Introduction to Linux

Srikeerthana Kuchi
13/05/2024
Contact: Srikeerthana.Kuchi@glasgow.ac.uk

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

Login/re-login

- Open (type in the search box) “Oracle VM Virtualbox” application on your windows machine
- Click on the pre-installed VM running Ubuntu 20.04
- Spacebar to activate login screen
- Password: manager
- Open “Terminal” application within the Ubuntu OS

First things first…

- Host (Right Ctrl) & F – Switch between full screen mode and scaled mode
- Ctrl & Shift & ++ – increase text size
- Copy shortcut: Ctrl & Shift & c
- Paste shortcut: Ctrl & Shift & v
- Ctrl+C – to end a process in the current terminal

Breaking down the CLI (Terminal)

- Terminal loads the command line interface (CLI) of the OS
- It 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 must 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
  - Sometimes 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 <filename>

Help!

- Manual pages: man (man ls)
  - enter/scroll/page down
  - ‘q’ to quit
  - 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
  - Organized 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

Copying data to current directory – work as we learn

- Using tab completion, type

```
cp -R course_data/Linux_data .
```

  - command = cp
  - option = -R
  - Two Arguments:
    - course_data/Linux_data
    - “.” – represents current working directory

Listing files– work as we learn

- ls
  - command used for 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
- The number following total is a representation of the cumulative disk space, in blocks, allocated for the files being displayed

Working directory and changing directory commands – work as we learn

- clear
  - Ctrl+l – clears the terminal screen
- history
  - shows all the commands used in the current terminal session
- 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 Linux_data/– 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 Linux_data/

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
  - ls -l

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
  - ls -l
  - Useful in saving disk space

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.fa
  - tail SARS-CoV-2.fa

Editing files – 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 "|" -k6n 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 "Influenza D" human_viruses.txt
- grep -v "Influenza D" human_viruses.txt

- 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 2024
  - -c - gives a count of the values

- Exercise: Count the number of countries that has had an outbreak in 2024

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 100 lines of the fasta sequence from the file “NC_048217.1_cds_YP_009824978.1_1.fa” and save the output into a new file “output.fa”
3. How many fasta files are there in the “Exercises” directory?
4. Extract all fasta header lines from the file “Betacoronaviruses.fa”.
5. How many fasta sequences are present in the file “Betacoronaviruses.fa”?
6. Copy the file “outbreak.csv” file from the “Linux_data” directory into “Exercises” directory.
7. From the file “outbreak.csv”, get the list of countries that had at least two outbreaks in 2024.
8. Find the 99th line of the file “NC_039207.1_cds_YP_009513009.1_2.fa” using only the ‘tail’ and ‘head’ command.
9. In the file “NC_039207.1_cds_YP_009513008.1_1.fa”, count the number of lines containing the sub-sequence “GGGG”.
10. How do you stop a process with pid 5678?
11. Re-execute your previous command using a keyboard shortcut.
12. Create a new directory named ‘Trial’ and move all files whose names begin with ‘NC_009021.1_cds_YP’ into the ‘Trial’ directory.
13. Which is the command used to remove or delete a file without a confirmation prompt?
14. _____ command is used to count the total number of lines, words and character in a file.
15. From the headers of all the fasta files (except the Betacoronaviruses.fa) present in the Exercises directory, extract fields 2 to 7 using the space delimiter.
16. Extract first 10 lines of “outbreak.csv”, sort them and save as “outbreak_1.csv”. Extract first 20 lines of outbreak.csv, sort them and save as “outbreak_2.csv”.
17. Extract common lines between the files “outbreak_1.csv” and “outbreak_2.csv” (use “comm” command, type “man comm” to get information) and save the output into a file named “output_1”.
18. Which command would you use to find the word “pattern” from the file, “filename.txt”? Using that command, extract the “BioProject” information from the file SARS-CoV-2.gb.
19. Use the file SARS-CoV-2.gb to extract protein identifiers (“protein_id”). Remove the pattern “protein_id” from the output.
20. Which option with the command “rm” is required to remove a directory?

Solutions for Exercises

1. cd Linux_data/Exercises/
2. head -100 NC_048217.1_cds_YP_009824978.1_1.fa > output.fa
3. ls *.fa | wc -l
4. grep “^>” Betacoronaviruses.fa
5. grep "^>" Betacoronaviruses.fa | wc -l
6. cp ../outbreak.csv .
7. cut -d, -f3 outbreak.csv | sort | uniq -c | grep -v 1
8. head -100 NC_039207.1_cds_YP_009513009.1_2.fa | tail -1
9. grep GGGG NC_039207.1_cds_YP_009513008.1_1.fa | wc -l
10. kill 5678
11. !!
12. mkdir Trial; mv NC_009021.1_cds_YP_00* Trial/.
13. rm –f <filename>
14. wc
15. grep "^>" *.fa | grep -v Betacoronaviruses.fa | cut -d" " -f2-7
16. head -10 outbreak.csv | sort > outbreak_1.csv; head -20 outbreak.csv | sort > outbreak_2.csv
17. comm -12 outbreak_1.csv outbreak_2.csv > output_1
18. grep "BioProject" SARS-CoV-2.gb
19. grep "protein_id" SARS-CoV-2.gb | cut -d"=" -f2
20. rm –r <dirname>