

ECE 331

Homework 1

See course web site for due date

Place your typed homework answers in vim. Print single sided with your name using a **mono space font**. No need to restate questions. Fully investigating questions is required for a higher grade. Please use the kernel coding style for all code. Please use your RPi for developing answers. Although code should be written and run on a RPi, it should run on ANY POSIX compliant OS. As always, all code shall be comment, conform to the Linux Kernel Coding Style, and error conditions shall be checked and appropriately handled.

1. Follow a vim tutorial. Several are available for example, <http://vim.wikia.com/wiki/Tutorial> or <http://www.openvim.com/>.

For your answer, indicate that the tutorial has been completed.

2. Review the Linux Kernel Coding Style document.

For your answer, indicate that the document has been reviewed and will be conformed to.

3. Using your RPi, write a C program using vim that prints all passed command line arguments.
 - a) For this part, include your source code in the **body** of your homework not a separate printout. Compile your program with `gcc -o args -Wall -g args.c`
 - b) Give the command to pass the filename `Game of Thrones $season 1 Episode 4.mp4` and all files that contain the string `\\` (four backslashes) anywhere in the filename. to your program. Information about command line arguments: <http://www.cprogramming.com/tutorial/c/lesson14.html>. Ignore the bad style at the preceding web site....
4. Write a C program that determines speed of memory on your RPi. Have your program accept one argument on the command line, namely the size of a buffer. Use `malloc()` to allocate a buffer of the size passed on the command line and then write the same value to every memory location. Time the amount of time to write to every memory location. Be sure to `free()` all allocated memory and handle errors. Use `clock_gettime()` for timing. Pass `CLOCK_MONOTONIC` for the `clockid`. Have your program print the buffer size and raw timing (that is start and end times). No calculations need be done. See the next question to see what to do with the program.

For your answer, include your source code. Do not attach a separate document with source code. Place all questions in order assigned.

5. Write a MATLAB or octave m-file that processes and plots the data from the previous question. Use your program to test every non-zero power of ten (1, 10, 100, ...) until memory is exhausted. Load the data into MATLAB or octave and plot the memory speed versus data size. Use appropriate data representation and plotting. (If you are really bored, try to do this in Mathematica - already installed on your RPi.)

For your answer, include your source code. Do not attach a separate document with source code. Place all questions in order assigned.

6. Find and look over the FHS. Answer the following questions .
 - a) Give the purpose of the `/sys` directory stated in your own words.

- b) Give the requirements for `/usr/lib`.
7. Login to the gitlab server at `gitlab.eece.maine.edu`. Generate an SSH key and upload the public key to gitlab. Follow the SSH keygen directions at <http://doc.gitlab.com/ce/ssh/README.html>. Then add the key to gitlab following the directions at <http://doc.gitlab.com/ce/gitlab-basics/create-your-ssh-keys.html>.

For your answer, indicate that this is complete.

8. Give the fewest command(s) for and do the following:
- a) Create an alias named `ll` that executes `ll -aLF --color`.
 - b) list all files sorted by extension.
 - c) move a file named `x` in the current directory to `/tmp` and rename the file `y`.
 - d) remove all files in the current directory that have the number 10 through 25 anywhere in their filenames.