CLI, BASH and Git notebook

UNIX and LINUX

They are operating systems similar to Windows. Some popular OS based on UNIX/LINUX are MacOS, Fedora, Ubuntu, Debian etc. For more info on the history on development of UNIX/LINUX can be found here

Every OS comes with a Kernel and Command line interpreter (Shell). Kernel handles allocation of time and memory to programs, also handles filestore and communications to and fro from the system. The shell acts as the interface between the user and the kernel. It interprets the commands provided by the user and prepares them for the kernel to execute them.

In UNIX/LINUX there are two major types of shells:

1) The Bourne shell. If you are using a Bourne-type shell, the default prompt is the $ character. There are again various subcategories for Bourne Shell :

- Bourne shell (sh)
- Korn shell (ksh)
- Bourne Again shell (bash)
- POSIX shell (sh)

2) The C shell. If you are using a C-type shell, the default prompt is the '\%' character. The different C-type shells follow:

- C shell (csh)
- TENEX/TOPS C shell (tcsh)

Note that we are not going to discuss in detail regarding the filesystem and UNIX OS in general, for people who are interested, please refer to this link for more information

Bash

Listing files and directories

When you first login, your current working directory is your home directory. Your home directory has the same name as your user-name, for example,

```
username@mypcname:~
```

and it is where your personal files and subdirectories are saved.

To find out what is in your home directory, type command `ls`

The ls command lists the contents of your current working directory. There may be no files visible in your home directory, in which case, the

ls does not, in fact, show all the files in your home directory, but only those ones whose name does not begin with a dot (.) Files beginning with a dot (.) are known as hidden files and usually contain important program configuration information. They are hidden because you should not change them unless you are very familiar with

To list all files in your home directory including those whose names begin with a dot, type ls -a

ls is an example of a command which can take options: -a is an example of an option. The options change the behaviour of the command. There are online manual pages that tell you which options a particular command can take, and how each option modifies the behaviour of the command.

Making Directories

We will make a subdirectory in your home directory to hold the files you will be creating and using in the course of this lecture and workshop. To make a subdirectory called "mini-astro-workshop" in your current working directory type $ mkdir mini-astro-workshop

To change your current directory to the directory you have just made, type $ cd mini-astro-workshop

Still in the "mini-astro-workshop" directory, type ls -a

As you can see, in the "mini-astro-workshop" directory (and in all other directories), there are two special directories called (.) and (..)

In UNIX, (.) means the current directory, so typing cd . means stay where you are (the "mini-astro-workshop" directory). This may not seem very useful at first, but using (.) as the name of the current directory will save a lot of typing in the long run.

(..) means the parent of the current directory, so typing cd .. , will take you one directory up the hierarchy (back to your home directory)

Note:

Note: typing cd with no argument always returns you to your home directory. This is very useful if you are lost in the file system.

Pathnames

Pathnames enable you to work out where you are in relation to the whole file-system. For example, to find out the absolute pathname of your home-directory.

Type cd to get back to your home-directory and then type pwd

The full pathname will look something like this - "/home/username/", which means that "username" (your home directory) is in the home sub-directory, which is in the top-level root directory called "/".

directory management:

- ls list files and directories
- ls -a list all files and directories
- mkdir make a directory
- cd dirname change to named directory
- cd change to home-directory
- cd ~ change to home-directory
- cd .. change to parent directory
- pwd display the path of the current directory

file management

- cp file1 file2 copy file1 and call it file2
- mv file1 file2 move or rename file1 to file2
- rm file remove a file
- rmdir directory remove a directory
- cat file display a file content
- less file display a file a page at a time
- head file display the first few lines of a file
- tail file display the last few lines of a file
- grep 'keyword' file search a file for keywords
- wc file count number of lines/words/characters in file

I/O

- command > file redirects standard output to a file
- command >> file append standard output to a file
- command < file redirect standard input from a file
- command1 | command2 & pipe the output of command1 to the input of command2
- cat file1 file2 > file0 concatenate file1 and file2 to file0
- sort sort data
- who list users currently logged in

file/text editors

- vi - For more details refer to [1] and [2]
- Nano - For more details refer to [1]
- emacs

others

- * match any number of characters
- ? match one character
- man command & read the online manual page for a command
- whatis command brief description of a command
- apropos keyword match commands with keyword in their man pages
- df reports on the space left on the file system
- du reports on the space used on the file system
- gzip filename compresses the size of a file
- zcat filename.gz will read gzipped files without needing to uncompress them first
- diff file1 file2 compares the contents of two files and displays the differences
- find searches through the directories for files
- history shell keeps an ordered list of all the commands that you have entered

Read the manual page for more information and usage by typing: man command

Screen

Screen or GNU Screen is a terminal command. In other words, it means that you can start a screen session and then open any number of windows (virtual terminals) inside that session. Processes running in Screen will continue to run when their window is not visible even if you get disconnected.

To check whether it's installed- $ `screen --version`

```
To install it: $ sudo apt install screen, on Ubuntu and Debian
               $ sudo yum install screen, on CentOS and Fedora
```

Screen commands:

screen -ls - lists all the screen that are active
screen -S session_name - to create a screen with certain name
screen -r - to reattach to a particular screen
screen -S -X quit - to quit and terminate the screen with that ID
Ctrl+a d To detach from the current screen

Examples: Creating a screen with session_name: maw, detach from the current screen, reattach and then finally delete it.

For managing the screen created:
Ctrl+a c Create a new window (with shell)
Ctrl+a " List all window
Ctrl+a 0 Switch to window 0 (by number)
Ctrl+a A Rename the current window
Ctrl+a S Split current region horizontally into two regions
Ctrl+a | Split current region vertically into two regions
Ctrl+a tab Switch the input focus to the next region
Ctrl+a Ctrl+a Toggle between the current and previous region
Ctrl+a Q Close all regions but the current one
Ctrl+a X Close the current region

References and Resources:

1. http://aendrizzi.gitpages.tpi.uni-jena.de/comphysi/
2. UNIX / Linux Tutorial for Beginners
3. How To Use Linux Screen
4. Linuxize: Linux Tips, Tricks and Tutorials
5.

Git

git clone <link to the repository> : to clone to the remote repository to your laptop
git add . : to add files which are to be committed and pushed to the remote repository
git status : to get status on differences between remote and local repository
git commit -m “message to commit” : to commit the changes
git push : to push updates to the remote repository from the laptop
git pull : to pull updates from the remote repository to the laptop

For more info on git refer to this link

Example:

Create a repository by going to your github account