

# **Sistemas Operativos**

## **Proyecto integrador**

Departamento de Ciencia y Tecnología

Universidad Nacional de Quilmes

Autor:

Sergio Loyola

[sergio.loyola@unq.edu.ar](mailto:sergio.loyola@unq.edu.ar)

Índice de contenido

1. Introduccion.....3

2- Objetivos.....4

3- Desarrollo.....5

4- Conclusiones.....6

# 1. Introduccion

El presente documento tiene como propósito informar sobre los alcances de la simulación de un sistema operativo en python, manifestando los problemas que se fueron encontrando durante el desarrollo del mismo y exponiendo los resultados obtenidos.

## 2- Objetivos

El objetivo del proyecto es emular un sistema operativo en python, con las siguiente características:

- CPU
- Dispatcher.
- Page Table.
- Colas.
- PCB.
- Content switch.
- Clock.
- Interrupciones
  - New.
  - Timeout.
  - IO.
  - IO\_end.
  - Kill.
  - PageFault
- Procesos y sus estados.
  - New.
  - Ready.
  - Running.
  - Waiting.
  - Terminate.
- Schedule
  - FCFS.
  - Prioty.
  - Round Robin
- Memoria.
  - Asignación continua.
    - Framentacion.
    - Compactacion.
    - Tipos de Ajuste.
      - First-Fit.
      - Worst-Fit.
      - Best-Fit.
  - Paginación.
    - Page Table
    - Algoritmo de Selecccion de Victima.
      - FIFO.
      - Second Chance.
      - -LRU.

### 3- Desarrollo

Al comenzar con la implementación para la emulación del sistema operativo en Python, me encontré con dos grandes obstáculos: no sabía programar en Python y tampoco conocía el concepto de objetos y como estos se utilizaban. Este fue el mayor problema con el que me encontré durante todo el proyecto de implementación. Observaba que cada vez que comenzaba una nueva etapa, necesitaba utilizar funciones del lenguaje que desconocía como, diccionarios, tuplas, etc.

Esto significo para mí empezar de cero, al menos tres veces. La ante ultima vez, desconocía el manejo de objetos. Para este punto, había realizado los tres “schedules” requeridos (FCFS, Priority y Round Robin) con funciones. Una vez me frustré bastante cuando, al llegar al aula (muy contento por cierto, pues había logrado programar utilizando funciones) me encontré con que, si bien mis schedules funcionaban, no era lo que debía aplicar, incluso pensé en dejar la materia, ya que los conceptos teóricos los entendía pero me costaba mucho llevarlos a la práctica e implementarlos.

Me amigue con el lenguaje y los objetos, y fué cuando finalmente pude comenzar con el desarrollo definitivo, logrando implementar, queue, pcb, schedule, irq, cpu, dispatcher, memory, clock.

Si bien pude lograr que el proyecto cumpla con todos los requerimientos y lineamientos solicitados, creo que el punto más destacable es el de “interrupciones”, donde desarrollé una programación ordenada y modular.

Considero al schedule Round Robin como el punto débil del proyecto, ya que, para rotar procesos dentro del CPU, utiliza ráfagas fijas y no variables como me hubiese gustado (para mi implementación, fijado en 2).

Otro punto débil del proyecto es no poder manejar dinámicamente los ticks del reloj y la ejecución de los programas (procesos).

No logré implementar el traslado de una pagina de la Swap a la memoria principal.

## **4- Conclusiones**

El desarrollo del proyecto me demandó bastante tiempo y esfuerzo, pero creo que el resultado fue el esperado (bajo los lineamientos indicados para su desarrollo) pues el mismo cumple con todos los requisitos que se solicitaron.

La implementación del proyecto me ayudo a afianzar y consolidar los conocimientos adquiridos sobre los sistemas operativos, aplicando en mi trabajo diario, por ejemplo, los conceptos de manejo de memoria RAM, que me ayudaron a entender y comprender su estructura y su funcionamiento; que antes desconocía y pensaba que se manejaba de forma automática. Ahora, que cuento con fundamentos, entiendo por qué.

Adicionalmente, aprendí a programar en Python, con el que puedo generar scripts para ejecución de tareas en los servidores con los que trabajo y administro diariamente.