IC150P (Lab)

February 2014 Semester

Lab Manual

February 2014
Indian Institute of Technology Mandi

IC150P Lab
February 2014 Semester

Course Instructors:
Dr. Satyajit Thakor, Dr. Tulika P. Srivastava,
Dr. Padmanabhan Rajan, Dr. Rajendra Ray

List of Essential Linux, Editor, and Debugger Commands

Getting Started:
To login, type your username (Your roll number in lowercase letters) at the **Login:** prompt, and your password (same as your username) at the **Password:** prompt. Open a **shell** window from **Applications -> Accessories -> Terminal** menu.

Linux Shell Commands:

mkdir *dirname*	Make a directory *dirname*
rmdir *dirname*	Remove the empty directory *dirname*
cd *dirname*	Change the current working directory to *dirname*
cd ..	Change the current working directory to the parent directory
cd ~	Change the current working directory to your home directory
pwd	Show your current working directory
mv *srcfile destfile*	Rename the *srcfile* as *destfile*
cp *srcfile destfile*	Copy one file, *srcfile* to *destfile*
cp *srcfile(s) destDir*	Copy many file, *srcfile(s)* to *destDir*
rm -i *file(s)*	Delete *file(s)*
ls -l	List files in the directory with their details (size, time of creation)
cat *filename*	Print the contents of the file *filename* to the screen
gcc –o *prog prog.c*	Compile the C program in the file *prog.c* and create the executable file *prog*

gcc –g –o *prog prog.c* Compile the C program in the file *prog.c* and create the executable file *prog* that can be used for debugging with **gdb**

./*prog* Run the program *prog*

Special characters in file and directory names:
***** - wildcard matches any string; **?** - matches any single character; **~** - your home directory

Editor:

The course will use a version of Ubuntu (a popular GNU/Linux distribution). Ubuntu comes with a number of editors that you can use for the purposes of this course. Typically, an editor will be used to input and save your C programs. Some editors that you can use include Emacs, Vim, gedit, kate, and nano. You can use any of these editors for this course (please become familiar with one of them).

Starting programming:

Usually programming follows an edit-compile-debug cycle until the program works the way you want it to. First you need to edit your program (using a text editor), then compile it (here we use gcc). Compilation may give erros, which you need to debug. Once the program compiles successfully, execute it to see if produces the desired output. If not, the debugging-editing-compilation cycle continues until you get the desired output.

You can start this process using a small “Hello Mandi” program.

Programming Assignment 0: Linux Commands and Text Editors

1. Login
2. Editor of your choice (e.g., gedit, vim or emacs)
3. Linux commands:
    1. Creation of directory
    2. Rename a file
    3. Copy a file
    4. Delete a file

Problem 0.1 (Use an editor of your choice):

- Use the editor to type a letter to your friend describing your first semester experiences at IIT Mandi
- It must be at least three paragraphs with seven or more sentences each.
- Delete the second and last sentences of the second paragraph
- Move the first sentence of third paragraph as the second sentence of second paragraph - you should not retype
- Copy the fourth sentence of first paragraph as the last sentence of second paragraph.
- Now read the letter and edit (delete and insert) necessary words/sentences so that it sounds sensible.

Problem 0.2 (Linux Commands):

- Save the letter of Problem-0 as a “file”.
- Use the **cat** command to print the file to the screen.
- You want to send the same letter to five more friends

- Make five copies of the same (use the cp command)
- Open the copies and change the names of your friends
- Delete the file containing the letter for your third friend (use “rm” command)
- You wanted to store these files in a separate place that you could remember. So, create a directory called “Assignment0” and move these files to it.

Problem 0.3

Here is a small C program. Edit it using a text editor, and save it as mandi.c:

```
#include<stdio.h>

main()

{

printf(“Hello Mandi\n”);

}
```

Compile it using gcc:

```
$ gcc mandi.c -o mandi
```

This creates an executable file “mandi” inyour current directory.

Execute the program using the command:

```
$ ./mandi
```