



Sistemas operativos

Cambios realizados en Gestión de Procesos

Alumno:

1.- Omar Santos Bernabé  
[osb0005@alu.ubu.es](mailto:osb0005@alu.ubu.es)

# Índice de contenido

1.Código.....	3
1.1.SJF.....	3
1.2.ROUND-ROBIN.....	3
1.3.FCFS.....	3
1.4.PRIORIDADES.....	4
1.5.SRPT.....	4
2.Salida por pantalla.....	6
3.Creación de informes.....	7
4.Documentación.....	8
5.Posibles correcciones.....	9
5.1.SJF.....	9
5.2.ROUND-ROBIN.....	9
5.3.FCFS.....	9
5.4.PRIORIDADES.....	9
5.5.SRPT.....	10

# **1.Código**

En el código no existen grandes cambios puesto que en gran parte de los algoritmos se encontraba ya muy simple y que funcionase bien, a excepción de un par de algoritmos que en ninguno de los ejemplos recibidos funcionaban bien.

## ***1.1.SJF***

El funcionamiento de este algoritmo era el correcto. Los cambios o mejoras que he tenido que realizar han sido básicamente realizar partes del algoritmo en alguna función a parte para aclarar el código. He tenido que estructurar el código de manera que la salida por pantalla fuera la misma que para otros algoritmos y que quedaran bien claros los datos. Del mismo modo he tenido que ampliar la documentación sobre el código puesto que había trozos de código no muy bien comentado por lo que he añadido comentarios para aclarar algunas funciones o dar una idea al próximo alumno que lo obtenga para así no inducir al fallo o a la duda.

## ***1.2.ROUND-ROBIN***

Este algoritmo es en el que menos cambios he realizado porque su manera de funcionar era la mas óptima y la más correcta de las opciones que tenía. Su salida por pantalla puede que sea algo diferente que la de los otros algoritmos porque a la hora de intentar estructurarlo de la misma forma surgía un error en el código porque al estar separado por funciones completamente pues había funciones que dependían unas de otras y por lo que al cambiar alguna cosa en una repercutía en otras y así en todo el programa dejando de funcionar o funcionando con datos, valores o resultados que no serían los correctos. Por ese mismo motivo sobre este algoritmo he realizado una documentación más extensa y explicativa para que no quedara ninguna duda sobre su funcionamiento a pesar de ser diferente al resto.

## ***1.3.FCFS***

Este algoritmo funcionaba a pesar de tener algún error de programación y que estaba algo lio de entender el código. Tuve que estructurarlo de una forma que fuera entendible y que se pareciera al resto de los algoritmos para así todos estas implementados de forma parecida. Los arreglos que tuve que realizar en el código fueron mínimos porque si realizaba su función pero había momentos que dependiendo que datos o tipos de datos se introdujeran saltaba un mensaje y terminaba el algoritmo o simplemente se ejecutaba mal. Igual que para otros algoritmos tuve que documentar partes del código que no quedaban claras y estructurar el código introduciendo funciones para comprobar aspectos. Tuve que

introducir en esas funciones formas de realizar un control de errores para los datos que se introducían porque no estaba implementado y podías introducir datos con los que el programa dejaría de funcionar y no saltaba ningún error hasta que se ejecutaba y en ejecución aparecía el error en la línea, variable o lugar del código correspondiente.

## ***1.4.PRIORIDADES***

La funcionalidad de este algoritmo era en un principio la correcta a pesar de que tuve problemas para corregir algunas líneas de código referentes a la gestión de memoria. Sobre este algoritmo tuve que realizar varios cambios referente al sistema de datos, la forma de introducir los datos o de coger los datos de un fichero. Del mismo modo tuve problemas a la hora de guardar los datos puesto que en un principio de creaban diversos ficheros con nombres no muy explicativos y tuve que minimizar la creación de ficheros de datos dejando solamente dos, uno que guardara los datos temporales y otros que fuera el informe que se abriría al final.

Tuve que estructurar toda la salida por pantalla y el código para que fuera similar al de otros algoritmos y se entendieran todos del mismo modo. Tuve que realizar algunos comentarios más en el código porque no estaban muy claras algunas de las líneas del código ni lo que se conseguía con ellas. De la misma forma que para otros algoritmos a pesar de que los dos de prioridades ya tuvieran algunas implementadas tuve que introducir más funciones que en definitiva lo que hacían era comprobar el control de errores y estructurar el código en diferentes funciones para tener mayor claridad para ver las cosas que se hacen en cada una.

## ***1.5.SRPT***

Es el algoritmo que más trabajo me ha llevado porque solamente tenía dos ejemplos de los cuales partir. Uno de ellos estaba bien realizado pero en el momento que eliminaba la opción de memoria daba errores que no tenían sentido y en líneas en las que no había nada de los cuales alguno no sabía arreglar y encontrarle sentido. Como ese no lo conseguía arreglar deje de continuar con ese archivo y comencé a trabajar con el otro que no funcionaba como era debido. Tuve que realizar cambios en la ordenación de los procesos y en la forma de interpretar los datos y la funcionalidad básica del algoritmo esta resuelta. Del mismo modo que con otros algoritmos repartí algunos trozos de código en funciones para que así fuera más claro de ver que hacía cada cosa. También tuve que comentar dichas funciones y comentar trozos de código que en un principio no quedaban muy claros.

He tenido que estructurar gran parte del código para que su forma de pedir datos, introducirlos, leer datos de fichero y otras funcionalidades del código fueran iguales a otros códigos de otros algoritmos. Con el mismo fin he estructurado la salida de pantalla de un modo que sea semejante al resto de algoritmos para que no hubiera una

gran diferencia entre la lectura de unos y de otros algoritmos y que viendo uno podría entenderse el funcionamiento de todos porque están igual implementados cada uno con sus necesidades. Tuve que introducir en el código un control de errores para que e el caso de meter los datos de manera manual no introdujeras datos inválidos para algunos aspectos y que en el caso de hacerlo saliera un mensaje de error para que introdujeras un nuevo dato válido y para que el programa no continuara hasta que todos los datos introducidos fueran los correctos para cada cosa, si se piden números números o si se piden nombre no introducir nombres iguales.

## 2.Salida por pantalla

A la hora de realizar la salida por pantalla me base en uno de los dos algoritmos que mejor estaban hechos, el SJF, y que de mejor forma se veía el funcionamiento del programa. De ese modo lo primero que introduje en ese algoritmo fue la licencia del Creative Commons. En cuanto a salida por pantalla de este algoritmo es lo que realice sumando a esto alguna tabulación o espacio para que quedase claro que era cada cosa y que hacía cada parte.

Lo primero que hice con el resto de algoritmos fue introducir del mismo modo la licencia del Creative Commons y a continuación realizar una cabecera del algoritmo que indicaría la versión que tiene el algoritmo, su último autor más que nada porque introduciendo el de todos los autores anteriores salía una cabecera muy extensa y no se veía claramente algunas de las cosas.

A la hora de introducir los datos hice que todos los algoritmos siguieran las mismas pautas y viendo su ejecución fueran las de todos iguales. Primero nos pedía el numero de procesos y la forma de introducirlos y una vez comenzaba a pedir los datos nos aparecía una tabla que colocaba los datos para un mayor entendimiento y que supiéramos que datos habían sido introducidos ya para evitar confusiones en el caso de que el numero de datos a introducir fuera muy largo.

Una vez se introducían los datos nos aparecía un pequeño listado tanto si introducíamos los datos de forma manual o por fichero. En ese listado podíamos ver los datos introducidos en el orden que se habían introducido y una vez terminaba la ejecución nos aparecía la salida del algoritmo, el orden de salida junto con algunos datos más para cada proceso, para todos los algoritmos la salida final era la misma para que de ese modo se pudiera ver claramente que el de más arriba es el primer en salir y el de más abajo el último.

Finalmente en la salida por pantalla a todos nos daba la opción de de abrir un informe que se iba realizando de forma no visible para el usuario.

Esta era la estructura básica de salida por pantalla de todos los algoritmos.

### 3.Creación de informes

La gran parte de los algoritmos no creaban un informe o un fichero de texto al final de su ejecución en el que se guardaran los datos. He creado un informe de cada uno de los algoritmos que se realiza a la vez que se está ejecutando el propio algoritmo.

En el se guarda una pequeña cabecera que incluye la licencia del creative commons y en la que aparece al igual que en la cabecera de la salida por pantalla solamente el nombre del ultimo alumno que ha realizado la modificación sobre dicho algoritmo.

A continuación de dicha cabecera se guardan los datos que se han utilizado, que van guardándose a medida que el algoritmo que introducimos los datos en la ejecución si estamos ejecutando de forma manual o el algoritmo los coge del fichero si no estamos introduciendo los datos de forma manual.

En el informe aparece en forma de lista según el orden en el que han sido introducidos al igual que pasa en la salida por pantalla del cada algoritmo justo antes de ejecutarse este.

Lo último que aparece en el informe es la salida del algoritmo, los procesos en el orden que terminan de ejecutarse junto con varios datos asociados a ellos igual que en la salida por pantalla de la ejecución.

En los informes también hay una diferencia respecto al Round-Robin puesto que es uno de los algoritmos que ya tenían implementada la creación de un informe. En el informe de este algoritmo sale básicamente lo mismo que en la salida por pantalla al ejecutar dicho algoritmo, lo que pasa que esta implementado con una pequeña descripción de lo que pasa en algunos instantes de tiempo que es relevante para la ejecución de dicho algoritmo.

## 4.Documentación

La documentación de muchos de los trabajos realizados que teníamos como ejemplos era bastante pobre en algunos caso y muy caótica en otros.

Algunas de esas documentaciones tenían incluso ejemplos de la realización de los algoritmos de procesos combinados con memoria y ningún ejemplo de los algoritmos de procesos por separado por lo que decidí realizar unas nuevas documentaciones empezando las de cero.

Estas documentaciones las estructuraba en diferentes partes;  
En la primera parte definía el algoritmo de una forma sencilla para que el alumno pudiera entender el funcionamiento básico de cada uno de los algoritmos.

A continuación realizaba ejercicios de forma manual explicando el funcionamiento del algoritmo. Estos ejercicios intentaban ser ejercicios suficientemente complicados para que se pudiera ver que e algoritmo funciona correctamente. En primer lugar realizaba un ejercicio con pocos procesos y que tuviera cierta complejidad y a continuación realizaba otro ejercicio con un numero más alto de procesos y que su complejidad pudiera ser mayor aun que en el anterior.

Seguidos a estos ejemplos explicaba el funcionamiento de los algoritmos realizando los mismos ejemplos que se habían hecho de forma manual. El primero de los ejemplos siempre era en el que introducíamos los datos de forma manual, de ahí que tuviera un numero menor de procesos. Mediante volcados de pantalla y explicaciones textuales iba explicando paso a paso el funcionamiento de los algoritmos y los detalles de su ejecución para que no pudieran tener dudas sobre esta. A continuación hacia lo mismo con el otro ejercicio lo que pasa que en vez de introduciendo los datos de forma manual los introducía desde un fichero, para que así futuros alumnos pudieran ver las dos formas existentes de introducir datos para cada algoritmo.

Finalmente para terminar el informe explicaba, en alguno de los algoritmos, no en todos, alguna función o linea del código que pudiera tener una cierta dificultad de comprensión solamente con la descripción que tenia el código por lo que realizaba una descripción algo más detallada y extensa que la que había.



## **5.Posibles correcciones**

### **5.1.SJF**

A mi parece este algoritmo no tiene muchas posibles mejoras en cuanto a ejecución y a salida por pantalla porque me parece una forma clara de entenderlo de la forma en la que se encuentra.

La posible mejora que se me ocurre es que pueda hacerse más eficiente o corto el código. Otra mejora podría ser reestructurar el código de forma que el alumno pueda entenderlo mejor, de la forma que está es una forma que a mi me parece correcta pero puede que para otra persona no y entienda como una forma mejor otra manera.

### **5.2.ROUND-ROBIN**

Este algoritmo no creo que tenga mejoras posibles porque es un algoritmo óptimo en su ejecución y del cual su código es fácil de entender a pesar de ser muy extenso.

Por sacar alguna posible mejora sería la de cambiar la forma de salida por pantalla de este algoritmo que puede que a alguien no le pueda gustar o le parezca difícil de seguir durante la ejecución.

### **5.3.FCFS**

La posible mejora que se me ocurre para este algoritmo sería la de modificar alguna condición de salida en la ejecución del algoritmo. En el caso de que dos procesos tengan la misma llegada no tiene criterio de ordenación, hay veces que ejecuta antes al más largo y otras veces al más corto. Buscando en Internet no encontré en ningún sitio que me dijera que criterio deben seguir en ese caso si ejecutar uno más largo antes o viceversa, por lo que yo lo deje tal cual porque el algoritmo en definitiva dice que se ejecuta primero el que antes llega, entonces dos procesos que lleguen a la vez puede ejecutarse cualquiera de los dos. Por eso propongo como mejor que si algún alumno entiende que debería tener un orden y que debería ejecutarse antes los mas cortos o antes los mas largos que lo implementara.

### **5.4.PRIORIDADES**

El funcionamiento de este algoritmo creo que es el correcto desde el criterio que he tomado para implementarlo. Su ejecución es la correcta y la salida por pantalla es clara y sencilla. Puede que a alguien no le parezca la suficientemente explicativa por lo que como posible mejora podrían hacer una salida por pantalla más clara.

### **5.5.SRPT**

Este algoritmo es el que más problemas me ha dado a la hora de ejecución. El código entiendo que está como los anteriores y me parece el correcto al igual que la salida por pantalla. Tiene un pequeño error que no he podido solucionar y es que hay veces que no cambia al proceso que se esta ejecutando por otro que tiene menor tiempo de ejecución. Esa sería una posible mejora de este algoritmo que yo creo que puede ser la única mejora que necesita el algoritmo. Como no e tenido ningún ejemplo de algoritmo que funcionara bien completamente esto es lo mejor que le he podido dejar faltando como mejora a corregir por otros alumnos el detalle de que hay veces que no cambia la ejecución.