UDACITY

Process Monitor

REVIEW
CODE REVIEW
HISTORY

Meets Specifications

Great work on completing the System Monitor Project!

You're previous review covered all the major bases in your code review before, so I won't risk any confusion by repeating them. However, you may find the following additional resources helpful as you continue on throughout the course:

Here are are few key talks from the recent Cpp Conference that you may find useful as you continue to the next steps of the course:

Object Oriented Programming: https://youtu.be/32tDTD9UJCE
Best Parts of C++: https://www.youtube.com/watch?v=iz5Qx18H6lg&list=WL&index=17&t=0s

You may find the following back to basics talks helpful in the next section:
RAII https://youtu.be/7Qgd9B1KuMQ
Move Semantics https://www.youtube.com/watch?v=St0MNEU5b0o

You may also find the following reading material helpful in your journey:
Code Complete, Steve McConnell
Clean Code, Robert C Martin
The Pragmatic Programmer, Andrew Hunt and Dave Thomas

Basic Requirements

The program must build an executable system monitor.

The program must build without generating compiler warnings.

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

The project should be organized into appropriate classes.

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.

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

Processor Requirements

The system monitor should display the CPU utilization.

Process Requirements

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

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

Fantastic work - you are now displaying the memory usage in the correct units (MB instead of KB).

↓ DOWNLOAD PROJECT

RETURN TO PATH

↓ DOWNLOAD PROJECT