Laboratory 1: installing Ubuntu LINUX & BASH basics

Booting your computer

You may, at your option, install Ubuntu LINUX 22.04.1 ('Jammy Jellyfish') directly on your computer, an external drive, in a virtual machine (instructions below), or on an inexpensive Single–Board Computer. If you (very foolishly) choose to use a USB flash disk for the entire class, be sure to save any files to an area of the disk that is designated for document storage or to an external device—otherwise they will be deleted when you shutdown. In addition, you may use the Ubuntu server version (live or legacy) for the full CLI experience, or the Ubuntu Desktop or Lubuntu Desktop for the GUI experience.

For VirtualBox virtual machines[1].

(1) Download the current version of VirtualBox from https://www.virtualbox.org/

(2) Install VirtualBox by following the provided instructions.

(3) Open VirtualBox.

(4) Create a new virtual machine ('Machine' » 'New...').

   (a) Provide a name, select 'Linux', and select 'Ubuntu (64 bit)'.

   (b) Give the virtual machine as much RAM as you can reasonably spare (the green portion of the selector bar, at least 2048 MB).

   (c) Create a virtual hard drive (use 'VDI' format, dynamically allocated, at least 32 GB).

(5) Modify the settings of the newly created virtual machine ('Machine' » 'Settings...').

   (a) Increase the number of processors as much as you reasonably can (the green portion of the selector bar).

   (b) Increase the video memory to the maximum (128 MB).

(6) Download Lubuntu Desktop version 22.04.1 LTS from https://lubuntu.me/downloads/

(7) Start the virtual machine ('Machine' » 'Start').

(8) Select the downloaded Lubuntu .iso file.

(9) Follow the installer instructions.

   (a) Choose a language you are fluent in.

   (b) Start Lubuntu.

   (c) Double click on the 'Install Lubuntu 22.04 LTS' icon.

   (d) Choose a language you are fluent in (again).

   (e) Select your time zone.

[1] For x86–based (Intel or AMD processor) computers. VirtualBox can run on ARM–based Macs, but does not work well (according to the internet).

(f) Select your keyboard layout (although this can be changed later, it causes a lot of frustration if not done correctly now).

(g) Select 'Erase disk' (this refers to your virtual disk).

(h) Choose your login and computer names as well as a password.

(i) Confirm your settings and install.

(j) Restart after the install is complete (do not worry about removing the virtual installer disk, just press 'Enter').

(10) After rebooting and logging into the virtual machine, you will need to install some additional software for the virtual machine to work properly.

(a) Open the terminal by clicking on the Lubuntu icon (a blue and white hummingbird; lower left of the screen), and selecting 'System Tools' » 'QTerminal'. You may wish to add the terminal to the quick launcher since you will be using it a lot (drag the icon from the menu to the quick launch area and release when a green plus symbol appears).

(b) Type `sudo apt update` to update the package information. Enter your password when prompted.

(c) Type `sudo apt upgrade` to install new software versions. Agree to the upgrade when prompted.

(d) Type `sudo apt autoremove` to remove old versions that are no longer needed. Agree when prompted.

(e) Type `sudo apt install build-essential openssh-server` to install needed packages. Agree when prompted.

(11) There are a set of kernel modules provided by VirtualBox to better integrate the virtual machine with the host system. These are not required, but make life much easier.

(a) Load the installer ('Devices' » 'Insert Guest Additions CD Image...'; select 'Open in file manager' when prompted).

(b) Type `cd /media/$USER/VBox_GAs_6.1.42/` to change to the guest additions installer.

(c) Type `sudo su` to switch to root user (this is dangerous and should only be done when absolutely necessary).

(d) Type `./VBoxLinuxAdditions.run` to install the guest additions.

(12) Type `reboot` to reboot the virtual machine.

For UTM virtual machines[1].

(1) Download the current version of UTM from https://mac.getutm.app

(2) Install UTM by following the provided instructions.

(3) Create a virtual machine with the '+' button.

(a) Give the virtual machine as much RAM as you can reasonably spare (at least 2048 MB).

[1] For ARM–based MacOS computers.

(b) Create a virtual hard drive (at least 32 GB).

(4) Follow steps (6)–(10)(e) above.

(5) Item `sudo apt install spice-vdagent spice-webdavd` to install additional needed packages. Agree when prompted.

(6) Type `sudo reboot` to reboot the virtual machine.

LOGIN. Login and explore a little. Ubuntu will most likely prompt you to update your software. Please do so. It is good practice to run an update before each lab session (type `sudo apt update; sudo apt upgrade; sudo apt autoremove` in the terminal). You may wish to increase the size of your virtual screen by right clicking on the screen icon in the VirtualBox control bar at the bottom of the screen.

BASH commands

There are many BASH terminal commands to learn. The following is an abbreviated list of the most useful ones. To learn about each command use `man` followed by the command name.

`apt` == install and manage software

`cd` == change directory

`chmod` == change file mode

`chown` == change file ownership

`cp` == copy a file

`file` == determine file type

`find` == find a file's location (by name, type, size, etc.)

`gzip` == compress or expand files

`less` == read a file one screen at a time

`ls` == list directory contents

`make` == build an application using a MakeFile

`man` == read the manual for a command

`mkdir` == make a directory

`mv` == move a file

`pwd` == print working directory

`rm` == remove a file

`scp` == secure file copy

`sftp` == secure file transfer protocol (connect to another computer)

`ssh` == secure shell hyper–terminal (connect to another computer)

`sudo` == use a pseudonym when executing a command

`tar` == tape archiver (the tape drive is optional on modern computers)

`wget` == web getter

`xargs` == extended arguments

`xz` == compress or expand files

Perhaps the most useful BASH command of all is not really a command: the <tab> key triggers the auto–compete function. Try typing a few letters (e.g. 'fu') press <tab> twice and a list of possible completions in the current directory and your PATH appear—if you have typed enough letters to uniquely identify something it is completed automatically with just one press of the <tab> key. This function can save you hours of typing and is a good way of preventing a number of common mistakes.

A close runner up to <tab> are the <up–arrow> and <down–arrow> keys: they store a history of the commands that you have typed so that you can run something again, or a slight modification, without retyping. Be careful. Everything that you do is stored in the '.bash_history' file, so if you do something bad everyone will know (e.g. if you are going to shut off the electric fences that keep the dinosaurs in their pens, and you do not want anyone to know it was you, it is best to turn off the history feature first).

Tasks

(1) Read all of the man pages for the programs listed on page B3.

(2) In the terminal, type `ls -lh /var/` and answer question (5).

(3) In the terminal, type `ls -lh /var/log` and answer question (6).

(4) In the terminal, type `less /var/log/syslog` and read through the file. Answer question (7).

(5) In the terminal, type `grep -a 'Started Session' /var/log/syslog | grep -c "$(date +'%b %d' | perl -pe 's/ 0/ /')"` and answer question (8).

(6) In the terminal, type `ls -lh /etc/ssl` and answer question (9).

(7) In the terminal, type `ls -lh /etc/ssl/certs/DigiCert_High_Assurance_EV_Root_CA.pem` and answer question (10).

(8) In the terminal, type `cat /etc/ssl/private/ssl-cert-snakeoil.key` and answer question (11).

(9) In the terminal, type `sudo cat /etc/ssl/private/ssl-cert-snakeoil.key` type your password when/if prompted and answer question (12).

(10) In the terminal, type `sudo cp /etc/ssl/private/ssl-cert-snakeoil.key $HOME/snake.pem` type your password when/if prompted and answer question (13).

(11) In the terminal, type `sudo chown $USER:$USER snake.pem` and answer question (14).

(12) Change the permissions of the file 'snake.pem' so that the owner and the group can read and write to the file, but no one else. Answer question (15).

Questions (https://forms.gle/B39m2cwQb3HJZyBX6)

(1) How many processors does your computer have (if you are using virtualization, answer for both the computer and the virtual machine)?

(2) How much RAM does it have (if you are using virtualization, answer for both the computer and the virtual machine)?

(3) How much RAM does Ubuntu require (just after startup without opening any programs)?

(4) How does this compare to your other operating system(s)?

(5) The 'log' directory:

  (a) Who owns the 'log' directory?
  (b) What group does the directory belong to?
  (c) Who can read files in the 'log' directory?

(6) The 'syslog' file:

  (a) Who owns the 'syslog' file?
  (b) What group does the 'syslog' file belong to?
  (c) Who can read the 'syslog' file?
  (d) Who can write to it?
  (e) Who can execute it?

(7) What is the purpose of the 'syslog' file?

(8) Your 'syslog' file:

  (a) How many times has the start of a session been logged today?
  (b) Explain what each element of the command does.
  (c) How would you modify the command to determine the number of logging events from yesterday?

(9) The 'certs' and 'private' directories:

  (a) Who owns the 'certs' directory?
  (b) What group does the 'certs' directory belong to?
  (c) Who can read files in the 'certs' directory?
  (d) Who owns the 'private' directory?
  (e) What group does the 'private' directory belong to?
  (f) Who can read files in the 'private' directory?

(10) For task (7):

(a) Who owns the ‘DigiCert_High_Assurance_EV_Root_CA.pem’ file?
(b) What group does the file belong to?
(c) Who can read the file?
(d) Who can write to it?
(e) Who can execute it?
(f) Is ‘DigiCert_High_Assurance_EV_Root_CA.pem’ a real file? Explain.

(11) For task (8), did the command work? Explain why or why not.

(12) For task (9), did the command work? Explain why or why not.

(13) For task (10):

(a) Explain what each element of the command does.
(b) Who owns the ‘snake.pem‘ file?
(c) What group does the ‘snake.pem’ file belong to?
(d) Who can read the ‘snake.pem’ file?
(e) Who can write to it?
(f) Who can execute it?

(14) For task (11):

(a) Now who owns the ‘snake.pem’ file?
(b) What group does the ‘snake.pem’ file belong to?
(c) Who can read the ‘snake.pem’ file?
(d) Who can write to it?
(e) Who can execute it?

(15) For task (12):

(a) What command did you use?
(b) How can you test to see if the command was successful?

Due at the start of class January 31.