

Prácticas de Sistemas Operativos

Informe de prácticas

Práctica de Control.

FCFS, fijas y no iguales, mejor

Código de grupo: 104

Alumnos:

1.- Jorge Navarro González

2.- Mario Ubierna San Mamés

Índice de contenido

<u>1.- Algoritmo.....</u>	<u>3</u>
<u>2.- Ejercicios a mano.....</u>	<u>4</u>
<u>Ejercicio 1: Ejercicio Sencillo.....</u>	<u>4</u>
<u>Ejercicio 2: Ejercicio Complejo.....</u>	<u>7</u>
<u>3.- Capturas de pantalla.....</u>	<u>11</u>
<u>4.- Cambios realizados sobre el SCRIPT del año pasado.....</u>	<u>17</u>

1.- Algoritmo

Nuestro algoritmo seleccionado es FCFS particiones fijas y no iguales ajuste al mejor.

FCFS (First Come First Served) significa que el primero que llega es el primero en ser enviado a la CPU, al ser no apropiativo, hasta que no se termina de ejecutar un proceso en la CPU, no comienza la ejecución del siguiente proceso. Este algoritmo es muy eficaz si los trabajos cortos se encuentran al principio, ya que en caso contrario, si los procesos largos están al principio entonces los procesos cortos son penalizados.

Particiones fijas y no iguales, principalmente significa que la memoria principal la vamos a dividir en una serie de particiones, las cuales tienen un tamaño diferente. Al utilizar particiones fijas, su tamaño no varía a lo largo de la ejecución del programa, ya que como su propio nombre indica se mantienen fijas durante la duración del mismo.

Finalmente en memoria realizamos el ajuste al mejor, esto quiere decir que el proceso se va a introducir en la partición que desperdicie menos memoria, permitiéndonos así que la fragmentación interna sea menor.

2.- Ejercicios a mano

Ejercicio 1: Ejercicio Sencillo

Para este ejercicio, realizaremos un ejercicio simple con dos procesos y tres particiones para ver como funcionaria lo básico del programa.

Tendremos inicialmente los siguientes procesos:

Procesos	Tiempo Llegada	Tiempo Ejecución	Memoria
p1	2	4	28
p2	1	6	16

Tendremos las siguientes 3 particiones en las cuales aún no hay ningún proceso metido en memoria:

Partición1=(50)	Partición2=(30)	Partición3=(20)
Vacía	Vacía	Vacía

Lo primero que hace nuestro algoritmo y que también debemos realizar a papel es ordenar los procesos por tiempo de llegada, en este caso:

Procesos	Tiempo Llegada	Tiempo Ejecución	Memoria
p2	1	6	16
p1	2	4	28

Las particiones aún siguen vacías ya que aún no hemos empezado a introducir ningún proceso:

Partición1=(50)	Partición2=(30)	Partición3=(20)
Vacía	Vacía	Vacía

Ahora vamos a empezar a meter cada uno de los procesos en sus respectivas particiones:

Tiempo igual a 1

En tiempo igual a 1 llega p2 que como ocupa 28 se meterá en la partición más pequeña en la que quepa, ya que estamos realizando el ejercicio con ajuste mejor.

Procesos	Tiempo Llegada	Tiempo Ejecución	Memoria
p2	1	6	16
p1	2	4	28

Particiones:

Partición1=(50)	Partición2=(30)	Partición3=(20)
Vacía	p2	Vacía

Tiempo igual a 2

Entra el proceso 1 a memoria, pero aún no se ejecuta ya que como es FCFS, hay que esperar a que termine el proceso que se encuentra ahora en ejecución. Y se mete en la partición de 20 ya que tiene que entrar en la más pequeña de las que quepa.

Procesos	Tiempo Llegada	Tiempo Ejecución	Memoria
p2	1	6	16
p1	2	4	28

Particiones:

Partición1=(50)	Partición2=(30)	Partición3=(20)
Vacía	p2	p1

Recordamos que el proceso que está en ejecución es el p2 mientras que p1 simplemente ha entrado en memoria pero no está ejecutándose en ningún momento.

Tiempo igual 7

En este instante ya ha finalizado el p2 por tanto queda en memoria p1 que cambiará su estado a ejecución, por tanto empezará a ejecutar los 4 tiempos que tiene.

Procesos	Tiempo Llegada	Tiempo Ejecución	Memoria
p2	1	6	16
p1	2	4	28

Particiones:

Partición1=(50)	Partición2=(30)	Partición3=(20)
Vacía	Vacía	p1

P1 empieza a ejecutarse en CPU.

Tiempo igual a 11

En este instante de tiempo p1 ya se ha ejecutado por completo y por tanto sale ya de memoria, y por tanto al finalizar p1 acaba la ejecución del programa.

Procesos	Tiempo Llegada	Tiempo Ejecución	Memoria
p2	1	6	16
p1	2	4	28

Particiones:

Partición1=(50)	Partición2=(30)	Partición3=(20)
Vacía	Vacía	Vacía

Ejercicio 2: Ejercicio Complejo

Para este segundo ejercicio, realizaremos el mismo proceso que en el ejercicio anterior, a diferencia de que en esta caso vamos a realizar el ejemplo con 5 procesos. Por otro lado vamos a usar 4 particiones.

Inicialmente tenemos los siguientes procesos:

Procesos	Tiempo Llegada	Tiempo Ejecución	Memoria
p1	1	3	9
p2	2	11	6
p3	4	5	17
p4	6	8	20
p5	1	2	50

Inicialmente tenemos las siguientes particiones, en las cuales no hay ningún proceso cargado en memoria:

Partición1=(60)	Partición2=(50)	Partición3=(25)	Partición4=(15)
Vacía	Vacía	Vacía	Vacía

Lo primero que debemos hacer en el algoritmo FCFS, es ordenar la tabla de los procesos, es decir, ordenar los procesos por orden de llegada. Una vez realizado esto nos quedaría una tabla similar a esta:

Procesos	Tiempo Llegada	Tiempo Ejecución	Memoria
p1	1	3	9
p5	1	2	50
p2	2	11	6
p3	4	5	17
p4	6	8	20

Tenemos las siguientes particiones, en las cuales no hay ningún proceso cargado en memoria:

Partición1=(60)	Partición2=(50)	Partición3=(25)	Partición4=(15)
Vacía	Vacía	Vacía	Vacía

Nuestro programa empieza la ejecución del algoritmo en el tiempo de llegada del primer proceso, es decir, en el tiempo de llegada de p1.

En Tiempo igual a 1:

Partición1=(60)	Partición2=(50)	Partición3=(25)	Partición4=(15)
Vacía	p5	Vacía	p1

Cuando en el tiempo real del programa es igual a 1, llegan dos procesos, el primero es p1 (como tiene una memoria de 9 y nuestro algoritmo es ajuste al mejor entonces se mete en la partición4) y el segundo proceso es p5 (tiene un tamaño de memoria igual a 50, por lo tanto se introduce en la partición que desperdicie menos memoria, en este caso es la partición2). Al estar llegar primero p1 es éste el que se ejecuta en la CPU.

En Tiempo igual a 2:

Partición1=(60)	Partición2=(50)	Partición3=(25)	Partición4=(15)
Vacía	p5	p2	p1

Cuando el tiempo de ejecución es igual a 2, llega un nuevo proceso a la memoria p2, aunque p2 tiene una memoria de 6 y la partición ideal en la que tendría que estar alojado este proceso es la partición4, al estar ocupada dicha partición por el proceso p1, utilizando el algoritmo FCFS ajuste al mejor, p2 se aloja en la partición3. Por otro lado continúa la ejecución de p1.

En Tiempo igual a 4:

Partición1=(60)	Partición2=(50)	Partición3=(25)	Partición4=(15)
p3	p5	p2	Vacía

En este instante de tiempo finaliza la ejecución de p1 y comienza la ejecución de p5. Además llega un nuevo proceso a la memoria, es el proceso p3. Dicho proceso tiene una memoria de 17, en la partición4 no cabe ya que dicha partición tiene un tamaño de 15, la partición2 y la partición3 están ocupadas, por lo que p3 solo se puede alojar en la partición1.

En Tiempo igual a 6:

Partición1=(60)	Partición2=(50)	Partición3=(25)	Partición4=(15)
p3	p4	p2	Vacía

En este momento p5 ha finalizado su ejecución y ha empezado la ejecución de p2, por otro lado ha llegado un nuevo proceso p4, el cual ocupa 20 de memoria. Por lo tanto solo cabe en la partición2 ya que en la partición4 no tiene el espacio suficiente, y el resto de particiones están ocupadas.

En el Tiempo igual a 17:

Partición1=(60)	Partición2=(50)	Partición3=(25)	Partición4=(15)
p3	p4	Vacía	Vacía

Cuando el tiempo de ejecución es igual a 17 finaliza la ejecución de p2 y comienza la ejecución de p3. Por otro lado p4 sigue esperando a que finalice el p3 para así poder comenzar su ejecución.

En el Tiempo igual a 22:

Partición1=(60)	Partición2=(50)	Partición3=(25)	Partición4=(15)
Vacía	p4	Vacía	Vacía

Finaliza la ejecución de p3 y comienza la ejecución de p4.

En el Tiempo igual a 30:

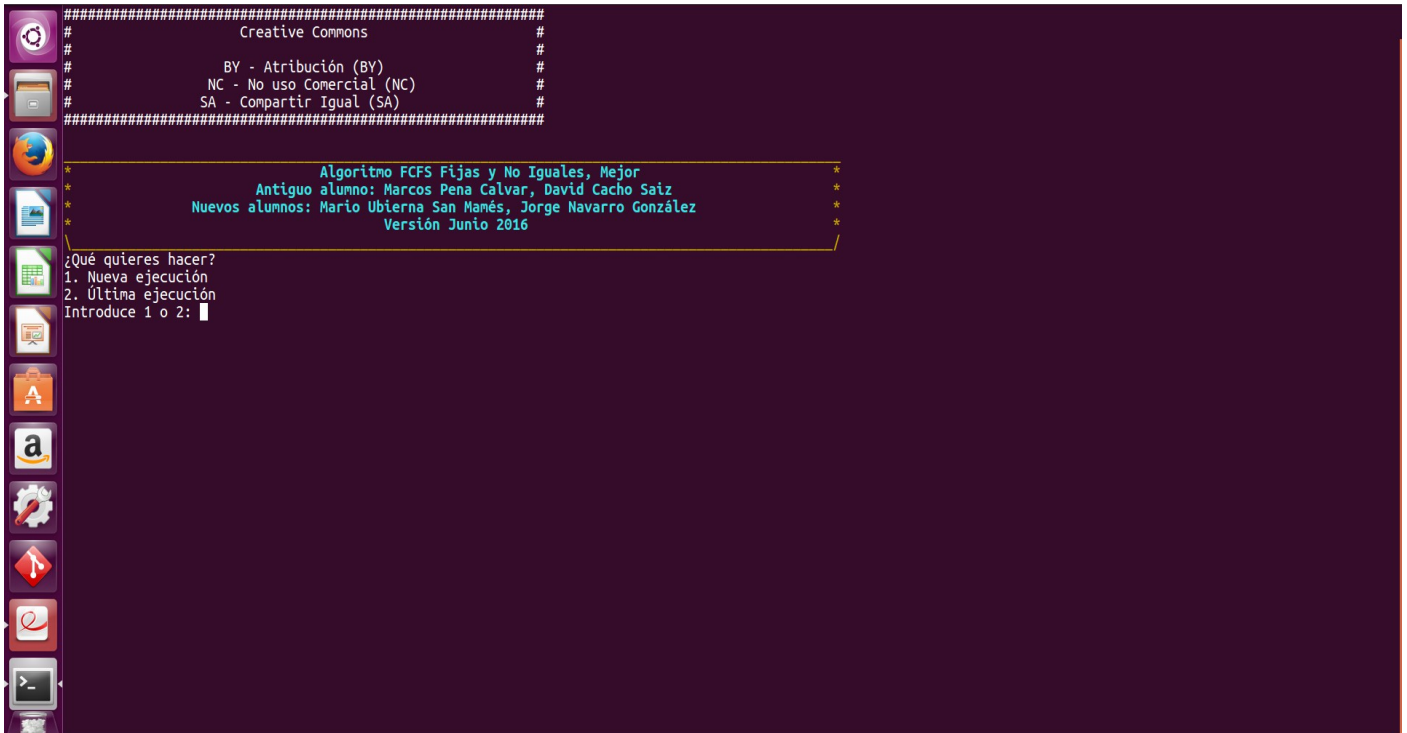
Partición1=(60)	Partición2=(50)	Partición3=(25)	Partición4=(15)
Vacía	Vacía	Vacía	Vacía

Finaliza la ejecución de p4, poniendo así fin al programa.

3.- Capturas de pantalla

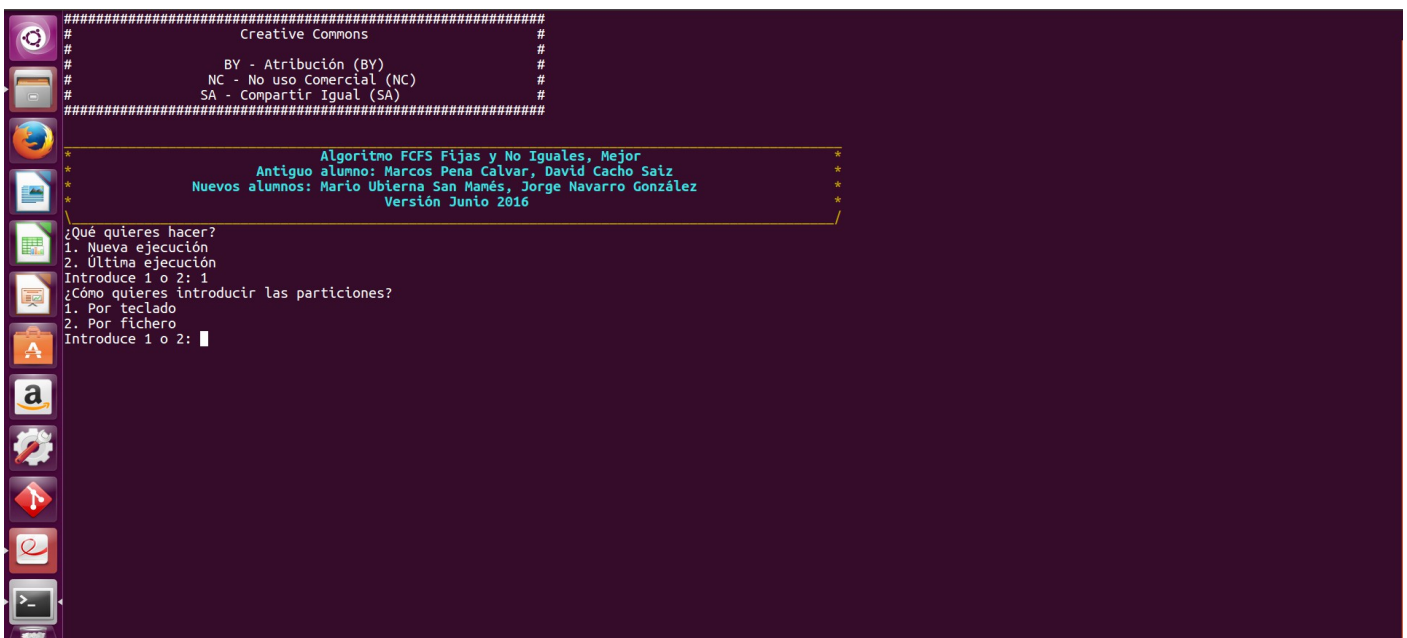
En este apartado vamos a realizar capturas de nuestro programa sobre el ejercicio realizado a mano número 2, además de enseñar también todo el menú con el fin de poder enseñar todas las opciones de las que disponemos.

Lo primero que tenemos en nuestro Script es un menú con nuestro algoritmo y los autores y la primera interacción con el usuario es que nos pide si queremos hacer una nueva ejecución, o por el contrario queremos ejecutar la anterior opción.



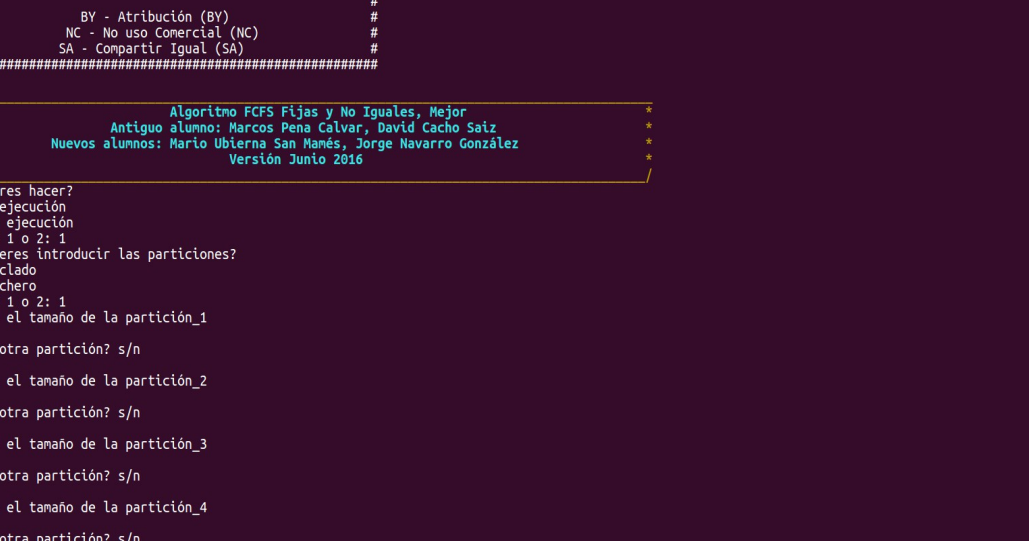
```
#####  
# Creative Commons #  
# #  
# BY - Atribución (BY) #  
# NC - No uso Comercial (NC) #  
# SA - Compartir Igual (SA) #  
#####  
  
* Algoritmo FCFS Fijas y No Iguales, Mejor *  
* Antiguo alumno: Marcos Pena Calvar, David Cacho Saiz *  
* Nuevos alumnos: Mario Ubierna San Mamés, Jorge Navarro González *  
* Versión Junio 2016 *  
/  
  
¿Qué quieres hacer?  
1. Nueva ejecución  
2. Última ejecución  
Introduce 1 o 2: |
```

Podremos seleccionar cualquiera de las dos, en caso de seleccionar la opción de última ejecución realizará directamente la última ejecución, si seleccionamos la nueva ejecución nos preguntará primero por las particiones, si queremos meterlas manualmente o por fichero.



```
#####  
# Creative Commons #  
# #  
# BY - Atribución (BY) #  
# NC - No uso Comercial (NC) #  
# SA - Compartir Igual (SA) #  
#####  
  
* Algoritmo FCFS Fijas y No Iguales, Mejor *  
* Antiguo alumno: Marcos Pena Calvar, David Cacho Saiz *  
* Nuevos alumnos: Mario Ubierna San Mamés, Jorge Navarro González *  
* Versión Junio 2016 *  
/  
  
¿Qué quieres hacer?  
1. Nueva ejecución  
2. Última ejecución  
Introduce 1 o 2: 1  
¿Cómo quieres introducir las particiones?  
1. Por teclado  
2. Por fichero  
Introduce 1 o 2: |
```

Como en el caso anterior si seleccionamos las particiones por fichero tendremos las particiones que vienen por defecto ya en el fichero, en caso de seleccionar las particiones desde teclado nos pedirá añadir los tamaños de las particiones, como en el ejercicio dos realizado pondremos cuatro particiones de 60, 50, 25 y 15.



```
jorge@Jorge-VB: ~/Sisop/TrabajoFinal/SCRIPT
#####
#                               #
#      Creative Commons         #
#                               #
#      BY - Atribución (BY)     #
#      NC - No uso Comercial (NC) #
#      SA - Compartir Igual (SA)  #
#                               #
#####

*                               *
*      Algoritmo FCFS Fijas y No Iguales, Mejor      *
*      Antiguo alumno: Marcos Pena Calvar, David Cacho Saiz      *
*      Nuevos alumnos: Mario Ubierna San Mamés, Jorge Navarro González      *
*                               *
*                               *

¿Qué quieres hacer?
1. Nueva ejecución
2. Última ejecución
Introduce 1 o 2: 1
¿Cómo quieres introducir las particiones?
1. Por teclado
2. Por fichero
Introduce 1 o 2: 1
Introduce el tamaño de la partición_1
60
¿Quieres otra partición? s/n
s
Introduce el tamaño de la partición_2
50
¿Quieres otra partición? s/n
s
Introduce el tamaño de la partición_3
25
¿Quieres otra partición? s/n
s
Introduce el tamaño de la partición_4
15
¿Quieres otra partición? s/n
s
```

Como podemos observar nos va pidiendo el tamaño de las particiones y una vez introducido nos preguntara si queremos meter otra partición más, lo siguiente que nos pedirá es si queremos introducir los procesos también a mano o por fichero, en nuestro caso lo haremos por fichero dándole los valores que tenemos en el ejercicio a mano.

```
Particion 1 60
Particion 2 50
Particion 3 25
Particion 4 15
Tipo de ejecución:
1. Manual
2. Por archivo
1
```

Estas serían las particiones y la opción para los procesos, si por archivo o manualmente.

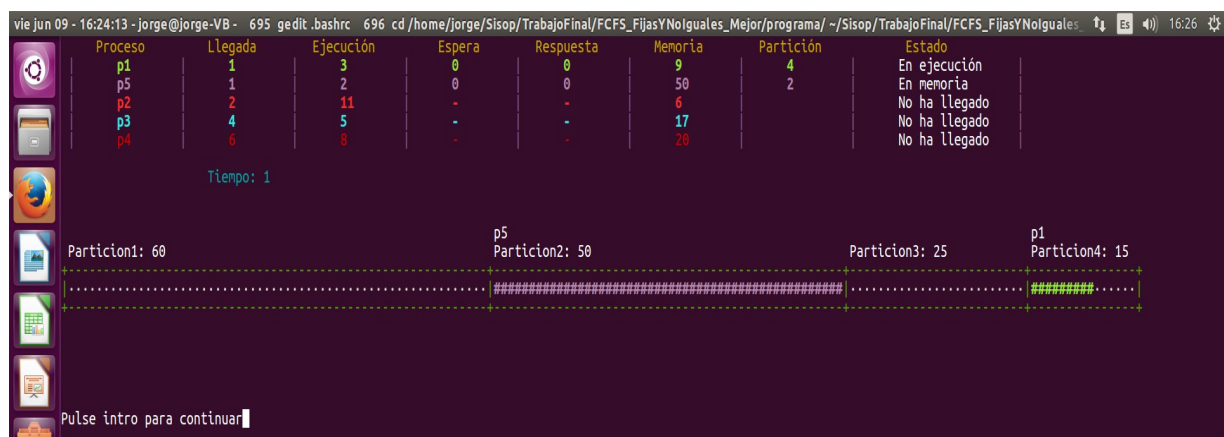
Para introducir los procesos a mano iríamos metiendo todos los datos y luego nos pondría una tabla arriba con los procesos ordenados por tiempo de llegada.

Proceso	T. Llegada	T.Ejecución	Memoria
p1	1	3	9
p5	1	2	50
p2	2	11	6
p3	4	5	17
p4	6	8	20

;Quieres más procesos? s/n

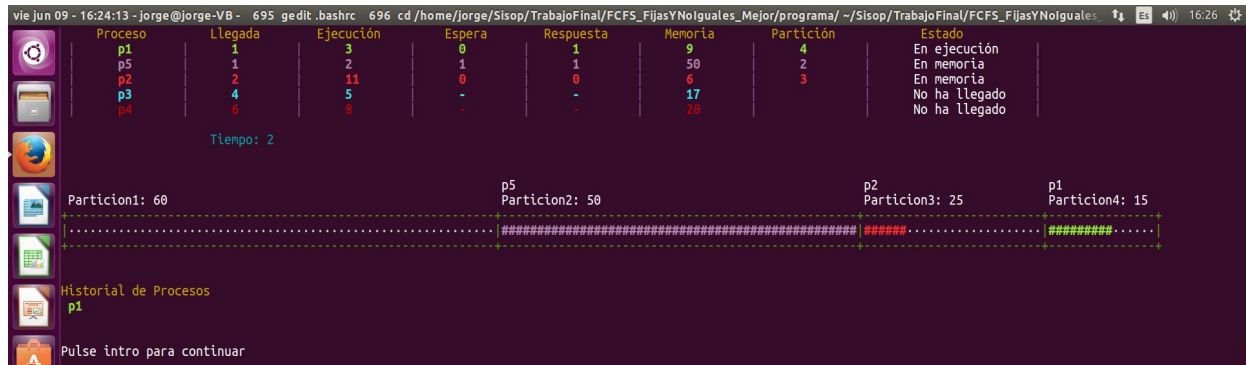
En este caso introduciremos que no queremos más procesos, como vemos ha ordenado los procesos por tiempo de llegada, para introducir los procesos nos pide el nombre, tiempo de llegada, tiempo de ejecución y memoria en orden como aparece en la tabla.

Lo siguiente es pasar ya a la ejecución. Lo primero que nos aparecerá será una tabla como la anterior pero esta vez dándole distintos colores a cada proceso de tal manera que luego abajo en la parte gráfica podamos distinguir a qué corresponde cada proceso.

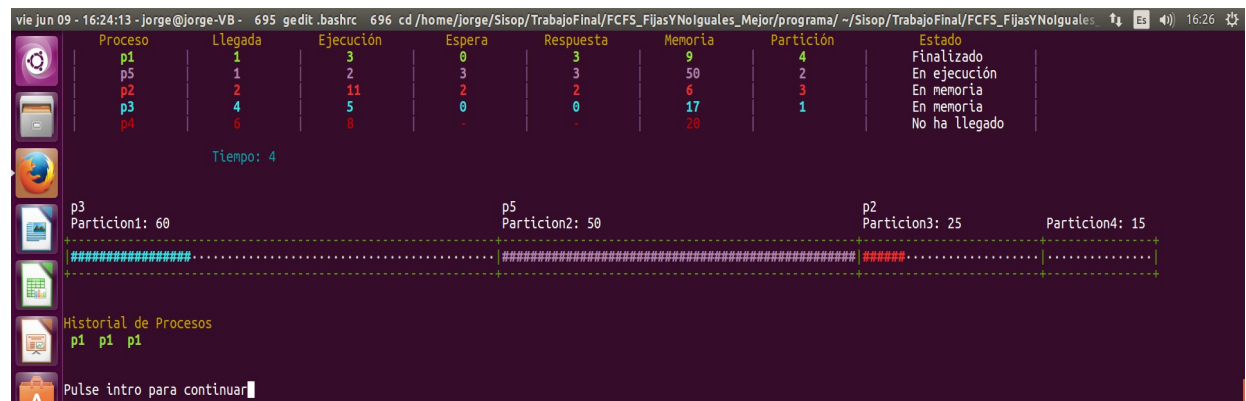


Como podemos ver, tenemos los procesos metidos en la partición correspondiente y tenemos el estado, para ver si esta en ejecución, en memoria, o si no ha llegado, también hay un estado más, que sería esperando, en caso de que haya llegado ya pero no pueda entrar aún en memoria. Como vemos el número de hastags se corresponde con el tamaño que ocupa la partición.

Ha pasado otra unidad de tiempo y vemos como p2 aunque ocupe 6 no puede entrar a la partición de 15 que es la más grande de todas, por tanto entra a la partición de 25 que es la siguiente más pequeña en la que puede entrar, como vemos abajo hay una barra de procesos con el avance que va teniendo el proceso y en que parte de la ejecución nos encontramos en ese momento.



Lo siguiente será en el siguiente cambio de estado, ya que no tenemos que estar dando enters cada unidad de tiempo sino cuando cambia el estado de algún proceso. Como podemos ver hemos avanzado dos unidades de tiempo, cuando se ha producido el cambio de estado de p3 entrando en memoria, que coincide con cuando p1 finaliza y p5 entra en ejecución. En el historial de procesos se ve como tenemos justo tres de p1 que es el tiempo de ejecución del mismo por tanto estamos justo en el momento en el que p1 acaba de terminar.

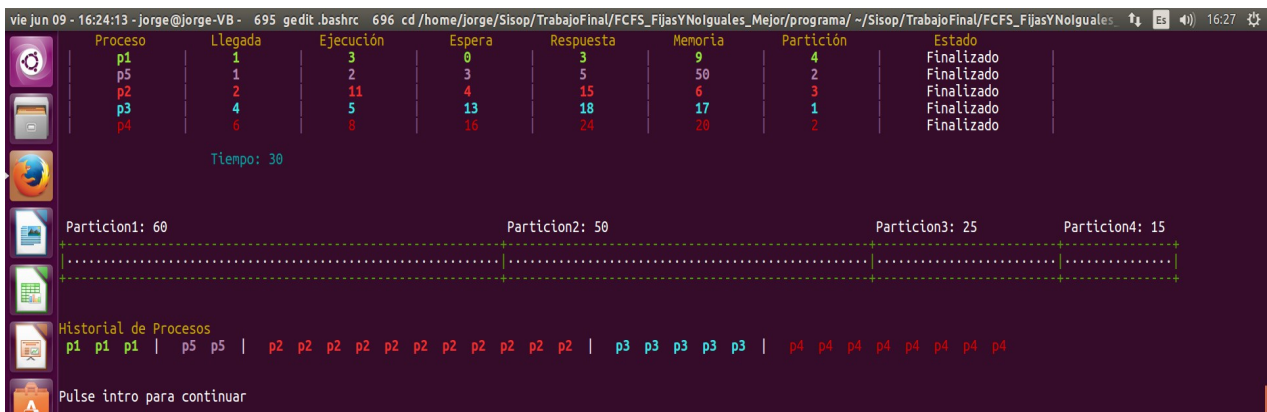


Lo siguiente será cuando se produzca otro cambio de estado.

Como vemos acaba de terminar ya p5 y se muestra en el historial de procesos, además coincide con p4 entrando a memoria, que como ocupa 20 pero la partición de 25 ya está ocupada, entra en la de 50.

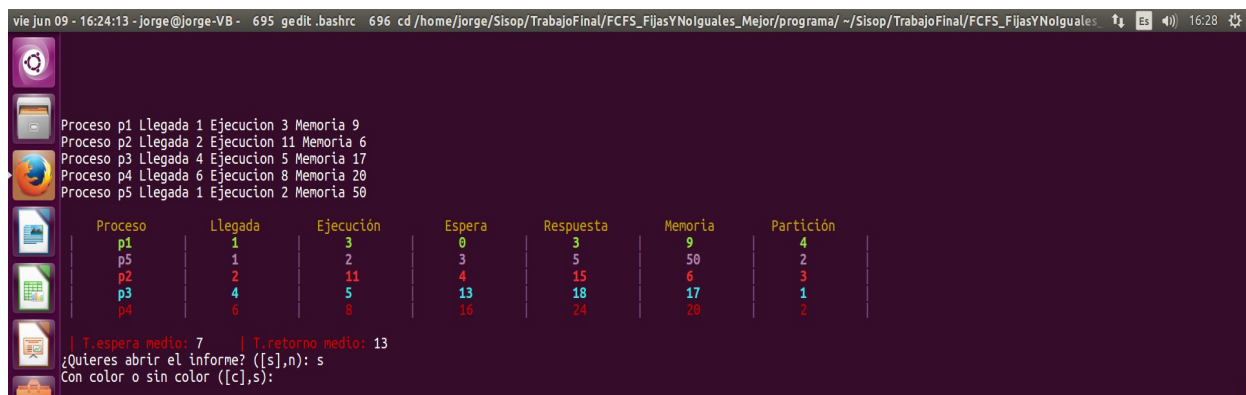
Como podemos ver hemos pasado del tiempo 6 a tiempo 17, que es cuando ya ha acabado p2 y por tanto cuando se ha producido un cambio de estado, lo siguiente según el ejercicio realizado a mano sería que acabara p3 que es el que se encuentra ahora en ejecución.

Como vemos p3 acaba de finalizar y solo quedaría que finalice p4 y salga de la memoria.

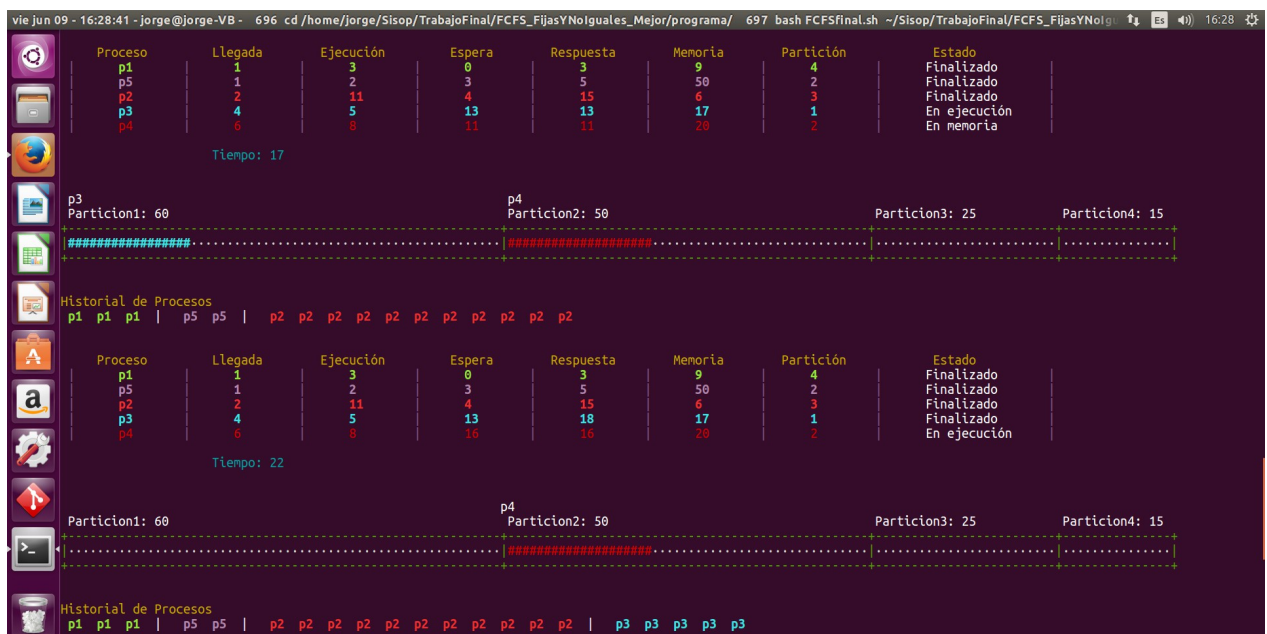


Ya ha salido p4 de la memoria y ha acabado todo de salir de memoria.

Ahora nos muestra una tabla con el resumen de toda la ejecución, con tiempo medio y tiempo de retorno.



Nos dirá si queremos abrir el informe con todas las pantallas que hemos pasado dando enters anteriormente y si queremos hacerlo en color o queremos dejarlo sin color. Pondré captura de la de color, la otra es lo mismo pero en blanco y negro.



Aquí tendríamos una parte de todo el informe con todos los pantallazos en color.

4.- Cambios realizados sobre el SCRIPT del año pasado

Nuestro script era FCFS con particiones fijas y no iguales y ajuste mejor, explicado ya al principio de este informe, en la página no había ningún script facilitado de este algoritmo pero si que teníamos uno parecido ya que era FCFS con particiones fijas y no iguales y ajuste al peor, aunque era un script parecido hemos tenido que realizar muchos cambios a dicho script ya que funcionaba bastante mal.

Uno de los cambios principales ha sido cambiar el algoritmo a ajuste mejor, aunque ese punto era una de las cosas que en el otro script estaba mejor hecha ya que si que hacia dicho ajuste, dicho esto tuvimos que cambiar toda la parte gráfica, ya que no hacia nada gráfico, lo poco que había de relleno de particiones, era algo parecido a lo que tenemos ahora con la diferencia que independientemente del tamaño del proceso, el script del año pasado siempre se rellenaba hasta la mitad, daba igual poner 1 de memoria en una partición de 80 que siempre se rellenaba hasta la mitad.

Además de tener que cambiar toda la parte de relleno de particiones, añadimos un historial de procesos que te va diciendo como van avanzando los procesos y así poder saber en todo momento cual es el proceso que acaba de ejecutarse en un instante de tiempo.

También añadimos respecto del script del año pasado que se pusieran colores por cada proceso, de tal manera que pudiera el usuario saber en todo momento donde va cada proceso con un golpe de vista, al ver los colores y poder relacionar cada color con un determinado proceso, en cuanto a los colores, también añadimos que pudiera el usuario abrir dos informes al final de toda la ejecución, uno con el informe en color, y otro con el informe en blanco y negro, de tal manera que el propio usuario pudiera elegir el que mas le conviene en cada momento.

Otro de los cambios importantes en nuestro script respecto al del año pasado, era el algoritmo de ordenación de procesos por tiempo de llegada, ya que si introducíamos por teclado 5 procesos p1, p2, p3, p4 y p5 con orden de llegada 1, 2, 2, 1, 0 respectivamente. Se guardaban en la tabla como p5-p4-p1-p3-p2, por tanto aunque si que estaban por tiempo de llegada, lo lógico era más bien que si hemos escrito antes el p1 que el p4, aunque tengan igual tiempo de llegada, salga antes el p1 no el p4.

También respecto del otro script hemos cambiado todos los tiempos de espera y de respuesta ya que estaban mal calculados y no te los daba además a tiempo real, sino que te ofrecía directamente el tiempo de espera final en cada caso y el tiempo de respuesta final.

Como resumen de todo lo añadido sería: los colores por cada proceso para que el usuario los pueda distinguir fácilmente, el ajuste mejor en vez de peor, el algoritmo de ordenación por tiempo de llegada, que pueda el usuario visualizar el informe en color o en blanco y negro y toda la parte gráfica del script, tanto las particiones, la manera en que se rellenan, que el año pasado lo hacía mal, como el historial de procesos que no estaba y lo hemos añadido, por último añadir que hemos cambiado también todos los tiempos de espera y respuesta para que estén bien calculados y a tiempo real.