

CMPT 214: Programming Principles and Practice

Term 1 2016-17

Lab 8 - Makefiles, UNIX again

At the beginning of your lab period, the lab instructor will briefly review the functionality of the UNIX `nice(1)` command. You will need to use `nice(1)` in question 2g of this lab. The `time(1)` command—including the meaning of elapsed time and user CPU time—will also be reviewed.

Solve or complete each of the tasks or problems below. For all problems involving the use of LINUX/UNIX commands, place the command or pipeline you used along with the resulting output (i.e. copy-and-paste from your terminal window) in a file called `lab8.txt`. However, do not include extraneous or superfluous commands or output; only include content relevant and essential to the specified task. Then, with a text editor, add to `lab8.txt` identifying information to clearly distinguish which commands/output/code/description correspond to each task/question. When done, hand in `lab8.txt` and your modified `Makefile` from question 2e through the `moodle` page for the lab. This lab is out of a total of 14 marks; the number of marks allocated to each question is indicated below. Marks may be docked for extraneous, irrelevant, or superfluous content or for not following directions. Your submission is due at 11:55 p.m. on Thursday, November 3. Note that specific platforms need to be used for some of the commands.

1. (1 mark) As you already know, information on UNIX commands can be obtained from their manual pages. However, some commands also contain additional information in “info pages”. Use the `info(1)` LINUX/UNIX command to view the info page for `make` on `tuxworld`. Among many other pieces of information, this info page lists possible values for the special variable `.FEATURES`—information not found in the manual page for `make`. Find the list of these values, and then copy-and-paste the description of the list of possible values for variable `.FEATURES` into `lab8.txt`.
2. Perform this question on your an `ismac` machine in Rm S311 or RmS315. Do **not** use `tuxworld` for this question. Include in `lab8.txt` a log of each step, except as noted.
 - (a) (1 mark) Download the file `queens14.cc` from the “Examples” portion of the course web pages and compile it manually (i.e. without using a `makefile`), calling the executable file `nqueens`. You will need this file in step 2g. You do not need to show a log of the download in `lab8.txt`.
 - (b) (1 mark) Use the `curl(1)` command to download the source code for the `factor` program from the following URL:
`http://www.acme.com/software/factor/factor_02Jun2016.tar.gz`
A bit of documentation for this program is available at this URL:
`http://www.acme.com/software/factor/`
 - (c) (1 mark) Use a UNIX command to `un-tar` (unpack) the file you downloaded in part 2b. Then change your current directory to the `factor` subdirectory that was just created. Notice that the source code for `factor` is written in C.

- (d) (1 mark) Use the `make(1)` command with appropriate options to print the commands that *would* be executed if you were to invoke `make` at this point (with no options or arguments). It is always a good idea to do this before using someone else's `makefile` just to make sure it isn't going to do anything you are not expecting or would not approve of.
- (e) (1 mark) You may have noticed that the output from step 2d showed a `cc` command involving the `"-s"` option. This option is deprecated, and should be removed from the set of `LDFLAGS`. Copy the original `Makefile` to `Makefile.org`. Then edit the `Makefile` to remove the use of the `"-s"` option. You do not need to submit a log of your editing. Finally, using `diff(1)` show the changes you made to the `Makefile`.
- (f) (2 marks) Use `make` to build the `factor` program. Do **not** `"make install"`. Test the `factor` program with a few inputs to get an idea of what the output looks like. Note that there may be another program called `factor` in the directories specified by your `PATH`; if so, you will need to explicitly specify the path to your newly-built `factor` (i.e. `./factor`) in order to run it. Make sure to test `factor` on a range of inputs (the documentation at <http://www.acme.com/software/factor/> will be necessary for this). Finally, change your working directory back to the one with `nqueens` in it.
- (g) For this part of the question, you are required to run the command `./nqueens 6` (this is the `nqueens` you compiled in step 2a) under three different conditions. For each of the conditions, you are to use the `time(1)` command to measure the amount of time taken by `nqueens`. The three conditions are as follows:
 - i. (1 mark) For the first condition, just run `nqueens` by itself, without running anything else concurrently. Use `time(1)` to run `nqueens`. Note that it might take a minute or two for the computation to complete.
 - ii. (2 marks) For the second condition, start ten instances of the command `"factor/factor 30000000-90000000 > /dev/null &"`. Next, use the command `"ps -0 nice"` to output the "nice number" of your processes, including these `factor` processes. Then give the same timed `nqueens` command as in part 2(g)i. Make sure these three steps are performed in rapid succession. After `nqueens` completes, kill all of the `factor` processes. The `killall(1)` command is useful for this.

Recall that to quickly re-invoke the last command given to the shell you can use `"!!"`. Alternatively, you can use the up-arrow key on the right side of the keyboard to retrieve the last command from the `HISTORY` buffer.
 - iii. (2 marks) For the third condition, run ten instances of the same `factor` command as in 2(g)ii (before running `nqueens`), but with the "nice number" of each `factor` process incremented by 10. Use the `nice(1)` command for this. After starting the ten instances of `factor`, execute the command `"ps -0 nice"` to show that the nice numbers of these processes have indeed been changed from the situation in part 2(g)ii. Then run the timed `nqueens` command (again using `time(1)`). Kill all of the `factor` processes after `nqueens` completes.
- (h) (1 mark) Were there any differences in the elapsed times taken in i, ii, and iii? What about the user CPU times? In `lab8.txt` comment on why these differences occurred (or did not occur).

You will be using the `factor` program in next week's lab, so make sure not to delete it.