

CPSC 8570: SECURITY IN ADVANCED NETWORKING TECHNOLOGIES

PROJECT 3

INSTALLATION OF CLICKOS TETBED

Description:

In this project, we are asked to follow a step-by-step guide to setup a virtual machine (VM) that runs on ClickOS and an exercise for simple demonstration of testing ClickOS. The step-by-step guide can be found at

<https://bitbucket.org/hongdal/clickos-setup/wiki/Tutorial%20of%20installing%20ClickOS%20testbed>

Initial Setup and configuration of VM:

1. The Oracle VirtualBox VM was downloaded from <https://www.virtualbox.org/wiki/Downloads> And was installed on my Windows machine
2. A pre-configured virtual machine image with Ubuntu 14.04.4 (LTS) 64-bit installed, was made available for us. This image was downloaded from http://people.cs.clemson.edu/~hongdal/Ubuntu_14.04.4-64bit.7z
3. Once the download was complete, the 'Ubuntu_14.04.4-64bit.7z' was extracted.
4. Ubuntu VM for ClickOS was setup on VirtualBox using the 'Ubuntu 14.04.4 (64bit).vdi' which was extracted in the previous step.
5. The username is 'osboxes' and password is 'osboxes.org'.
6. We configured the VM and set the memory to 1024 MB and also set the number of processors to 2.
7. All the commands were run with root privileges. We used the command `sudo su -` and used the username and password mentioned in step 5 to switch to root

Install Xen:

Xen is one of the most popular virtualized platforms. In this project, we are going to use Xen to support our Network Function Virtualization (NFV). More information about the Xen project can be found at

<http://www.xenproject.org/>

1. We install all the dependencies that are required for Xen:

```
# apt-get update  
# apt-get install build-essential wget  
# apt-get install bcc bin86 gawk bridge-utils iproute libcurl3 libcurl4-openssl-dev bzip2  
module-init-tools transfig tgif  
# apt-get install texinfo texlive-latex-base texlive-latex-recommended
```

```
# apt-get install texlive-fonts-extra texlive-fonts-recommended pciutils-dev mercurial
# apt-get install make gcc libc6-dev zlib1g-dev python python-dev python-twisted
# apt-get install libncurses5-dev patch libsdl-dev libjpeg62-dev
# apt-get install libvncserver-dev
# apt-get install iasl libbz2-dev e2fslibs-dev git-core uuid-dev ocaml ocaml-findlib
# apt-get install libx11-dev bison flex xz-utils libyajl-dev
# apt-get install gettext libpixmap-1-dev
# apt-get build-dep xen
# apt-get install libglib2.0-dev
# apt-get install libyajl-dev
```

2. Create a new directory called 'tutorial' where all the files are downloaded, built and run:

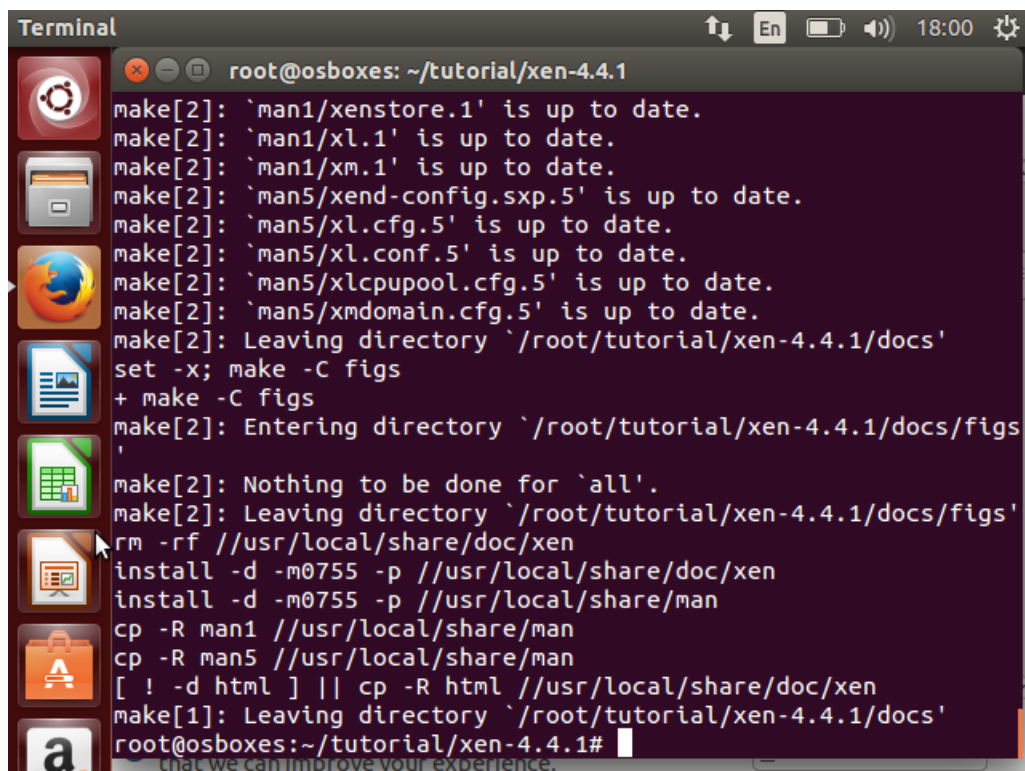
```
# mkdir tutorial
```

3. Download Xen-4.4.1 in the tutorial directory:

```
# cd tutorial
# wget http://people.cs.clemson.edu/~hongdal/xen-4.4.1.tar.gz
# tar -xf xen-4.4.1.tar.gz
```

4. Compile and build Xen:

```
# export XEN_ROOT=`pwd`/xen-4.4.1
# cd $XEN_ROOT
# ./configure
# make world
# make install
```



```
Terminal
root@osboxes: ~/tutorial/xen-4.4.1
make[2]: `man1/xenstore.1' is up to date.
make[2]: `man1/xl.1' is up to date.
make[2]: `man1/xm.1' is up to date.
make[2]: `man5/xend-config.sxp.5' is up to date.
make[2]: `man5/xl.cfg.5' is up to date.
make[2]: `man5/xl.conf.5' is up to date.
make[2]: `man5/xlcpupool.cfg.5' is up to date.
make[2]: `man5/xmdomain.cfg.5' is up to date.
make[2]: Leaving directory `/root/tutorial/xen-4.4.1/docs'
set -x; make -C figs
+ make -C figs
make[2]: Entering directory `/root/tutorial/xen-4.4.1/docs/figs'
make[2]: Nothing to be done for `all'.
make[2]: Leaving directory `/root/tutorial/xen-4.4.1/docs/figs'
rm -rf //usr/local/share/doc/xen
install -d -m0755 -p //usr/local/share/doc/xen
install -d -m0755 -p //usr/local/share/man
cp -R man1 //usr/local/share/man
cp -R man5 //usr/local/share/man
[ ! -d html ] || cp -R html //usr/local/share/doc/xen
make[1]: Leaving directory `/root/tutorial/xen-4.4.1/docs'
root@osboxes:~/tutorial/xen-4.4.1#
```

5. Configure your system to boot from Xen:

Before configuring the system to boot from Xen, it is highly recommended that we take a snapshot of the VM, so that we can restore back if the system crashes. This can be done using the 'update-rc.d' commands as shown:

```
# /sbin/ldconfig
# update-rc.d xencommons defaults 19 18
# update-rc.d xendomains defaults 21 20
# update-rc.d xen-watchdog defaults 22 23
```

6. Modify the linux bootloader – 'grub' to boot as commandline only interface:

We open the file at '/etc/default/grub' and update some of the configuration lines to the following appropriate values:

```
GRUB_DEFAULT=3
GRUB_HIDDEN_TIMEOUT=0
```

Update grub and reboot the system:

```
# update-grub
# reboot
```

7. Verify Xen has been installed successfully:

After rebooting, in my case, the system automatically started a command line Interface (CLI) as the virtual machine does not support GUI in Xen. If the system starts a graphic user interface (GUI) automatically we need to switch to a command line interface (CLI). Press *control + alt + F1* to switch to a CLI. input your username and password to log in. After logging in, switch to root privilege.

The following command can be used to verify that Xen has been successfully installed:

```
# xl list
```

Example output of xl list is:

Name	ID	Mem	VCPUs	State	Time(s)
Domain-0	0	4098	32	r-----	23926.7

```
click (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Ubuntu 14.04.4 LTS osboxes tty1
osboxes login: osboxes
Password:
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 4.2.0-27-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

osboxes@osboxes:~$ sudo su -
[sudo] password for osboxes:
root@osboxes:~# xl list
Name                      ID    Mem VCPUs    State    Time(s)
Domain-0                  0    945     1    r-----    12.2
root@osboxes:~# _
```

Building ClickOS:

1. Our first task in building the ClickOS is to get the source codes:

All the clickos codes are downloaded into our tutorial directory:

```
# cd tutorial
# wget http://people.cs.clemson.edu/~hongdal/clickos.tar
# wget http://people.cs.clemson.edu/~hongdal/mini-os.tar
# wget http://people.cs.clemson.edu/~hongdal/cosmos.tar
# wget http://people.cs.clemson.edu/~hongdal/toolchain.tar
# tar -xf clickos.tar
# tar -xf mini-os.tar
# tar -xf cosmos.tar
# tar -xf toolchain.tar
```

2. Setting up the environment variables:

First we need to make sure 'pwd' outputs the path – '/root/tutorial'.

```
# export XEN_ROOT=$(pwd)/xen-4.4.1/"
# export MINIOS_ROOT=$XEN_ROOT"extras/mini-os/"
# export CLICKOS_ROOT=$(pwd)/clickos/"
# export TOOLCHAIN_ROOT=$(pwd)/toolchain/"
# export COSMOS_ROOT=$(pwd)/cosmos/"
```

Evertime we reboot the system or switch to another user, it is important to check if these variables are still intact. This is done by typing
`echo $VARIABLE-NAME`

Example:

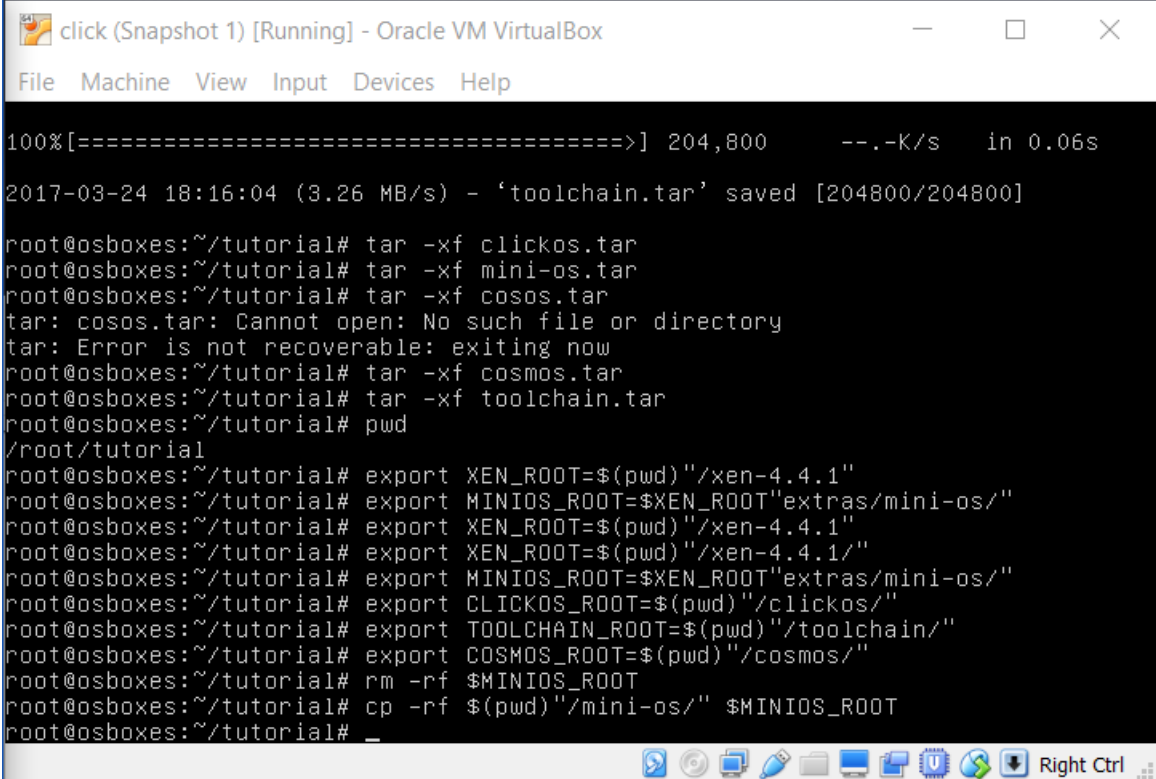
`echo $XEN_ROOT` to check variable `$XEN_ROOT`

3. Replacing mini-os:

MiniOS is an operating system that comes with Xen. ClickOS is built based on MiniOS. We need to replace Xen's mini-os with ClickOS's minios.

```
# rm -rf $MINIOS_ROOT
```

```
# cp -rf $(pwd)/mini-os/" $MINIOS_ROOT
```



```
100%[=====>] 204,800 --.-K/s in 0.06s
2017-03-24 18:16:04 (3.26 MB/s) - 'toolchain.tar' saved [204800/204800]

root@osboxes:~/tutorial# tar -xf clickos.tar
root@osboxes:~/tutorial# tar -xf mini-os.tar
root@osboxes:~/tutorial# tar -xf cosos.tar
tar: cosos.tar: Cannot open: No such file or directory
tar: Error is not recoverable: exiting now
root@osboxes:~/tutorial# tar -xf cosmos.tar
root@osboxes:~/tutorial# tar -xf toolchain.tar
root@osboxes:~/tutorial# pwd
/root/tutorial
root@osboxes:~/tutorial# export XEN_ROOT=$(pwd)/xen-4.4.1
root@osboxes:~/tutorial# export MINIOS_ROOT=$XEN_ROOT"extras/mini-os/"
root@osboxes:~/tutorial# export XEN_ROOT=$(pwd)/xen-4.4.1
root@osboxes:~/tutorial# export XEN_ROOT=$(pwd)/xen-4.4.1/"
root@osboxes:~/tutorial# export MINIOS_ROOT=$XEN_ROOT"extras/mini-os/"
root@osboxes:~/tutorial# export CLICKOS_ROOT=$(pwd)/clickos/"
root@osboxes:~/tutorial# export TOOLCHAIN_ROOT=$(pwd)/toolchain/"
root@osboxes:~/tutorial# export COSMOS_ROOT=$(pwd)/cosmos/"
root@osboxes:~/tutorial# rm -rf $MINIOS_ROOT
root@osboxes:~/tutorial# cp -rf $(pwd)/mini-os/" $MINIOS_ROOT
root@osboxes:~/tutorial# _
```

4. Building toolchain:

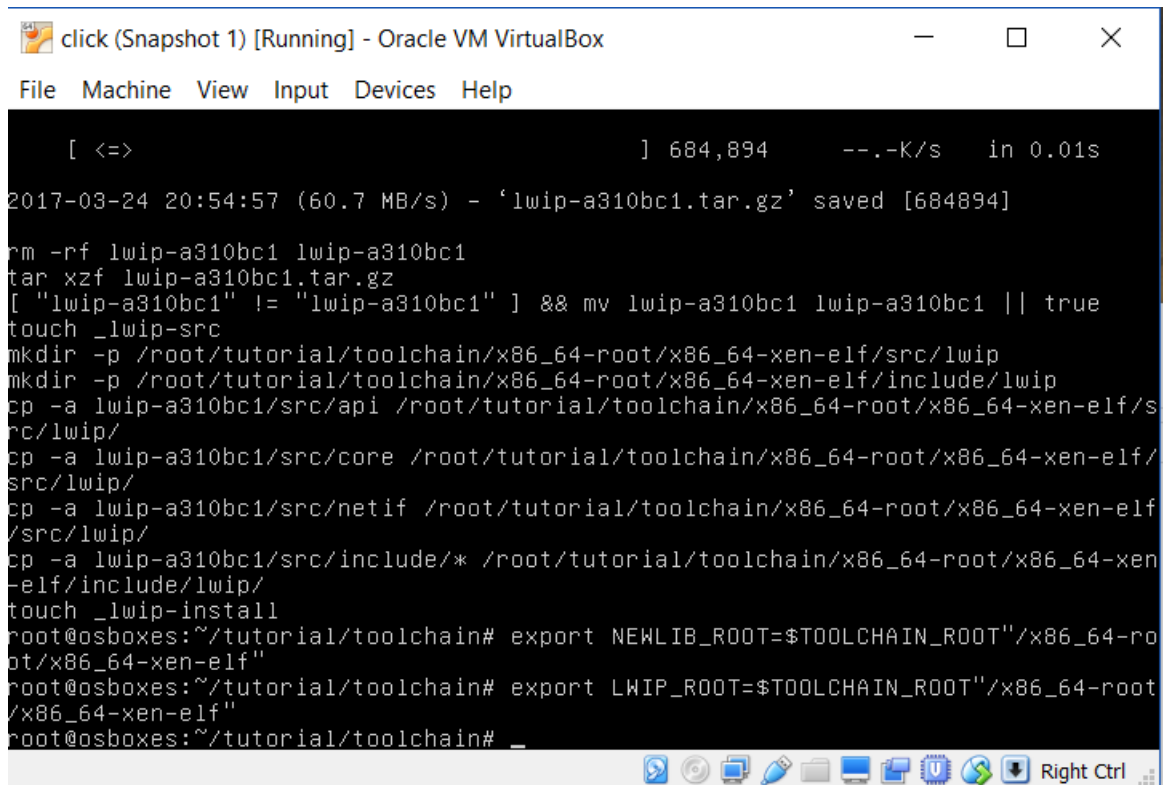
This toolchain is for ClickOS's C++ library

```
# cd $TOOLCHAIN_ROOT
```

```
# make
```

```
# export NEWLIB_ROOT=$TOOLCHAIN_ROOT"/x86_64-root/x86_64-xen-elf"
```

```
# export LWIP_ROOT=$TOOLCHAIN_ROOT"/x86_64-root/x86_64-xen-elf"
```



```
click (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

[ <=> ] 684,894 --.-K/s in 0.01s
2017-03-24 20:54:57 (60.7 MB/s) - 'lwip-a310bc1.tar.gz' saved [684894]

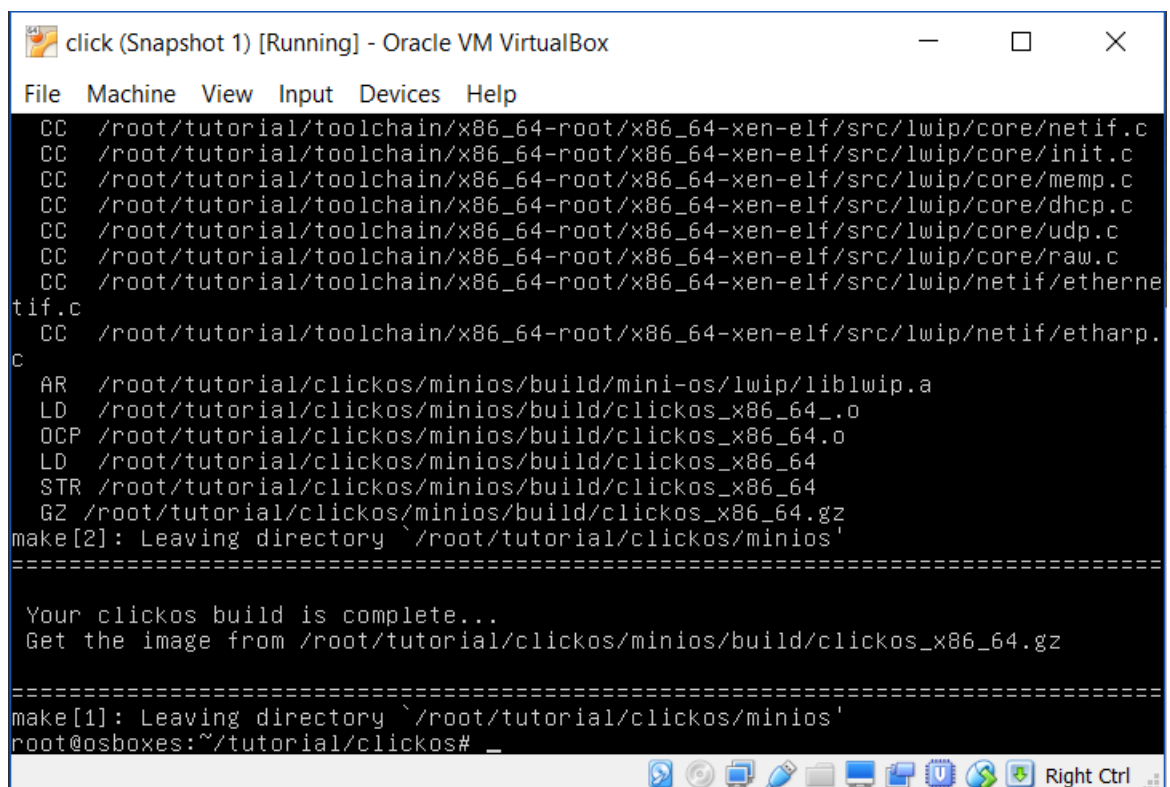
rm -rf lwip-a310bc1 lwip-a310bc1
tar xzf lwip-a310bc1.tar.gz
[ "lwip-a310bc1" != "lwip-a310bc1" ] && mv lwip-a310bc1 lwip-a310bc1 || true
touch _lwip-src
mkdir -p /root/tutorial/toolchain/x86_64-root/x86_64-xen-elf/src/lwip
mkdir -p /root/tutorial/toolchain/x86_64-root/x86_64-xen-elf/include/lwip
cp -a lwip-a310bc1/src/api /root/tutorial/toolchain/x86_64-root/x86_64-xen-elf/s
rc/lwip/
cp -a lwip-a310bc1/src/core /root/tutorial/toolchain/x86_64-root/x86_64-xen-elf/
src/lwip/
cp -a lwip-a310bc1/src/netif /root/tutorial/toolchain/x86_64-root/x86_64-xen-elf
/src/lwip/
cp -a lwip-a310bc1/src/include/* /root/tutorial/toolchain/x86_64-root/x86_64-xen
-elf/include/lwip/
touch _lwip-install
root@osboxes:~/tutorial/toolchain# export NEWLIB_ROOT=$TOOLCHAIN_ROOT"/x86_64-ro
ot/x86_64-xen-elf"
root@osboxes:~/tutorial/toolchain# export LWIP_ROOT=$TOOLCHAIN_ROOT"/x86_64-root
/x86_64-xen-elf"
root@osboxes:~/tutorial/toolchain# _
```

5. Building ClickOS kernel:

```
# cd $CLICKOS_ROOT
```

```
# ./configure --enable-minios --with-xen=$XEN_ROOT --with-minios=$MINIOS_ROOT
```

```
# make minios
```



```
click (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

CC /root/tutorial/toolchain/x86_64-root/x86_64-xen-elf/src/lwip/core/netif.c
CC /root/tutorial/toolchain/x86_64-root/x86_64-xen-elf/src/lwip/core/init.c
CC /root/tutorial/toolchain/x86_64-root/x86_64-xen-elf/src/lwip/core/memp.c
CC /root/tutorial/toolchain/x86_64-root/x86_64-xen-elf/src/lwip/core/dhcp.c
CC /root/tutorial/toolchain/x86_64-root/x86_64-xen-elf/src/lwip/core/udp.c
CC /root/tutorial/toolchain/x86_64-root/x86_64-xen-elf/src/lwip/core/raw.c
CC /root/tutorial/toolchain/x86_64-root/x86_64-xen-elf/src/lwip/netif/etherne
tif.c
CC /root/tutorial/toolchain/x86_64-root/x86_64-xen-elf/src/lwip/netif/etharp.
c
AR /root/tutorial/clickos/minios/build/mini-os/lwip/liblwip.a
LD /root/tutorial/clickos/minios/build/clickos_x86_64.o
OCP /root/tutorial/clickos/minios/build/clickos_x86_64.o
LD /root/tutorial/clickos/minios/build/clickos_x86_64
STR /root/tutorial/clickos/minios/build/clickos_x86_64
GZ /root/tutorial/clickos/minios/build/clickos_x86_64.gz
make[2]: Leaving directory `/root/tutorial/clickos/minios'
=====
Your clickos build is complete...
Get the image from /root/tutorial/clickos/minios/build/clickos_x86_64.gz
=====
make[1]: Leaving directory `/root/tutorial/clickos/minios'
root@osboxes:~/tutorial/clickos# _
```

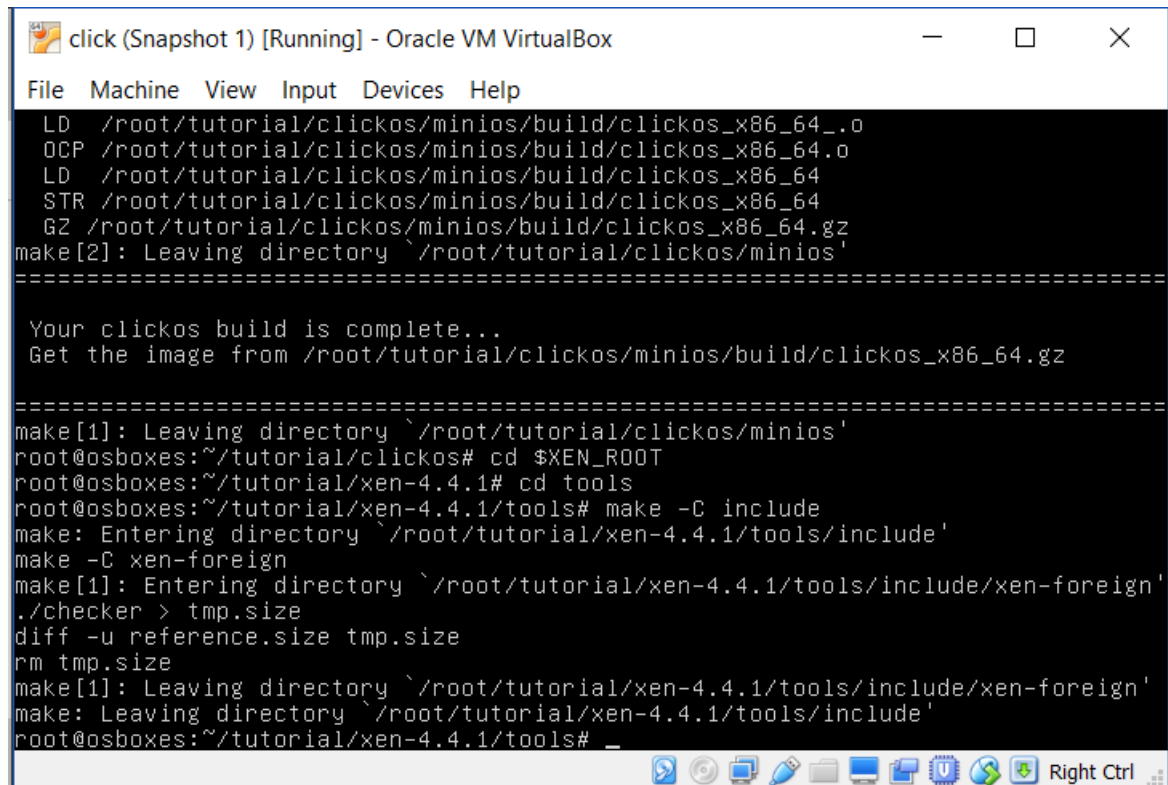
6. Building cosmos:

cosmos is a tool used for booting ClickOS kernel.

```
# cd $XEN_ROOT
```

```
# cd tools
```

```
# make -C include
```



```
# cd $COSMOS_ROOT
```

```
# make DOMLIB=xl
```

We need to set the 'PATH' environmental variable

```
# export PATH=$PATH":/$COSMOS_ROOT/dist/bin"
```

Install Open vSwitch:

We will be using Open vSwitch (OVS) as a software switch. More information about OVS can be found at

<http://openvswitch.org/>

1. Install Open vSwitch:

```
# cd tutorial
```

```
# wget http://openvswitch.org/releases/openvswitch-2.5.0.tar.gz
```

```
# tar -xf openvswitch-2.5.0.tar.gz
```

```
# cd openvswitch-2.5.0/
```

```
# ./configure --with-linux=/lib/modules/`uname -r`/build
```

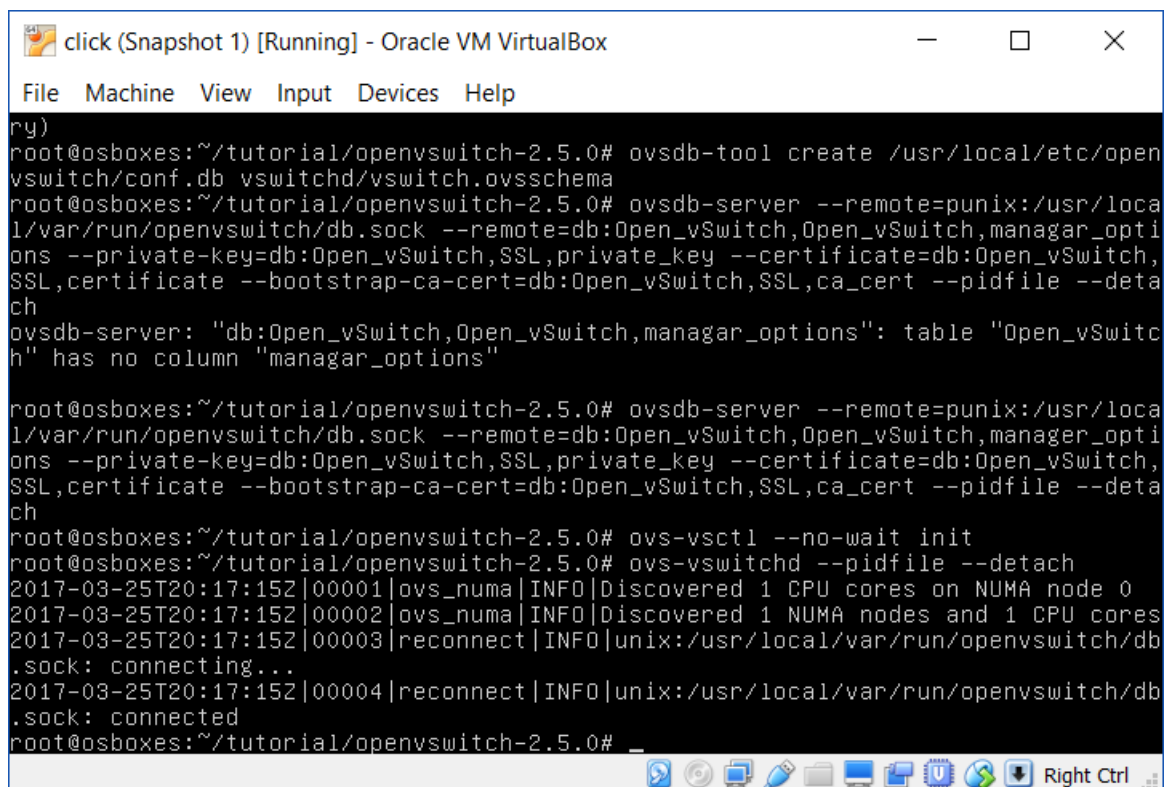
```
# make
```

```
# make install
# make modules_install
```

2. Configure and run OVS:

The following commands need to be executed in the directory –

```
'tutorial/openvswitch-2.5.0/'
# /sbin/modprobe openvswitch
# mkdir -p /usr/local/etc/openvswitch
# ovsdb-tool create /usr/local/etc/openvswitch/conf.db vswitchd/vswitch.ovsschema
# ovsdb-server --remote=punix:/usr/local/var/run/openvswitch/db.sock --
remote=db:Open_vSwitch,Open_vSwitch,manager_options --private-
key=db:Open_vSwitch,SSL,private_key --certificate=db:Open_vSwitch,SSL,certificate --
bootstrap-ca-cert=db:Open_vSwitch,SSL,ca_cert --pidfile --detach
# ovs-vsctl --no-wait init
# ovs-vswitchd --pidfile --detach
```



```
click (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@osboxes:~/tutorial/openvswitch-2.5.0# ovsdb-tool create /usr/local/etc/open
vswitchd/vswitch.ovsschema
root@osboxes:~/tutorial/openvswitch-2.5.0# ovsdb-server --remote=punix:/usr/loca
l/var/run/openvswitch/db.sock --remote=db:Open_vSwitch,Open_vSwitch,manager_opti
ons --private-key=db:Open_vSwitch,SSL,private_key --certificate=db:Open_vSwitch,
SSL,certificate --bootstrap-ca-cert=db:Open_vSwitch,SSL,ca_cert --pidfile --deta
ch
ovsdb-server: "db:Open_vSwitch,Open_vSwitch,manager_options": table "Open_vSwitc
h" has no column "manager_options"
root@osboxes:~/tutorial/openvswitch-2.5.0# ovsdb-server --remote=punix:/usr/loca
l/var/run/openvswitch/db.sock --remote=db:Open_vSwitch,Open_vSwitch,manager_opti
ons --private-key=db:Open_vSwitch,SSL,private_key --certificate=db:Open_vSwitch,
SSL,certificate --bootstrap-ca-cert=db:Open_vSwitch,SSL,ca_cert --pidfile --deta
ch
root@osboxes:~/tutorial/openvswitch-2.5.0# ovs-vsctl --no-wait init
root@osboxes:~/tutorial/openvswitch-2.5.0# ovs-vswitchd --pidfile --detach
2017-03-25T20:17:15Z|00001|ovs_numa|INFO|Discovered 1 CPU cores on NUMA node 0
2017-03-25T20:17:15Z|00002|ovs_numa|INFO|Discovered 1 NUMA nodes and 1 CPU cores
2017-03-25T20:17:15Z|00003|reconnect|INFO|unix:/usr/local/var/run/openvswitch/db
.sock: connecting...
2017-03-25T20:17:15Z|00004|reconnect|INFO|unix:/usr/local/var/run/openvswitch/db
.sock: connected
root@osboxes:~/tutorial/openvswitch-2.5.0#
```

Everytime the system is rebooted; we need to repeat the above commands to reconfigure the OVS.

3. Create a bridge on OVS:

```
# ovs-vsctl add-br ovs-lan
```

We use – ‘*ovs-vsctl show*’ to verify that the OVS is running correctly.

Example screen dump of 'ovs-vsctl show':

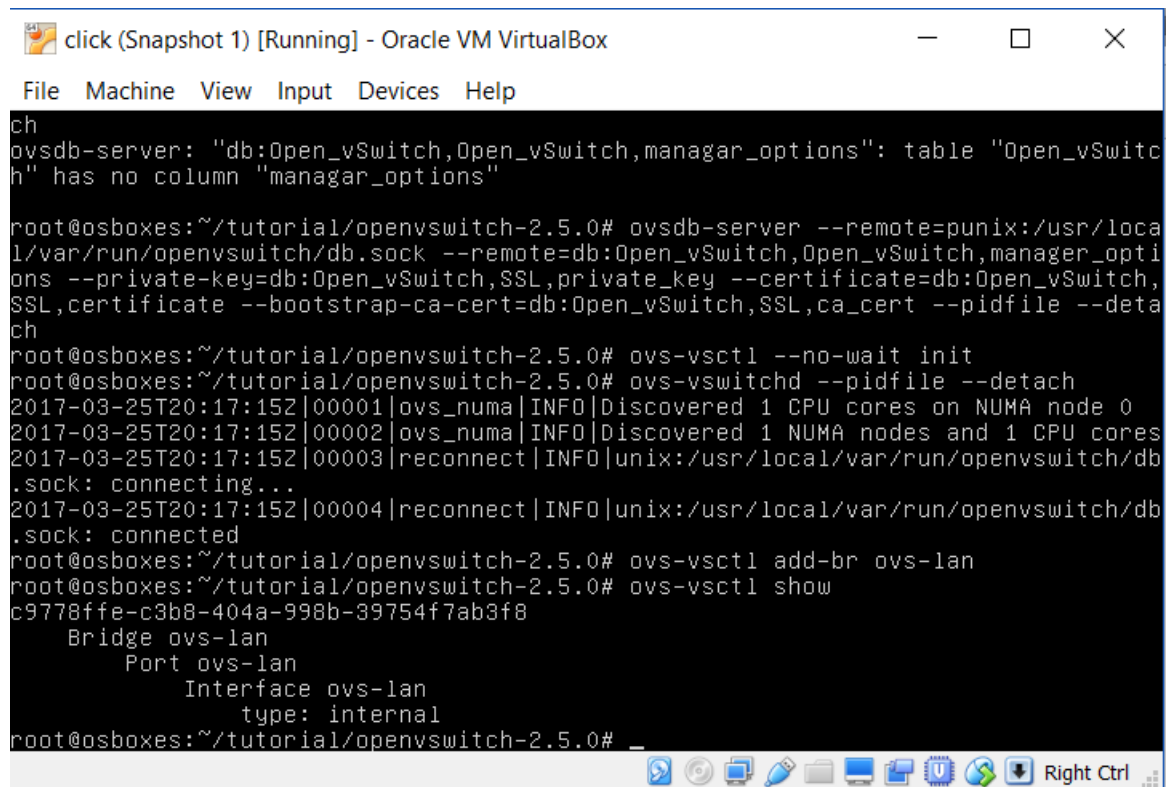
e6d82000-68d3-49cc-a5d0-3e51508ad772

Bridge ovs-lan

Port ovs-lan

Interface ovs-lan

type: internal



```
ch
ovsdb-server: "db:Open_vSwitch,Open_vSwitch,managar_options": table "Open_vSwitch" has no column "managar_options"

root@osboxes:~/tutorial/openvswitch-2.5.0# ovsdb-server --remote=punix:/usr/local/var/run/openvswitch/db.sock --remote=db:Open_vSwitch,Open_vSwitch,manager_options --private-key=db:Open_vSwitch,SSL,private_key --certificate=db:Open_vSwitch,SSL,certificate --bootstrap-ca-cert=db:Open_vSwitch,SSL,ca_cert --pidfile --detach
ch
root@osboxes:~/tutorial/openvswitch-2.5.0# ovs-vsctl --no-wait init
root@osboxes:~/tutorial/openvswitch-2.5.0# ovs-vswitchd --pidfile --detach
2017-03-25T20:17:15Z|00001|ovs_numa|INFO|Discovered 1 CPU cores on NUMA node 0
2017-03-25T20:17:15Z|00002|ovs_numa|INFO|Discovered 1 NUMA nodes and 1 CPU cores
2017-03-25T20:17:15Z|00003|reconnect|INFO|unix:/usr/local/var/run/openvswitch/db.sock: connecting...
2017-03-25T20:17:15Z|00004|reconnect|INFO|unix:/usr/local/var/run/openvswitch/db.sock: connected
root@osboxes:~/tutorial/openvswitch-2.5.0# ovs-vsctl add-br ovs-lan
root@osboxes:~/tutorial/openvswitch-2.5.0# ovs-vsctl show
c9778ffe-c3b8-404a-998b-39754f7ab3f8
    Bridge ovs-lan
        Port ovs-lan
            Interface ovs-lan
                type: internal
root@osboxes:~/tutorial/openvswitch-2.5.0#
```

Starting a ClickOS Instance:

1. Create a configuration file for Xen:

Change your working directory to - '*tutorial/clickos/minios*'.

We will use an existing Xen configuration file and update some of the values in '*config.xen*' file:

```
kernel = './build/clickos_x86_64'
```

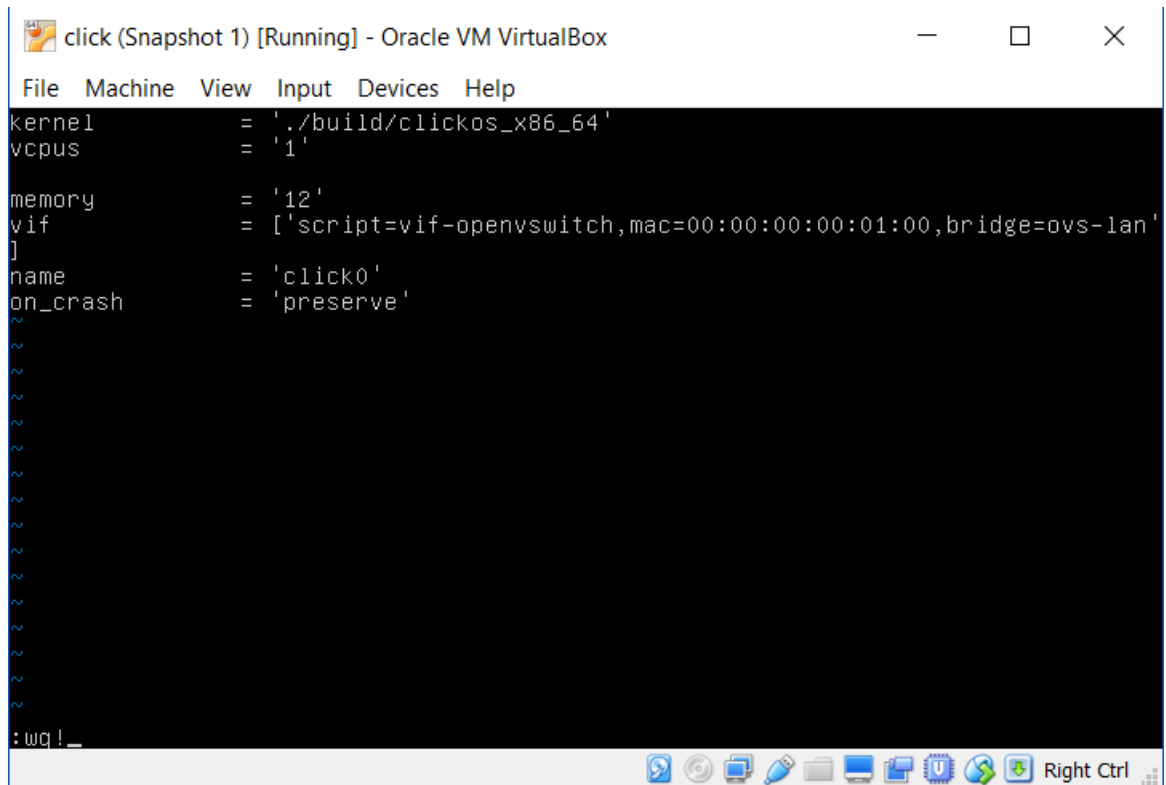
```
vcpus = '1'
```

```
vif = ['script=vif-openvswitch,mac=00:00:00:00:01:00,bridge=ovs-lan']
```

```
name = 'click0'
```

We delete the following line from '*config.xen*' file as it may create some problems:

```
cpus = '2'
```



2. Create configuration file for Click:

Click is a scripting language that can define network functions based on Click Modular.

Refer to the Click Element website, to write more complicated network functions using Click. Link to Click Element website is:

<http://read.cs.ucla.edu/click/elements>

Some examples could be found here:

<http://read.cs.ucla.edu/click/examples>

The Click configuration file describes the functionality that a ClickOS instance will implement.

As a simple test, the following scripts describe a simple network function that receives packets from the network interface, prints the packets, and sends the packets back to the network. To achieve this, we create the following script as a file – *mirror.click*:

FromDevice -> Print('OK') -> EtherMirror -> ToDevice

3. Start ClickOS Instance:

We will use 'cosmos' to start a ClickOS instance. Check whether '\$COSMOS_ROOT' is intact by typing 'echo \$COSMOS_ROOT' and set it appropriately.

xl create config.xen

\$COSMOS_ROOT/dist/bin/cosmos start click0 mirror.click

Example screendump of 'xl list':

<i>Name</i>	<i>ID</i>	<i>Mem</i>	<i>VCPUs</i>	<i>State</i>	<i>Time(s)</i>
<i>Domain-0</i>	<i>0</i>	<i>4098</i>	<i>32</i>	<i>r----</i>	<i>23926.7</i>
<i>click0</i>	<i>1</i>	<i>12</i>	<i>1</i>	<i>r----</i>	<i>234.5</i>

A screenshot of the Oracle VM VirtualBox interface. The title bar reads "click (Snapshot 1) [Running] - Oracle VM VirtualBox". Below it are standard window controls (minimize, maximize, close). A menu bar contains "File Machine View Input Devices Help". The main area shows a black terminal window with white text output from several commands.

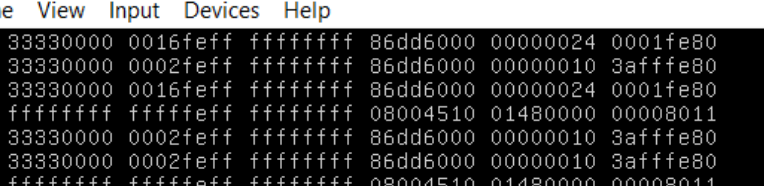
```
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
  
"mirror.click" 2 lines, 54 characters written  
root@osboxes:~/tutorial/clickos/minios# xl create config.xen  
Parsing config from config.xen  
root@osboxes:~/tutorial/clickos/minios# $COSMOS_ROOT/dist/bin/cosmos start click0 mirror.click  
Domain ID for click0: 4  
Location of click script: mirror.click  
root@osboxes:~/tutorial/clickos/minios# xl list
```

The last command produces the following table:

Name	ID	Mem	VCPUs	State	Time(s)
Domain-0	0	914	1	r-----	381.7
click0	4	12	1	------	8.9

The prompt returns to root@osboxes:~/tutorial/clickos/minios#. At the bottom right of the screen, there's a taskbar with various icons and the text "Right Ctrl".

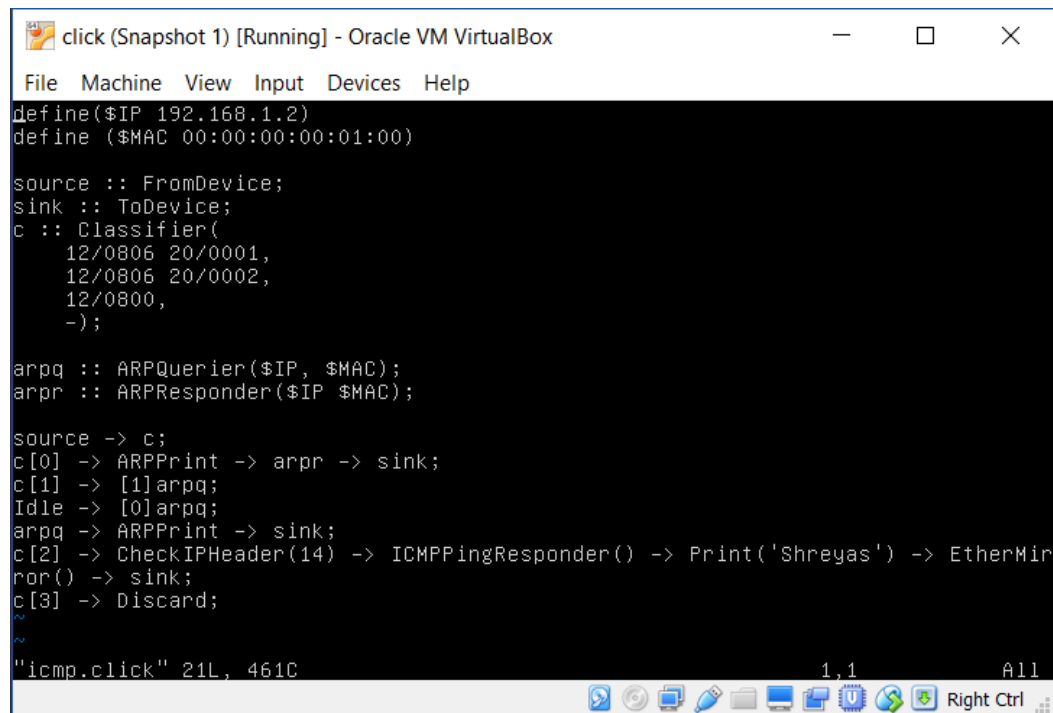
To watch the output of `click0`, type `'xl console click0'`. To quit the console, press `'control + j'`



Exercise:

ICMP Responder:

In this exercise, we are given a template script to create a Click configuration file – *'icmp.click'*. We need to set the IP address and MAC address appropriately and make it print our name on the click0 console window. The screendump of the icmp.click file after making the changes to print my name is as shown below:



```
click (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
define($IP 192.168.1.2)
define ($MAC 00:00:00:00:01:00)

source :: FromDevice;
sink :: ToDevice;
c :: Classifier(
    12/0806 20/0001,
    12/0806 20/0002,
    12/0800,
    -);

arpq :: ARPQuerier($IP, $MAC);
arpr :: ARPResponder($IP $MAC);

source -> c;
c[0] -> ARPPrint -> arpr -> sink;
c[1] -> [1]arpq;
Idle -> [0]arpq;
arpq -> ARPPrint -> sink;
c[2] -> CheckIPHeader(14) -> ICMPPingResponder() -> Print('Shreyas') -> EtherMirror() -> sink;
c[3] -> Discard;
~
~
"icmp.click" 21L, 461C 1,1 All
```

We need to kill existing ClickOS instance at click0 using:

```
xl destroy click0
```

We then start a new instance at click0:

```
# xl create config.xen
```

```
# $COSMOS_ROOT/dist/bin/cosmos start click0 icmp.click
```

We need to add our VM's network interface OVS so that you can ping the ClickOS instance from your VM:

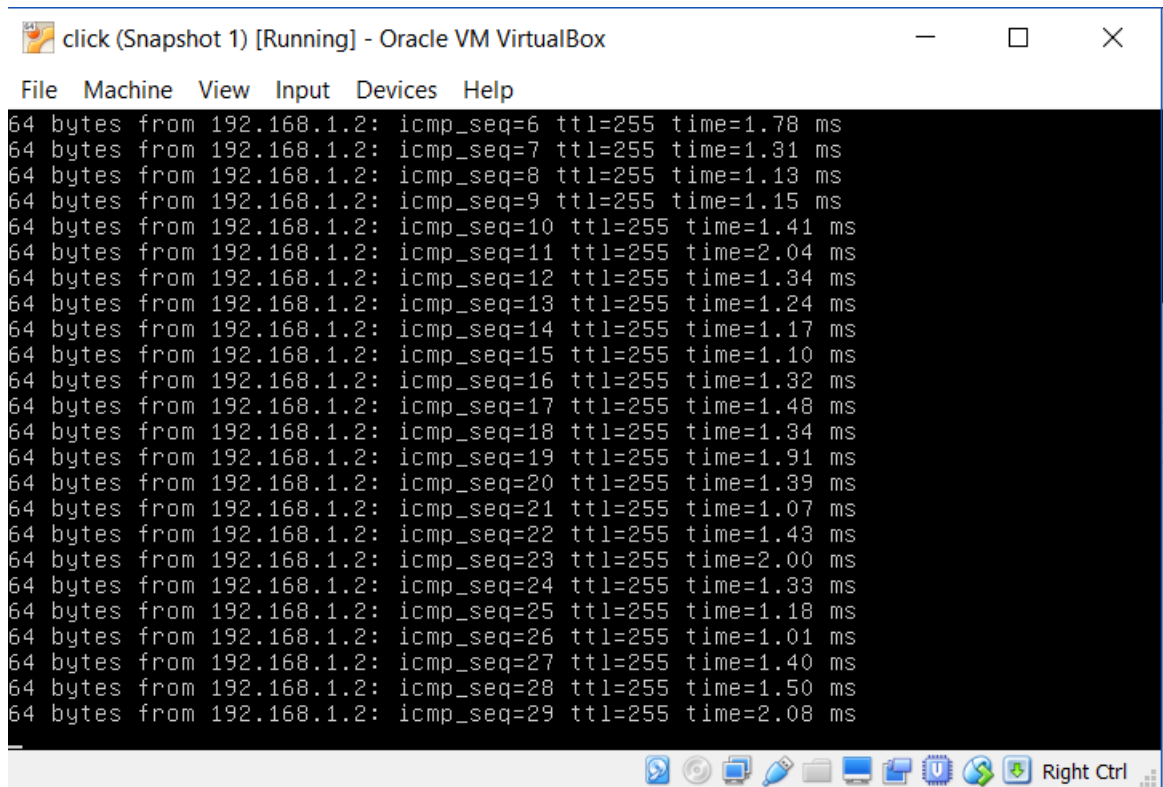
```
# ifconfig eth0 0
```

```
# ovs-vsctl add-port ovs-lan eth0
```

```
# ifconfig ovs-lan 192.168.1.1 netmask 255.255.255.0 up
```

Now we need to ping ClickOS instance from our VM:

```
# ping 192.168.1.2
```

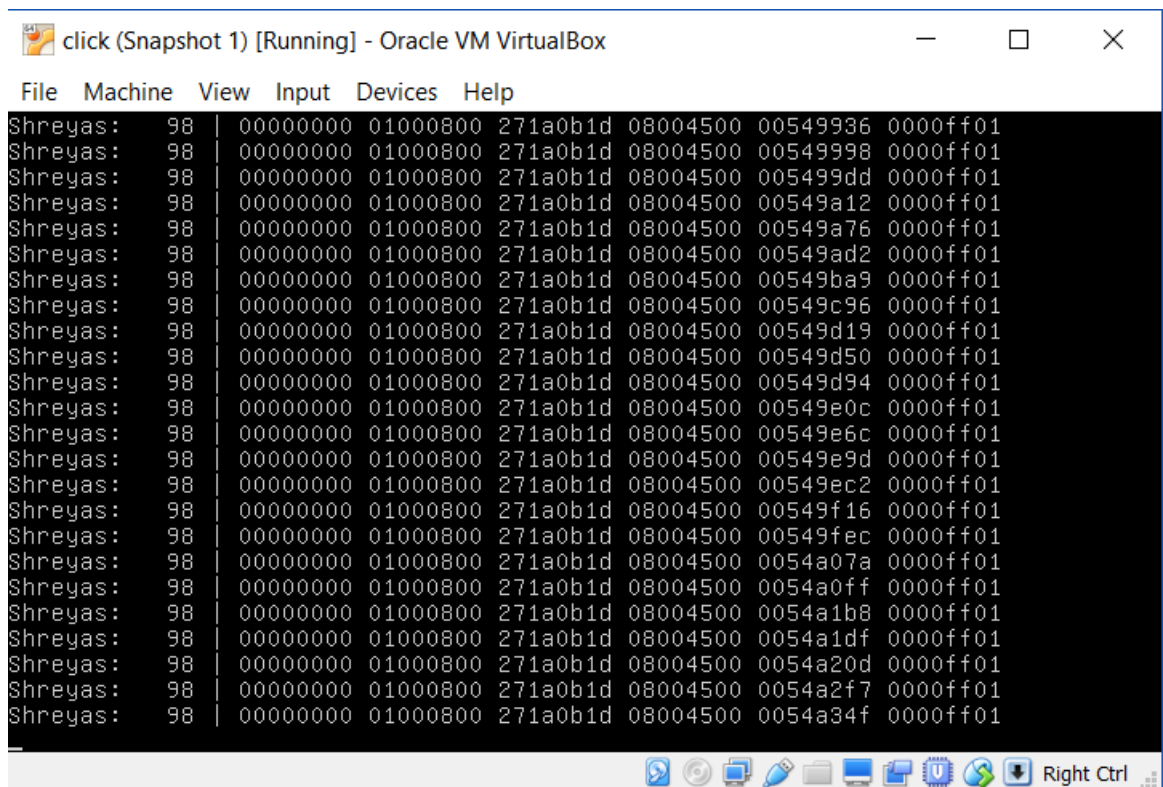


The screenshot shows a VirtualBox window titled "click (Snapshot 1) [Running] - Oracle VM VirtualBox". The console displays a series of ICMP echo requests from 192.168.1.2 to 192.168.1.2, with sequence numbers 6 through 29. Each line shows the packet size (64 bytes), source and destination IP addresses, sequence number, TTL (255), and round-trip time in milliseconds. The times vary between approximately 1.01 ms and 2.08 ms. The bottom of the window shows a taskbar with various icons and a "Right Ctrl" button.

```
File Machine View Input Devices Help
64 bytes from 192.168.1.2: icmp_seq=6 ttl=255 time=1.78 ms
64 bytes from 192.168.1.2: icmp_seq=7 ttl=255 time=1.31 ms
64 bytes from 192.168.1.2: icmp_seq=8 ttl=255 time=1.13 ms
64 bytes from 192.168.1.2: icmp_seq=9 ttl=255 time=1.15 ms
64 bytes from 192.168.1.2: icmp_seq=10 ttl=255 time=1.41 ms
64 bytes from 192.168.1.2: icmp_seq=11 ttl=255 time=2.04 ms
64 bytes from 192.168.1.2: icmp_seq=12 ttl=255 time=1.34 ms
64 bytes from 192.168.1.2: icmp_seq=13 ttl=255 time=1.24 ms
64 bytes from 192.168.1.2: icmp_seq=14 ttl=255 time=1.17 ms
64 bytes from 192.168.1.2: icmp_seq=15 ttl=255 time=1.10 ms
64 bytes from 192.168.1.2: icmp_seq=16 ttl=255 time=1.32 ms
64 bytes from 192.168.1.2: icmp_seq=17 ttl=255 time=1.48 ms
64 bytes from 192.168.1.2: icmp_seq=18 ttl=255 time=1.34 ms
64 bytes from 192.168.1.2: icmp_seq=19 ttl=255 time=1.91 ms
64 bytes from 192.168.1.2: icmp_seq=20 ttl=255 time=1.39 ms
64 bytes from 192.168.1.2: icmp_seq=21 ttl=255 time=1.07 ms
64 bytes from 192.168.1.2: icmp_seq=22 ttl=255 time=1.43 ms
64 bytes from 192.168.1.2: icmp_seq=23 ttl=255 time=2.00 ms
64 bytes from 192.168.1.2: icmp_seq=24 ttl=255 time=1.33 ms
64 bytes from 192.168.1.2: icmp_seq=25 ttl=255 time=1.18 ms
64 bytes from 192.168.1.2: icmp_seq=26 ttl=255 time=1.01 ms
64 bytes from 192.168.1.2: icmp_seq=27 ttl=255 time=1.40 ms
64 bytes from 192.168.1.2: icmp_seq=28 ttl=255 time=1.50 ms
64 bytes from 192.168.1.2: icmp_seq=29 ttl=255 time=2.08 ms
```

To watch the outputs of the ClickOS instance:

```
# xl console click0
```



The screenshot shows a VirtualBox window titled "click (Snapshot 1) [Running] - Oracle VM VirtualBox". The console displays a series of memory addresses and their corresponding values, likely representing the state of the ClickOS instance. Each line shows a memory address (e.g., 00000000, 01000800, 271a0b1d, 08004500, 00549936, 0000ff01) and a value (e.g., 00000000, 01000800, 271a0b1d, 08004500, 00549998, 0000ff01). The values are repeated for each address. The bottom of the window shows a taskbar with various icons and a "Right Ctrl" button.

```
File Machine View Input Devices Help
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 00549936 0000ff01
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 00549998 0000ff01
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 005499dd 0000ff01
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 00549a12 0000ff01
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 00549a76 0000ff01
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 00549ad2 0000ff01
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 00549ba9 0000ff01
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 00549c96 0000ff01
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 00549d19 0000ff01
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 00549d50 0000ff01
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 00549d94 0000ff01
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 00549e0c 0000ff01
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 00549e6c 0000ff01
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 00549e9d 0000ff01
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 00549ec2 0000ff01
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 00549f16 0000ff01
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 00549fec 0000ff01
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 0054a07a 0000ff01
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 0054a0ff 0000ff01
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 0054a1b8 0000ff01
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 0054a1df 0000ff01
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 0054a20d 0000ff01
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 0054a2f7 0000ff01
Shreyas: 98 | 00000000 01000800 271a0b1d 08004500 0054a34f 0000ff01
```