

Proyecto Final Sistemas Operativos

Este proyecto implementa un simulador de algoritmos de calendarización (FIFO, SJF, SRTF, RR y Priority) y mecanismos de sincronización (mutex y semáforo) con interfaz gráfica en Tkinter.

Github: https://github.com/vgcarlol/PROY2_SO

Requisitos

- **Python 3.7+**
- Módulos estándar: `tkinter`, `random`, `csv`
- No se requieren librerías externas.

Estructura de archivos

```
PROY2_SO/
├── gui.py           # Interfaz gráfica del simulador
├── scheduler.py     # Implementación de algoritmos y sincronización
├── utils.py         # Lectura de archivos, cálculo de métricas y estructuras
└── auxiliares
    ├── main.py      # Ejecución por consola para pruebas y depuración
    ├── README.md    # Esta documentación
    └── data/        # (Opcional) carpeta con ejemplos de archivos de entrada
        ├── procesos.txt
        ├── recursos.txt
        └── acciones.txt
```

Ejecución del simulador

Interfaz gráfica

1. Abra una terminal y vaya a la carpeta del proyecto:

```
cd /ruta/a/PROY2_SO
```

2. Ejecute:

```
python gui.py
```

3. En la pestaña **Calendarización**:

- Pulse **Cargar Procesos** y seleccione un archivo `.txt` o `.csv` con procesos.
- (Opcional) Marque **Ingresar manualmente** para editar directamente el listado.
- Seleccione los algoritmos deseados y, si usa RR, ajuste el **Quantum**.

- Pulse **Iniciar Simulación** para ver el Gantt animado y las métricas.
4. En la pestaña **Sincronización**:
- Pulse **Cargar Recursos** para cargar un **.txt** con capacidades.
 - Pulse **Cargar Acciones** para cargar un **.txt** con las operaciones de los procesos.
 - Elija **Mutex** o **Semáforo**.
 - Pulse **Iniciar Simulación** para ver el timeline de accesos/esperas.

Línea de comandos

Para pruebas rápidas sin GUI:

```
python main.py
```

Mostrará por consola el diagrama de Gantt y la simulación de sincronización.

Formatos de archivo de entrada

Procesos (**procesos.txt** o **.csv**)

Cada línea define un proceso:

```
PID, BT, AT, PRIORITY
```

- **PID**: identificador (e.g. **P1**)
- **BT**: burst time (entero)
- **AT**: arrival time (entero)
- **PRIORITY**: prioridad (entero, 0 = alta prioridad)

Ejemplo:

```
P1, 8, 0, 2  
P2, 4, 1, 1  
P3, 2, 2, 3
```

Recursos (**recursos.txt**)

Cada línea define un recurso y su contador inicial:

```
RESOURCE_NAME, COUNT
```

- **RESOURCE_NAME**: nombre (e.g. **R1**)
- **COUNT**: número de unidades (entero)

Ejemplo:

```
R1, 1  
R2, 2
```

Acciones (**acciones.txt**)

Cada línea describe una operación en un ciclo:

PID, ACTION, RESOURCE, CYCLE

- **ACTION:** **READ** o **WRITE**
- **CYCLE:** ciclo de reloj (entero)

Ejemplo:

```
P1, READ, R1, 0
P2, WRITE, R1, 0
P3, READ, R2, 1
```