

(60-140) Lab Exercises #1

— Getting familiar with software and starting C programming

September 18, 2017

1. UNIX Exercise:

Both Cygwin (on PC) and Ubuntu (on CS system) are essentially variants of UNIX. While a list of common UNIX *commands and control sequences* is provided at the end of this file, the following shows a couple of examples for the use of `ls` command. While the base directory is `/cygdrive/c` in Cygwin for the `C:` drive on PC, it refers to `/home/your_user_name` in Ubuntu, where `user_name` is your user name for login. The following line of command allows you to see the directories and files in the base directory, in which `base_dir` needs to be replaced by either `/cygdrive/c` or `/home/your_user_name` depending on the system you are using.

```
$ ls base_dir
```

If you have a directory named `Program Files` in the base directory, the following command lists the files under this directory.

```
$ ls base_dir/"Program Files"
```

For the exercise of UNIX, you need to accomplish at least the following tasks.

- a. Start a Cygwin terminal.
- b. At the terminal prompt `$`, create a `fall2017` directory in the base directory `base_dir`.
- c. In a similar manner, create a `60140` directory in the newly created `fall2017` directory.
- d. Change your working directory to the `/cygdrive/c/fall2017/60140` directory.
- e. Check the name of your current working directory.
- f. List all the files of the current directory, including hidden “dot” files.
- g. Take a screenshot of the interaction, and submit it online.

2. C Programming Exercise

Accomplish the following tasks.

- a. Start Notepad++.
- b. Type in the program in Section 2.1 on page 10 of the textbook with a modification to change the `printf` statement to the following, where `my_name` needs to be replaced by your full name.

```
printf("Hello World!\nMy name is my name.\n");
```

- c. Save the file in the format of a C source file as `modified_pun.c` in your working directory, i.e., `/home/your user name/fall2017/60140`, and submit it online.
- d. As you are already in the working directory when completing the UNIX Exercise, you can now compile the C program with the following command to produce an executable.

```
$ gcc -Wall -std=c99 modified_pun.c -o my_prog
```

- e. Run the executable with the following command to produce the output.

\$./my_prog

3. RAPTOR Exercise:

- a. Start RAPTOR, and load the “10_area.rap” that you had created in Lab #0.
- b. Run the flowchart with a radius of ‘5’.
- c. Play with RAPTOR and, without changing the flowchart, make the program to display the area value with the precision set to two digits after the decimal point.

*Hint: toggle a breakpoint to stop the program before generating the output message, and then call the built-in function **Set_Precision(2)** by typing in the command in the MasterConsole textbox.*

- d. Modify the flowchart to include one function call that sets precision in the program itself right before printing the result of calculation.
- e. Save the modified flowchart to “11_area.rap”, and submit it online.

Evaluation:

All online submissions must be completed before due time, which will be kept on record. In addition, every student is required to show/demonstrate his/her complete exercises to a GA/TA at the end of this lab, or at the beginning of the next lab after completing online submission. The demonstration includes showing the submitted flowchart and/or C codes, compiling the C program, and trying out the flowchart and/or C program with different input values. The maximum marks for this lab is 15, with 10 for the lab work (submission and demonstration) and 5 for lab attendance.

COMMON COMMAND AND CONTROL SEQUENCES

The following commands can be entered at the UNIX system command prompt. Press the ⟨Return⟩ key after each command. Use lower case letters unless otherwise indicated.

Command	Description
WORKING WITH DIRECTORIES	
<code>pwd</code>	Gives you the name of your current working directory.
<code>cd <i>dirname</i></code>	Changes your working directory to <i>dirname</i> . If you are not in <i>dirname</i> 's parent directory, you must use <i>dirname</i> 's full pathname.
<code>cd ..</code>	Changes your working directory to the parent of the directory that you are in when you issue this command. If you are at the root, you will remain there.
<code>cd</code>	Changes your working directory to your home directory.
<code>ls</code>	Lists the files in your working directory.
<code>ls -l</code>	Lists the files in your working directory, giving lengthier information about them — including file protection and access privileges.
<code>ls -a</code>	Lists the files in your working directory and shows your hidden “dot” files, as well, including your <code>.login</code> file.
<code>ls -lt</code>	Same as <code>ls -l</code> , but orders the files by time modified.
<code>mkdir <i>dirname</i></code>	Creates a directory called <i>dirname</i> . Your working directory will be the parent of this new directory, unless you specify another pathname.
<code>rmdir <i>dirname</i></code>	Deletes a directory called <i>dirname</i> , if the directory is completely empty. If you are not in <i>dirname</i> 's parent directory, you must use <i>dirname</i> 's full pathname.
WORKING WITH FILES	
<code>pico <i>filename</i></code>	Uses the editor <code>pico</code> to open a file called <i>filename</i> . If a file of that name does not already exist, it starts a new one.
<code>vi <i>filename</i></code>	Uses the editor <code>vi</code> to open a file called <i>filename</i> .
<code>cp <i>file1 file2</i></code>	Makes a copy of a file called <i>file1</i> and names the copy <i>file2</i> . If there already is a file called <i>file2</i> , it will be replaced.
<code>cp -i <i>file1 file2</i></code>	Makes a copy of <i>file1</i> and calls the copy <i>file2</i> . If there already is a file called <i>file2</i> , the system queries you before replacing it.
<code>rm <i>filename</i></code>	Deletes a file called <i>filename</i> . If you are not in the same directory as <i>filename</i> , you must specify <i>filename</i> 's pathname.
<code>rm -i <i>filename</i></code>	Deletes a file after confirmation. Type <code>y</code> to confirm, <code>n</code> to cancel.
<code>more <i>filename</i></code>	Lists the contents of the file called <i>filename</i> on your screen. Press the spacebar to scroll forward in the file. Type <code>b</code> to scroll back one screenful. Type <code>q</code> to quit.
<code>lpr -P <i>pr fname</i></code>	Sends a file called <i>fname</i> to a printer called <i>pr</i> .

ADDITIONAL COMMANDS	
history	Lists your previous commands. !n Executes command number n in the list that you receive as output of the history command.
man <i>command</i>	Obtains online information about the command of your choice. These are the UNIX system's online "man pages". Press the spacebar to scroll to the next screenful. Type b to scroll back a screenful. Type q to quit the man facility.
man -k <i>key</i>	Obtains online information when you do not know the command name. Replace <i>key</i> with a keyword for the function you wish to perform.
learn	Initiates online instruction in file manipulation, vi , C , and other topics.
passwd	Initiates the password-changing program. Respond, as prompted, with your old and new passwords.
pine	Starts the Pine mail program.
tin	Starts the network news reader tin . Use the up- and down-arrow keys to highlight a newsgroup (or message) of interest; press Enter key to select it. Type ? for help, and q to quit.
logout	Logs you off the UNIX system. If you have a "stopped process", the system will prompt you. Enter fg to resume the process or logout a second time to abort the process and log off the system.
KEYS AND TERMINAL EMULATION	
In the following, a caret (^) indicates the "control" key (often marked CTRL or CTL). Hold the "control" key down while typing the letter shown next to it.	
Backspace	To backspace, use the backspace key (often marked BKSP) or, if that does not work, the delete key (often marked DEL).
^c	Aborts the current process — e.g., a long screen listing.
^s	Temporarily suspends a listing until you type ^q .
^q	Resumes a listing after it has been suspended with ^s .
MORE CONTROL KEY SEQUENCES (ADVANCED USAGE)	
^u	Erases the whole line of characters you have typed thus far at the command line.
^d	This is the end-of-file indication.
^z	Suspends the execution of a process/"job". You may then have the process continue in the background (type bg immediately after). Type fg to resume the job or bring it to the foreground. To kill a suspended job, type jobs , then kill %n , where n is the number of the suspended job. (Note: use of this suspend character should be avoided until you have some familiarity with the system.)
^l	Redraws the screen.