Using Linux in the Lab: A How To Guide

<u>Opening the Terminal:</u>

❖ When on a lab computer:

Open the terminal: Ctrl, Alt, T

Close the terminal: Ctrl, D

❖ When using Mobaxterm:

Click on “Sessions” tab, then click on “New User” tab. In the box that opens, click on the “SSH” tab.

Enter the computer id for “Remote Host.”

TCML computer IDs:

- 132.68.176.162
- 132.68.177.108
- 132.68.177.143
- 132.68.177.52

Enter your username for “Specify User Name.” Then hit OK.

ONCE YOUR ON THE TERMINAL:

Login with your password (for the first time – then you can have your password remembered). **When entering your password, you will not see any characters show up on the screen.

<u>The Manual:</u>

There are 8 sections to the manual:

Section	*Contains*	*Description*
1	User Commands	Commands that can be run from the shell by a normal user (typically no administrative privileges are needed)
2	System Calls	Programming functions used to make calls to the Linux kernel
3	C Library Functions	Programming functions that provide interfaces to specific programming libraries
4	Devices and Special Files	File system nodes that represent hardware devices or software devices
5	File Formats and Conventions	The structure and format of file types or specific configuration files
6	Games	Games available on the system
7	Miscellaneous	Overviews of miscellaneous topics such as protocols, filesystems, etc.
8	System Administration Tools and Daemons	Commands that require root or other administrative privileges to use

You will probably be using section 1 (which is the default) most often.

- Using the Manual:
  - To search for a specific term, enter: man -k <search term>
  - To open a specific page from any section (other than section 1), enter: man <#> <page name>
    - Replace <#> with whichever number is relevant (2-8)
    - Replace <page name> with whichever page name you are looking for
  - To search a specific User Command, enter: man <command name>
    - It will only work if the command is found in Section 1
    - If the command does not show up in man and you are sure it is correct, try entering: help <command name>

- Reading the Manual:
  - SYNOPSIS - The manual will show you how to write the command with the following punctuation/styling: *THING should be replaced with whatever is appropriate (such as OPTION, FILE…)*
    - [THING] → THING is optional (you can leave it out and the command will still work)
    - <THING> → THING is required (the command will not work if this is left out)
    - THING… → THING can be repeated (for example you can use multiple options)
    - THING1 | THING2 → Use THING1 **OR** THING2 (you CANNOT use both)
  - DESCRIPTION – describes what the command does
    - It than lists all the short-form and long-form options and describes what they do

Command Formatting:

All commands follow the same basic structure: <command name> [-options] arguments.

Inputs are case sensitive and spacing often matters!

- **Command name**

The command name should be one word followed by a space.

To check whether a command is valid on the shell type: which <command name>. If a path is returned than the command name is valid and if not "command not found" will be returned.

- **Options**

NOT all commands need an option to work.

To check what options are available for a specific command use the manual (man).

- Short-form options
  The option[s] must have a dash - preceding it. There should be no space between the dash and the option letter[s]
  for example: <command name> -a -b -c
  OR you can combine multiple short-form options,
  for example: <command name> -abc (no space between the letters)
  both forms of the examples above will give the same result.
- Long-form options
  If it is a long-form option, it will be preceded my two dashes –
  For example: <command name> --alpha –beta
  You cannot combine multiple long-form options with two dashes preceding them (WRONG example: <command name> --alpha beta)

❖ Arguments

NOT all commands need an argument to work.

To check what arguments are available for a specific command/option use the manual (man)

- Arguments for commands
  The argument is a type of input that the command operates on. NOT all commands need an argument. Some commands can take an unlimited number of inputs, some can take a specific number, and some cannot have any.
- Arguments for options
  Sometimes an option can take its own arguments (inputs). When this is the case place a space between the option and the argument.

Some Basic Commands

Command	*Description*
echo	Prints command line arguments to standard output
date	Displays the current date and time
cal	Displays a calendar
cat	Combine files and send a joined file to standard output. Also useful for seeing the contents of one file in the terminal
history	Show commands previously entered (can also press the up arrow on the keyboard to see previous commands)
!!	Run the previous command again
![#]	Run the command from a specific line of the command history (replace [#] with whichever number line the command is on)

The Linux File System

Some standard files in the Root of the file tree:

Directory (Folder)	*Purpose*
/	The very start of the file tree (the Root). Holds everything else.
/tcmldrive	Contains databases relevant to the lab.
/home	Where the home directories for regular users are stored. For example: /home/user_name
/bin	Contains common Linux user commands (binaries). For example: cat, cal, date.
/sbin	Contains administrative commands (binaries) for the Root user.
/root	The home folder for the Root user (similar to the administrator on Windows).
/etc	Contains administrative configuration files. The format for many of these configurations can be found in section 5 of the manual.

Basic commands for navigating the file system:

Remember, you can use the man command to find out more

Remember, the commands are case sensitive

Command	*Description*
pwd	Show the absolute path (starting at the Root) of the current working directory (the one you are in) on the terminal.

cd [new location]	Change your current working directory to the new location (new directory/file) you entered in the []. If no location is provided, you will go to your user home directory: /home/user_name
ls [location]	Lists the contents of the optional location (directory) you entered. If no location was entered, get the contents of the current working directory. (If you want more information about list include the option -l)
find	Lists the all contents of below the point of the file tree it is told to start at (it will not just list the contents of a given directory but will list all the contents of the directory within the starting one to an infinite depth; for example: find / will list EVERYTHING). It is therefore best to use this command with options such as (the single dash is needed): -maxdepth 4 (will only go up to 4 levels from the starting directory) -name "*.txt" (search for items matching a specific name. The name should be enclosed by quotation marks. In the example the command will list all files under the starting directory that end with .txt) -iname "[item]" (works the same as -name but is NOT case sensitive) -exec <command> {} \: (will enable you to execute another command on the items that it lists. For example, find /Documents -exec cp {} ~/Desktop \: would copy all the items (directories and files) in your Documents folder to your Desktop folder)
touch <file>	Create a new blank file with the name you gave between the <>
mkdir <directory>	Create a new empty directory with the name you gave between the <>
rm <item>	Remove items. If you don't use an option you can only remove files. To remove a directory, you need to -r option. If you remove a directory using the -r option you WILL also remove all the files that are within it (so be careful).
wget <paste the link>	Copy links from the internet and move the data to the terminal. In order to PASTE on Linux right click on the mouse (ctrl ^v does not work)