OS Lab 1401 – 2

Concurrency project

Our goal is to demonstrate the applications of concurrency in an operating system. To do this, we make a very simple simulation of an operating system, using CLI (Command Line Interface) or similar, text-based interface menus.

We want our OS to run a number of processes, as instructed using our interface. These processes, and our OS, should run concurrently, using multithreading or multiprocessing.

The choice of programming language is up to you, as long as you use concurrency in it.

Our OS has access to a finite amount of resources. These include CPU (processing power), ram (memory), and network bandwidth. Each of these can be represented by a number (such as an integer)

Assume the following specs:

CPU: 100 (%)

Ram: 6 (GB)

Bandwidth: 10 (Mbps)

Each process requires some resource(s) to work, and it receives those resources from the operating system. After it is complete, it returns those resources back to the OS.

If there aren't enough resources for a process that we want to run, you should use a task scheduling algorithm (like FCFS) to handle this situation. (Other task schedulers have bonus points.)

We want these processes from our OS:

1- **Process check:**
   - running this process will show us all currently running processes, their process ID, and the amount of resources each of them are using.
   - This process prints its results and ends immediately, and doesn't use any resources.

2- **Process manager:**
   - This process will take a process ID, and a control command. The OS must then apply the command to the target process. Given the "kill" command, the target process must stop all its functions and return all of its resources back to the OS. (Other commands have bonus points)
   - This process returns its results and ends immediately, and doesn't use resources.

3- **VPN:**
   - The process does nothing (sleeps or is idle)
   - Once started, it doesn't end on its own, and has to be terminated using the process manager.
   - It uses a 2Mbps network bandwidth while it is running.

4- **Mine:**
   - If VPN is running, it prints a success message after sleeping for 30 seconds.
   - Otherwise it prints a network error message and the process finishes.
   - If it runs, the process uses 80% of CPU, 4GB ram and 8Mbps network bandwidth.

5- **Counter:**
   - The process counts up to 10,000 and prints the last number when complete.
   - It uses 10% CPU and 3GB ram.