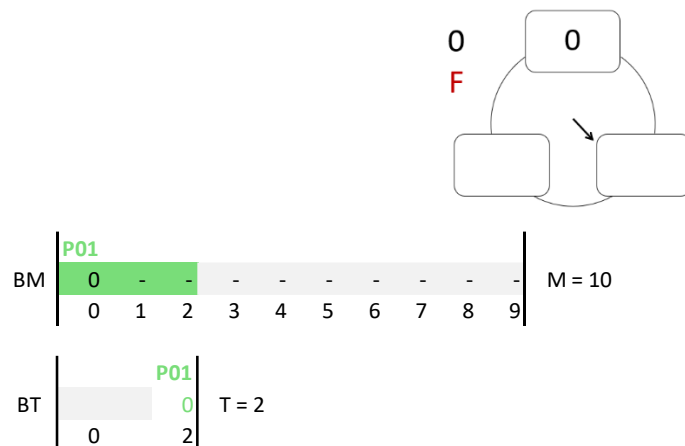
T=0

Se trata del instante inicial. Como aún no ha llegado ningún proceso, la memoria se encuentra vacía.



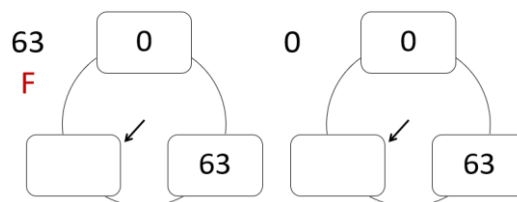
T=2

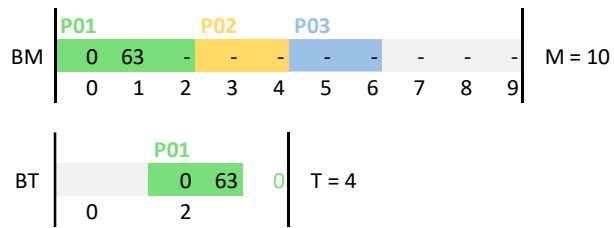
Llega el proceso **P01** al sistema y se introduce en memoria. Como ocupa 3 marcos, ahora hay 7 marcos libres en memoria. Se introduce en la CPU al ser el único proceso en memoria y comienza a ejecutarse. Este proceso introduce su primera página, produciéndose 1 fallo de paginación. La manecilla del reloj apunta al siguiente marco.



T=4

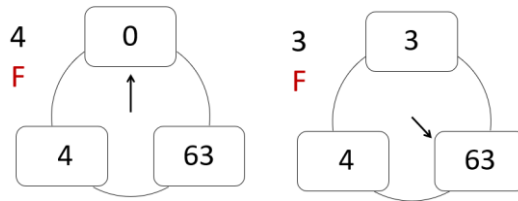
Llegan los procesos **P02** y **P03** al sistema y entran en memoria. Como cada uno ocupa 2 marcos, ahora hay 3 marcos libres en la memoria principal. Ambos procesos permanecerán en memoria a esperas de que la CPU quede libre. El proceso **P01** sigue ejecutándose y en este intervalo ha introducido sus 2 páginas siguientes, produciendo solo 1 fallo de paginación, puesto que la página 0 ya se encontraba cargada en la memoria principal. Cuando no se produce fallo, la manecilla no avanza.



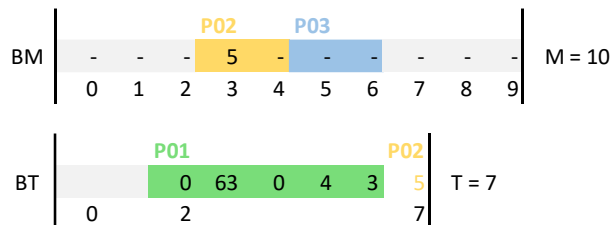
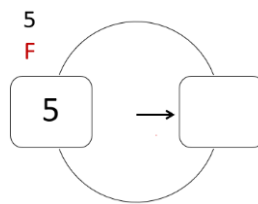


T=7

El proceso **P01** ha terminado de ejecutar sus páginas y se han producido 2 fallos de paginación adicionales. Finaliza su ejecución con 4 fallos totales y libera los espacios que había ocupado en memoria.

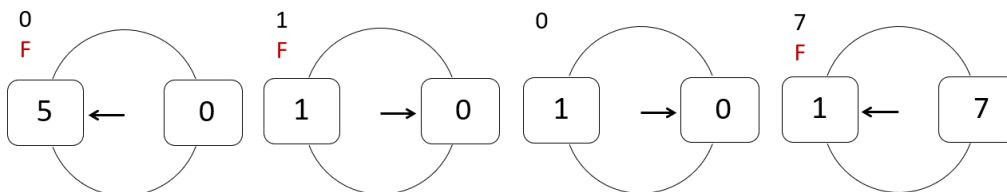


A continuación, el proceso que pasará a ejecutarse en este instante será **P02**, puesto que es el que primero ha llegado a memoria y el algoritmo utilizado es FCFS. Introduce su primera página en memoria y se produce un fallo de paginación.

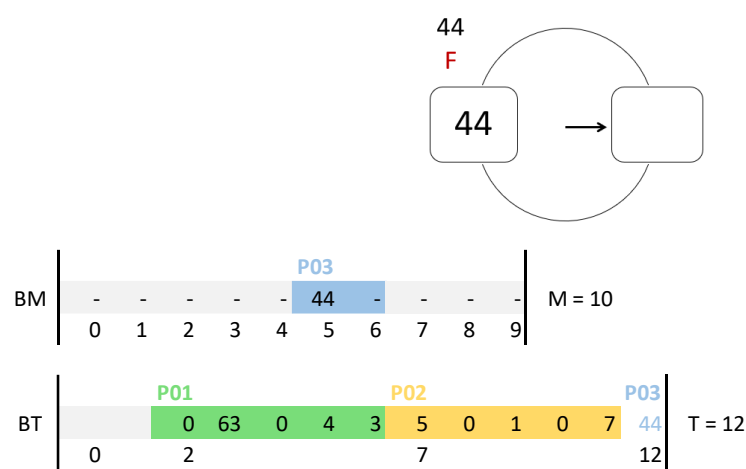


T=12

El proceso **P02** termina de ejecutarse con 4 fallos de paginación totales. Abandona la CPU y libera los marcos M3 y M4 de la memoria principal.

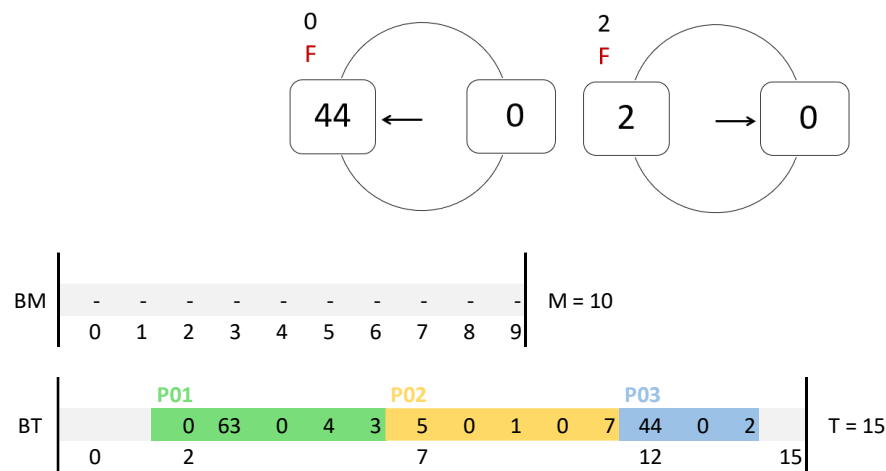


Finalmente entra el proceso **P03** al procesador para comenzar a ejecutarse a partir de este instante. Carga su primera página en memoria y se produce 1 fallo de paginación, por lo que avanza la manecilla al siguiente marco.



T=15

El proceso **P03** ha finalizado su ejecución con 3 fallos de página en total. Libera los marcos M5 y M6 que había ocupado en la memoria principal. Ya no quedan más procesos por llegar al sistema. Tanto la memoria principal como la CPU se encuentran vacías.



4.1.2. Ejecución en el script

A continuación se muestra cómo se visualizaría el ejemplo anterior en el script. Aprovechando que el primer instante $T=0$ está vacío, se explicará a continuación las diferentes estructuras que componen los volcados a pantalla.

T=0

FCFS/SJF - PAGINACIÓN - RELOJ - MEMORIA CONTINUA - NO REUBICABLE														
1	T=0 Algoritmo utilizado = FCFS Memoria del sistema = 1000 Tamaño de página = 100 Número de marcos = 10													
	Ref	Tll	Tej	nMar	Tesp	Tret	Trej	Mini	Mfin	Estado	Dirección-Página			
	P01	2	5	3	-	-	-	-	-	Fuera de sist.	98-0	6325-63	23-0	452-4 314-3
	P02	4	5	2	-	-	-	-	-	Fuera de sist.	523-5	66-0	120-1	38-0 785-7
	P03	4	3	2	-	-	-	-	-	Fuera de sist.	4444-44	10-0	241-2	
3	Tiempo Medio de Espera: 0.0 Tiempo Medio de Retorno: 0.0													
	M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M=10													
	BM - - - - - - - - - - M=10													
	BT T=0													
	0													

1) $T=_$ indica el instante de tiempo actual. El programa hará volcados a pantalla en los instantes en los que ocurra un evento importante, es decir, momentos como entradas de procesos a memoria, entrada/salida de la CPU, etc.

2) Es una tabla con el resumen de los procesos y sus parámetros. A medida que se vaya avanzando en el tiempo, se actualizarán sus tiempos de espera, retorno, marcos asociados y estado.

3) Son los tiempos medios de espera y retorno. Se van actualizando en cada instante a partir de la información de los procesos.

4) Es la tabla de marcos de página. Actualmente se encuentra vacía pero, a medida que entran procesos, muestra su referencia (P01, P02...), los marcos que ocupa, qué páginas hay alojadas en cada marco y subraya al marco al que apunta la manecilla.

5) La banda de memoria representa los espacios de la memoria principal. En un primer instante está vacía, pero a medida que se introducen procesos, muestra los mismos parámetros que la tabla de marcos (salvo el subrayado para la manecilla).

6) La banda de tiempo representa qué proceso ha estado ejecutándose en cada instante y cuáles son las páginas que ha ido cargando en memoria.

Adicionalmente, a partir de que se empiece la ejecución, se irá mostrando un log de eventos en la parte superior de la pantalla, que informa de los acontecimientos que han tenido lugar en cada instante. Además, cada vez que un proceso finaliza su ejecución, se imprimirá por pantalla un resumen de sus fallos de paginación y los marcos en los que han tenido lugar.

Si a un marco le apunta la manecilla, éste se subrayará (tanto en reloj como en segunda oportunidad). En reloj también se pondrá su bit a 1. Las páginas que hayan causado un fallo en un instante (o las que cambian su bit a 1, en caso de segunda oportunidad) se sombrearán del color del proceso.

T=2

FCFS/SJF - PAGINACIÓN - RELOJ - MEMORIA CONTINUA - NO REUBICABLE

T=2 Algoritmo utilizado = FCFS Memoria del sistema = 1000 Tamaño de página = 100 Número de marcos = 10

Entra el proceso **P01** a memoria a partir del marco 0

Entra el proceso **P01** al procesador

Ref	Tll	Tej	nMar	Tesp	Tret	Trej	Mini	Mfin	Estado	Dirección-Página
P01	2	5	3	0	0	5	0	2	En ejecución	98-0 6325-63 23-0 452-4 314-3
P02	4	5	2	-	-	-	-	-	Fuera de sist.	523-5 66-0 120-1 38-0 785-7
P03	4	3	2	-	-	-	-	-	Fuera de sist.	4444-44 10-0 241-2

Tiempo Medio de Espera: 0.0 Tiempo Medio de Retorno: 0.0

P01

M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 | M=10
0 - -

BM | P01 |
| 0 - - - - - - - - - - | M=10
| 0 3 |

BT | P01 |
| 0 0 | T=2
| 0 2 |

T=4

FCFS/SJF - PAGINACIÓN - RELOJ - MEMORIA CONTINUA - NO REUBICABLE

T=4 Algoritmo utilizado = FCFS Memoria del sistema = 1000 Tamaño de página = 100 Número de marcos = 10

Entra el proceso **P02** a memoria a partir del marco 3

Entra el proceso **P03** a memoria a partir del marco 5

Sigue ejecutándose el proceso **P01**

Ref	Tll	Tej	nMar	Tesp	Tret	Trej	Mini	Mfin	Estado	Dirección-Página
P01	2	5	3	0	2	3	0	2	En ejecución	98-0 6325-63 23-0 452-4 314-3
P02	4	5	2	0	0	5	3	4	En memoria	523-5 66-0 120-1 38-0 785-7
P03	4	3	2	0	0	3	5	6	En memoria	4444-44 10-0 241-2

Tiempo Medio de Espera: 0.0 Tiempo Medio de Retorno: .66

P01 P02 P03
M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 | M=10
0 63 - - - - -

BM | P01 P02 P03 |
| 0 63 - - - - - - - - - - | M=10
| 0 3 5 7 |

BT | P01 |
| 0 0 63 0 | T=4
| 0 2 |

T=7

FCFS/SJF - PAGINACIÓN - RELOJ - MEMORIA CONTINUA - NO REUBICABLE

T=7 Algoritmo utilizado = FCFS Memoria del sistema = 1000 Tamaño de página = 100 Número de marcos = 10

El proceso **P01** ha finalizado y ha transcurrido este tiempo: 5

P01 -> Tiempo Entrada: 2 Tiempo Salida: 7 Tiempo Restante: 0

Entra el proceso **P02** al procesador

Ref	Tll	Tej	nMar	Tesp	Tret	Trej	Mini	Mfin	Estado	Dirección-Página
P01	2	5	3	0	5	0	-	-	Finalizado	98-0 6325-63 23-0 452-4 314-3
P02	4	5	2	3	3	5	3	4	En ejecución	523-5 66-0 120-1 38-0 785-7
P03	4	3	2	3	3	3	5	6	En memoria	4444-44 10-0 241-2

Tiempo Medio de Espera: 2.00 Tiempo Medio de Retorno: 3.66

Se han producido 4 fallos de página en la ejecución de **P01**

	0	63	0	4	3
M0	0-0	0-0	0-0	0-1	3-0
M1		63-0	63-0	63-0	63-1
M2				4-0	4-0
	F	F		F	F

P02 P03
M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 | M=10
5 - - -

BM | P02 P03 |
| - - - 5 - - - - - - - - - - | M=10
| 0 3 5 7 |

BT | P01 P02 |
| 0 0 63 0 4 3 5 | T=7
| 0 2 7 |

T=12

FCFS/SJF - PAGINACIÓN - RELOJ - MEMORIA CONTINUA - NO REUBICABLE

T=12 Algoritmo utilizado = FCFS Memoria del sistema = 1000 Tamaño de página = 100 Número de marcos = 10

El proceso P02 ha finalizado y ha transcurrido este tiempo: 5

P02 -> Tiempo Entrada: 7 Tiempo Salida: 12 Tiempo Restante: 0

Entra el proceso P03 al procesador

Ref	Tll	Tej	nMar	Tesp	Tret	Trej	Mini	Mfin	Estado	Dirección-Página
P01	2	5	3	0	5	0	-	-	Finalizado	98-0 6325-63 23-0 452-4 314-3
P02	4	5	2	3	8	0	-	-	Finalizado	523-5 66-0 120-1 38-0 785-7
P03	4	3	2	8	8	3	5	6	En ejecución	4444-44 10-0 241-2

Tiempo Medio de Espera: 3.66 Tiempo Medio de Retorno: 7.00

Se han producido 4 fallos de página en la ejecución de P02

M3	M4	M5	M6	M7	M8	M9	M10
5-0	5-1	1-0	1-0	1-0	1-1	1-1	1-1
0-0	0-0	0-1	0-1	0-1	7-0	7-0	7-0
F	F	F	F	F	F	F	F

P03
M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 | M=10
44 -

BM | - - - - - | M=10
| 0 5 7 |

BT | P01 P02 P03 | T=12
| 0 2 7 12 |

T=15

FCFS/SJF - PAGINACIÓN - RELOJ - MEMORIA CONTINUA - NO REUBICABLE

T=15 Algoritmo utilizado = FCFS Memoria del sistema = 1000 Tamaño de página = 100 Número de marcos = 10

El proceso P03 ha finalizado y ha transcurrido este tiempo: 3

P03 -> Tiempo Entrada: 12 Tiempo Salida: 15 Tiempo Restante: 0

Ref	Tll	Tej	nMar	Tesp	Tret	Trej	Mini	Mfin	Estado	Dirección-Página
P01	2	5	3	0	5	0	-	-	Finalizado	98-0 6325-63 23-0 452-4 314-3
P02	4	5	2	3	8	0	-	-	Finalizado	523-5 66-0 120-1 38-0 785-7
P03	4	3	2	8	11	0	-	-	Finalizado	4444-44 10-0 241-2

Tiempo Medio de Espera: 3.66 Tiempo Medio de Retorno: 8.00

Se han producido 3 fallos de página en la ejecución de P03

M5	M6	M7	M8	M9	M10
44-0	44-1	2-0	2-0	2-0	2-0
0-0	0-0	0-1	0-1	0-1	0-1
F	F	F	F	F	F

M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 | M=10

BM | - - - - - | M=10
| 0 5 7 12 15 |

BT | P01 P02 P03 | T=15
| 0 2 7 12 15 |

Cuando acaba la ejecución del algoritmo, se muestra un resumen de los tiempos de espera, retorno y fallos de paginación de cada proceso junto a los tiempos medios finales.

RESUMEN FINAL

T.Espera -> Tiempo que el proceso no ha estado ejecutándose en la CPU desde que entra en memoria 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

Fallos Pág. -> Número de fallos de página que han ocurrido en la ejecución de cada proceso

Proc.	T.Espera	Inicio/Fin	T.Retorno	Fallos Pág.
P01	0	2/7	5	4
P02	3	4/12	8	4
P03	8	4/15	11	3

Tiempo total transcurrido en ejecutar todos los procesos: 15

Media tiempo espera de todos los procesos: 3.66

Media tiempo retorno de todos los procesos: 8.00

4.2. Ejercicio complejo

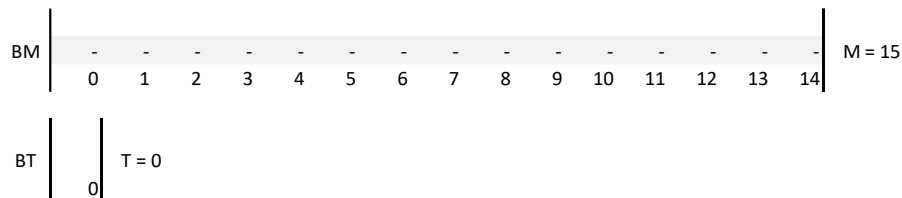
Ref	Til	Tej	nMar	Dirección-Página
P01	1	5	2	6123-61 5894-58 12-0 456-4 6123-61
P02	4	16	7	111-1 121- 414-4 531-5 676-6 717-7 102-1 835-8 98-0 19999-199 45-0 722-7 389-3 444-4 521-5 666-6
P03	4	10	8	111-1 212-2 121-1 4444-44 515-5 131-1 6666-66 525-5 717-7 10-0
P04	6	8	2	4444-44 515-5 131-1 6666-66 525-5 717-7 10-0 251-2
P05	30	9	7	453-4 78522-785 965-9 421-4 555-5 234-2 965-9 421-4 741-7
P06	30	10	2	111-1 38-0 121-1 6542-65 54-0 852-8 4569-45 123-1 6542-65
P07	65	9	2	74-0 38-0 121-1 852-8 50-0 4569-45 23-0 6542-65 238-2

Algoritmo: SJF
 Memoria sistema = 1500
 Tamaño página = 100
 Nº marcos memoria = 15

4.2.1. Ejemplo a mano

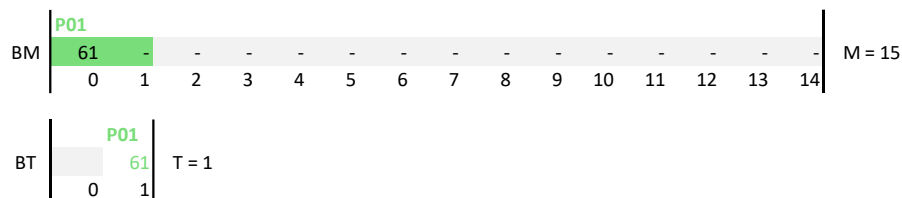
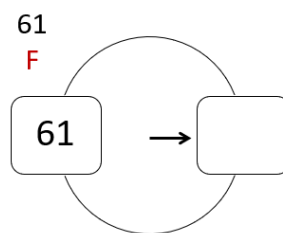
T=0

Se trata del instante inicial. Como aún no ha llegado ningún proceso, la memoria se encuentra vacía.



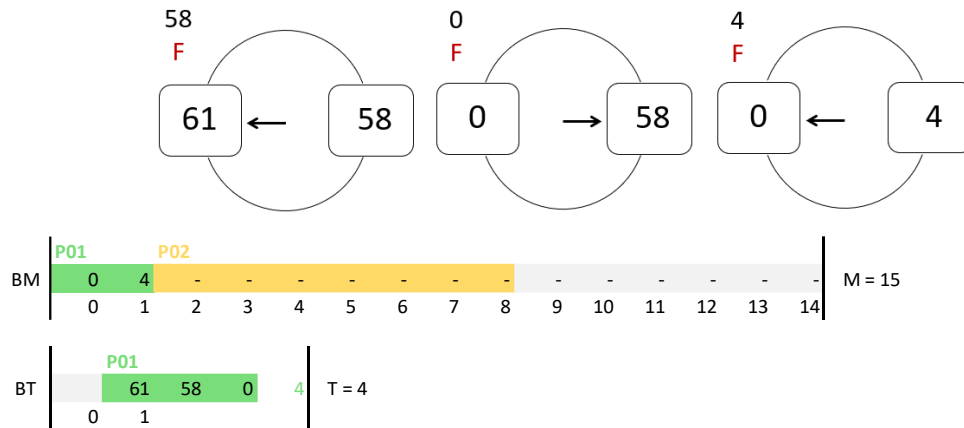
T=1

Llega el proceso **P01** al sistema y entra en memoria. Ocupa los marcos M0 y M1, dejando 13 marcos vacíos para el resto de los procesos. Entra en la CPU y comienza a ejecutarse. Carga su primera página en memoria, lo que provoca un fallo de paginación. La manecilla del reloj apunta ahora al siguiente marco.



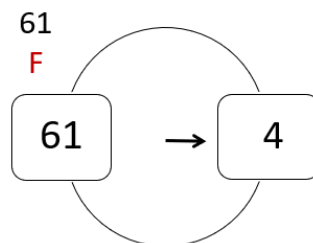
T=4

Llegan los procesos **P02** y **P03** al sistema. El proceso **P02** se introduce en memoria, a esperas de que la CPU quede libre. Ahora hay 6 marcos libres en la memoria principal. El proceso **P03** ocupa 7 marcos, por lo que no cabe en memoria. Se bloquea y se quedará en espera hasta que haya hueco suficiente en la memoria principal para que entre. El proceso **P01** continúa ejecutándose y ha introducido 3 páginas más, produciendo 3 fallos de paginación.

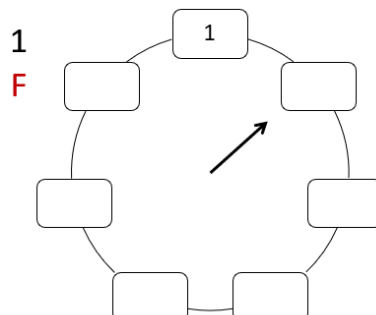


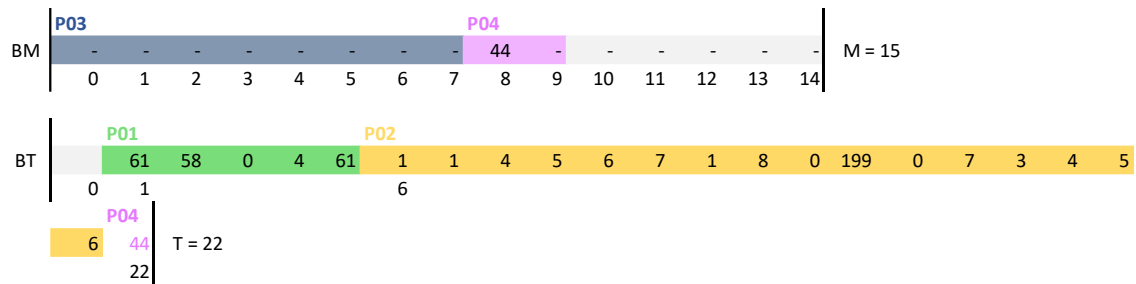
T=6

El proceso **P01** ha ejecutado su última página y ha dejado la CPU libre. En total ha tenido 5 fallos de paginación.



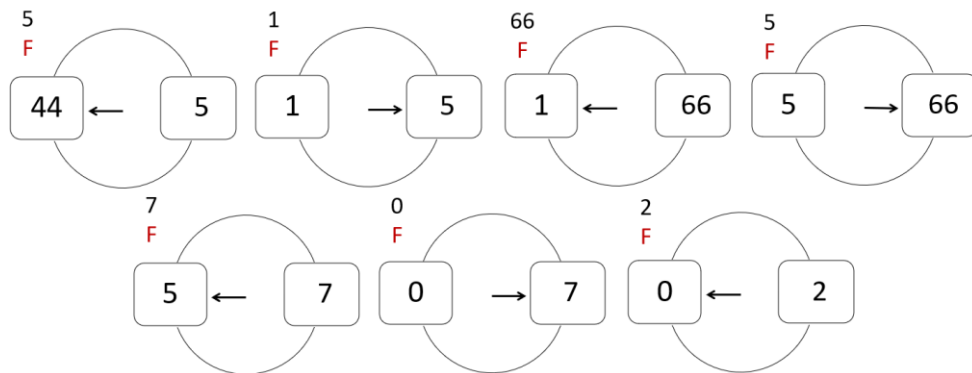
Entra ahora a ejecutarse el proceso **P02**, puesto que es el único que se encuentra en memoria, y carga su primera página, produciendo 1 fallo de paginación. En este instante también ha llegado al sistema el proceso **P04**, pero recordemos que, antes que él, el proceso **P03** se encontraba bloqueado, a esperas de que hubiera hueco suficiente en memoria. Los 2 marcos libres que ha dejado el proceso **P01** no son suficientes para alojar a **P03**, a pesar de que haya 9 marcos libres. Debido a que la memoria es continua, el proceso no puede dividirse. Como él no puede entrar, tampoco puede hacerlo **P04**, debido a que la cola de entrada a memoria es una cola FIFO.



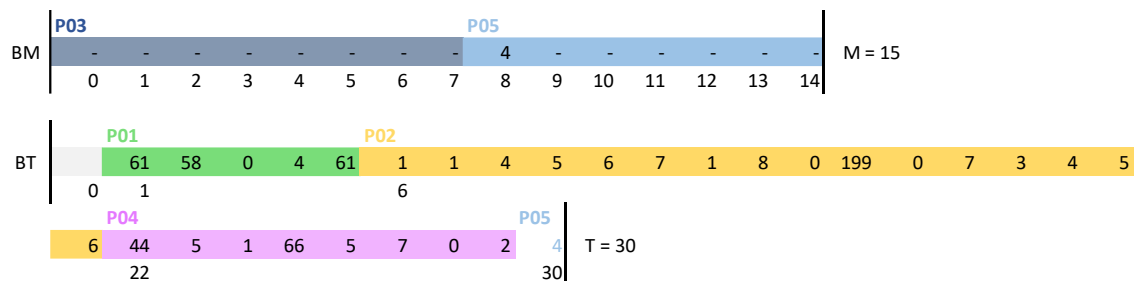
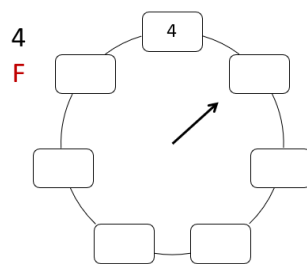


T=30

Finaliza su ejecución el proceso **P04** con 8 fallos de paginación y deja libre la CPU.

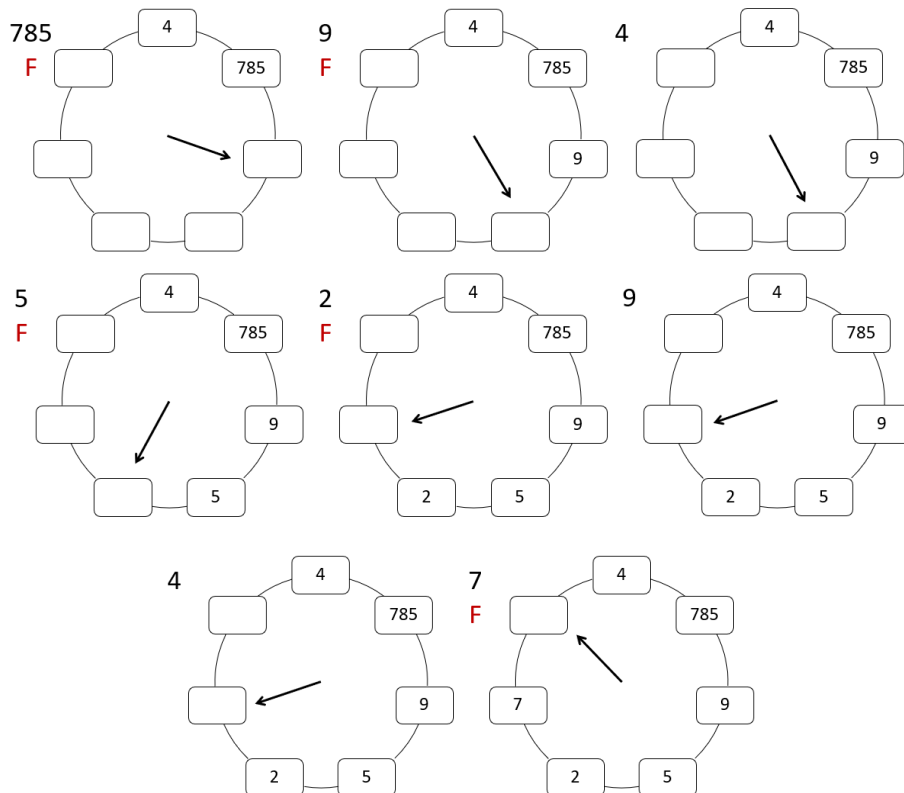


En este instante también han llegado al sistema los procesos **P05** y **P06**. De entre ellos dos, solo **P05** entra a memoria por ser el primero en llegar (recordemos cola FIFO de entrada a memoria), y ocupa los 7 marcos que quedaban vacíos en la memoria, dejando así 0 marcos libres. Es por lo tanto que **P06** no puede entrar y se queda en espera. El proceso **P05** comienza a ejecutarse por tener menor tiempo de CPU que **P03** (9 instantes frente a 10) y carga su primera página.

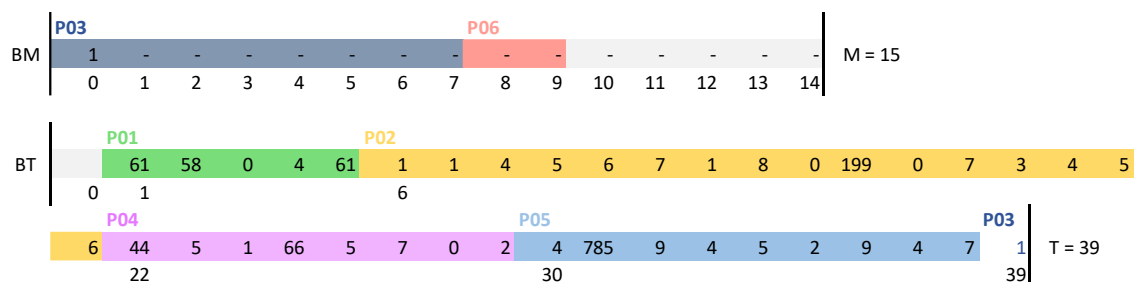
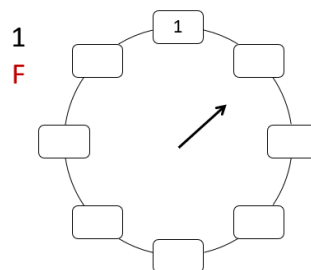


T=39

Finaliza su ejecución el proceso **P05** con 6 fallos de página y deja libres los 7 marcos de memoria que estaba ocupando, haciendo que pueda entrar a memoria **P06**, el cual ya había llegado al sistema pero no tenía sitio para alojarse.

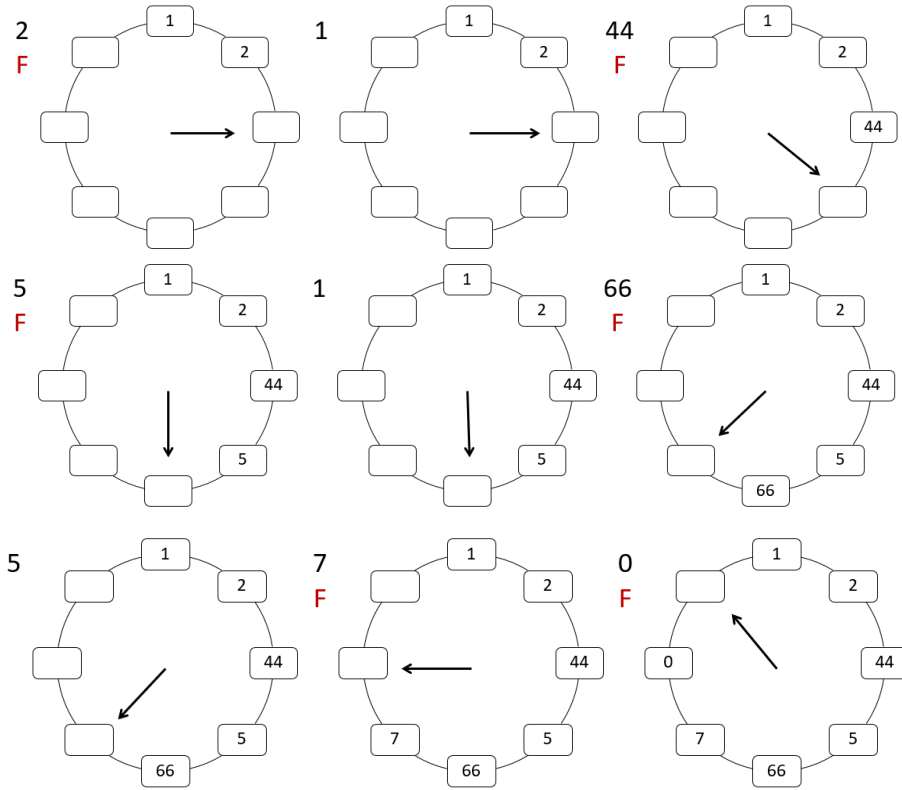


Se comienza a ejecutar el proceso **P03**, que ya llevaba bastante tiempo en memoria pero por su largo tiempo de CPU no había sido seleccionado para ejecutarse. Esta vez, ambos procesos en memoria tiene tiempo de ejecución 10, pero, ante un empate, se emplea el algoritmo FIFO, por lo que se ejecutará el primero que llegó (como ya se ha mencionado, **P03**). Carga su primera página en memoria.

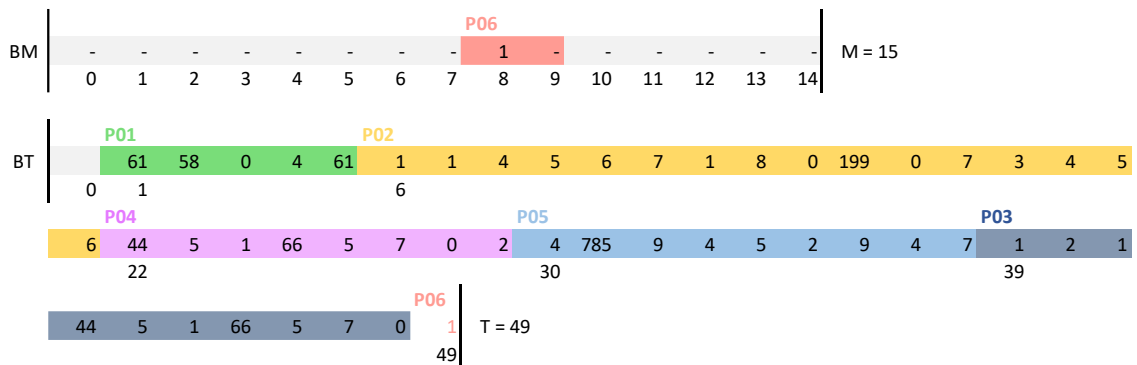
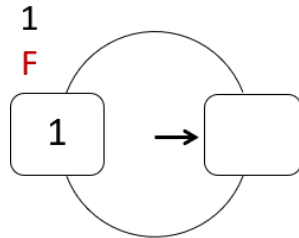


T=49

Finaliza la ejecución de **P03** con 7 fallos de paginación. El proceso abandona la CPU y libera los espacios que ocupaba en la memoria.

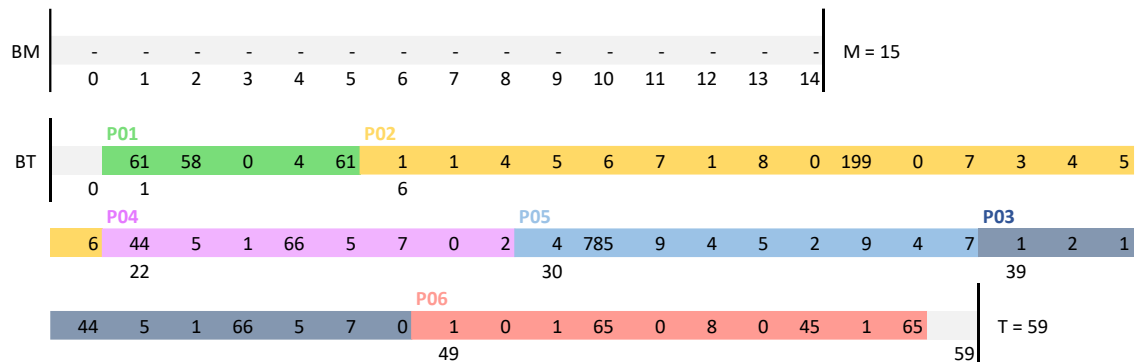
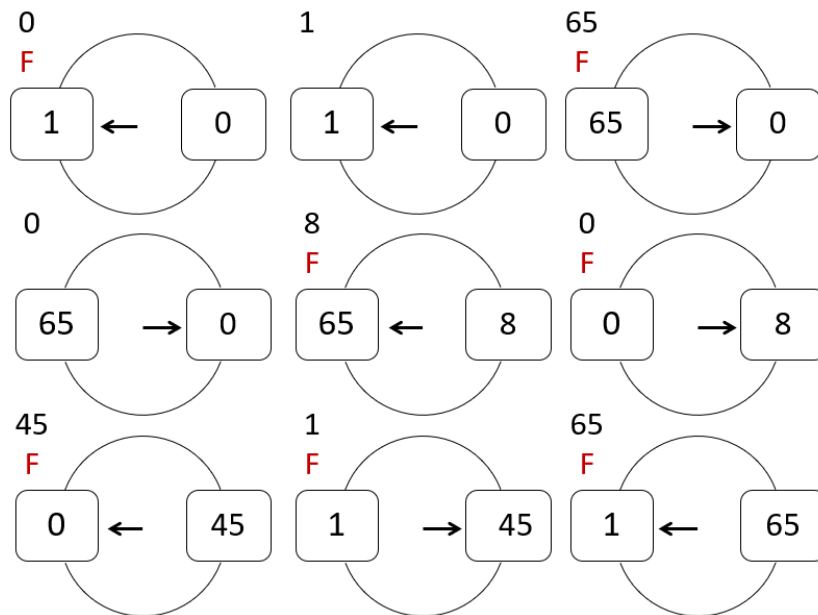


Pasa a ejecutarse el proceso **P06** y carga su primera página.



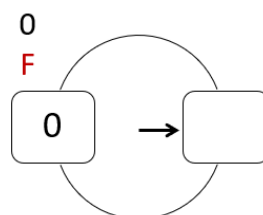
T=59

Finaliza la ejecución de **P06** con 8 fallos de página. Tanto la memoria como la CPU quedan vacías, pero la ejecución del algoritmo aún no ha terminado, puesto que queda un último proceso.



T=65

Llega al sistema el proceso **P07** y se introduce en memoria. Al ser el único proceso en la cola de entrada al procesador, pasa a ejecutarse e introduce su primera página.



En la banda de tiempo se puede observar que ha habido varios instantes de tiempo en los que no se ha estado ejecutando ningún proceso.

4.2.2. Ejecución en el script

T=0

```
FCFS/SJF - PAGINACIÓN - RELOJ - MEMORIA CONTINUA - NO REUBICABLE
T=0 Algoritmo utilizado = SJF Memoria del sistema = 1500 Tamaño de página = 100 Número de marcos = 15
Ref Tll Tej nMar Tesp Tret Trej MInl Mfin Estado Dirección-Página
P01 1 5 2 - - - - - Fuera de sist. 6123-61 5894-58 12-0 456-4 6123-61
P02 4 16 7 - - - - - Fuera de sist. 111-1 121-1 414-4 531-5 676-6 717-7 102-1 835-8 98-0 19999-199 45-0 722-7 389-3 444-4 521-5 666-6
P03 4 10 8 - - - - - Fuera de sist. 111-1 212-2 121-1 4444-44 515-5 131-1 6666-66 525-5 717-7 10-0
P04 6 8 2 - - - - - Fuera de sist. 4444-44 515-5 131-1 6666-66 525-5 717-7 10-0 251-2
P05 30 9 7 - - - - - Fuera de sist. 453-4 78522-785 965-9 421-4 555-5 234-2 965-9 421-4 741-7
P06 30 10 2 - - - - - Fuera de sist. 111-1 38-0 121-1 6542-65 54-0 852-8 50-0 4569-45 123-1 6542-65
P07 65 9 2 - - - - - Fuera de sist. 74-0 38-0 121-1 852-8 50-0 4569-45 23-0 6542-65 238-2
Tiempo Medio de Espera: 0.0 Tiempo Medio de Retorno: 0.0
```

M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 M12 M13 M14 | M=15

BM | - - - - - - - - - - - - - - - | M=15
0

BT | |
0 | T=0

T=1

```
FCFS/SJF - PAGINACIÓN - RELOJ - MEMORIA CONTINUA - NO REUBICABLE
T=1 Algoritmo utilizado = SJF Memoria del sistema = 1500 Tamaño de página = 100 Número de marcos = 15
Entra el proceso P01 a memoria a partir del marco 0
Entra el proceso P01 al procesador
Ref Tll Tej nMar Tesp Tret Trej MInl Mfin Estado Dirección-Página
P01 1 5 2 0 0 5 0 1 En ejecución 6123-61 5894-58 12-0 456-4 6123-61
P02 4 16 7 - - - - - Fuera de sist. 111-1 121-1 414-4 531-5 676-6 717-7 102-1 835-8 98-0 19999-199 45-0 722-7 389-3 444-4 521-5 666-6
P03 4 10 8 - - - - - Fuera de sist. 111-1 212-2 121-1 4444-44 515-5 131-1 6666-66 525-5 717-7 10-0
P04 6 8 2 - - - - - Fuera de sist. 4444-44 515-5 131-1 6666-66 525-5 717-7 10-0 251-2
P05 30 9 7 - - - - - Fuera de sist. 453-4 78522-785 965-9 421-4 555-5 234-2 965-9 421-4 741-7
P06 30 10 2 - - - - - Fuera de sist. 111-1 38-0 121-1 6542-65 54-0 852-8 50-0 4569-45 123-1 6542-65
P07 65 9 2 - - - - - Fuera de sist. 74-0 38-0 121-1 852-8 50-0 4569-45 23-0 6542-65 238-2
Tiempo Medio de Espera: 0.0 Tiempo Medio de Retorno: 0.0
```

P01
M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 M12 M13 M14 | M=15
61 -

BM | P01 |
0 2 | M=15

BT | P01 |
0 1 | T=1

T=4

```
FCFS/SJF - PAGINACIÓN - RELOJ - MEMORIA CONTINUA - NO REUBICABLE
T=4 Algoritmo utilizado = SJF Memoria del sistema = 1500 Tamaño de página = 100 Número de marcos = 15
Entra el proceso P02 a memoria a partir del marco 2
Segue ejecutándose el proceso P01
Ref Tll Tej nMar Tesp Tret Trej MInl Mfin Estado Dirección-Página
P01 1 5 2 0 3 2 0 1 En ejecución 6123-61 5894-58 12-0 456-4 6123-61
P02 4 16 7 0 0 16 2 8 En memoria 111-1 121-1 414-4 531-5 676-6 717-7 102-1 835-8 98-0 19999-199 45-0 722-7 389-3 444-4 521-5 666-6
P03 4 10 8 0 0 10 - - En espera 111-1 212-2 121-1 4444-44 515-5 131-1 6666-66 525-5 717-7 10-0
P04 6 8 2 - - - - - Fuera de sist. 4444-44 515-5 131-1 6666-66 525-5 717-7 10-0 251-2
P05 30 9 7 - - - - - Fuera de sist. 453-4 78522-785 965-9 421-4 555-5 234-2 965-9 421-4 741-7
P06 30 10 2 - - - - - Fuera de sist. 111-1 38-0 121-1 6542-65 54-0 852-8 50-0 4569-45 123-1 6542-65
P07 65 9 2 - - - - - Fuera de sist. 74-0 38-0 121-1 852-8 50-0 4569-45 23-0 6542-65 238-2
Tiempo Medio de Espera: 0.0 Tiempo Medio de Retorno: 1.00
```

P01 P02
M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 M12 M13 M14 | M=15
0 4 - - - - - - -

BM | P01 P02 |
0 2 - - - - - 9 | M=15

BT | P01 |
0 1 | T=4

T=6

FCFS/SJF - PAGINACIÓN - RELOJ - MEMORIA CONTINUA - NO REUBICABLE

T=6 Algoritmo utilizado = SJF Memoria del sistema = 1500 Tamaño de página = 100 Número de marcos = 15

El proceso P01 ha finalizado y ha transcurrido este tiempo: 5

P01 -> Tiempo Entrada: 1 Tiempo Salida: 6 Tiempo Restante: 0

Entra el proceso P02 al procesador

Ref	Tll	Tej	nMar	Tesp	Tret	Trej	Mini	Mfin	Estado	Dirección-Página
P01	1	5	2	0	5	0	-	-	Finalizado	6123-61 5894-58 12-0 456-4 6123-61
P02	4	16	7	2	2	16	2	8	En ejecución	111-1 121-1 414-4 531-5 676-6 717-7 102-1 835-8 98-0 19999-199 45-0 722-7 389-3 444-4 521-5 666-6
P03	4	10	8	2	2	10	-	-	En espera	111-1 212-2 121-1 4444-44 515-5 131-1 6666-66 525-5 717-7 10-0
P04	6	8	2	0	0	8	-	-	En espera	4444-44 515-5 131-1 6666-66 525-5 717-7 10-0 251-2
P05	30	9	7	-	-	-	-	-	Fuera de sist.	453-4 78522-785 965-9 421-4 555-5 234-2 965-9 421-4 741-7
P06	30	10	2	-	-	-	-	-	Fuera de sist.	111-1 38-0 121-1 6542-65 54-0 852-8 50-0 4569-45 123-1 6542-65
P07	65	9	2	-	-	-	-	-	Fuera de sist.	74-0 38-0 121-1 852-8 50-0 4569-45 23-0 6542-65 238-2

Tiempo Medio de Espera: 1.00 Tiempo Medio de Retorno: 2.25

Se han producido 5 fallos de página en la ejecución de P01

M0	61	58	0	4	61
M0	61-0	61-1	0-0	0-1	61-0
M1		58-0	58-1	4-0	4-1
	F	F	F	F	F

M0	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M=15
		1	-	-	-	-	-	-	-	-	-	-	-	-	-

BM	P02	M=15
	0 1 2 3 4 5 6 7 8 9	

BT	P01	P02	T=6
	0 1 2 3 4 5 6		

T=22

FCFS/SJF - PAGINACIÓN - RELOJ - MEMORIA CONTINUA - NO REUBICABLE

T=22 Algoritmo utilizado = SJF Memoria del sistema = 1500 Tamaño de página = 100 Número de marcos = 15

El proceso P02 ha finalizado y ha transcurrido este tiempo: 16

P02 -> Tiempo Entrada: 6 Tiempo Salida: 22 Tiempo Restante: 0

Entra el proceso P03 a memoria a partir del marco 0

Entra el proceso P04 a memoria a partir del marco 8

Entra el proceso P04 al procesador

Ref	Tll	Tej	nMar	Tesp	Tret	Trej	Mini	Mfin	Estado	Dirección-Página
P01	1	5	2	0	5	0	-	-	Finalizado	6123-61 5894-58 12-0 456-4 6123-61
P02	4	16	7	2	18	0	-	-	Finalizado	111-1 121-1 414-4 531-5 676-6 717-7 102-1 835-8 98-0 19999-199 45-0 722-7 389-3 444-4 521-5 666-6
P03	4	10	8	18	18	10	0	7	En memoria	111-1 212-2 121-1 4444-44 515-5 131-1 6666-66 525-5 717-7 10-0
P04	6	8	2	16	16	8	8	9	En ejecución	4444-44 515-5 131-1 6666-66 525-5 717-7 10-0 251-2
P05	30	9	7	-	-	-	-	-	Fuera de sist.	453-4 78522-785 965-9 421-4 555-5 234-2 965-9 421-4 741-7
P06	30	10	2	-	-	-	-	-	Fuera de sist.	111-1 38-0 121-1 6542-65 54-0 852-8 50-0 4569-45 123-1 6542-65
P07	65	9	2	-	-	-	-	-	Fuera de sist.	74-0 38-0 121-1 852-8 50-0 4569-45 23-0 6542-65 238-2

Tiempo Medio de Espera: 9.00 Tiempo Medio de Retorno: 14.25

Se han producido 12 fallos de página en la ejecución de P02

M2	1-0	1-0	1-0	1-0	1-0	1-0	1-0	1-0	1-0	1-0	1-0	1-0	1-0	1-0	1-0
M3			4-0	4-0	4-0	4-0	4-0	4-0	4-0	4-0	4-0	4-0	4-0	4-0	4-0
M4				5-0	5-0	5-0	5-0	5-0	5-0	5-0	5-0	5-0	5-0	5-0	5-0
M5					6-0	6-0	6-0	6-0	6-0	6-0	6-0	6-0	6-0	6-0	6-0
M6						7-0	7-0	7-0	7-0	7-0	7-0	7-0	7-0	7-0	7-0
M7							8-0	8-0	8-0	8-0	8-0	8-0	8-0	8-0	8-0
M8								9-0	9-0	9-0	9-0	9-0	9-0	9-0	9-0
	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F

M0	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M=15
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

BM	P03	P04	M=15
	0 1 2 3 4 5 6 7 8 9 10		

BT	P01	P02	P04	T=22
	0 1 2 3 4 5 6	7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22	23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44	

T=30

FCFS/SJF - PAGINACIÓN - RELOJ - MEMORIA CONTINUA - NO REUBICABLE

T=30 Algoritmo utilizado = SJF Memoria del sistema = 1500 Tamaño de página = 100 Número de marcos = 15

El proceso P04 ha finalizado y ha transcurrido este tiempo: 8

P04 -> Tiempo Entrada: 22 Tiempo Salida: 30 Tiempo Restante: 0

Entra el proceso P05 a memoria a partir del marco 8

Entra el proceso P05 al procesador

Ref	Tll	Tej	nMar	Tesp	Tret	Trej	Mini	Mfin	Estado	Dirección-Página
P01	1	5	2	0	5	0	-	-	Finalizado	6123-61 5894-58 12-0 456-4 6123-61
P02	4	16	7	2	18	0	-	-	Finalizado	111-1 121-1 414-4 531-5 676-6 717-7 102-1 835-8 98-0 19999-199 45-0 722-7 389-3 444-4 521-5 666-6
P03	4	10	8	26	26	10	0	7	En memoria	111-1 212-2 121-1 4444-44 515-5 131-1 6666-66 525-5 717-7 10-0
P04	6	8	2	16	24	0	-	-	Finalizado	4444-44 515-5 131-1 6666-66 525-5 717-7 10-0 251-2
P05	30	9	7	0	0	9	8	14	En ejecución	453-4 78522-785 965-9 421-4 555-5 234-2 965-9 421-4 741-7
P06	30	10	2	0	0	10	-	-	En espera	111-1 38-0 121-1 6542-65 54-0 852-8 50-0 4569-45 123-1 6542-65
P07	65	9	2	-	-	-	-	-	Fuera de sist.	74-0 38-0 121-1 852-8 50-0 4569-45 23-0 6542-65 238-2

Tiempo Medio de Espera: 7.33 Tiempo Medio de Retorno: 12.16

Se han producido 8 fallos de página en la ejecución de P04

M8	44-0	44-1	1-0	1-1	5-0	5-1	0-0	0-1
M9		5-0	5-1	66-0	66-1	7-0	7-1	2-0
	F	F	F	F	F	F	F	F

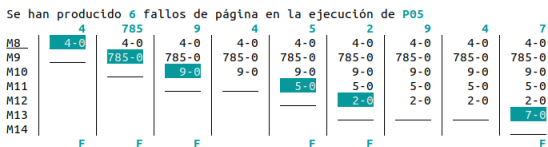
M0	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M=15
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

BM	P03	P05	M=15
	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14		

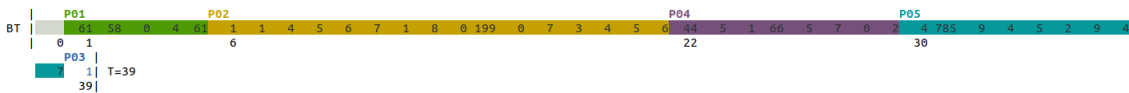
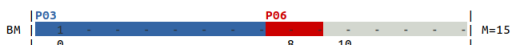
BT	P01	P02	P04	P05	T=30
	0 1 2 3 4 5 6	7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22	23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234	

T=39

FCFS/SJF - PAGINACIÓN - RELOJ - MEMORIA CONTINUA - NO REUBICABLE
T=39 Algoritmo utilizado = SJF Memoria del sistema = 1500 Tamaño de página = 100 Número de marcos = 15
El proceso P05 ha finalizado y ha transcurrido este tiempo: 9
P05 -> Tiempo Entrada: 30 Tiempo Salida: 39 Tiempo Restante: 0
Entra el proceso P06 a memoria a partir del marco 8
Entra el proceso P03 al procesador
Ref Tll Tej nMar Tesp Tret Trej Mini Mfin Estado Dirección-Página
P01 1 5 2 0 5 0 - - Finalizado 6123-61 5894-58 12-0 456-4 6123-61
P02 4 16 7 2 18 0 - - Finalizado 111-1 121-1 414-4 531-5 676-6 717-7 102-1 835-8 98-0 19999-199 45-0 722-7 389-3 444-4 521-5 666-6
P03 4 10 8 35 10 0 7 En ejecución 111-1 212-2 121-1 4444-44 515-5 131-1 6666-66 525-5 717-7 10-0
P04 6 8 2 16 24 0 - - Finalizado 4444-44 515-5 131-1 6666-66 525-5 717-7 10-0 251-2
P05 30 9 7 0 9 0 - - Finalizado 453-4 78522-785 965-9 421-4 555-5 234-2 965-9 421-4 741-7
P06 30 10 2 9 9 10 8 9 En memoria 111-1 38-0 121-1 6542-65 54-0 852-8 50-0 4569-45 123-1 6542-65
P07 65 9 2 - - - - - Fuera de sist. 74-0 38-0 121-1 852-8 50-0 4569-45 23-0 6542-65 238-2
Tiempo Medio de Espera: 10.33 Tiempo Medio de Retorno: 16.66

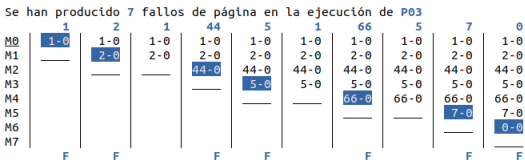


P03 P06
M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 M12 M13 M14 | M=15
1 - - - - - - - - - - - - - - -

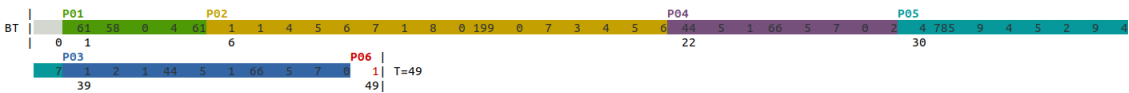
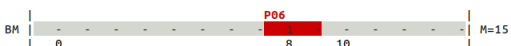


T=49

FCFS/SJF - PAGINACIÓN - RELOJ - MEMORIA CONTINUA - NO REUBICABLE
T=49 Algoritmo utilizado = SJF Memoria del sistema = 1500 Tamaño de página = 100 Número de marcos = 15
El proceso P03 ha finalizado y ha transcurrido este tiempo: 10
P03 -> Tiempo Entrada: 39 Tiempo Salida: 49 Tiempo Restante: 0
Entra el proceso P06 al procesador
Ref Tll Tej nMar Tesp Tret Trej Mini Mfin Estado Dirección-Página
P01 1 5 2 0 5 0 - - Finalizado 6123-61 5894-58 12-0 456-4 6123-61
P02 4 16 7 2 18 0 - - Finalizado 111-1 121-1 414-4 531-5 676-6 717-7 102-1 835-8 98-0 19999-199 45-0 722-7 389-3 444-4 521-5 666-6
P03 4 10 8 35 45 0 - - Finalizado 111-1 212-2 121-1 4444-44 515-5 131-1 6666-66 525-5 717-7 10-0
P04 6 8 2 16 24 0 - - Finalizado 4444-44 515-5 131-1 6666-66 525-5 717-7 10-0 251-2
P05 30 9 7 0 9 0 - - Finalizado 453-4 78522-785 965-9 421-4 555-5 234-2 965-9 421-4 741-7
P06 30 10 2 19 19 10 8 9 En ejecución 111-1 38-0 121-1 6542-65 54-0 852-8 50-0 4569-45 123-1 6542-65
P07 65 9 2 - - - - - Fuera de sist. 74-0 38-0 121-1 852-8 50-0 4569-45 23-0 6542-65 238-2
Tiempo Medio de Espera: 12.00 Tiempo Medio de Retorno: 20.00



P06
M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 M12 M13 M14 | M=15
1 - - - - - - - - - - - - - - -



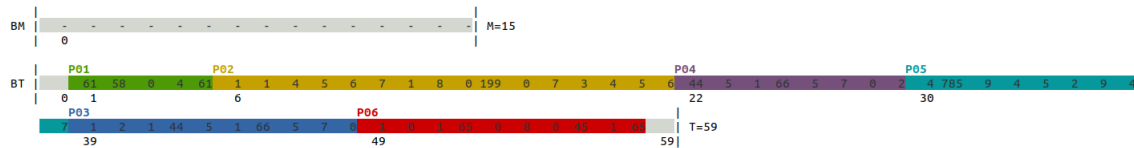
T=59

FCFS/SJF - PAGINACIÓN - RELOJ - MEMORIA CONTINUA - NO REUBICABLE
T=59 Algoritmo utilizado = SJF Memoria del sistema = 1500 Tamaño de página = 100 Número de marcos = 15
El proceso **P06** ha finalizado y ha transcurrido este tiempo: 10
P06 -> Tiempo Entrada: 49 Tiempo Salida: 59 Tiempo Restante: 0
Ref Tll Tej nMar Tesp Tret Trej Mini Mfin Estado Dirección-Página
P01 1 5 2 0 5 0 - - Finalizado 6123-61 5894-58 12-0 456-4 6123-61
P02 4 16 7 2 18 0 - - Finalizado 111-1 121-1 414-4 531-5 676-6 717-7 102-1 835-8 98-0 19999-199 45-0 722-7 389-3 444-4 521-5 666-6
P03 4 10 8 35 45 0 - - Finalizado 111-1 212-2 121-1 4444-44 515-5 131-1 6666-66 525-5 717-7 10-0
P04 6 8 2 16 24 0 - - Finalizado 4444-44 515-5 131-1 6666-66 525-5 717-7 10-0 251-2
P05 30 9 7 0 9 0 - - Finalizado 453-4 78522-785 965-9 421-4 555-5 234-2 965-9 421-4 741-7
P06 30 10 2 19 29 0 - - Finalizado 111-1 38-0 121-1 6542-65 54-0 852-8 50-0 4569-45 123-1 6542-65
P07 65 9 2 - - - - Fuera de sist. 74-0 38-0 121-1 852-8 50-0 4569-45 23-0 6542-65 238-2
Tiempo Medio de Espera: 12.00 Tiempo Medio de Retorno: 21.66

Se han producido 8 fallos de página en la ejecución de **P06**

M8	1-0	0	1-1	65-0	65-0	65-1	0-0	0-1	1-0	65
M9	0-0	0-0	0-0	0-1	0-1	8-0	8-1	45-0	45-1	65-0
	F	F	F	F	F	F	F	F	F	F

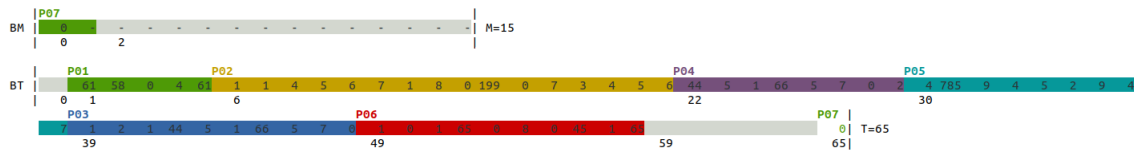
M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 M12 M13 M14 | M=15



T=65

FCFS/SJF - PAGINACIÓN - RELOJ - MEMORIA CONTINUA - NO REUBICABLE
T=65 Algoritmo utilizado = SJF Memoria del sistema = 1500 Tamaño de página = 100 Número de marcos = 15
Entra el proceso **P07** a memoria a partir del marco 0
Entra el proceso **P07** al procesador
Ref Tll Tej nMar Tesp Tret Trej Mini Mfin Estado Dirección-Página
P01 1 5 2 0 5 0 - - Finalizado 6123-61 5894-58 12-0 456-4 6123-61
P02 4 16 7 2 18 0 - - Finalizado 111-1 121-1 414-4 531-5 676-6 717-7 102-1 835-8 98-0 19999-199 45-0 722-7 389-3 444-4 521-5 666-6
P03 4 10 8 35 45 0 - - Finalizado 111-1 212-2 121-1 4444-44 515-5 131-1 6666-66 525-5 717-7 10-0
P04 6 8 2 16 24 0 - - Finalizado 4444-44 515-5 131-1 6666-66 525-5 717-7 10-0 251-2
P05 30 9 7 0 9 0 - - Finalizado 453-4 78522-785 965-9 421-4 555-5 234-2 965-9 421-4 741-7
P06 30 10 2 19 29 0 - - Finalizado 111-1 38-0 121-1 6542-65 54-0 852-8 50-0 4569-45 123-1 6542-65
P07 65 9 2 0 9 0 1 En ejecución 74-0 38-0 121-1 852-8 50-0 4569-45 23-0 6542-65 238-2
Tiempo Medio de Espera: 10.28 Tiempo Medio de Retorno: 18.57

P07
M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 M12 M13 M14 | M=15
0 -



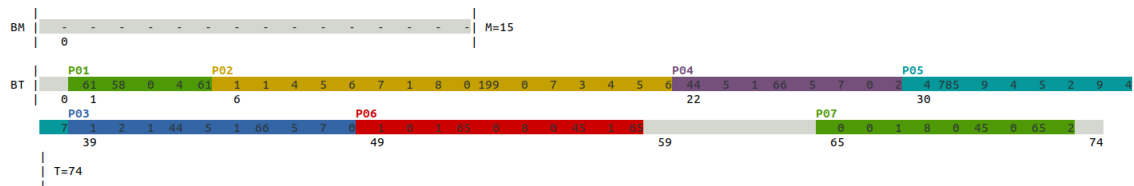
T=74

FCFS/SJF - PAGINACIÓN - RELOJ - MEMORIA CONTINUA - NO REUBICABLE
T=74 Algoritmo utilizado = SJF Memoria del sistema = 1500 Tamaño de página = 100 Número de marcos = 15
El proceso **P07** ha finalizado y ha transcurrido este tiempo: 9
P07 -> Tiempo Entrada: 65 Tiempo Salida: 74 Tiempo Restante: 0
Ref Tll Tej nMar Tesp Tret Trej Mini Mfin Estado Dirección-Página
P01 1 5 2 0 5 0 - - Finalizado 6123-61 5894-58 12-0 456-4 6123-61
P02 4 16 7 2 18 0 - - Finalizado 111-1 121-1 414-4 531-5 676-6 717-7 102-1 835-8 98-0 19999-199 45-0 722-7 389-3 444-4 521-5 666-6
P03 4 10 8 35 45 0 - - Finalizado 111-1 212-2 121-1 4444-44 515-5 131-1 6666-66 525-5 717-7 10-0
P04 6 8 2 16 24 0 - - Finalizado 4444-44 515-5 131-1 6666-66 525-5 717-7 10-0 251-2
P05 30 9 7 0 9 0 - - Finalizado 453-4 78522-785 965-9 421-4 555-5 234-2 965-9 421-4 741-7
P06 30 10 2 19 29 0 - - Finalizado 111-1 38-0 121-1 6542-65 54-0 852-8 50-0 4569-45 123-1 6542-65
P07 65 9 2 0 9 0 - - Finalizado 74-0 38-0 121-1 852-8 50-0 4569-45 23-0 6542-65 238-2
Tiempo Medio de Espera: 10.28 Tiempo Medio de Retorno: 19.85

Se han producido 7 fallos de página en la ejecución de **P07**

M0	0-0	0-0	0-1	8-0	8-1	45-0	45-0	45-1	2-0
M1	1-0	1-0	1-1	0-0	0-1	0-1	65-0	65-1	
	F		F	F	F	F	F	F	

M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 M12 M13 M14 | M=15



Finalmente se muestra un resumen de los tiempos de espera y de retorno de cada proceso, así como su número de fallos de paginación. En la parte superior, se definen estos conceptos por si hubiera dudas sobre qué abarcan. También aparecen los tiempos medios de espera y retorno, los cuales se han ido recalculando en cada instante a partir de la información de los procesos.

RESUMEN FINAL

T.Espera -> Tiempo que el proceso no ha estado ejecutándose en la CPU desde que entra en memoria 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
 Fallos Pág. -> Número de fallos de página que han ocurrido en la ejecución de cada proceso

Proc.	T.Espera	Inicio/Fin	T.Retorno	Fallos Pág.
P01	0	1/6	5	5
P02	2	4/22	18	12
P03	35	4/49	45	7
P04	16	6/30	24	8
P05	0	30/39	9	6
P06	19	30/59	29	8
P07	0	65/74	9	7

Tiempo total transcurrido en ejecutar todos los procesos: 74
 Media tiempo espera de todos los procesos: 10.28
 Media tiempo retorno de todos los procesos: 19.85

5. MEJORAS REALIZADAS

A continuación se listarán las mejoras realizadas sobre el script **P - 05180 - FCFS-SJF-Pag-SegOp-C-NR**, de las implementaciones de 2021-2022.

5.1. Ejecución

- Implementado algoritmo de planificación SJF (el anterior trabajo solo hacía FCFS).
- Cambiada Segunda Oportunidad a Reloj, pero se ha seguido dejando como opción para el algoritmo de reemplazo.
- Reescrita casi por completo la función principal del algoritmo (antes 'FCFS', ahora 'ejecucion'). Se ha acortado muchísimo, ya no se encuentra dividida la ejecución del primer proceso de la del resto (originaba bastantes problemas). Se han creado funciones auxiliares para inicializar las variables y gestionar la finalización de un proceso.
- Modificadas todas las funciones de la cola ('mueveCola', 'actualizaCola'...) porque utilizaban procedimientos poco eficientes.
- El número de fallos de paginación ahora es correcto. Antes, se calculaban en una función de NRU (siendo mi script base de Segunda Oportunidad), pero tampoco se llegaban a visualizar nunca debido a que el resumen final estaba comentado y no se mostraba después de finalizar la ejecución.
- El modo de ejecución 4 'Completo (solo resumen)' ahora funciona correctamente y lo manda todo a los informes (antes se visualizaba la tabla de marcos y no mostraba el resumen al final).

5.2. Eventos

- Solucionado error de instantes de tiempo sin procesos. Si había un hueco de tiempo entre dos procesos en el que no se ejecutaba nada, todo empezaba a fallar y se detenía la ejecución por completo.
- Ahora también se imprimen los eventos intermedios. Con “intermedios” me refiero a que tienen lugar mientras hay otro proceso ejecutándose (como los eventos de entrada a memoria). La ejecución del código anterior estaba planteada de manera que, una vez entraba un proceso, no ocurría nada más hasta su finalización (los “eventos importantes” eran solo los de entrada y salida de la CPU). La notificación de que un proceso se había metido en memoria se mostraba por pantalla cuando el proceso ejecutándose finalizase, en lugar de detenerlo en el instante en el que realmente entraba y que se pudiera visualizar cómo había cambiado la situación del proceso ejecutándose (líneas de memoria y tiempo, páginas ejecutadas, coeficientes, etc.).
- Relacionado con lo anterior, ahora también se notifica qué proceso sigue ejecutándose, si se trata de un evento intermedio.
- Creadas las variables ‘logEventos’ y ‘logEventosBN’, que almacenan el texto de los eventos que han tenido lugar en un instante de tiempo. Almacenándolos así, se pueden mostrar por pantalla de forma ordenada en la función que se ha creado para eso (descrita en el apartado de volcados).

5.3. Volcados a pantalla/informes

- Nueva tabla de fallos de paginación. Más clara, se imprime más rápido y tiene unidades variables y saltos de línea si no cupiera. Los marcos ahora se corresponden con los del proceso (antes siempre empezaban en M0). Se imprime una línea inferior indicando si hay fallo (‘F’). La página que ha causado el fallo de paginación se sombrea. En segunda oportunidad también se sombrea si el bit ha cambiado a 1. En reloj, el bit será 1 si le apunta la manecilla y 0 en otro caso.
- La tabla de páginas y bandas de memoria y tiempo también se han modificado y se imprimen mucho más rápido. Ahora sus unidades son variables y, si no cabe el número de columnas correspondiente a una unidad, se hace un salto de línea y se guardan correctamente los márgenes.
- Al cambiar los eventos, se ha descubierto que la función ‘diagramaResumen’ no subrayaba bien las páginas que se habían ejecutado si se trataba de un evento por medio de un proceso ejecutándose (subrayaba solo la primera o todas), ahora ya se subrayan correctamente.
- Creada función ‘volcadoAPantalla’ que recoge todos los volcados y los muestra en función de qué modo de ejecución se esté utilizando (manual, automático...) y de si se trata o no de un evento importante).
- Los colores usados durante todo el script se han almacenado en variables para que sean más cómodos de manejar y evitar las secuencias de escape (aunque aún quedan algunas sueltas pero son mínimas).
- Corregidos errores volcados informes (caracteres erróneos, volcados sin sentido de versiones anteriores, no imprimía los mIni y mFin...). Cambiada también la forma en la que se envían las tablas al informeBN (antes “describía”, por ejemplo: ‘Tejec: 4’, ahora se envía como por pantalla).

5.4. Ficheros

- Incluida opción 7 de entrada en el menú: “Aleatorio Rangos Total” y sus respectivas funcionalidades.
- Cambios en la organización de los directorios y nombres de ficheros para que sigan la estructura planteada en el fichero de correcciones (FDatos, FLast, FRangos, etc.).
- Las preguntas de dónde guardar datos/rangos ahora se hacen siempre antes de mostrar los datos.
- Si no existen los ficheros de datos/rangos de última ejecución o DatosRangosAleatorioTotal.txt, se informa y se sale del programa. Antes se procedía a la ejecución del algoritmo y, lógicamente, detenía el script abruptamente.
- Modificado el fichero de ayuda. Ahora es más completo, ordenado e incluye el nuevo método de entrada ‘Aleatorio total’. Se han eliminado las explicaciones obsoletas acerca de los tiempos de ejecución como parámetro, puesto que, ya desde implementaciones anteriores, el tiempo de ejecución de un proceso es determinado por su número de páginas.

5.5. Código

- Se ha indentado el código por completo para mejorar la navegación por él. Además, si se trabaja en entornos como Visual Studio Code, se pueden minimizar las funciones y esto ayuda mucho a la hora de visualizarlo. También se ha reorganizado y ordenado, dentro de cada categoría (entrada de datos, algoritmo...), con el orden lógico de ejecución y también orden de prioridad (las principales van por delante de las auxiliares).
- Toda función cuenta con un par o varias líneas descriptivas que explican su propósito. Si su funcionamiento depende de los argumentos que se les han pasado, también informa sobre qué resultado tiene cada argumento. Adicionalmente se han marcado las funciones con “function”, cuya utilización no es necesaria en bash, pero es útil a la hora de buscarlas con Ctrl+F, para diferenciar la función original de sus llamadas.
- Eliminadas las muchas líneas comentadas, variables repetitivas y funciones inutilizadas. Cambiados nombres de variables y funciones para que sean más descriptivos.
- Acortadas bastante las funciones de entrada de datos, creadas funciones auxiliares que sirven para más de un modo de entrada.
- Las funciones de lectura de datos/rangos desde ficheros eran muchas y se repetían. Se han fusionado en ‘leeDatosFichero’ y ‘leeRangosFichero’. Para diferenciar de si se tratan de los ficheros de última ejecución o no, se utilizan parámetros.
- Si se visualiza la ayuda no se sale del script sino que vuelve al menú inicial. También se ha añadido una tercera opción para salir directamente.

6. MEJORAS POR REALIZAR

- Dibujar el reloj en el script para sustituir a la tabla de fallos de página.
- Mejorar robustez en la entrada de datos. Que se compruebe mejor que no hay errores en el fichero y que dé más pistas sobre qué se ha introducido mal (en la mayoría de los casos solo se comprueba que es un número no negativo).
- Adicionalmente, acortar las funciones de entrada manual para rangos/datos o dividir las en subfunciones para que sea más navegable y modular.
- Unidades variables para las columnas de 'diagramaResumen' (la tabla que muestra los procesos, tiempos de llegada, etc.), que se imprima más rápido si es posible y que si no caben las direcciones de página se respete el margen de un espacio en la siguiente línea.
- Fusionar algunas funciones que actúan similar como las de 'imprimeProcesos...', 'aleatorioEntre...'. También hay algunas pocas variables que representan los mismos datos pero son diferentes y podrían fusionarse, mayoritariamente las relativas a volcados a pantalla. Tanto para las funciones como para las variables, se indica en algunos comentarios del script si son candidatas para fusionar con otra.
- Si el nombre del fichero que se ha escogido para guardar los datos/rangos existe, que se informe al usuario y se le pregunte si quiere modificarlo.
- Añadir muchos más colores de procesos (podría modificarse el vector 'colorines' con los colores de la paleta de 256).

7. CONCLUSIÓN

Esta práctica ha resultado útil para desarrollar y entender mejor los conceptos planteados en la asignatura acerca de la gestión de procesos y memoria en un SO. Personalmente, me ha ayudado a comprender mejor las cuestiones relacionadas con la paginación y la memoria virtual, a pesar de que al principio suene todo un poco complejo, luego se ve muy claro.

Se trata de una práctica a la que hay que dedicarle, como bien nos indican continuamente, horas y horas. El primer reto al que te enfrentas es el de elegir y entender el código sobre el que vas a realizar las modificaciones. En mi caso, tuve que limpiar casi todo el código. Es por esto por lo que he intentado dejar el script lo más claro y organizado posible (lo he reducido unas 2000 líneas aun habiendo añadido muchas funciones, por lo que serán más en realidad), puesto que ha sido mil veces más cómodo tanto para mí como espero que para la próxima persona que trabaje con él.

Los primeros días de trabajo son frustrantes puesto que es algo completamente nuevo y, antes de ponerte a hacer cambios, primero tienes que entenderlo y localizar dónde está cada cosa, lo que conlleva bastante tiempo y parece que no se avanza. Sin embargo, una vez empiezas a tenerlo todo situado y a resolver errores, lo vas viendo más claro.

Se puede concluir por lo tanto que se trata de una práctica larga e intrincada, pero que ayuda bastante a reforzar los conceptos de la asignatura y acerca al alumno a trabajar con códigos de mayor extensión y complejidad.

8. REFERENCIAS

Como se ha indicado anteriormente, el código base sobre el que se han realizado las modificaciones ha sido la implementación de 2021-2022: **P - 05180 - FCFS-SJF-Pag-SegOp-C-NR**, por César Rodríguez Villagrà, con Rodrigo Pérez Ubierna como autor previo a él.

Para los conceptos explicados durante el informe, me he apoyado en los apuntes de la asignatura de Sistemas Operativos por José Manuel Sáiz Díez y las modificaciones han sido realizadas siguiendo las instrucciones del profesor de la asignatura, el cual me ha ido indicando las mejoras a realizar.