UDACITY

‹ Return to "C++" in the classroom

DISCUSS ON STUDENT HUB

Process Monitor

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

You put a lot of effort into this project and I think it was worth it because you must have learnt a lot from it. Congratulations!
Keep up the good work for the next challenging project.

Here are some interesting Youtube videos from which you can deepen your knowledge about OOP, too:

- C++ Tutorial: Object Oriented Programming
- C++ Beginner Game Programming
- Code it yourself: Role-playing Game
- Polymorphism

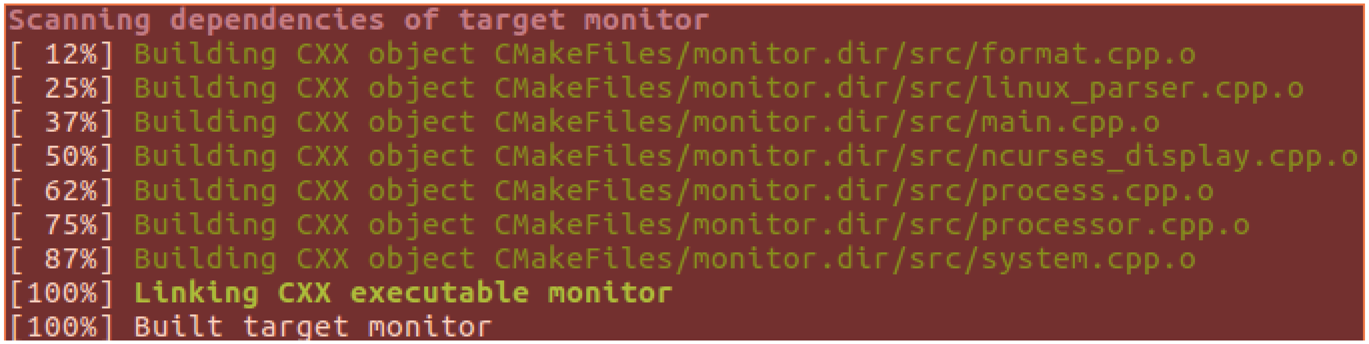
Basic Requirements

The program must build an executable system monitor.

You provided all necessary files and the project can be compiled without an error. Great job.

The program must build without generating compiler warnings.

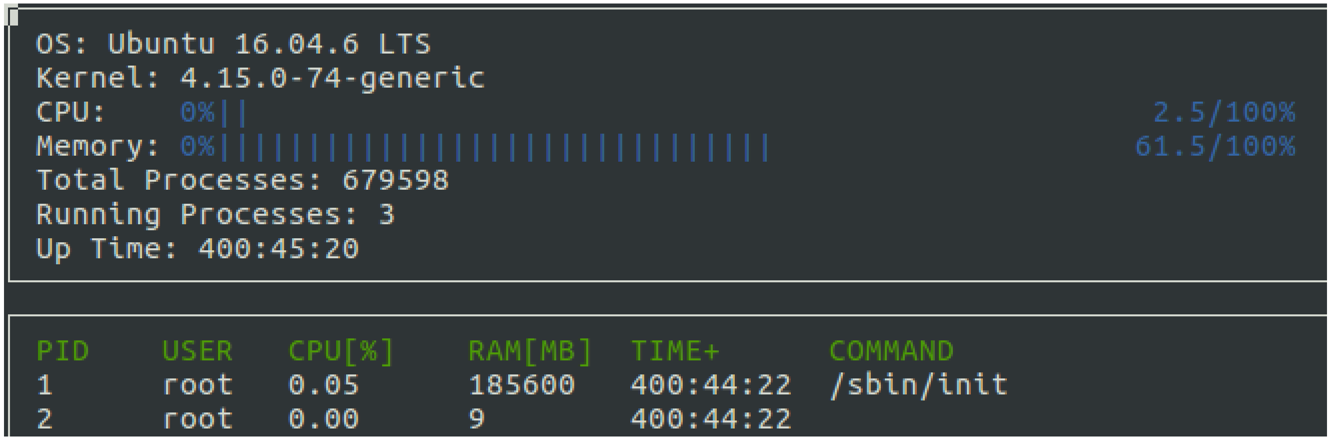
During the compilation I got no warning messages. Nice work.



```
Scanning dependencies of target monitor
[ 12%] Building CXX object CMakeFiles/monitor.dir/src/format.cpp.o
[ 25%] Building CXX object CMakeFiles/monitor.dir/src/linux_parser.cpp.o
[ 37%] Building CXX object CMakeFiles/monitor.dir/src/main.cpp.o
[ 50%] Building CXX object CMakeFiles/monitor.dir/src/ncurses_display.cpp.o
[ 62%] Building CXX object CMakeFiles/monitor.dir/src/process.cpp.o
[ 75%] Building CXX object CMakeFiles/monitor.dir/src/processor.cpp.o
[ 87%] Building CXX object CMakeFiles/monitor.dir/src/system.cpp.o
[100%] Linking CXX executable monitor
[100%] Built target monitor
```

The system monitor must run continuously without error, until the user terminates the program.

The project runs without error. I let it run more than a minute and it worked and reflected that status of my computer. Well done.



```
OS: Ubuntu 16.04.6 LTS
Kernel: 4.15.0-74-generic
CPU:    0%||                                                  2.5/100%
Memory: 0%|||||||||||||||||||||||||||||||                    61.5/100%
Total Processes: 679598
Running Processes: 3
Up Time: 400:45:20

PID    USER   CPU[%]    RAM[MB]  TIME+      COMMAND
1      root   0.05      185600   400:44:22  /sbin/init
2      root   0.00      9        400:44:22
```

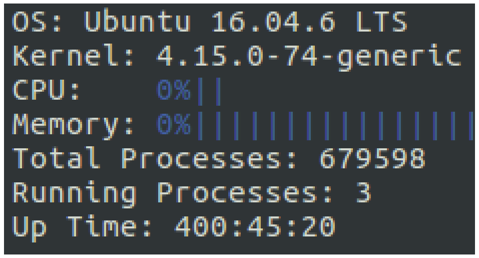
The project should be organized into appropriate classes.

The project is organized into classes (Process, Processor, System) with separated header (.h) and source (.cpp) files. Nice work.

System Requirements

The system monitor program should list at least the operating system, kernel version, total number of processes, number of running processes, and up time.

All the necessary information is shown by the system monitor.



```
OS: Ubuntu 16.04.6 LTS
Kernel: 4.15.0-74-generic
CPU:    0%||
Memory: 0%|||||||||||||||
Total Processes: 679598
Running Processes: 3
Up Time: 400:45:20
```

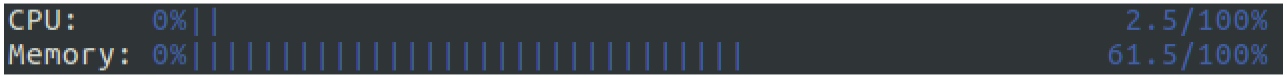
The System class should be composed of at least one other class.

Nice work having the `processes_` std::vector containing objects of the Process class in the System class.

Processor Requirements

The system monitor should display the CPU utilization.

The CPU (and memory) utilization is displayed correctly, it reflects the current status of the system. Well done.



```
CPU:    0%||                                                  2.5/100%
Memory: 0%|||||||||||||||||||||||||||||||                    61.5/100%
```

Process Requirements

The system monitor should display a partial list of processes running on the system.

The monitor shows the system processes as well. Nice work.

The system monitor should display the PID, user, CPU utilization, memory utilization, up time, and command for each process.

All the necessary information is shown in a tabular format. Well done.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review