'Frequently Asked Questions for Lab 2'

Q. What exactly is the point of problem 2, the two files look the same to me?

Open the two files side by side. Try to find out how they send messages across the sender and receiver, where is the message stored. Look up what does mqd_t, mq_attr, mq_open etc and explain what do they achieve.

Q. What is the purpose of this Lab?

This Lab is mainly focused on creating your own logging mechanism for the various subsystems in your project. By subsystems we mean modules in your project ; for eg. one subsystem could be dealing with reading the values from the sensors, another subsystem could be involved in performing computations on these values.

Logs are like printf's which we usually add for debugging, but they are a little more sophisticated. Look at debug macros such as __FUNC__ etc, which can be included in the logs that you print. It would be good to **also have a time-stamp** included in each of the logs.

You can now use these logs to understand the sequence of events that occurred in your system. You can also set logging levels, such as HIGH, MEDIUM, LOW. So when you make the logging level as high, only the messages with the level as high would be printed. When you make the current log level as MEDIUM, all the HIGH and MEDIUM level logs are printed, when you make the level as LOW,all the HIGH, MEDIUM and LOW messages are printed. The rationale behind having different levels is that logging eats up considerable amount of CPU time, and hence by having these levels, you can control how much of your CPU time is taken up in logging; . When you are debugging your code you can have the log level as LOW, so that you get all the log messages, when you are done developing your code, you can make the level as HIGH, and hence log only the most important messages. You **should write a function that allows to dynamically change the system's current logging level to any one of the standard levels**.You can also have a functionality, where you can configure the logging levels for each of the individual subsystems. For eg. you may enable all the logs for one subsystem and completely disable log for the other subsystem. This would be a 'nice to have' feature for the logging mechanism in your Final Project :-)

A sample log could look like

SENSORS_SUBSYSTEM::HIGH:: Read all the sensor values :: Line no:39 :: Function:read_input
PROCESS_SUBSYSTEM::LOW:: Acquiring lock on the input buffer :: Line no 22 :: Function:compute_mean

For the purpose of this lab, you can have the POSIX timers as one subsystem and two_tasks.c as another subsystem,think of something else for the third subsystem.Select events in each of these subsystems, and then assign each event some level based on its importance.You may also go ahead and write your own subsystems, if you want.

If you have any questions regarding this, feel free to post it on the Google group, or you may email the Professor and CC the TA's.

Q. Do I have to use PIT, can't I just use tickGet?

Yes, you have to use PIT. tickGet() does not have a high enough resolution to be used for this lab. Though you can start with tickGet() and have your frame work up and running.

Q. Where is the file in which I log all the messages stored?

While creating a new target connection,look at all the steps to understand when you create a file on the target, where exactly is that file stored on the host machine.

This page is last updated on October 8, 2013

<< Back