# Process Management in Python

---

This project demonstrates various **Operating System process management concepts** using Python. It includes process creation, command execution, zombie/orphan processes, and process prioritization — all implemented with simple, well-documented code.

## ■ Requirements

- Works best on **Linux or macOS** (since Windows does not support `os.fork()` and `/proc`)
- Requires **Python 3.8 or later** - No extra dependencies — just pure Python!

## ■■ How to Run It

1■■ **Clone the repository:** ```bash git clone https://github.com//.git cd ```

2■■ **Check your Python version:** ```bash python3 --version ```

3■■ **Run the script:** ```bash python3 process_management.py ```

You'll see each task running one after another, with clear console messages explaining what's happening.

## ■ What Each Task Does

| Task | Description | Key Concept | |------|-------------|-------------| | 1 | Creates multiple child processes using `os.fork()` | Process creation | | 2 | Executes system commands (`ls`, `date`, `ps`) | Command execution | | 3 | Demonstrates zombie and orphan processes | Process states | | 4 | Reads info from `/proc/[pid]` | Process inspection | | 5 | Runs multiple CPU tasks with different priorities | Process scheduling |

## ■ Want to See a Zombie Process?

During **Task 3**, open another terminal and type: ```bash ps -el | grep defunct ``` You'll notice the child process listed as `` briefly before it's cleaned up.

## ■ End of Execution

Once all tasks complete, you'll see: ``` ======== All Tasks Completed ======== ``` That's your confirmation that everything ran perfectly!

## ■■ Tips & Notes

- For best results, **run this from a terminal**, not an IDE. - On Windows, you can still run it using **WSL (Windows Subsystem for Linux)**: ```bash wsl python3 process_management.py ```

---

■ *Created with care to help you understand how real operating systems manage processes — step by step.*