

Asignación: Informe Parcial 1

2322101, BERRIO PARRA LAURA CELESTE

Operating Systems

Jefferson A.

Cali, 2024-II

1. El algoritmo **MLQ (Multi-Level Queue Scheduling)** organiza los procesos en múltiples colas, cada una con diferentes prioridades y algoritmos de planificación. Los procesos se asignan a una cola específica de acuerdo a las especificaciones de los archivos de entrada.

Cada cola puede tener su propio algoritmo de planificación, como **FCFS**, **SJF**, **STCF** o **Round Robin**. Las colas de mayor prioridad tienen acceso preferencial a la CPU, es decir, los procesos en una cola de mayor prioridad se ejecutan antes que los de colas de menor prioridad. Solo cuando una cola de mayor prioridad está vacía, se consideran las colas de menor prioridad, de lo contrario se ejecuta en orden cada cola hasta finalizar completamente sus procesos.

De esta manera, MLQ es un algoritmo jerárquico y expropiativo, ya que los procesos pueden ser interrumpidos si un proceso de una cola más alta llega al sistema. Esto permite combinar distintos tipos de planificación según las necesidades del sistema.

Cada proceso tiene los siguientes atributos:

- Etiqueta (PID) : Identificación del proceso (p1, p2, etc.).
- Burst Time (BT) : Tiempo total que el proceso necesita para ejecutarse.
- Arrival Time (AT) : Momento en el que el proceso llega al sistema.
- Queue (Q) : Cola a la que pertenece el proceso (nivel de prioridad en el algoritmo MLQ).
- Prioridad (P) : Prioridad del proceso dentro de su cola, siendo 5 la más alta y 1 la más baja.

a. Ejecución de dos entradas

- Entrada 1 (RR (1)), (RR (3)), SJF

A screenshot of a code editor with three tabs: 'procesos.txt', 'main.py', and 'salida.txt'. The 'procesos.txt' tab is active and shows a text file with 8 lines of data. The first two lines are comments starting with '#'. The following six lines represent process data in the format 'pid; bt; at; q; p'. The last line is an empty line.

```
1 # Archivo: mlq002.txt
2 # 09102024
3 p1; 20; 0; 1; 5
4 p2; 10; 2; 1; 4
5 p3; 15; 4; 2; 3
6 p4; 5; 6; 3; 2
7 p5; 8; 8; 3; 1
8
```

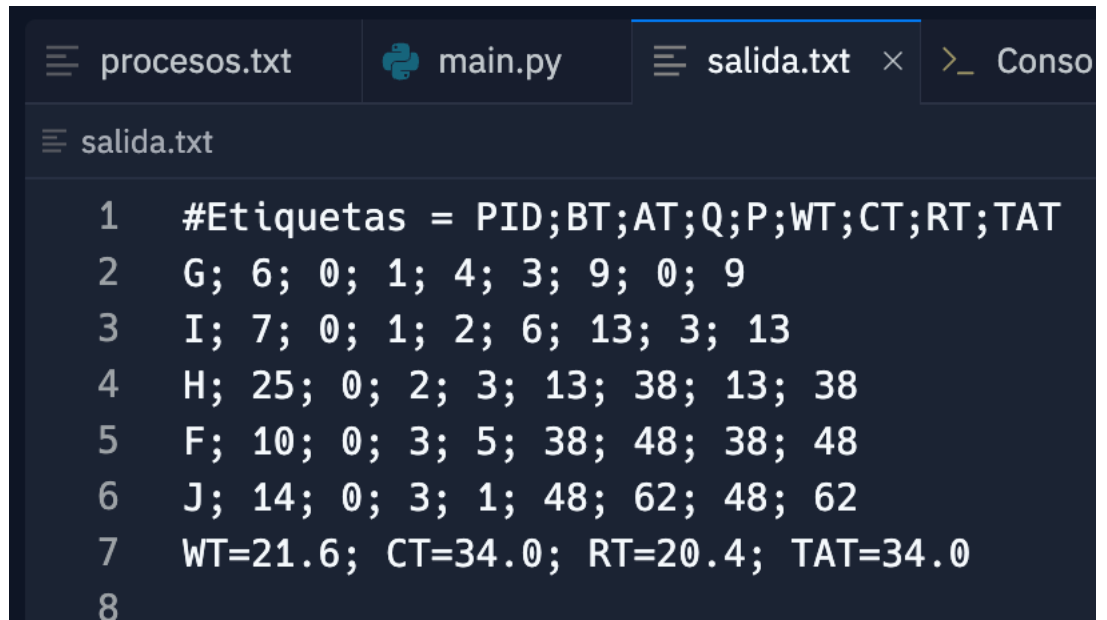
- Salida 1

```
procesos.txt  main.py  salida.txt × +
salida.txt
1  # Archivo: mlq002.txt
2  #Etiquetas = PID;BT;AT;Q;P;WT;CT;RT;TAT
3  p2; 10; 2; 1; 4; 9; 21; 2; 19
4  p1; 20; 0; 1; 5; 10; 30; 0; 30
5  p3; 15; 4; 2; 3; 26; 45; 30; 41
6  p4; 5; 6; 3; 2; 39; 50; 45; 44
7  p5; 8; 8; 3; 1; 42; 58; 50; 50
8  WT=25.2; CT=40.8; RT=25.4; TAT=36.8
9
```

- Entrada 2 (RR (3)), (RR (5)), FCFS

```
procesos.txt ×  main.py  salida.txt
procesos.txt
1  # Archivo: mlq006.txt
2  # 08112023
3  F; 10; 0; 3; 5
4  G; 6; 0; 1; 4
5  H; 25; 0; 2; 3
6  I; 7; 0; 1; 2
7  J; 14; 0; 3; 1
8
```

- Salida 2



The screenshot shows a code editor with three tabs: 'procesos.txt', 'main.py', and 'salida.txt'. The 'salida.txt' tab is active, displaying the following output:

```
1 #Etiquetas = PID;BT;AT;Q;P;WT;CT;RT;TAT
2 G; 6; 0; 1; 4; 3; 9; 0; 9
3 I; 7; 0; 1; 2; 6; 13; 3; 13
4 H; 25; 0; 2; 3; 13; 38; 13; 38
5 F; 10; 0; 3; 5; 38; 48; 38; 48
6 J; 14; 0; 3; 1; 48; 62; 48; 62
7 WT=21.6; CT=34.0; RT=20.4; TAT=34.0
8
```

2. VIDEO EXPLICATIVO

https://drive.google.com/file/d/1AqCuC3SDM2q0io1Bx8XciTvYKJsXHc2F/view?usp=drive_link