Rohan Kapur
Lab 1
CSE 150 Spring 2023
Due: 4/22/23

Prelab

1. The “id” command will show what groups you’re a member of
2. The “$?” environment variable holds the exit code of the previous command.
3. The “Ctrl-Z” key combination will suspend a running process and place it in the background.
4. The command “uname -rn” can be used to retrieve the current kernel version and the network “nodename” for the machine.
   (Ref: https://www.cyberciti.biz/faq/howto-check-linux-kernel-version/)
5. The path “.” is the current working directory, “..” is the parent directory, and “~” is the user’s home directory. The path “/” when not preceded by anything refers to the system’s root directory.
6. The “pid” is the current process ID. The command “ps” can be used to retrieve the ID for a process (Ref: https://linuxhandbook.com/find-process-id/)
7.
```
cat /etc/passwd | cut -d ':' -f 7
```
   (Ref: https://unix.stackexchange.com/a/312281/184297)
8. “sudo” executes the command specified as the superuser, whereas “su root” *switches* the current user to the superuser
9. You would schedule a “cronjob” by specifying the command to be run using the “crontab” command, which installs a “crontab” file under the “/usr/lib/cron/tabs/” directory that specifies the job to be executed on the specified schedule
10.
```
IFS=$'\n' # Set the "internal field separator" variable
          # which the shell uses to split words in loops
for file in `ls`; do
    for line in `cat $file | sed -n 'n;p'`; do
        echo "$file:$line"
    done
done
unset IFS # Unset the IFS variable to restore the default
          # word-splitting behavior
```
    References:
    - https://unix.stackexchange.com/q/184863/184297 (IFS variable)
    - https://stackoverflow.com/a/41745739/5661257 (select even-numbered lines using “sed”)

Lab

1. Refer to file “RohanKapur-topo.py”

2. The following is a screenshot of my *dump* and *pingall* command outputs for my python topology script

```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=573010>
<Host h2: h2-eth0:10.0.0.2 pid=573012>
<Host h3: h3-eth0:10.0.0.3 pid=573014>
<Host h4: h4-eth0:10.0.0.4 pid=573016>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None,s1-eth4:None pid=573021>
<Controller c0: 127.0.0.1:6653 pid=573003>
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
```

   The *dump* output shows the IP addresses and process IDs of the four virtual host processes spawned, as well as the IP address of localhost (lo) and the ethernet (s1-eth) interfaces on the one virtual network switch process (s1) the four hosts are connected to. There are also the process IDs for s1 and controller (c0) for the OpenFlow container. The controller gives the IP address and port (6653) and the process ID associated with the controller process.
   The *pingall* output shows the result of sending ping packets all four hosts, which in this case was successful for all 4*3=12 packets sent since all of them were acknowledged by the receiving hosts.

3. The *iperf* command was hanging for me, but running *iperfudp* generated the following output:

```
mininet> iperfudp
*** Iperf: testing UDP bandwidth between h1 and h4
*** Results: ['10M', '10.5 Mbits/sec', '10.5 Mbits/sec']
```

   This shows that the connection bandwidth between hosts h1 and h4 is 10.5 Mbits/sec in both directions for a 10MB packet sent

4. The “of” filter wasn’t qworking for me since the OpenFlow packets all had “OFPT_PACKET_X” types, so I had to change the filter to “openflow_v1” to filter the OpenFlow packets

   A. After running the command *h1 ping -c 5 h2* between hosts h1 and h2, the following 7 *of_packet_in* messages showed up

303	20.458438222	10.0.0.1	10.0.0.2	OpenFlow	184	Type: OFPT_PACKET_IN
304	20.458546267	10.0.0.1	10.0.0.2	OpenFlow	190	Type: OFPT_PACKET_OUT
310	20.458900861	10.0.0.2	10.0.0.1	OpenFlow	184	Type: OFPT_PACKET_IN
311	20.458941904	127.0.0.1	127.0.0.1	OpenFlow	148	Type: OFPT_FLOW_MOD
312	20.458946613	10.0.0.2	10.0.0.1	OpenFlow	190	Type: OFPT_PACKET_OUT
320	21.459687698	10.0.0.1	10.0.0.2	OpenFlow	184	Type: OFPT_PACKET_IN
321	21.459873829	127.0.0.1	127.0.0.1	OpenFlow	148	Type: OFPT_FLOW_MOD
322	21.459897246	10.0.0.1	10.0.0.2	OpenFlow	190	Type: OFPT_PACKET_OUT
351	25.478705226	5a:22:2…	22:ec:99:d…	OpenFlow	128	Type: OFPT_PACKET_IN
352	25.478728851	22:ec:9…	5a:22:2f:1…	OpenFlow	128	Type: OFPT_PACKET_IN
353	25.478806978	127.0.0.1	127.0.0.1	OpenFlow	148	Type: OFPT_FLOW_MOD
354	25.478814229	5a:22:2…	22:ec:99:d…	OpenFlow	134	Type: OFPT_PACKET_OUT
355	25.478818729	127.0.0.1	127.0.0.1	OpenFlow	148	Type: OFPT_FLOW_MOD
356	25.478821687	22:ec:9…	5a:22:2f:1…	OpenFlow	134	Type: OFPT_PACKET_OUT
362	25.479216490	22:ec:9…	5a:22:2f:1…	OpenFlow	128	Type: OFPT_PACKET_IN
363	25.479226032	5a:22:2…	22:ec:99:d…	OpenFlow	128	Type: OFPT_PACKET_IN
364	25.479443497	127.0.0.1	127.0.0.1	OpenFlow	148	Type: OFPT_FLOW_MOD
365	25.479447830	22:ec:9…	5a:22:2f:1…	OpenFlow	134	Type: OFPT_PACKET_OUT
366	25.479451039	127.0.0.1	127.0.0.1	OpenFlow	148	Type: OFPT_FLOW_MOD
367	25.479452789	5a:22:2…	22:ec:99:d…	OpenFlow	134	Type: OFPT_PACKET_OUT

The pink highlighting indicates *ICMP* (Internet Control Media Protocol) being used (notice how the source and destination are IP addresses), whereas the gold highlighting indicates *ARP* (Address Resolution Protocol) being used by OpenFlow (note how the source and destination are MAC addresses in this case). The ICMP is used for sending and receiving ping messages, whereas ARP is used for tying IP addresses to a host's MAC (media access control) address in a local area network.

B. The source and destination IP addresses of the *of_packet_in* and *of_packet_out* entries alternate between 10.0.0.1 (matching host *h1*) and 10.0.0.2 (hatching host *h2*), and the same applies to the destination IP address for both entry types.

303	20.458438222	10.0.0.1	10.0.0.2	OpenFlow	184	Type: OFPT_PACKET_IN
304	20.458546267	10.0.0.1	10.0.0.2	OpenFlow	190	Type: OFPT_PACKET_OUT
310	20.458900861	10.0.0.2	10.0.0.1	OpenFlow	184	Type: OFPT_PACKET_IN
311	20.458941904	127.0.0.1	127.0.0.1	OpenFlow	148	Type: OFPT_FLOW_MOD
312	20.458946613	10.0.0.2	10.0.0.1	OpenFlow	190	Type: OFPT_PACKET_OUT

C. The following 76 entries show up when I use the display filter "icmp && not of" and run the *pingall* command among all four hosts. These all seem to ping entries between the different hosts.

302	20.457721783	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0xc55a, seq=1/256…
306	20.458738523	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0xc55a, seq=1/256…
307	20.458740648	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0xc55a, seq=1/256…
308	20.458741815	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0xc55a, seq=1/256…
309	20.458813817	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) reply id=0xc55a, seq=1/256…
314	20.459034907	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) reply id=0xc55a, seq=1/256…
319	21.459278019	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0xc55a, seq=2/512…
324	21.460127587	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0xc55a, seq=2/512…
325	21.460159504	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) reply id=0xc55a, seq=2/512…
326	21.460325718	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) reply id=0xc55a, seq=2/512…
327	22.460916266	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0xc55a, seq=3/768…
328	22.461208567	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0xc55a, seq=3/768…
329	22.461240317	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) reply id=0xc55a, seq=3/768…
330	22.461243984	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) reply id=0xc55a, seq=3/768…
331	23.462968575	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0xc55a, seq=4/102…
332	23.463004076	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0xc55a, seq=4/102…
333	23.463030536	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) reply id=0xc55a, seq=4/102…
334	23.463033911	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) reply id=0xc55a, seq=4/102…
335	24.490992154	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0xc55a, seq=5/128…
336	24.491016447	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0xc55a, seq=5/128…
337	24.491040697	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) reply id=0xc55a, seq=5/128…
338	24.491041989	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) reply id=0xc55a, seq=5/128…

5140	1643.723415612	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0xdc2b, seq=1/256, ttl=
5144	1643.726826840	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0xdc2b, seq=1/256, ttl=
5145	1643.726836257	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0xdc2b, seq=1/256, ttl=
5146	1643.726851590	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0xdc2b, seq=1/256, ttl=
5147	1643.727006670	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) reply id=0xdc2b, seq=1/256, ttl=
5152	1643.728030776	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) reply id=0xdc2b, seq=1/256, ttl=
5153	1643.732915977	10.0.0.1	10.0.0.3	ICMP	100	Echo (ping) request id=0x8c0e, seq=1/256, ttl=
5156	1643.733183972	10.0.0.1	10.0.0.3	ICMP	100	Echo (ping) request id=0x8c0e, seq=1/256, ttl=
5157	1643.733185597	10.0.0.1	10.0.0.3	ICMP	100	Echo (ping) request id=0x8c0e, seq=1/256, ttl=
5158	1643.733186763	10.0.0.1	10.0.0.3	ICMP	100	Echo (ping) request id=0x8c0e, seq=1/256, ttl=
5159	1643.733266845	10.0.0.3	10.0.0.1	ICMP	100	Echo (ping) reply id=0x8c0e, seq=1/256, ttl=
5164	1643.733991915	10.0.0.3	10.0.0.1	ICMP	100	Echo (ping) reply id=0x8c0e, seq=1/256, ttl=
5165	1643.736397037	10.0.0.1	10.0.0.4	ICMP	100	Echo (ping) request id=0x588f, seq=1/256, ttl=
5168	1643.736629158	10.0.0.1	10.0.0.4	ICMP	100	Echo (ping) request id=0x588f, seq=1/256, ttl=
5169	1643.736631408	10.0.0.1	10.0.0.4	ICMP	100	Echo (ping) request id=0x588f, seq=1/256, ttl=
5170	1643.736632283	10.0.0.1	10.0.0.4	ICMP	100	Echo (ping) request id=0x588f, seq=1/256, ttl=
5171	1643.736702948	10.0.0.4	10.0.0.1	ICMP	100	Echo (ping) reply id=0x588f, seq=1/256, ttl=
5176	1643.737072941	10.0.0.4	10.0.0.1	ICMP	100	Echo (ping) reply id=0x588f, seq=1/256, ttl=
5177	1643.756543579	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) request id=0xe812, seq=1/256, ttl=
5182	1643.757506936	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) request id=0xe812, seq=1/256, ttl=
5183	1643.757542393	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) reply id=0xe812, seq=1/256, ttl=
5188	1643.758234880	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) reply id=0xe812, seq=1/256, ttl=
5189	1643.761085661	10.0.0.2	10.0.0.3	ICMP	100	Echo (ping) request id=0x150f, seq=1/256, ttl=
5194	1643.761403821	10.0.0.2	10.0.0.3	ICMP	100	Echo (ping) request id=0x150f, seq=1/256, ttl=
5195	1643.761891854	10.0.0.3	10.0.0.2	ICMP	100	Echo (ping) reply id=0x150f, seq=1/256, ttl=
5200	1643.762106142	10.0.0.3	10.0.0.2	ICMP	100	Echo (ping) reply id=0x150f, seq=1/256, ttl=
5201	1643.763840318	10.0.0.2	10.0.0.4	ICMP	100	Echo (ping) request id=0x4860, seq=1/256, ttl=
5206	1643.764059189	10.0.0.2	10.0.0.4	ICMP	100	Echo (ping) request id=0x4860, seq=1/256, ttl=
5207	1643.764073480	10.0.0.4	10.0.0.2	ICMP	100	Echo (ping) reply id=0x4860, seq=1/256, ttl=
5212	1643.764221269	10.0.0.4	10.0.0.2	ICMP	100	Echo (ping) reply id=0x4860, seq=1/256, ttl=
5213	1643.776225170	10.0.0.3	10.0.0.1	ICMP	100	Echo (ping) request id=0x28e8, seq=1/256, ttl=
5218	1643.778455671	10.0.0.3	10.0.0.1	ICMP	100	Echo (ping) request id=0x28e8, seq=1/256, ttl=
5219	1643.778505961	10.0.0.1	10.0.0.3	ICMP	100	Echo (ping) reply id=0x28e8, seq=1/256, ttl=

5224	1643.779182824	10.0.0.1	10.0.0.3	ICMP	100	Echo (ping) reply id=0x28e8, seq=1/256, ttl=
5225	1643.780828210	10.0.0.3	10.0.0.2	ICMP	100	Echo (ping) request id=0x01dd, seq=1/256, ttl=
5230	1643.780988540	10.0.0.3	10.0.0.2	ICMP	100	Echo (ping) request id=0x01dd, seq=1/256, ttl=
5231	1643.781013623	10.0.0.2	10.0.0.3	ICMP	100	Echo (ping) reply id=0x01dd, seq=1/256, ttl=
5236	1643.781143579	10.0.0.2	10.0.0.3	ICMP	100	Echo (ping) reply id=0x01dd, seq=1/256, ttl=
5237	1643.782550761	10.0.0.3	10.0.0.4	ICMP	100	Echo (ping) request id=0x9e4c, seq=1/256, ttl=
5242	1643.782729549	10.0.0.3	10.0.0.4	ICMP	100	Echo (ping) request id=0x9e4c, seq=1/256, ttl=
5243	1643.782737008	10.0.0.4	10.0.0.3	ICMP	100	Echo (ping) reply id=0x9e4c, seq=1/256, ttl=
5248	1643.783732781	10.0.0.4	10.0.0.3	ICMP	100	Echo (ping) reply id=0x9e4c, seq=1/256, ttl=
5249	1643.792391411	10.0.0.4	10.0.0.1	ICMP	100	Echo (ping) request id=0xa003, seq=1/256, ttl=
5254	1643.792680614	10.0.0.4	10.0.0.1	ICMP	100	Echo (ping) request id=0xa003, seq=1/256, ttl=
5255	1643.792701614	10.0.0.1	10.0.0.4	ICMP	100	Echo (ping) reply id=0xa003, seq=1/256, ttl=
5260	1643.793050857	10.0.0.1	10.0.0.4	ICMP	100	Echo (ping) reply id=0xa003, seq=1/256, ttl=
5261	1643.794559538	10.0.0.4	10.0.0.2	ICMP	100	Echo (ping) request id=0x677f, seq=1/256, ttl=
5266	1643.794752076	10.0.0.4	10.0.0.2	ICMP	100	Echo (ping) request id=0x677f, seq=1/256, ttl=
5267	1643.794764534	10.0.0.2	10.0.0.4	ICMP	100	Echo (ping) reply id=0x677f, seq=1/256, ttl=
5272	1643.794926114	10.0.0.2	10.0.0.4	ICMP	100	Echo (ping) reply id=0x677f, seq=1/256, ttl=
5273	1643.796020427	10.0.0.4	10.0.0.3	ICMP	100	Echo (ping) request id=0x21e6, seq=1/256, ttl=
5278	1643.796394920	10.0.0.4	10.0.0.3	ICMP	100	Echo (ping) request id=0x21e6, seq=1/256, ttl=
5279	1643.796401628	10.0.0.3	10.0.0.4	ICMP	100	Echo (ping) reply id=0x21e6, seq=1/256, ttl=
5284	1643.796542084	10.0.0.3	10.0.0.4	ICMP	100	Echo (ping) reply id=0x21e6, seq=1/256, ttl=