

Introduction to Linux

Srikeerthana Kuchi

MRC-University of Glasgow Centre for Virus Research

09/10/2022

Unix - History

- Unix is an operating system developed by AT&T Bell labs (1969-1971)
- Collaborators worked on MULTICS (Multiplexed Information and Computing Service)
- Ken Thompson and others developed a much smaller OS called UNICS (Uniplexed Information Computing Service), later named to Unix
- Rewritten in C programming language in 1972 by Dennis Ritchie

Linux - History

- 1975 – Unix licensed to outside world (educational institutions, corporate companies, government agencies)
- Unix Version 5 – first distributed version as source code
- Linux – developed by Linus Benedict Torvalds, an open source Unix like OS in 1991
- Various Linux distributions include Ubuntu, openSUSE, Fedora, Red Hat etc.
- Mac OSX, mobile devices such as iOS, Android and Kindle use different variants of Unix

Breaking down the CLI (Terminal)

- VM – desktop edition of Ubuntu
- Open Terminal, which loads the command line interface (CLI) of the OS
- Terminal lets you interact with the shell

- `username@computername:~$`
 - `username` – name of the user
 - `computername` – name of the computer/host
 - `:` - separator
 - `~` - tilde symbol – shows that the user is working in the home directory
 - `$` - dollar symbol – user is a regular user (root user has a `#` symbol displayed)

Shell

- OS shell uses a CLI/GUI (graphical user interface) to access OS services
- Outermost layer surrounding the OS kernel and acts as an interface between the user and the system
- Common Shells
 - sh: Thompson shell (1971)
 - sh: Bourne shell (1977) (replaced previous shell)
 - csh: C shell (1979)
 - tcsh: Tabbed C shell (1979)
 - ksh: Korn shell (1982)
 - **bash: Bourne-Again shell (1987)**
 - zsh: Z shell (1990)

What are commands? Commands have **turquoise** background

- Commands are single words/words combined by “_” or “-” that are typed in CLI, received by the shell and processed by the OS
- Rules of command options and arguments
 - Commands are case sensitive
 - Options have to follow command
 - Options can start with a single hyphen and a character or a double hyphen and a word
 - Single character options can be combined
 - Some times options need a value (cut -f 1)
 - Argument can be one or more inputs
 - You can write more than one command separating with a semicolon (;)

ls -a (or) **ls - -all**
ls -a course_data/

Copy text – Ctrl + Shift + c
Paste text – Ctrl + Shift + v

Help!

- Manual pages: man (`man ssh`)
 - Most of the commands have manual pages
 - Gives summary of a command
 - Gives all available options
 - Gives examples
 - Gives developer information
- Information: info
 - More detailed information than man
 - Available in newer versions

Tweaks to remember

- Directories
 - Directories in Unix – equivalent to folders on a PC/Mac
 - Organised in a hierarchy
- Tab completion
 - Bash shell on most Linux distros supports tab completion
 - For example, to run the firefox command, type “fir”/”fire” and press tab for auto-completion
 - Double tapping tab provides options to choose; type fi and double tap to see all available options

Working directory – work as we learn

- Type

```
cd course_data/  
cd Introduction_to_Linux_Unix_Text_processing/
```

- Avoid errors by using tab completion as follows:

```
cd cou (Press 'tab' once)  
cd Int (Press 'tab' once)  
cd Introduction_to_L (Press 'tab' once)
```

Working directory and changing directory commands – work as we learn

- `pwd`
 - Print working directory (`pwd`)
 - This command returns the path of the current working directory
- `cd`
 - Changing directories
 - From present working directory to the specified directory
 - Example :
 - `cd Exercises/` – changes the working directory to the specified directory
 - `pwd`
 - `cd ..` – changes to the parent directory from which the previous `cd` command was typed in (to navigate up one directory level)
 - `cd /` - changes to the root directory
 - `cd` – changes to the home directory (specified by `~` symbol in the terminal)
 - `cd course_data/Introduction_to_Linux_Unix_Text_processing`

Listing files– work as we learn

- `ls`
 - Listing files
 - Directories – blue; files – white;
 - `ls -l` – long list files/directories
 - Information (from left to right):
File permissions, number of links, owner's name, group's name, number of bytes, last modified time, file/directory name
 - `ls -R` – recursive listing
 - `ls -a` – include hidden files

Creating/removing files and directories – work as we learn

- mkdir
 - Make directory – creates a directory in the working directory
 - `mkdir Practice` – creates a directory named Practice
 - `ls -l` – list all files/directories
- rmdir
 - Removes the specified directory
 - `rmdir Practice` – removes the Practice directory
- touch
 - Updates the access time of the specified file to the current time
 - Creates one if the file does not exist
 - `touch temp-file` – creates a file named temp-file; if the file exists, changes the access time to the current time
 - `ls -l` – check if the file is created/check the time

Alert:
Please remember
once a file or
directory is
deleted, it will
not go to
“Recycle bin” in
Linux and there is
no way you can
recover it.

Creating/removing files and directories – work as we learn

- `rm`
 - Removes files from the system
 - `rm temp-file` – removes the file temp-file
 - `-r` removes directories recursively
 - `-f` never prompt
- `cp`
 - `touch temp1` – creates a file named temp1
 - `cp temp1 temp2` – make a copy of temp1 as temp2
 - `-R` – recursive copy in case of copying directories

Moving and renaming files/directories – work as we learn

- `mv` – move/rename a file or a directory
 - `mkdir temp` – creates a directory named temp
 - `mv temp1 temp/.` – moves the file temp1 into the temp directory
 - `mv temp2 temp3` – renames temp2 to temp3

Create symbolic links to files – work as we learn

- `ln` – create links to a file or a directory
 - `ln -s temp/temp1 .` – creates a link to the specified file in the current directory
 - Useful in saving disk space

Helpful commands – work as we learn

- history
 - **history** – shows all the commands used in the current terminal session
- clear
 - **clear** – clears the terminal and provides a clean window to work on

Unix Text Processing

Srikeerthana Kuchi

MRC-University of Glasgow Centre for Virus Research

09/10/2022

Viewing files – work as we learn

- cat
 - Concatenate command combines files and prints onto standard output
 - `cat SARS-CoV-2.fa` – prints the file onto the screen
- more/less
 - Commands to view files
 - `more SARS-CoV-2.fa` – shows the contents of the file
 - Press Enter to view the file further
 - q to quit

Viewing files – work as we learn

- head/tail
 - Shows first and last 10 lines respectively
 - `head SARS-CoV-2.gb`
 - `tail SARS-CoV-2.gb`

File editors – work as we learn

- Non-graphical text editors
 - ed
 - emacs
 - vi
 - nano

nano – work as we learn

- nano
 - Graphical editor
 - Commands executed through keyboard
 - Modifier is the Ctrl key
 - nano – opens a standard blank nano window
 - Options
 - Ctrl + X – exits nano; returns to command line
 - Ctrl + O – writes the contents of the text buffer to file
 - Ctrl + R - reads file
 - Ctrl + T – opens the file navigator

Text processing in Linux – work as we learn

- cut
 - Command line utility to cut sections from a file
 - `cut -c1-10 SARS-CoV-2.fa` – cut 10 characters from each line of the file
 - `-d` – based on the delimiter
 - `-f` – based on the field number
- `head human_viruses.txt` – viruses that have human hosts, genbank ids and genome length.
 - `cut -d'|' -f2 human_viruses.txt` – cuts the file by delimiter “|” and prints 2nd column onto standard output

Text processing in Linux – work as we learn

- sort
 - Sorts the input
- Few options:
 - -t: field separator
 - -n: numeric sort
 - -k: sort with a key (field)
 - -r: reverse sort
 - -u: print unique entries
- `sort -t'|' -nrk6 human_viruses.txt` – sorts the human viruses by the genome length field, delimited by “|” symbol

Text processing in Linux – work as we learn

- grep
 - Searches the input for a given pattern/text
- Few options:
 - -A: after context
 - -B: before context
 - -C: before and after context
 - -c: count
 - -l: file with match
 - -i: ignore case
 - -o: only match
 - -v: invert match
 - -w: word match

Text processing in Linux – work as we learn

- `grep "Hepatitis" human_viruses.txt`
- `grep -v "Hepatitis" human_viruses.txt`
- Linux commands support BRE - special characters - pattern in data
- `grep "Torque teno midi virus . DNA" human_viruses.txt`
- `ls -l temp?`
- `ls -l temp*`

“.”- dot character that matches any single character at a given position

“?”- question mark character that matches one occurrence

“*”- asterisk matches zero or more occurrences of the preceding character

Text processing in Linux – work as we learn

- Pipes
 - Powerful and efficient way to combine commands.
 - “|” in Linux acts as a link between commands, redirects output of first command as an input to the next
 - Nest as many commands as we would like to
 - `sort -t”|” -nk6 human_viruses.txt | head -10` – prints smallest 10 human viruses

Exercise: print largest 10 human viruses

Text processing in Linux – work as we learn

- `wc`
 - Word count – counts lines, words or characters
 - `wc -l outbreak.csv`
 - `cat outbreak.csv | wc -l`
- `uniq`
 - Extracts unique lines from the input
 - Used in combination with sort command
 - `cut d"," -f3 outbreak.csv | sort | uniq` – prints the unique list of countries that has had an outbreak in 2022
 - `-c` - gives a count of the values

Exercise: Count the number of countries that has had an outbreak in 2022

I/O control in Linux – work as we learn

- Output of a command – sent to standard output i.e terminal
- To redirect to a file, use the “>”
 - `ls > list` – creates a file named “list” with all the file names in the directory; if exists, overwrites it; >> to append
 - `cat list` – prints the contents of the file “list”
- To redirect standard error, use “2>”
- To redirect both stdout and stderr, use “&>”

Process control – work as we learn

- Commands that take longer – put to background by appending the command with “&”
- Completion indicated by “Done”
- `gzip list &` - compresses the file “list” in the background
- `jobs` – list of currently running jobs in the terminal

Command line shortcuts

- Up/Down arrows: Previous commands
- !!: Reruns previous command
- Tab: Auto complete
- Tab+Tab: All available options
- Ctrl+a: Move cursor to start of line
- Ctrl+e: Move cursor to end of line
- Alt+: Alternates between terminals
- Ctrl+l: Clear screen ((or Command+k on Mac)
- Ctrl+c: Terminates the running program
- Ctrl+z: Suspends the running program
- Ctrl+w: Removes a previous word
- Ctrl+d: Logout
- Ctrl+u: Removes till the beginning

Exercises

1. Open a new terminal and navigate into Exercises directory.
2. Extract first 15 lines from the file “HM067743.1_cds_ADQ37313.1_1.fa” and save the output into “output.fa”
3. How many fasta files are there in the directory?
4. Extract all header lines from the file all.fa
5. How many sequences are there in the file all.fa?
6. Get the list of countries (excluding multi country outbreaks) that had an outbreak in 2022 (Copy outbreak.csv into Exercises directory)
7. Find the number of outbreaks (exclude multi country outbreaks using invert match grep (-v)) in each month of 2022 (Input: outbreak.csv).