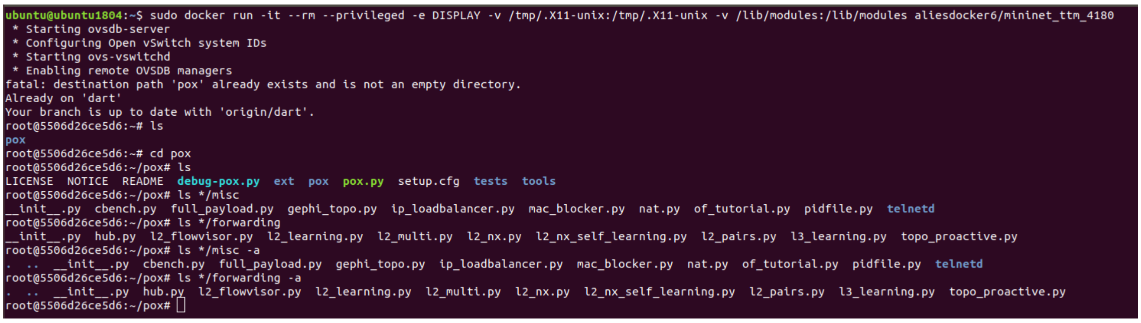
TTM4180 - Applied Networking
Lab 1: Virtualization and Linux

By S. Xiao Fernández Marín

T1: Use *cd, ls, cat* and find the content of the directory *misc* and *forwarding* inside the pox directories.

Q1: Which type of files do you find in *misc* and *forwarding*? Include a screenshot of the output from *ls*.



```
ubuntu@ubuntu1804:~$ sudo docker run -it --rm --privileged -e DISPLAY -v /tmp/.X11-unix:/tmp/.X11-unix -v /lib/modules:/lib/modules aliesdocker6/mininet_ttm_4180
 * Starting ovsdb-server
 * Configuring Open vSwitch system IDs
 * Starting ovs-vswitchd
 * Enabling remote OVSDB managers
fatal: destination path 'pox' already exists and is not an empty directory.
Already on 'dart'
Your branch is up to date with 'origin/dart'.
root@5506d26ce5d6:~# ls
pox
root@5506d26ce5d6:~# cd pox
root@5506d26ce5d6:~/pox# ls
LICENSE  NOTICE  README  debug-pox.py  ext  pox  pox.py  setup.cfg  tests  tools
root@5506d26ce5d6:~/pox# ls */misc
__init__.py  cbench.py  full_payload.py  gephi_topo.py  ip_loadbalancer.py  mac_blocker.py  nat.py  of_tutorial.py  pidfile.py  telnetd
root@5506d26ce5d6:~/pox# ls */forwarding
__init__.py  hub.py  l2_flowvisor.py  l2_learning.py  l2_multi.py  l2_nx.py  l2_nx_self_learning.py  l2_pairs.py  l3_learning.py  topo_proactive.py
root@5506d26ce5d6:~/pox# ls */misc -a
.  ..  __init__.py  cbench.py  full_payload.py  gephi_topo.py  ip_loadbalancer.py  mac_blocker.py  nat.py  of_tutorial.py  pidfile.py  telnetd
root@5506d26ce5d6:~/pox# ls */forwarding -a
.  ..  __init__.py  hub.py  l2_flowvisor.py  l2_learning.py  l2_multi.py  l2_nx.py  l2_nx_self_learning.py  l2_pairs.py  l3_learning.py  topo_proactive.py
root@5506d26ce5d6:~/pox# 
```

Figure 1: Files in misc and forwarding directories

In the Figure 1 we can see that we have python files in both directories and a folder called telnetd.

Using the command *cat* we can see that there are programs to manage packages and switches.

Q2: Use *’$ man mn’* and explain in your own words what *mn* is and what it is used for.

The mn command creates a Mininet network, it can create:

1. **Parametrized topologies:** Using the command *–topo=TOPO* and utilizing the flags linear, minimal, reversed, single, torus or tree. Where linear=LinearTopo, torus=TorusTopo, tree=TreeTopo, single=SingleSwitchTopo, reversed=SingleSwitchReversedTopo and minimal =MinimalTopo.
2. **Invoke the Mininet CLI:** Adding the command *mn*, it opens the command-line interface, ”*mininet >*”.
3. **Run tests:** *–test=,TEST/* using as TEST: cli, build, pingall, pingpair, iperf, all, iperfudp or none.

Q3: What is *’the Shell’*, in the context of Linux, and what is it used for?

The shell in Linux is an interpreted language where text lines are read (ending in \n, analyzed and processed. The lines are read from:

- Keyboard input
- Shell-script
- Arguments with options bash -c. (i.e. *ls -a*)

It manages the interaction between the user and the OS, requesting an input, interpreting that input for the OS and managing any output the OS could give.

Q4: What is the relation between *sh* and *bash*? What does *bash* stand for?

- **sh:** A shell used as one of the first implementations of Unix. It has this name as the developer was called Stephen Bourne. Consequently, multiple shells have been arise:zsh (Z shell), ash (almquist shell), bash (Bourne again shell), dash (Debian almquist shell) or ksh (Korn shell).

- **bash:** Bourne Again SHell is the interpreter of commands assigned for the users of the OS Linux. It is the most used shell and it is characterized for having more functionalities than the first ones Bourne Shell added, like for example $HOME/.bashrc and .bash_profile.

The main difference between bash and sh is that bash supports features of sh and provides more extensions on top of that.

Q5: All global Linux commands like *ls* and *cd* are binary files located on the computer somewhere. How does Linux figure out where to look for these files when you execute them? *(hint: google Linux PATH).*

The files are stored in the directories */usr/local/sbin: /usr/local/bin: /usr/sbin: /usr/bin: /sbin: /bin.* When we type a command in the shell, it uses the variable $PATH to know where to look for the executable files, as it has the list of directories that the bash shell searches when we write it in the terminal.

Q6: What is the advantage of using this virtual machine in our labs?

It includes all the tools needed for completing the lab so every student has the same setup, which is easier for us as we do not have to install everything from zero. This could be bad for us as maybe we install a version of a program that is not the right one. It also saves a lot of time because you do not have to set up and configure the environment.

One thing I like is that you can also save the machine state, so if you want to save the state of the VM you can, thing that it would be more difficult to achieve in a dual boot.