

Linux Hands-On Session

1 Introduction

This document is intended for a practice session for a new Linux user enrolled in MECH 4822. It covers basic material used in this course and generally useful commands. Commands are shown in **typewriter font**, and must be typed precisely as shown with attention to lower and uppercase letters and spaces. Commands that are typed in a Linux shell are shown on a line starting with the “\$” character. You do not type this character; it represents the Linux shell prompt. For example, the command to list files in a directory (folder) would be shown as

```
$ ls -l
```

Note that there is a space after the command “ls”. This space could be shown explicitly with the “_” character to represent the space:

```
$ ls_-l
```

but the space character will be omitted for clarity in the remainder of these notes.

After some of the commands below, example screen shots will be given which indicate roughly what you should expect to see at your terminal. The userID for the examples is “engsjo”; your userID should appear in place of engsjo in those examples.

These notes cover the bash shell on Linux.

2 Basic Commands

2.1 Listing Files

(Note: A sample result for the next seven commands is given following those commands.)

To move to your “root” directory (the place in your file system where you login) type:

```
$ cd
```

```
$ ls -l
```

2.2 Directory Location

At any time, to find out what directory you are presently in, type:

```
$ pwd
```

2.3 Creating a Directory (Folder)

To create a directory for this course type:

```
$ cd
```

```
$ pwd
```

```
$ mkdir mech-4822
```

To move to that course directory, type:

```
$ cd mech-4822
```

Show that you are now in that directory:

```
$ pwd
```

2.4 Copying files

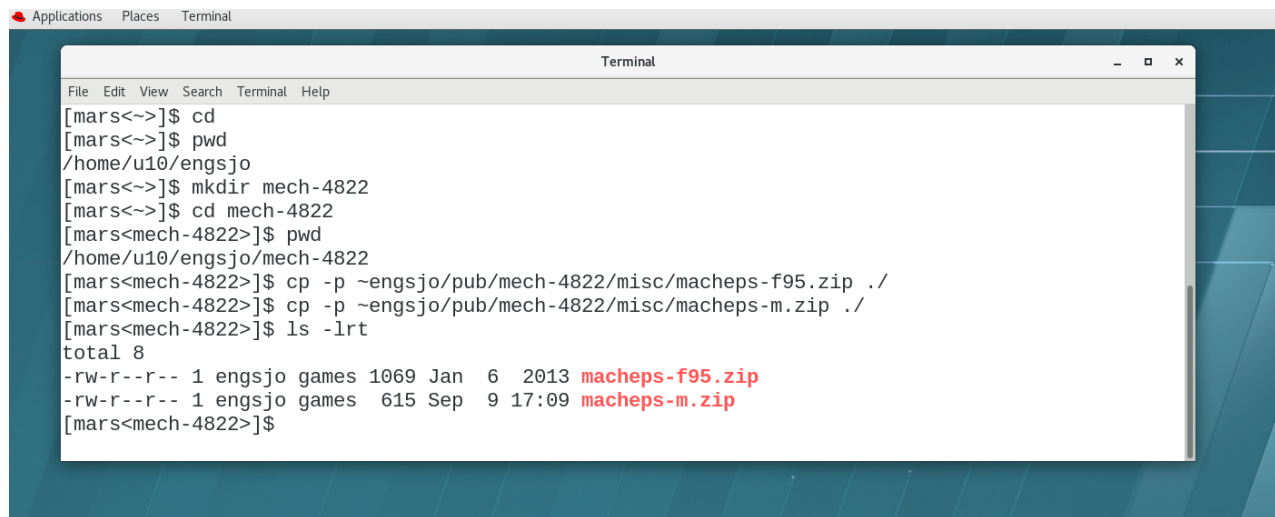
```
$ cp -p ~engsjo/pub/mech-4822/misc/macheps-f95.zip ./
```

```
$ cp -p ~engsjo/pub/mech-4822/misc/macheps-m.zip ./
```

List the files in the directory, sorted by time (newest last):

```
$ ls -lrt
```

The commands from the previous two subsections should produce an output like:



```
Applications Places Terminal
Terminal
File Edit View Search Terminal Help
[mars<->]$ cd
[mars<->]$ pwd
/home/u10/engsjo
[mars<->]$ mkdir mech-4822
[mars<->]$ cd mech-4822
[mars<mech-4822>]$ pwd
/home/u10/engsjo/mech-4822
[mars<mech-4822>]$ cp -p ~engsjo/pub/mech-4822/misc/macheps-f95.zip ./
[mars<mech-4822>]$ cp -p ~engsjo/pub/mech-4822/misc/macheps-m.zip ./
[mars<mech-4822>]$ ls -lrt
total 8
-rw-r--r-- 1 engsjo games 1069 Jan  6 2013 macheps-f95.zip
-rw-r--r-- 1 engsjo games  615 Sep  9 17:09 macheps-m.zip
[mars<mech-4822>]$
```

3 Other Useful Tools and Programs

3.1 Extracting files from a ZIP file

Unzip the files that were copied above:

```
$ unzip macheps-m.zip
```

```
$ unzip macheps-f95.zip
```

3.2 Viewing a File With a Pager

These commands allow you to look at a text file a page (one terminal screen) at a time without editing the file. It is fast and simple for checking the contents of a file.

For both commands: press the spacebar to page through the file. **When using “less”, type “q” to quit.**

```
$ more macheps.f95
```

```
$ less macheps.f95
```

3.3 Editing a File

```
$ gedit machepts.f95 &
```

```
$ emacs machepts.f95 &
```

Note that the ampersand character (&) at the end of the command line text tells the Linux shell to put the command *in the background*. This gives you the command line back for other commands.

If, when you start to edit a file with emacs, you find the window is split into two buffers (*i.e.*, an error occurred when it started up), then exit from that window and do the following (if there was no error, you do not need to do the following):

```
$ cd
```

```
$ cat .emacs
```

If the file is empty, no text will be displayed. If there is text in the file, use the following command to edit the file:

```
$ gedit .emacs
```

While editing the `.emacs` file, insert two semi-colons (;;) at the start of every line that has the # at the start. Save the file and exit. Now you should only have one windows when you start emacs.

3.4 Plotting with gnuplot

Note: Creating plots with gnuplot requires a graphical environment (*e.g.* X Windows). These commands should work when logged into a workstation console. If, however, you log in to a machine using **putty** or other terminal emulation program, you will not normally get a graphical environment (*i.e.* it is a text-based terminal session). Graphical commands will work if you log into a machine using an “X Window emulation” program (*e.g.*, **ThinLinc**).

3.4.1 Examples from Chapter 1

```
$ cp -p ~engsjo/pub/mech-4822/gp-examples/eg-plt-notes.zip ./
```

```
$ unzip eg-plt-notes.zip
```

```
$ gnuplot
```

```
gnuplot> plot sin(x)
```

```
gnuplot> load 'eg1-2.plt'
```

```
gnuplot> set term postscript
```

```
gnuplot> set output 'testplot.ps'
```

```
gnuplot> replot
```

```
gnuplot> q
```

```
$ ls -lrt
```

- To view the postscript file created above:

```
$ evince testplot.ps
```

- To convert a postscript file to a pdf file:

```
$ ps2pdf testplot.ps
```

- To view a PDF file:

```
$ evince testplot.pdf
```

3.4.2 Examples from the Appendix in the Supplementary Notes

```
$ cp -p ~engsjo/pub/mech-4822/gp-examples/gp-appendix-examples.zip ./
```

```
$ unzip gp-appendix-examples.zip
```

Explore using gnuplot by loading these examples in gnuplot and comparing the results with the figures in the Appendix of the Supplementary notes.

3.5 Redirection and Pipes

Examples

- Take the output that you would normally see at the screen from `ls -l *.txt` and send it to the file `txt-files.dat`.

```
$ ls -l *.txt > txt-files.dat
```

In the command above, the “*” (asterisk) is called a “wild card”. It represents any string. So, the command looks for all files with any name but having file extension “.dat”.

- View the results of a file listing in a pager:

```
$ ls -alrt | less
```

- View the list of aliases (shortcuts) sorted by name in a pager:

```
$ alias | sort | less
```

Other reference material : Linux reference cards on UMLearn.

4 Other Useful Commands

- Claiming extra disk space (one time only):

```
$ workspacetool claim
```

- Check your disk usage (to see if you are over quota:

```
$ quota -v
```

- Check the details of your disk usage:

```
$ du -h
```

- Do a remote login on a machine using secure shell:

```
$ ssh gaia.cc.umanitoba.ca
```

(Use `exit` to log out of your remote login session.)

- To list the set of processes on the computer:

```
$ top
```

(type `q` to quit).

- To compare two files:

```
$ diff -w -C0 file1.txt file2.txt
```

- To find a file on your account with of type `.dat` (the string after `name` is in double quotes; this string can be changed to find other files):

```
$ find . -name "*.dat"
```

- To delete a file:

```
$ rm filename
```

- To rename a file (can include a change in path):

```
$ mv oldfilename newoldfilename
```

- To search for a string in all files with extension `txt`:

```
$ grep string *.txt
```

The string must be inside double quotes if it contains spaces:

```
$ grep -i " call " *.f95
```

Note that the `-i` option makes the search case-insensitive.

- To make a shortcut (symbolic link) to a folder

```
$ ln -s actual-directory shortcut-name
```

- To make a file executable (e.g. a shell script)

```
$ chmod u+x filename
```

- To create a 7zip file

```
$ 7za archive-filename file-list
```

- To list a 7zip file contents

```
$ 7za l archive-filename.7z
```

- To extract contents from a 7zip file

```
$ 7za x archive-filename.7z
```

- Free File Sync:

```
$ FFS &
```

- Inkscape drawing software:

```
$ inkscape &
```

- Gimp image manipulation software:

```
$ gimp &
```

- check a command:

```
$ which command
```

```
$ whereis command
```

5 Customising your account

Some customisation of your account must be done in order to set up your account to run the commercial CFD software to be used for the assignments and the design project in this course. The instructor will provide more details of this customisation.

As a start, use the following command to copy over a file of customisations into your root directory.

1. It is important to move back to your root director:

```
$ cd
```

2. Copy the necessary files:

```
$ cp -p ~engsj0/pub/mech-4822/misc/add-to-bashrc-mech-4822.sh ./
```

```
$ cp -p ~engsj0/pub/mech-4822/misc/set_ls_colors.sh ./
```

```
$ cp -p ~engsj0/pub/mech-4822/misc/default_colors ./
```

3. Do the following once only (makes a backup of your `.bashrc` file):

```
$ cp -p .bashrc original.bashrc
```

4. Do the following once only (appends the text file of additional commands to your `.bashrc` file):

```
$ cat add-to-bashrc-mech-4822.sh >> .bashrc
```

5. Right click on the desktop and open a new terminal to make sure that these customisations have not generated any errors.

6 Running MATLAB

This uses the file `macheps.m` that was copied over earlier.

The file `macheps.m` is a MATLAB file that calculates machine epsilon for double precision. Start MATLAB and open and run this file to compare what it produces with the output of the fortran version. To start MATLAB, at a terminal command line type:

```
$ matlab &
```

7 Building and running a FORTRAN Program on Linux

Note that an executable program that you create from source code using the commands below will run only on a machine of the same category (architecture and OS). If you have problems running a program, you may need to delete the “.o” files and the “.exe” file and re-build the program (if it is a different category from when you last built the program).

To compile the fortran source code file `macheps.f95` on a Linux machine, type:

```
$ gfortran -c -g -C -fimplicit-none macheps.f95
```

To link the program object code with system libraries and create an executable program, type:

```
$ gfortran -g -C -o macheps.exe macheps.o
```

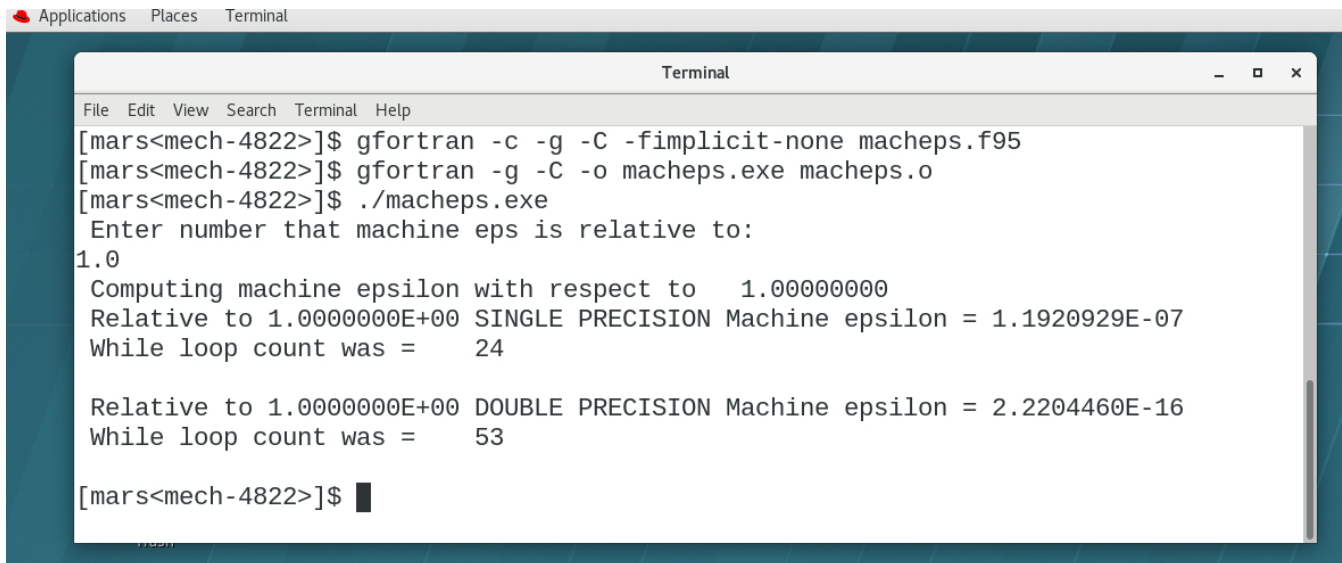
These commands create executable `macheps.exe`.

To run the program type:

```
$ ./macheps.exe
```

(and type in a number (for example, 1.0) in response to the prompt)

The results of the previous commands should look like (up to the prompt from running the `macheps` program):



```
Applications Places Terminal
Terminal
File Edit View Search Terminal Help
[mars<mech-4822>]$ gfortran -c -g -C -fimplicit-none machepts.f95
[mars<mech-4822>]$ gfortran -g -C -o machepts.o machepts.o
[mars<mech-4822>]$ ./machepts.exe
Enter number that machine eps is relative to:
1.0
Computing machine epsilon with respect to 1.000000000
Relative to 1.0000000E+00 SINGLE PRECISION Machine epsilon = 1.1920929E-07
While loop count was = 24

Relative to 1.0000000E+00 DOUBLE PRECISION Machine epsilon = 2.2204460E-16
While loop count was = 53

[mars<mech-4822>]$
```

To list the files now in your current directory, type:

```
$ ls -lrt
```

You should now have the program that you will use to perform the computations needed for Self Study Assignment 1 Question 1. A hard copy of the program output is stored in the file `machepts.out`.

When you are done with the program, to clean up files that are no longer needed, use:

```
$ rm *.o
```

```
$ rm *.exe
```


8 Lists of Linux Computers Available

The complete address of all of the following computers is obtained by appending `.cc.umanitoba.ca` to the name.

8.1 Linux Computers

Name	OS	Function
gaia	RHEL 8	login server
mars	RHEL 8	login server
venus	RHEL 8	login server
cc01	RHEL 8	compute server
cc02	RHEL 8	compute server
cc03	RHEL 8	compute server
cc04	RHEL 8	compute server
cc05	RHEL 8	compute server
cc06	RHEL 8	compute server
cc06	RHEL 8	compute server
cc07	RHEL 8	compute server
cc08	RHEL 8	compute server
cc10	RHEL 8	compute server
cc11	RHEL 8	compute server

The first four are accessed using ThinLinc (connecting to `ccldesktop.cc.umanitoba.ca`)
Can log into others using `ssh`. For example:

```
ssh cc03.cc.umanitoba.ca
```

To determine the load on all the IST Linux servers before choosing one for a long compute job, use the following command:

```
ccload
```

To determine the version of Linux your system is running:

```
cat /etc/system-release
```

This should produce something like:

```
Red Hat Enterprise Linux release 8.8 (Ootpa)
```