Installation and Usage

Ordonnanceur-multi-tache-de-processus

December 14, 2025

Contents

1 Overview

This document explains how to install, build, and run the Multi-Process Scheduler Simulator.

The simulator visualizes process scheduling algorithms (TUI and GUI modes). Scheduling algorithms are implemented as separate modules and dynamically loaded at runtime.

Repository layout (key files)

- `src/main.c` — application entry point, CLI and menu handling
- `src/processus.h` — `Processus` struct and shared definitions
- `src/affichage.c` — Gantt chart and display logic
- `src/politiques/` — scheduling algorithm implementations
- `src/processus.txt` — example process input file
- `build/` — compiled artifacts created by `make`

2 Prerequisites

Recommended platform: Linux / WSL (Windows Subsystem for Linux).

Tools required:

- `gcc` (C compiler)
- `make`

3 Build (compile) the project

1. Open a terminal in the `src/` directory:

   ```
   cd path/to/Ordonnanceur-multi-tache-de-processus/src
   ```

2. Run `make`:

   ```
   make
   ```

   This compiles the main program and the scheduling modules. The compiled shared libraries are placed into `build/politiques/` and the executable `main` is produced.

4 Process input file format

Each process is one line with the following fields separated by spaces:

```
process-name arrival-time execution-time initial-priority
```

Example (`processus.txt`):

```
P1 0 5 1
P2 1 3 2
P3 2 4 1
```

5 Run the simulator

There are two main modes:

5.1 TUI (Terminal UI) mode

Show a terminal-based menu and results in the terminal. Provide the process file as argument:

```
./main --tui processus.txt
```

5.2 GUI mode

Open the graphical interface:

```
./main --gui processus.txt
```

6 Add a new scheduling algorithm

1. Create a `.c` file in `src/politiques/`, e.g. `my_algorithm.c`.
2. Include the shared header and implement `ordonnancer`:

   ```
   #include "../processus.h"

   void ordonnancer(Processus T[], int n) {
       // scheduling logic here
   }
   ```

3. Run `make` again; the makefile auto-detects new files and compiles them to `build/politiques/`.

7 Troubleshooting

- `make:  command not found` — install build tools with your package manager.
- `Permission denied` when running `./main` — run `chmod +x main` or run via WSL/appropriate shell.
- `process file not found` — ensure path is correct; use absolute path if needed.
- Compilation errors in `politiques/` — confirm each module implements `void ordonnancer(Process` `T[], int n)` and includes `../processus.h`.

8 Contact / Contributing

Feel free to open issues or PRs for adding algorithm implementations, improving visuals or GUI, or automating PDF generation via CI.