

Objetivo

Debe crear un programa en lenguaje C en el que se implementen los 5 diferentes algoritmos de *scheduling* dados en clases.

Entrada

El programa debe recibir como parámetros, además del nombre del algoritmo, los valores necesarios para la ejecución de los mismos. Cada algoritmo tendrá valores por defecto que serán usados en caso de no recibir ningún parámetro adicional.

Nota: el formato de entrada de los parámetros será definido por cada equipo.

Nota: los valores por defecto de cada algoritmo serán definidos por cada equipo.

Ejemplo 1:

```
$ gcc my-scheduler.c  
$ ./a.out MLFQ file.txt 5 3 10 20 30 100
```

Ejemplo 2:

```
$ gcc my-scheduler.c  
$ ./a.out -file file.txt -algorithml MLFQ -quantum 5 -levels 3 ...
```

El programa siempre recibirá como parámetro el nombre de un fichero texto en el que aparecerán, uno por línea, los diferentes procesos que el algoritmo tiene que planificar.

La estructura de cada línea es la siguiente.

1. un número entero que representa el momento en el que el proceso llega a la cola.
2. un número entero que representa el tiempo que se va a demorar ese proceso en ejecutarse (algunos algoritmos no deben tener en cuenta ese valor)
3. pares de números [opcional]
 - a) el primero número del par es el momento en el que el proceso va a realizar una llamada a una operación de entrada/salida (este número siempre va a ser un número entre cero y el tiempo máximo de ejecución)
 - b) el segundo número del par es el tiempo que se va a demorar la ejecución de la operación de entrada/salida.

Nota: la cantidad de pares de números puede ser desde cero hasta 10.

Los parámetros adicionales de cada scheduler son los siguientes:

FIFO

- No tiene parámetros adicionales

STF

- No tiene parámetros adicionales

STCF

- <quantum> - define el tiempo máximo que puede correr un proceso de forma ininterrumpida

RR

- <quantum> - define el tiempo máximo que puede correr un proceso de forma ininterrumpida

MLFQ

- <quantum> - define el tiempo máximo que puede correr un proceso de forma ininterrumpida
- <time-to-reset> - define cada cuanto tiempo todos los procesos son trasladados a la cola de mayor prioridad
- <queues> cantidad de colas

- <allotment> el tiempo máximo que puede correr un proceso en una cola (uno por cola)

Ejemplo 1:

```
$ ./scheduler file.txt MLFQ 5 200 3 5 10 15
```

Ejemplo 2:

```
$ ./scheduler -file file.txt -algorithm MLFQ -quantum 5 -time-to-reset 200 ...
```

Salida

Al terminar la ejecución se deben imprimir estadísticas de la ejecución de los procesos. Dentro de estas estadísticas debe aparecer como mínimo

- Response Time Máximo
- Response Time Promedio
- Turnaround Time Máximo
- Turnaroung Time Promedio

Nota: además de esos 4 valores, cada equipo puede decidir mostrar mas información al final de la corrida.

Nota: se permite además imprimir otras cosas durante la ejecución del proyecto.

Entrega

Todas las tareas se envían por correo electrónico a pqr@matcom.uh.cu antes de las 11:59pm del martes 10 de febrero.

En todos los casos, deben enviar un fichero .zip con código fuente de la aplicación y un fichero README.txt donde se expliquen los detalles de ejecución del proyecto, las opciones de compilación, dependencias, parámetros, etc.

En el fichero README.txt deben aparecer los nombres completos de los 2 integrantes del equipo

Revisión

la revisión de cada equipo será luego programada durante la semana por cada profesor de clases prácticas y alumnos ayudantes.

Nota: los estudiantes se revisan con el proyecto que enviaron por correo.