Project: Process and Resource Management

Protocol for Project Submission and Testing

Documentation

- Submit your source code to EEE by the due date (see course web page if you need help with EEE submissions)
- The code you submit must be *exactly* the code you use for your testing. If you need to make any changes after submission then inform the TA prior to testing.
- No other documentation is necessary.

Testing

- You need to see the TA during one of the time windows on the due date (time and place will be announced by email).
- You should bring your own laptop for the test. If you do not have a laptop or prefer not to use it, then you can run your program on one of the lab computers.
- Your program must be able to read a text file (.txt extension, similar to the one posted on the web) from a USB memory stick, and write a text file (similar to the one posted on the web) to the same memory stick.
- You will be asked to perform the following steps
    - start your program (it must be the exact program that you already submitted to EEE)
    - your program should read the text file called input.txt from a memory stick that will be given to you
    - it should write all output to a single text file on the same memory stick; the name of the file should be nnn.txt, where nnn is your 8-digit student number
- You only get one chance to run the test, except when there is some minor problem that results in a catastrophic failure and can be fixed on the spot, e.g., the program crashes and produces no results.
- We will evaluate the output of your program and report the results to you (not during the demo session). You can then see the TA during office hours to see the test cases you failed. If you have a valid justification for why your results are different, we may accept the results or award additional credit.
- I suggest that you test the protocol before coming to the demo session to avoid unnecessary delays/problems:
    - copy the sample input file from the web page onto a memory stick
    - run the above protocol
    - check your memory stick to make sure it contains a file nnn.txt that matches the output file on the web page

Expectations

- There will be only one input file. It contains a series of test sequences. Each sequence (except the first one) starts with an “init” command and ends with a blank line.
- All commands in the input file will have a correct format. Specifically, only existing opcodes will be given (cr, req, to, etc.) followed by the appropriate number of parameters, each separated by a blank space (as shown in the sample input file).
- However, commands may request operations that are illegal. This includes commands that would never succeed (e.g. requesting 5 units of any resource); commands that violate semantics (e.g. releasing a resource held by a different process); or commands that would cause problems in the future (e.g. creating multiple processes with the same name). It is part of your assignment to identify and catch all such illegal operations. If any such a command is detected, then ignore it, write “error” into the output file, and continue with the next command (see the sample output).
- The output file should contain a separate line for each test sequence (not for each command). Each line should start with “init”; this should be followed by a series of single characters separated by blanks, where each character is the name of the currently running process. If a command requests an illegal operation then output the string “error” (see the sample output). To facilitate grading, do not output any additional information.