UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA

Sistemas Operativos

Tarea 2

ALUMNOS:

Cuevas Quintana Amir

De la Rosa Flores Fernando

PROFESOR;

Gunnar Wolf

GRUPO 6

2025-1

Objetivo:

- Realizar 5 ejecuciones de procesos aleatorios y usar algoritmos de planificación de modo que se comparen sus ejecuciones entre sí, imprimiendose estas de manera contigua

Entornos y dependencias:

Bibliotecas:

```
import random
from tabulate import tabulate
from colorama import Fore, Style
```

- El programa fue realizado en Python 3.12.3, usando el IDE Visual Studio Code de Microsoft, usando un Sistema Operativo Windows 11.
- Se debe tener en cuenta que igual se debe instalar la biblioteca colorama 0.4.6 y tabulate 0.9.0, las demás ya vienen con Python.

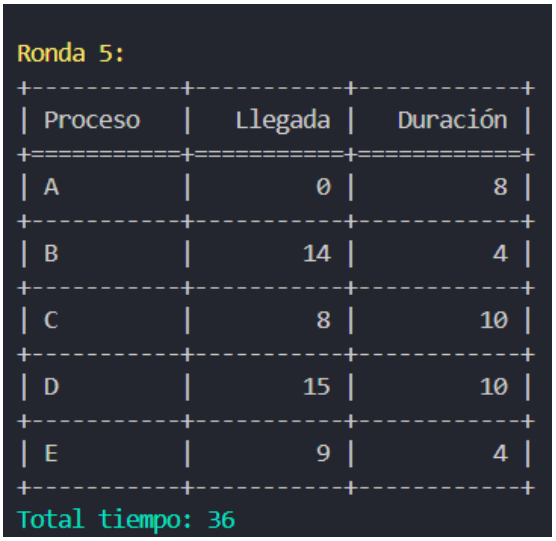
Buscando en Internet, para una distribución Debian de Linux, en esta página se muestra como instalar Python para Debian 11 Instalar Python 3.11.1 en Debian 11 y derivados - weblinus y python - How to install Tkinter on debian sid? - Stack Overflow

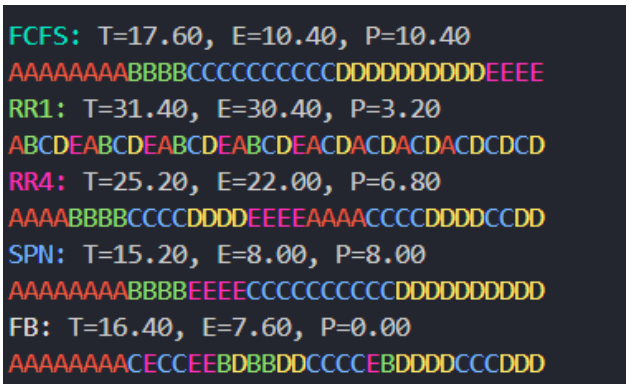
Especificaciones:

- Que genere una lista de procesos y que se vea en una salida como es que cada proceso se ve respecto a los diferentes organizadores.

Solución:

- Se realizo una función para cada algoritmo (A excepción de Round Robin) siguiendo los criterios de cada uno, además se le dio un color a cada proceso y se puso en una “tabla” a estos para que se viera de una manera más ordenada, para él como los procesos son organizados por cada algoritmo, a cada algoritmo también se le dio un color y se da una impresión, de forma que los caracteres totales son iguales al tiempo total de los procesos, se implementaron los algoritmos de First Come, First Served (FCFS) First Come First Serve – CPU Scheduling (Non-Preemptive) - GeeksforGeeks, RR1 y RR4 (Round Robin para 1 quantum y 4 quantums) Round Robin Scheduling with different arrival times - GeeksforGeeks, SPN (Shortest Process Next) Shortest Job First (or SJF) CPU Scheduling Non-preemptive algorithm using Segment Tree - GeeksforGeeks y FB (Feedback) Multilevel Feedback Queue Scheduling (MLFQ) CPU Scheduling - GeeksforGeeks





Este es un ejemplo de una ronda, sin embargo el código genera 5 para poder ver diferentes ejecuciones.

Ejecución:

Para poder hacer uso del programa basta con ejecutar el archivo llamado ‘Tarea2.py’ desde la terminal o con ayuda de algun IDE, para Linux, encontramos este enlace [How to execute python file in linux - Stack Overflow]

No es necesario hacer modificaciones al código.