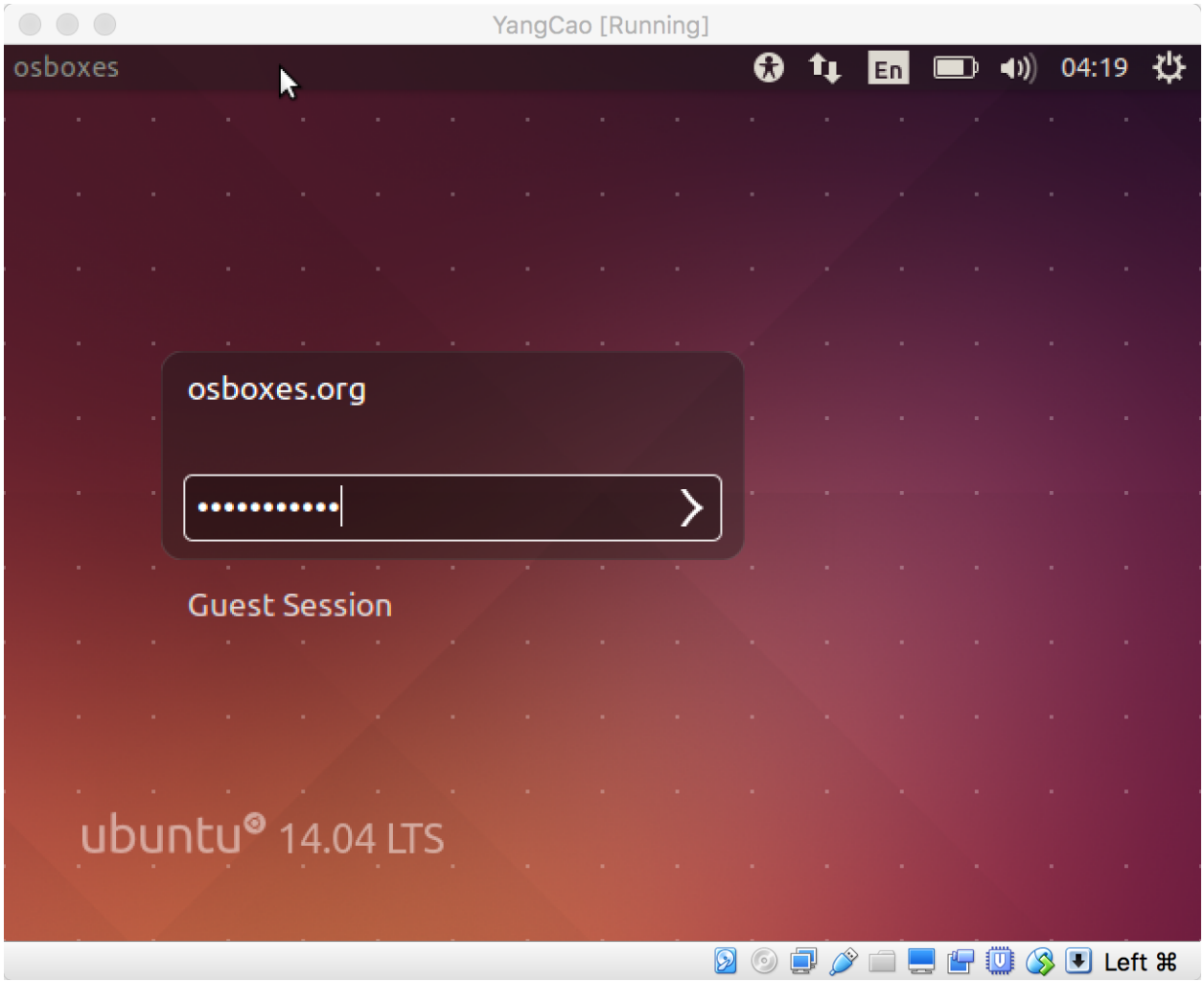
Project #3
Install ClickOS testbed

Yang Cao
C14189452

The primary objective of this project is to set up a virtual machine which runs ClickOS and test ClickOS using simple exercises.

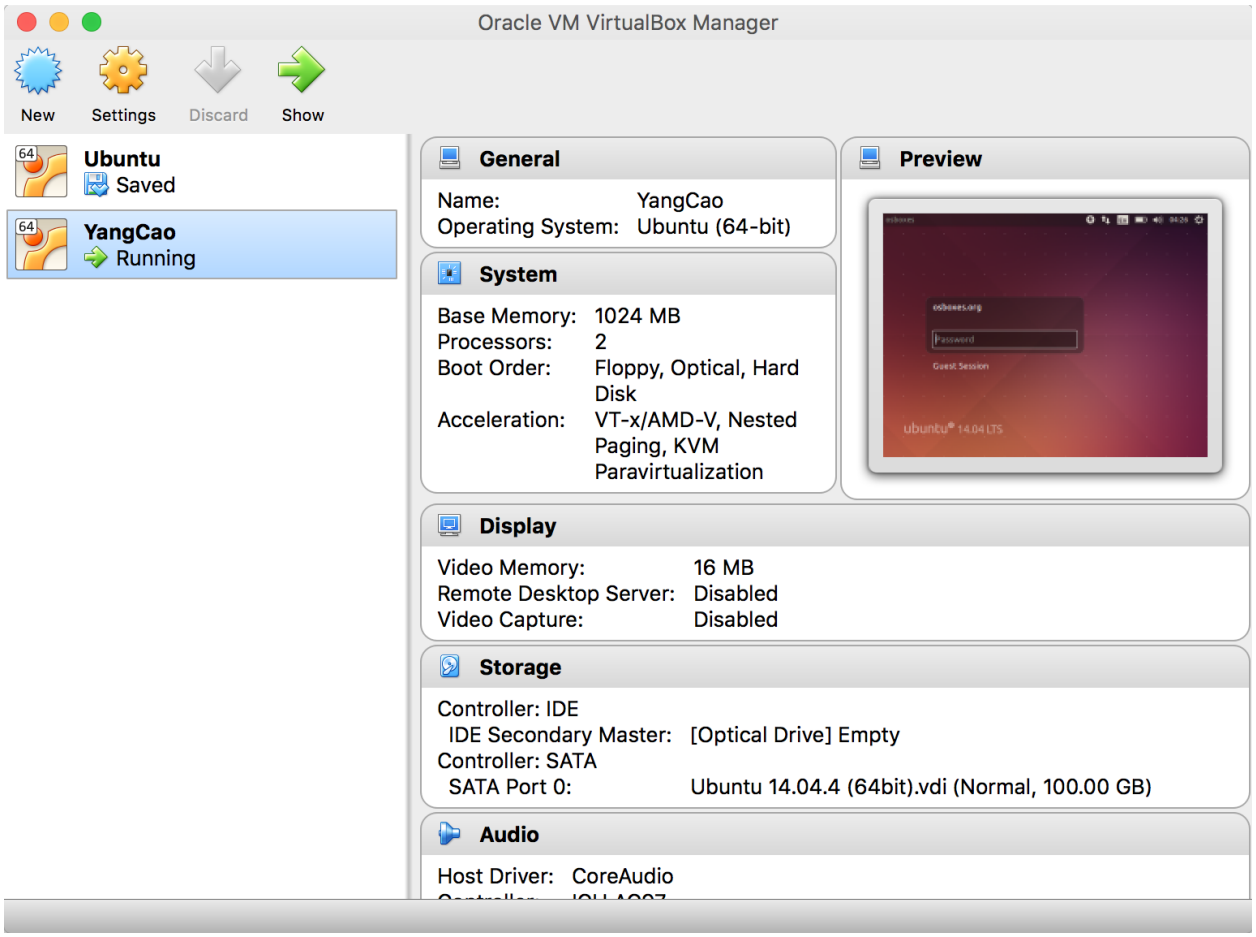
1. Download VM image and install VirtualBox

I downloaded a virtual machine image with Ubuntu 14.04.4 (LTS) 64-bit and added it into my VirtualBox. Also, I renamed it as “YangCao”.



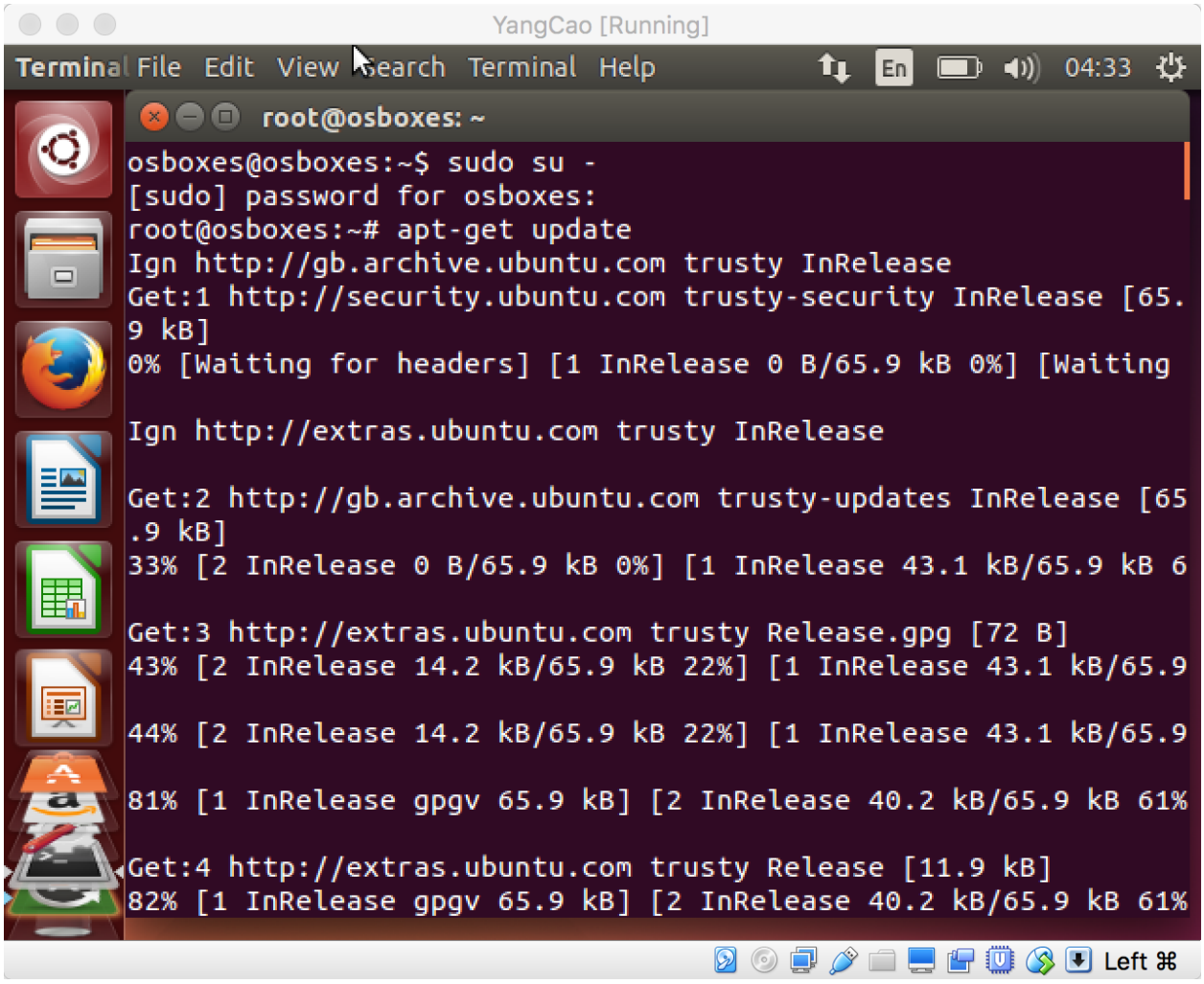
2. Configure the VM

According to what is required in this part, I set the memory of this VM image to 1024MB and the number of processors to 2. The following screenshot shows the result.



3. Install Xen

In this task, Xen is needed to support the Network Function Virtualization (NFV). So first of all, I needed to switch into the root privilege by executing “sudo su -”. Then I installed dependencies for Xen use a series of commands as what the following screenshots show.




```
root@osboxes:~# apt-get install build-essential wget
```

```
root@osboxes:~# apt-get install bcc bin86 gawk bridge-utils ipr
oute libcurl3 libcurl3 libcurl4-openssl-dev bzip2 module-init-t
ools transfig tgif
```

```
root@osboxes:~# apt-get install texinfo texlive-latex-base texl
ive-latex-recommended
```

```
root@osboxes:~# apt-get install texlive-fonts-extra texlive-fon
ts-recommended pciutils-dev mercurial
```

```
root@osboxes:~# apt-get install make gcc libc6-dev zlib1g-dev p
ython python-dev python-twisted
```

```
root@osboxes:~# apt-get install libncurses5-dev patch libsdl-de
v libjpeg62-dev
```

```
root@osboxes:~# apt-get install libvncserver-dev
```

```
root@osboxes:~# apt-get install iasl libbz2-dev e2fslibs-dev gi
t-core uuid-dev ocaml-findlib
```

```
root@osboxes:~# apt-get install libx11-dev bison flex xz-utils
libyajl-dev
```

```
root@osboxes:~# apt-get install gettext libpixman-1-dev
```

```
root@osboxes:~# apt-get build-dep xen
```

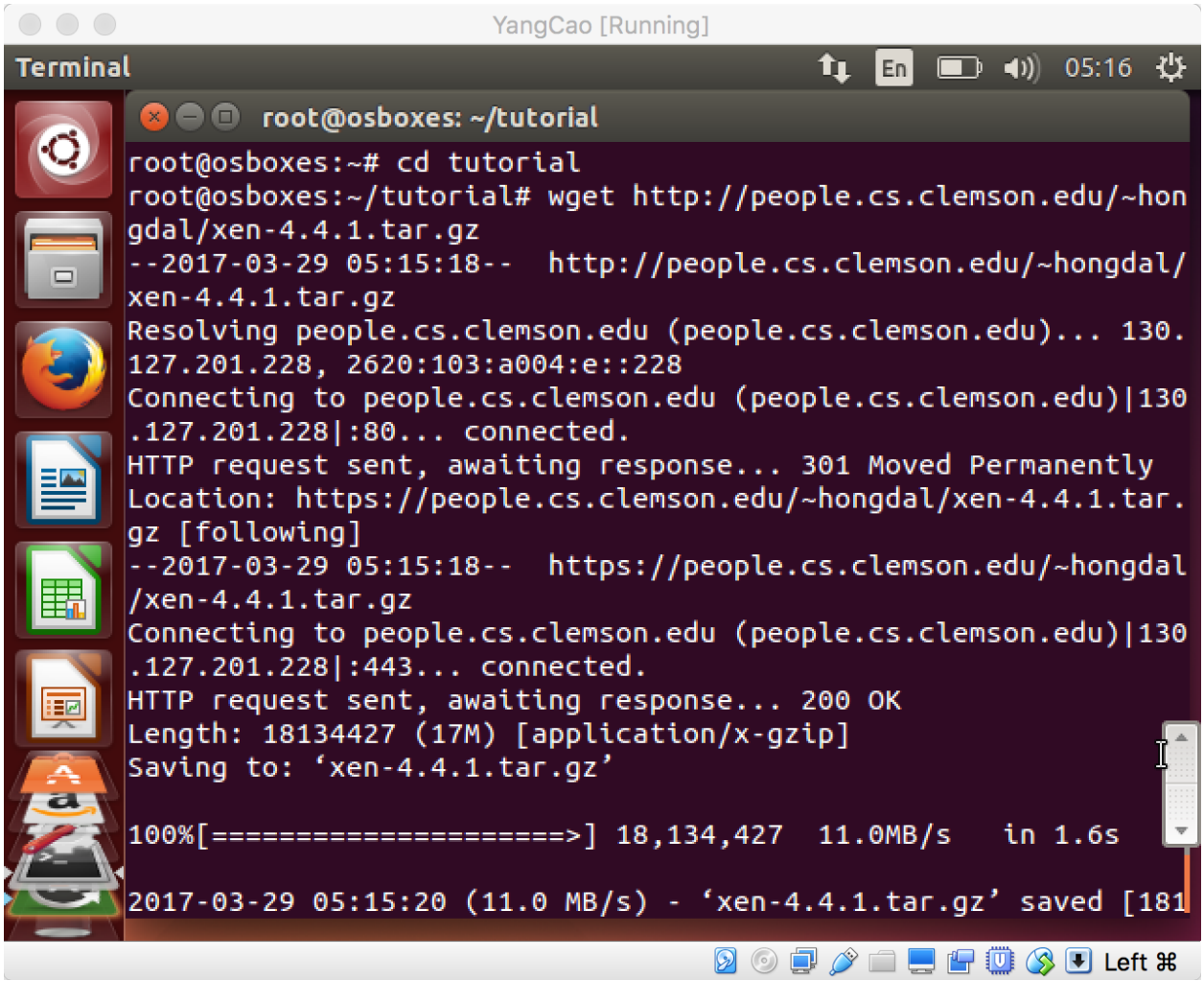
```
root@osboxes:~# apt-get install libglib2.0-dev
```

```
root@osboxes:~# apt-get install libyajl-dev
```

Then I made a directory called tutorial.

```
root@osboxes:~# mkdir tutorial
```

Download Xen-4.4.1 in the tutorial directory:




```
root@osboxes:~/tutorial# tar -xf xen-4.4.1.tar.gz
```

Compile and build Xen:

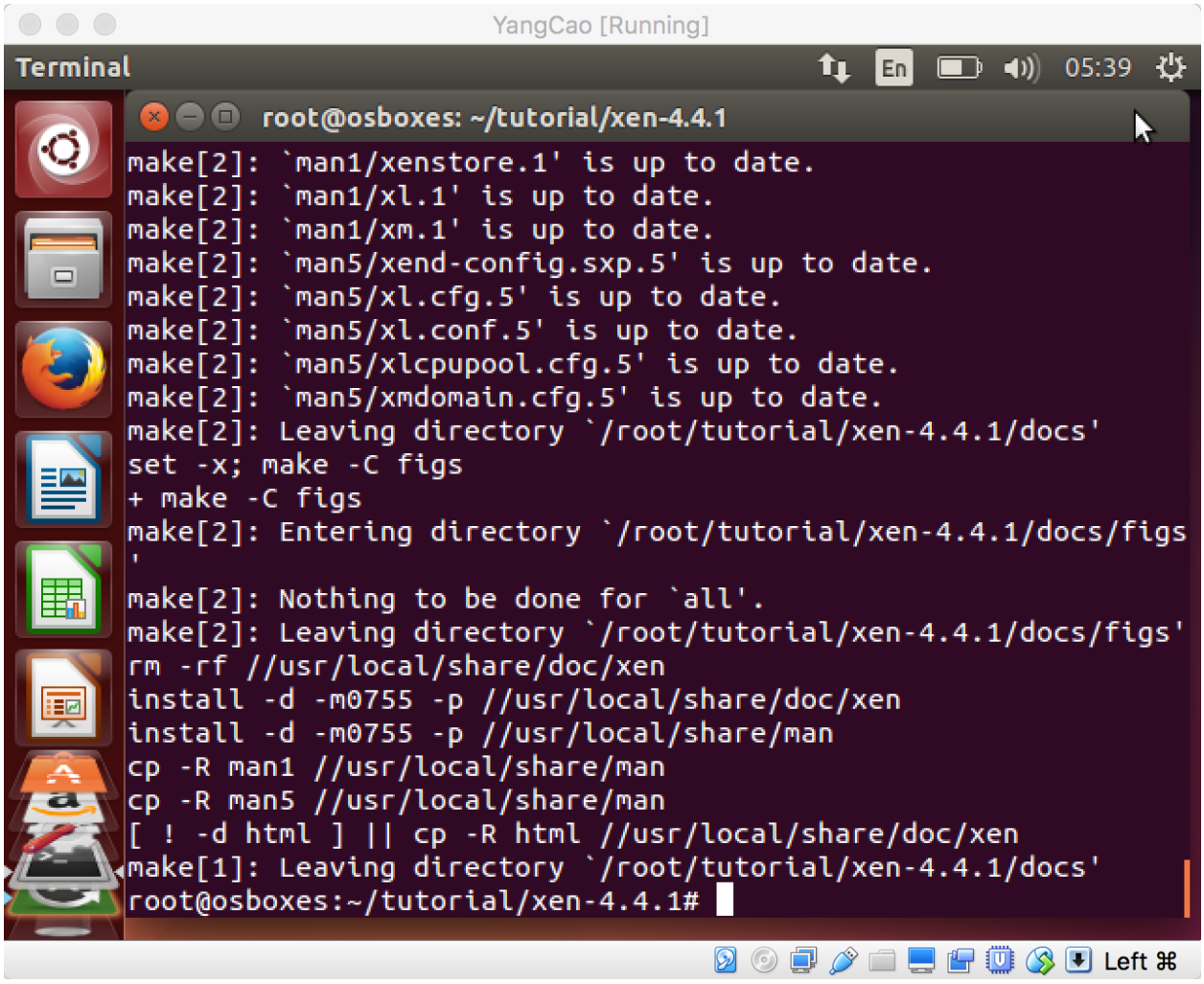
```
root@osboxes:~/tutorial# export XEN_ROOT=`pwd`/xen-4.4.1
root@osboxes:~/tutorial# cd $XEN_ROOT
root@osboxes:~/tutorial/xen-4.4.1# ./configure
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
Will build the following subsystems:
  xen
  kernels
  tools
  stubdom
  docs
configure: creating ./config.status
config.status: creating ./config/Toplevel.mk
=== configuring in tools (/root/tutorial/xen-4.4.1/tools)
configure: running /bin/bash ./configure --disable-option-check
ing '--prefix=/usr/local'  --cache-file=/dev/null --srcdir=.
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
```

```
root@osboxes:~/tutorial/xen-4.4.1# make world
```

```
root@osboxes:~/tutorial/xen-4.4.1# make install
```

Configure the system to boot from Xen:

I took a snapshot of the VM before doing this part.



And then I executed a series of commands as what the following screenshots show.

```
root@osboxes:~/tutorial/xen-4.4.1# /sbin/ldconfig
root@osboxes:~/tutorial/xen-4.4.1# update-rc.d xencommons defau
lts 19 18
update-rc.d: warning: default start runlevel arguments (2 3 4 5
) do not match xencommons Default-Start values (2 3 5)
 Adding system startup for /etc/init.d/xencommons ...
   /etc/rc0.d/K18xencommons -> ../init.d/xencommons
   /etc/rc1.d/K18xencommons -> ../init.d/xencommons
   /etc/rc6.d/K18xencommons -> ../init.d/xencommons
   /etc/rc2.d/S19xencommons -> ../init.d/xencommons
   /etc/rc3.d/S19xencommons -> ../init.d/xencommons
   /etc/rc4.d/S19xencommons -> ../init.d/xencommons
   /etc/rc5.d/S19xencommons -> ../init.d/xencommons
```

```
root@osboxes:~/tutorial/xen-4.4.1# update-rc.d xencommons defau
lts 21 20
update-rc.d: warning: default start runlevel arguments (2 3 4 5
) do not match xencommons Default-Start values (2 3 5)
 System start/stop links for /etc/init.d/xencommons already exi
st.
```

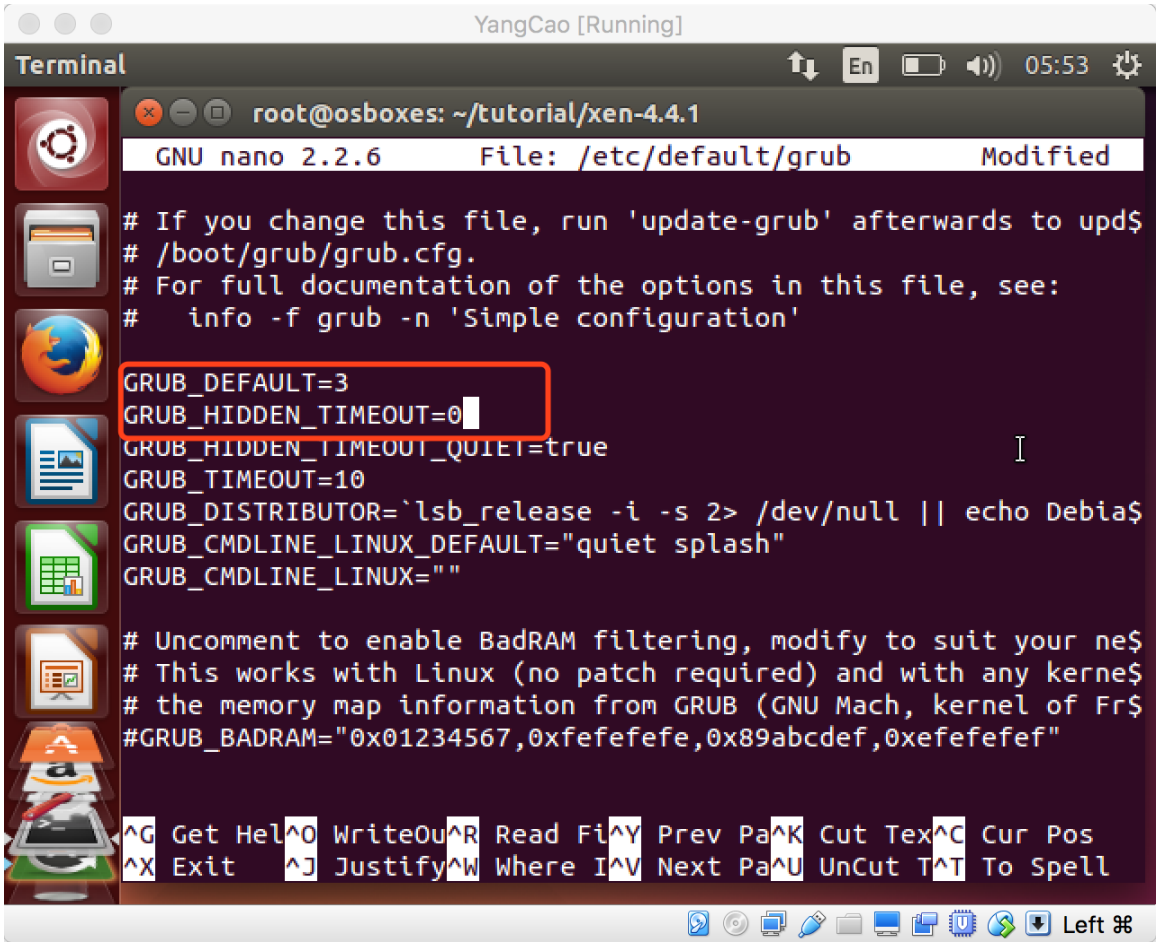
```
root@osboxes:~/tutorial/xen-4.4.1# update-rc.d xen-watchdog def
aults 22 23
update-rc.d: warning: default start runlevel arguments (2 3 4 5
) do not match xen-watchdog Default-Start values (2 3 5)
 Adding system startup for /etc/init.d/xen-watchdog ...
   /etc/rc0.d/K23xen-watchdog -> ../init.d/xen-watchdog
   /etc/rc1.d/K23xen-watchdog -> ../init.d/xen-watchdog
   /etc/rc6.d/K23xen-watchdog -> ../init.d/xen-watchdog
   /etc/rc2.d/S22xen-watchdog -> ../init.d/xen-watchdog
   /etc/rc3.d/S22xen-watchdog -> ../init.d/xen-watchdog
   /etc/rc4.d/S22xen-watchdog -> ../init.d/xen-watchdog
   /etc/rc5.d/S22xen-watchdog -> ../init.d/xen-watchdog
root@osboxes:~/tutorial/xen-4.4.1#
```

Update some of the configuration lines at /etc/default/grub to the appropriate values:
Firstly, I installed nano as a text editor.

```
root@osboxes:~/tutorial/xen-4.4.1# apt-get install nano
Reading package lists... Done
Building dependency tree
Reading state information... Done
nano is already the newest version.
The following package was automatically installed and is no lo
ger required:
  libjpeg62
Use 'apt-get autoremove' to remove it.
0 to upgrade, 0 to newly install, 0 to remove and 283 not to up
grade.
```

Then, I needed to change the values of `GRUB_DEFAULT` and `GRUB_HIDDEN_TIMEOUT` into the required ones. The results are showed in the following screenshots.

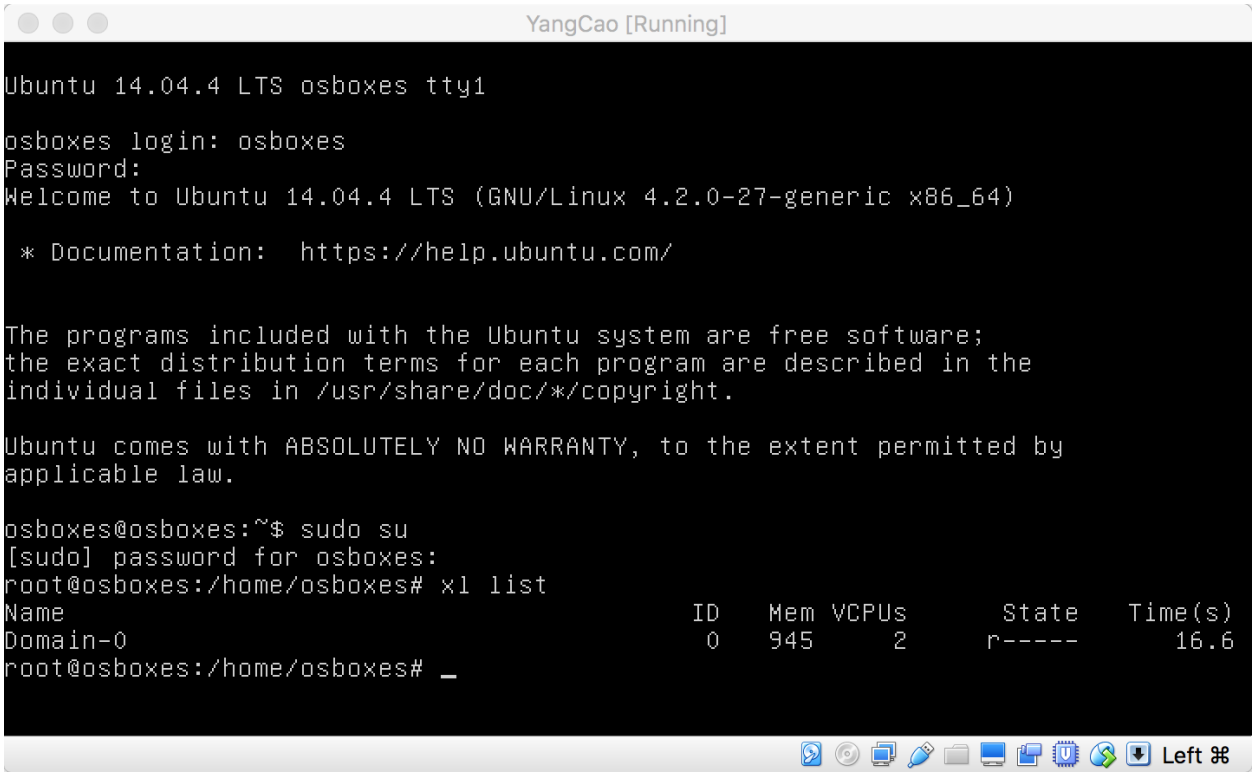
```
root@osboxes:~/tutorial/xen-4.4.1# nano /etc/default/grub
```



Update grub and reboot the system:

```
root@osboxes:~/tutorial/xen-4.4.1# update-grub
Generating grub configuration file ...
Warning: Setting GRUB_TIMEOUT to a non-zero value when GRUB_HID
DEN_TIMEOUT is set is no longer supported.
Found linux image: /boot/vmlinuz-4.2.0-27-generic
Found initrd image: /boot/initrd.img-4.2.0-27-generic
dpkg: warning: version '/boot/xen.gz' has bad syntax: version n
umber does not start with a digit
Found linux image: /boot/vmlinuz-4.2.0-27-generic
Found initrd image: /boot/initrd.img-4.2.0-27-generic
Found linux image: /boot/vmlinuz-4.2.0-27-generic
Found initrd image: /boot/initrd.img-4.2.0-27-generic
Found linux image: /boot/vmlinuz-4.2.0-27-generic
Found initrd image: /boot/initrd.img-4.2.0-27-generic
Found linux image: /boot/vmlinuz-4.2.0-27-generic
Found initrd image: /boot/initrd.img-4.2.0-27-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
```

Verify Xen has been installed successfully:
After rebooting, the system started a graphic user interface (GUI) automatically. However, the virtual machine did not support GUI in Xen. So I needed to switch to a command line interface (CLI). After logging in, I switched to the root privilege. And I used “xl list” command to verify if Xen had been installed successfully. The expected outputs are showed in following screenshot.



4. Build ClickOS

Get source codes:
I downloaded clickos codes into the tutorial directory. The steps and results are showed in the following screenshots.

```
root@osboxes:/home/osboxes# cd ~/tutorial
root@osboxes:~/tutorial# wget http://people.cs.clemson.edu/~hongdal/clickos.tar
--2017-03-29 06:07:31--  http://people.cs.clemson.edu/~hongdal/clickos.tar
Resolving people.cs.clemson.edu (people.cs.clemson.edu)... 130.127.201.228, 2620
:103:a004:e::228
Connecting to people.cs.clemson.edu (people.cs.clemson.edu)|130.127.201.228|:80.
.. connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://people.cs.clemson.edu/~hongdal/clickos.tar [following]
--2017-03-29 06:07:31--  https://people.cs.clemson.edu/~hongdal/clickos.tar
Connecting to people.cs.clemson.edu (people.cs.clemson.edu)|130.127.201.228|:443
... connected.
HTTP request sent, awaiting response... 200 OK
Length: 41185280 (39M) [application/x-tar]
Saving to: ‘clickos.tar’

100%[======================================>] 41,185,280  13.5MB/s   in 2.9s

2017-03-29 06:07:34 (13.5 MB/s) - ‘clickos.tar’ saved [41185280/41185280]
```

```
root@osboxes:~/tutorial# wget http://people.cs.clemson.edu/~hongdal/mini-os.tar
--2017-03-29 06:09:18--  http://people.cs.clemson.edu/~hongdal/mini-os.tar
Resolving people.cs.clemson.edu (people.cs.clemson.edu)... 130.127.201.228, 2620
:103:a004:e::228
Connecting to people.cs.clemson.edu (people.cs.clemson.edu)|130.127.201.228|:80.
.. connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://people.cs.clemson.edu/~hongdal/mini-os.tar [following]
--2017-03-29 06:09:23--  https://people.cs.clemson.edu/~hongdal/mini-os.tar
Connecting to people.cs.clemson.edu (people.cs.clemson.edu)|130.127.201.228|:443
... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1208320 (1.2M) [application/x-tar]
Saving to: ‘mini-os.tar’

100%[======================================>] 1,208,320   --.-K/s   in 0.1s

2017-03-29 06:09:23 (9.86 MB/s) - ‘mini-os.tar’ saved [1208320/1208320]
```

```
root@osboxes:~/tutorial# wget http://people.cs.clemson.edu/~hongdal/cosmos.tar
--2017-03-29 06:10:52--  http://people.cs.clemson.edu/~hongdal/cosmos.tar
Resolving people.cs.clemson.edu (people.cs.clemson.edu)... 130.127.201.228, 2620
:103:a004:e::228
Connecting to people.cs.clemson.edu (people.cs.clemson.edu)|130.127.201.228|:80.
.. connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://people.cs.clemson.edu/~hongdal/cosmos.tar [following]
--2017-03-29 06:10:52--  https://people.cs.clemson.edu/~hongdal/cosmos.tar
Connecting to people.cs.clemson.edu (people.cs.clemson.edu)|130.127.201.228|:443
... connected.
HTTP request sent, awaiting response... 200 OK
Length: 296960 (290K) [application/x-tar]
Saving to: ‘cosmos.tar’

100%[======================================>] 296,960     --.-K/s   in 0.04s

2017-03-29 06:10:52 (7.06 MB/s) - ‘cosmos.tar’ saved [296960/296960]
```

```
root@osboxes:~/tutorial# wget http://people.cs.clemson.edu/~hongdal/toolchain.ta
r
--2017-03-29 06:12:01--  http://people.cs.clemson.edu/~hongdal/toolchain.tar
Resolving people.cs.clemson.edu (people.cs.clemson.edu)... 130.127.201.228, 2620
:103:a004:e::228
Connecting to people.cs.clemson.edu (people.cs.clemson.edu)|130.127.201.228|:80.
.. connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://people.cs.clemson.edu/~hongdal/toolchain.tar [following]
--2017-03-29 06:12:02--  https://people.cs.clemson.edu/~hongdal/toolchain.tar
Connecting to people.cs.clemson.edu (people.cs.clemson.edu)|130.127.201.228|:443
... connected.
HTTP request sent, awaiting response... 200 OK
Length: 204800 (200K) [application/x-tar]
Saving to: 'toolchain.tar'

100%[======================================>] 204,800     --.-K/s   in 0.02s

2017-03-29 06:12:02 (7.83 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 cosmos.tar
root@osboxes:~/tutorial# tar -xf toolchain.tar
```

Set environment variables:

```
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/"
```

Replace mini-os:

MiniOS is an operating system which comes with Xen. ClickOS is built based on MiniOS. Thus, I needed to replace Xen's mini-os with ClickOS's minios.

```
root@osboxes:~/tutorial# rm -rf $MINIOS_ROOT
root@osboxes:~/tutorial# cp -rf $(pwd)"/mini-os/" $MINIOS_ROOT
```

Build toolchain:

I needed to enter the toolchain directory first by executing “cd $TOOLCHAIN_ROOT” command. Then I executed the following commands orderly.

```
root@osboxes:~/tutorial/toolchain# make_
```

```
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"
```

Build ClickOS kernel:

```
root@osboxes:~/tutorial/toolchain# cd $CLICKOS_ROOT
```

```
root@osboxes:~/tutorial/clickos# ./configure --enable-minios --with-xen=$XEN_ROO
T --with-minios=$MINIOS_ROOT_
```

```
root@osboxes:~/tutorial/clickos# make minios_
```

Build cosmos:

```
================================================================================

 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# cd $XEN_ROOT
root@osboxes:~/tutorial/xen-4.4.1# cd tools
root@osboxes:~/tutorial/xen-4.4.1/tools# make -C include
make: Entering directory `/root/tutorial/xen-4.4.1/tools/include'
make -C xen-foreign
make[1]: Entering directory `/root/tutorial/xen-4.4.1/tools/include/xen-foreign'
./checker > tmp.size
diff -u reference.size tmp.size
rm tmp.size
make[1]: Leaving directory `/root/tutorial/xen-4.4.1/tools/include/xen-foreign'
make: Leaving directory `/root/tutorial/xen-4.4.1/tools/include'
root@osboxes:~/tutorial/xen-4.4.1/tools# cd $COSMOS_ROOT
root@osboxes:~/tutorial/cosmos# make DOMLIB=xl_
```

YangCao [Running]

```
root@osboxes:~/tutorial/clickos# cd $XEN_ROOT
root@osboxes:~/tutorial/xen-4.4.1# cd tools
root@osboxes:~/tutorial/xen-4.4.1/tools# make -C include
make: Entering directory `/root/tutorial/xen-4.4.1/tools/include'
make -C xen-foreign
make[1]: Entering directory `/root/tutorial/xen-4.4.1/tools/include/xen-foreign'
./checker > tmp.size
diff -u reference.size tmp.size
rm tmp.size
make[1]: Leaving directory `/root/tutorial/xen-4.4.1/tools/include/xen-foreign'
make: Leaving directory `/root/tutorial/xen-4.4.1/tools/include'
root@osboxes:~/tutorial/xen-4.4.1/tools# cd $COSMOS_ROOT
root@osboxes:~/tutorial/cosmos# make DOMLIB=xl
  MKD /root/tutorial/cosmos/build
  MKD /root/tutorial/cosmos/dist
  MKD /root/tutorial/cosmos/dist/bin
  MKD /root/tutorial/cosmos/dist/lib
  MKD /root/tutorial/cosmos/build/libxcl
  MKD /root/tutorial/cosmos/build/libcosmos
  MKD /root/tutorial/cosmos/build/cosmos
  CC  /root/tutorial/cosmos/src/main.c
  CC  /root/tutorial/cosmos/src/clickos.c
  CC  /root/tutorial/cosmos/src/domain_xl.c
  LD  /root/tutorial/cosmos/dist/bin/cosmos
root@osboxes:~/tutorial/cosmos# _
```

Left ⌘

Set the PATH environment variable:

```
root@osboxes:~/tutorial/cosmos# export PATH=$PATH":/$COSMOS_ROOT/dist/bin"
```

5. Install Open vSwitch

ClickOS should work with Linux bridge, VALE bridge, Open vSwitch, and many other software switches. I needed to use Open vSwitch (OVS) as the software switch. The following commands show how to install OVS.

```
root@osboxes:~/tutorial/cosmos# cd ~/tutorial
root@osboxes:~/tutorial# wget http://openvswitch.org/releases/openvswitch-2.5.0.
tar.gz_
```

```
root@osboxes:~/tutorial# tar -xf openvswitch-2.5.0.tar.gz
root@osboxes:~/tutorial# cd openvswitch-2.5.0/
root@osboxes:~/tutorial/openvswitch-2.5.0# ./configure --with-linux=/lib/modules
/`uname -r`/build_
```

YangCao [Running]

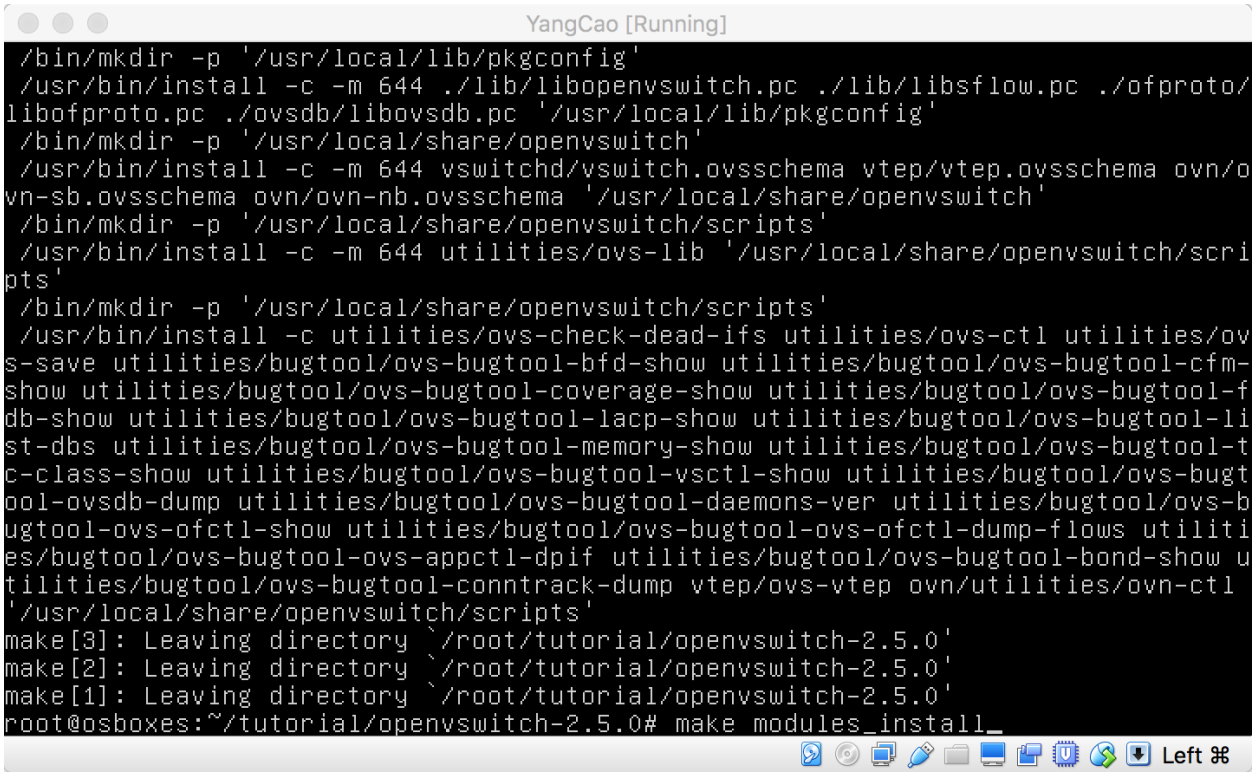
```
config.status: creating ofproto/libofproto.sym
config.status: creating lib/libsflow.sym
config.status: creating lib/libopenvswitch.sym
config.status: creating ovn/lib/libovn.sym
config.status: creating vtep/libvtep.sym
config.status: creating Makefile
config.status: creating datapath/Makefile
config.status: creating datapath/linux/Kbuild
config.status: creating datapath/linux/Makefile
config.status: creating datapath/linux/Makefile.main
config.status: creating tests/atlocal
config.status: creating lib/libopenvswitch.pc
config.status: creating lib/libsflow.pc
config.status: creating ofproto/libofproto.pc
config.status: creating ovsdb/libovsdb.pc
config.status: creating include/openvswitch/version.h
config.status: creating config.h
config.status: executing tests/atconfig commands
config.status: executing depfiles commands
config.status: executing libtool commands
config.status: executing include/openflow/openflow.h.stamp commands
config.status: executing utilities/bugtool/dummy commands
config.status: executing ovn/dummy commands
config.status: executing ovn/utilities/dummy commands
root@osboxes:~/tutorial/openvswitch-2.5.0# make_
```

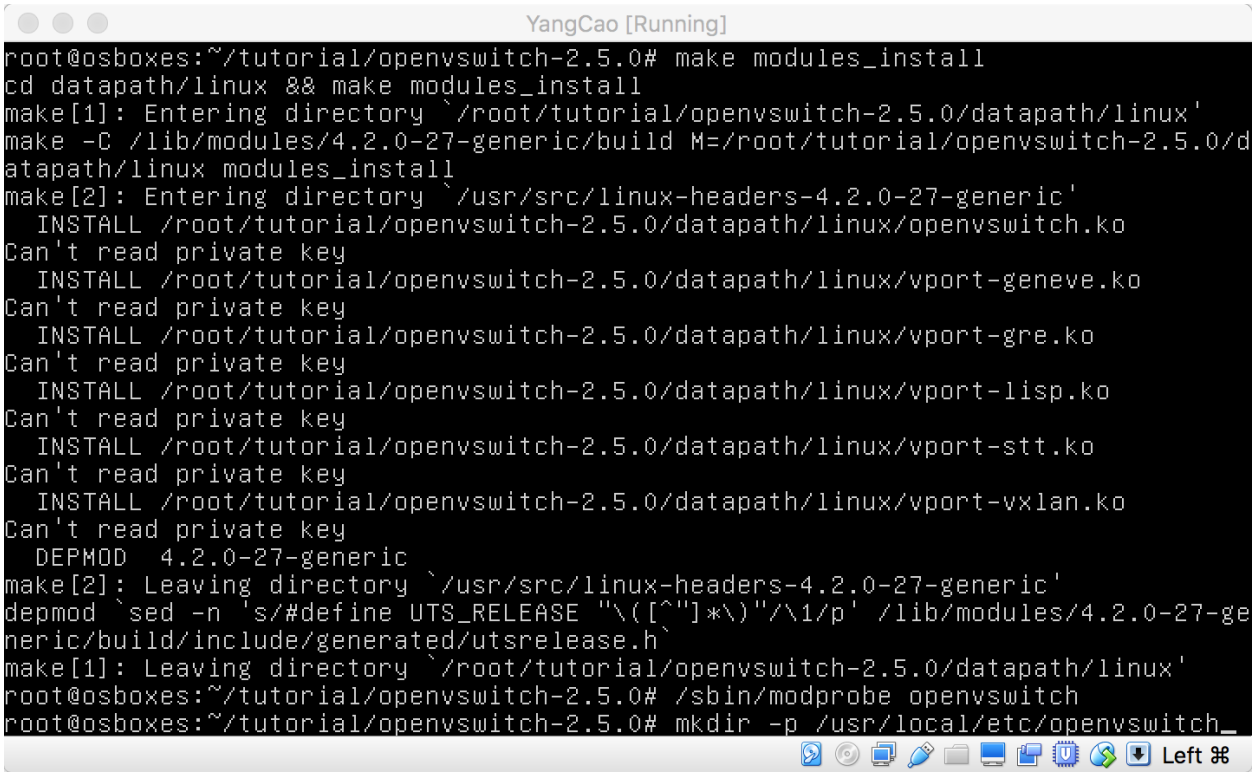
Left ⌘

YangCao [Running]

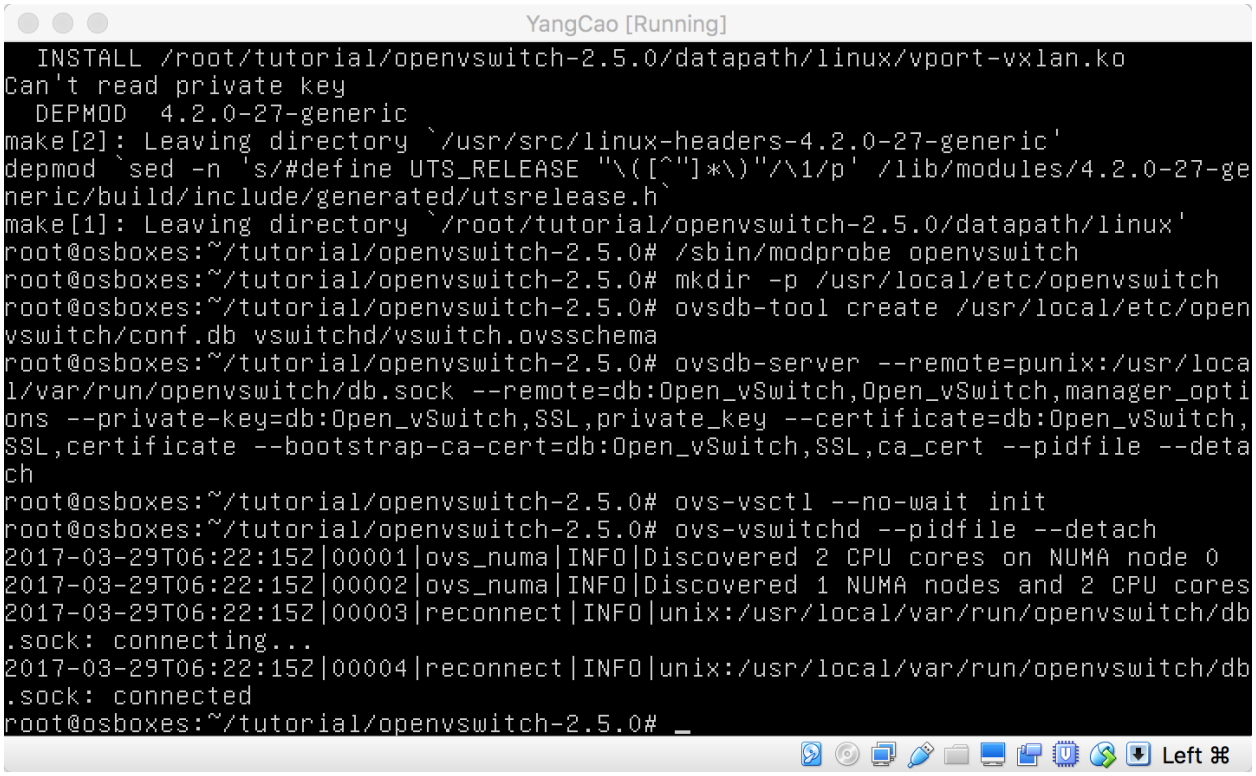
```
build-aux/check-structs -I./include include/openflow/openflow-1.2.h && \
        touch include/openflow/openflow-1.2.hstamp
PYTHONPATH=./python":"$PYTHONPATH PYTHONDONTWRITEBYTECODE=yes /usr/bin/python ./
build-aux/check-structs -I./include include/openflow/openflow-1.3.h && \
        touch include/openflow/openflow-1.3.hstamp
PYTHONPATH=./python":"$PYTHONPATH PYTHONDONTWRITEBYTECODE=yes /usr/bin/python ./
build-aux/check-structs -I./include include/openflow/openflow-1.4.h && \
        touch include/openflow/openflow-1.4.hstamp
PYTHONPATH=./python":"$PYTHONPATH PYTHONDONTWRITEBYTECODE=yes /usr/bin/python ./
build-aux/check-structs -I./include include/openflow/openflow-1.5.h && \
        touch include/openflow/openflow-1.5.hstamp
PYTHONPATH=./python":"$PYTHONPATH PYTHONDONTWRITEBYTECODE=yes /usr/bin/python ./
build-aux/check-structs -I./include include/openflow/openflow-common.h && \
        touch include/openflow/openflow-common.hstamp
PYTHONPATH=./python":"$PYTHONPATH PYTHONDONTWRITEBYTECODE=yes /usr/bin/python ./
build-aux/check-structs -I./include include/openflow/openflow.h && \
        touch include/openflow/openflow.hstamp
./build-aux/cksum-schema-check vswitchd/vswitch.ovsschema vswitchd/vswitch.ovssc
hema.stamp
./build-aux/cksum-schema-check vtep/vtep.ovsschema vtep/vtep.ovsschema.stamp
./build-aux/cksum-schema-check ovn/ovn-nb.ovsschema ovn/ovn-nb.ovsschema.stamp
./build-aux/cksum-schema-check ovn/ovn-sb.ovsschema ovn/ovn-sb.ovsschema.stamp
make[2]: Leaving directory `/root/tutorial/openvswitch-2.5.0'
make[1]: Leaving directory `/root/tutorial/openvswitch-2.5.0'
root@osboxes:~/tutorial/openvswitch-2.5.0# make install_
```

Left ⌘



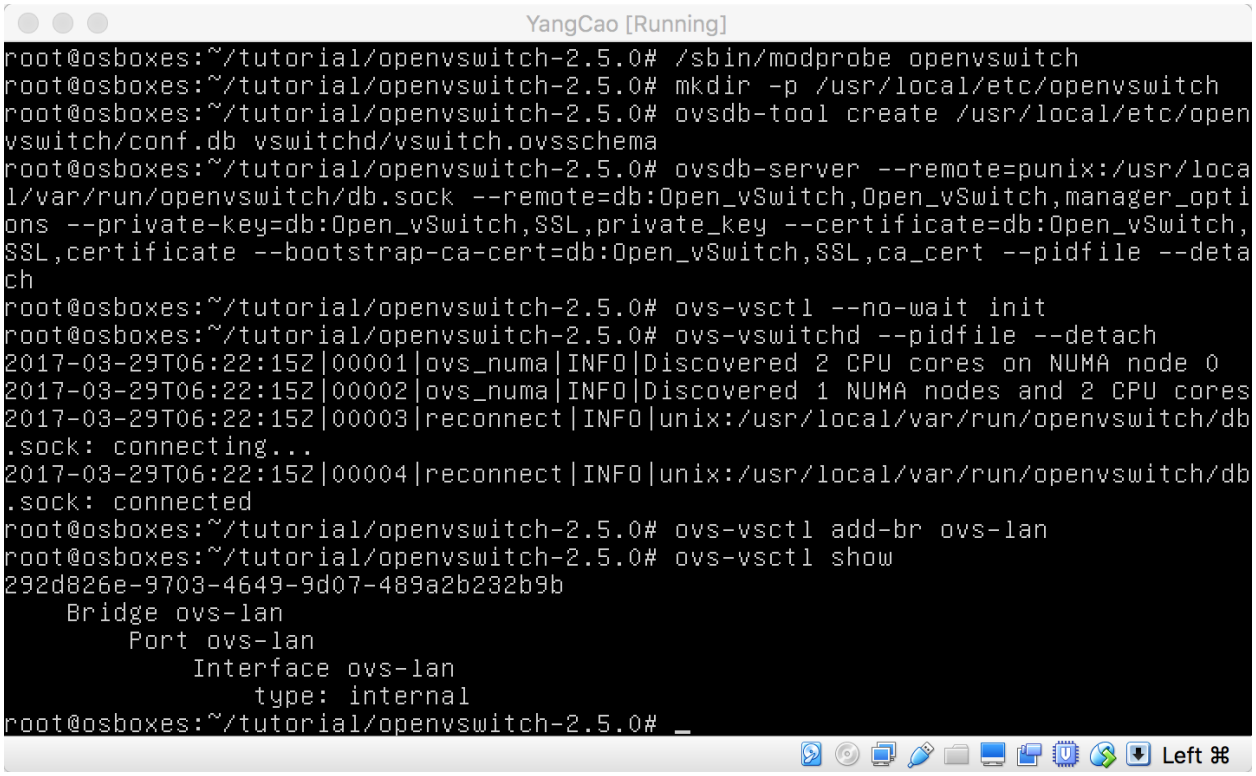


Configure and run OVS:



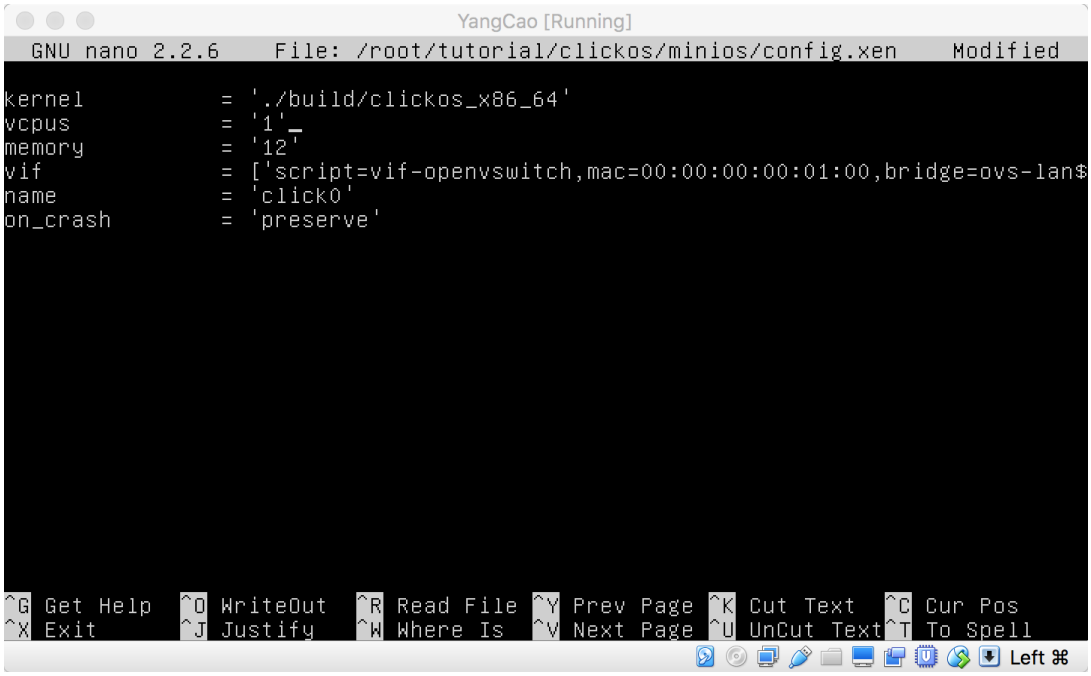
Create a bridge on OVS:

I used command “ovs-vsctl show” to verify that OVS was running correctly. The outputs showed in the following screenshot are expected.

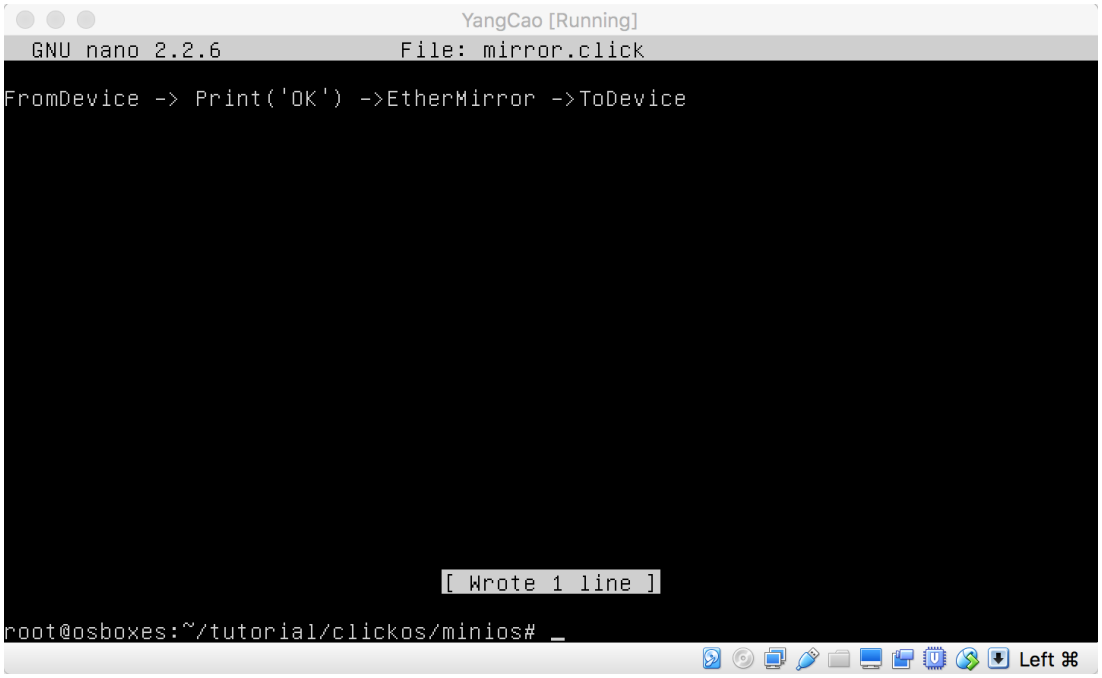


6. Start a ClickOS instance

Create configuration file for Xen:
I changed my working directory to “tutorial/clickos/minios” and updated some of the values in config.xen. The file after modified is showed in the following screenshot.



Create configuration file for Click:
I created and saved the example script as a file, mirror.click.

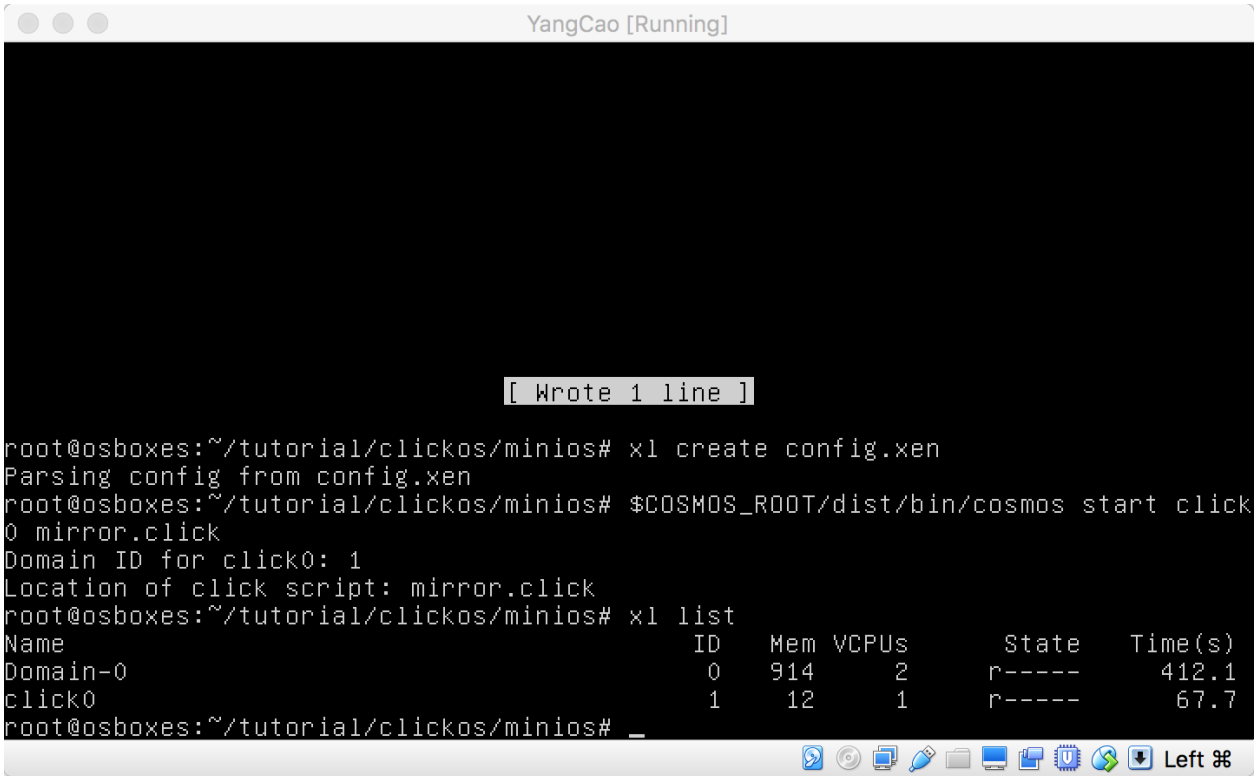


Start ClickOS instance:

I used cosmos to start a ClickOS instance and checked whether $COSMOS_ROOT was intact by typing “echo $COSMOS_ROOT” and set it appropriately.

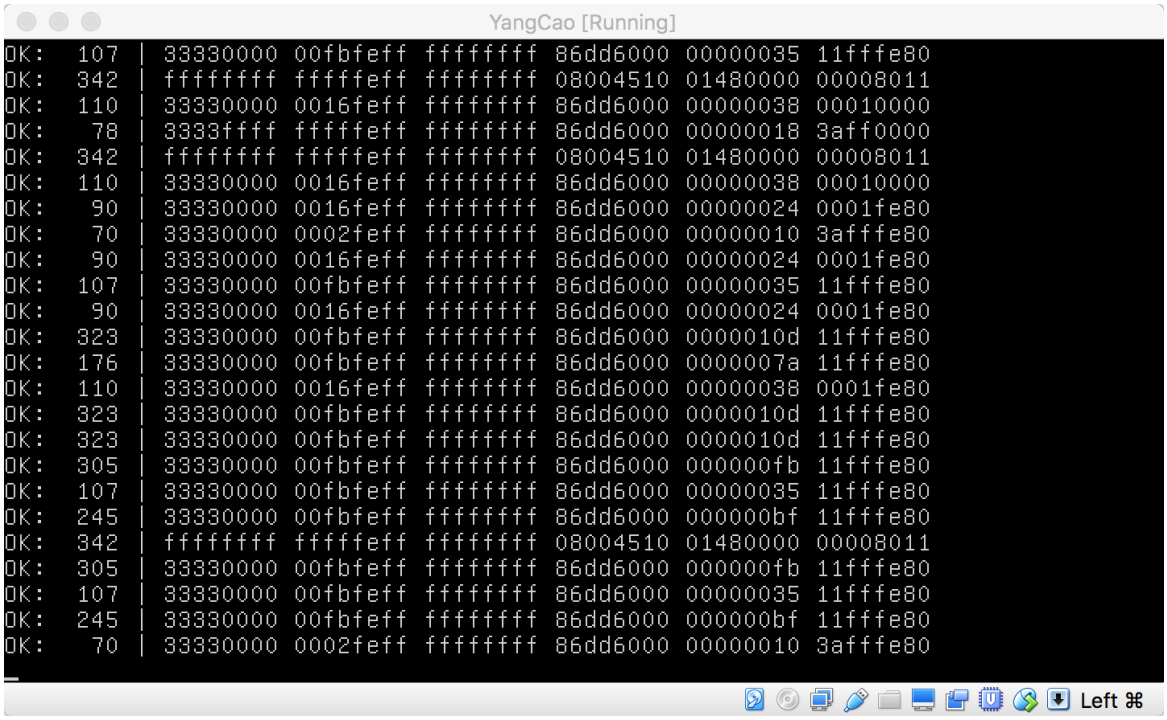
```
root@osboxes:~/tutorial/clickos/minios# xl create config.xen
Parsing config from config.xen
root@osboxes:~/tutorial/clickos/minios# $COSMOS_ROOT/dist/bin/cosmos start click
0 mirror.click
Domain ID for click0: 1
Location of click script: mirror.click
root@osboxes:~/tutorial/clickos/minios# _
```

To verify that the ClickOS instance was running, I used a “xl list” command.



To watch the output of click0, I executed “xl console click0” command and got the expected result showed in the following screenshot.

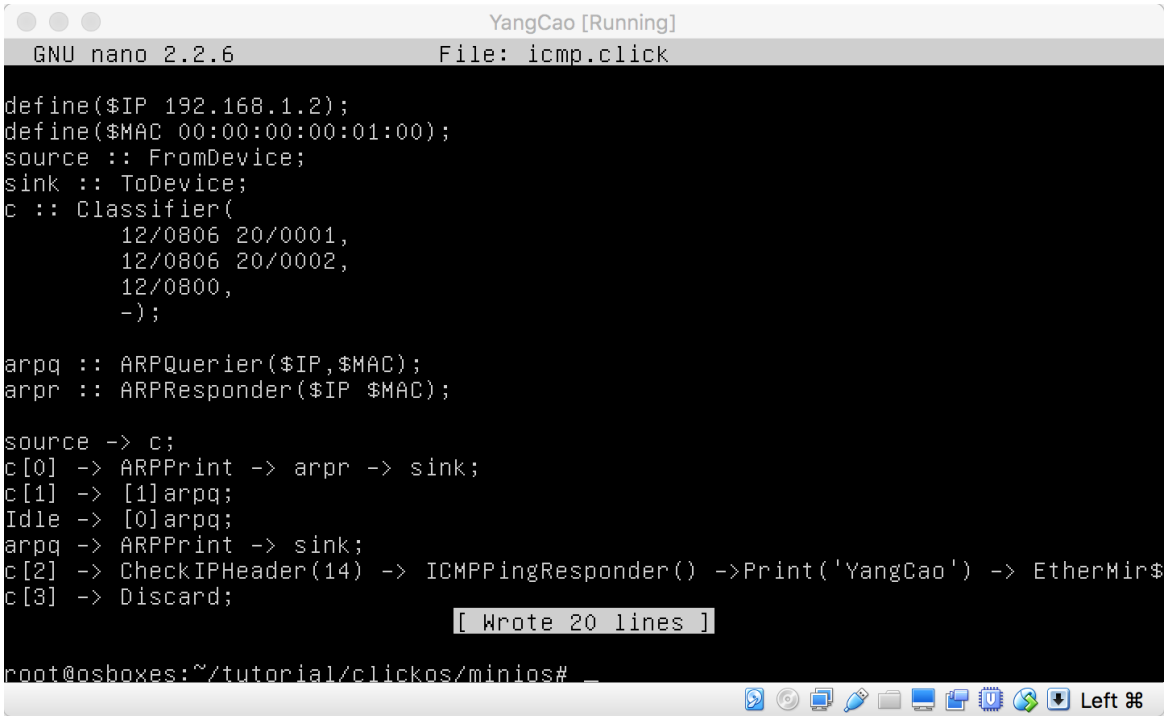
```
root@osboxes:~/tutorial/clickos/minios# xl console click0_
```



7. Exercise

ICMP Responder:

I created a Click configuration file as icmp.click and set the IP and MAC addresses appropriately Also, I made my name printed.



Kill existing ClickOS instances by executing “xl destroy click0” command. Then I needed to start a new instance:

```
root@osboxes:~/tutorial/clickos/minios# xl destroy click0
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 click
0 icmp.click
Domain ID for click0: 3
Location of click script: icmp.click
root@osboxes:~/tutorial/clickos/minios# _
```

Then I added my VM's network interface (assuming it's eth0, with IP, 192.168.1.1) to OVS so that I could ping the ClickOS instance from my VM:

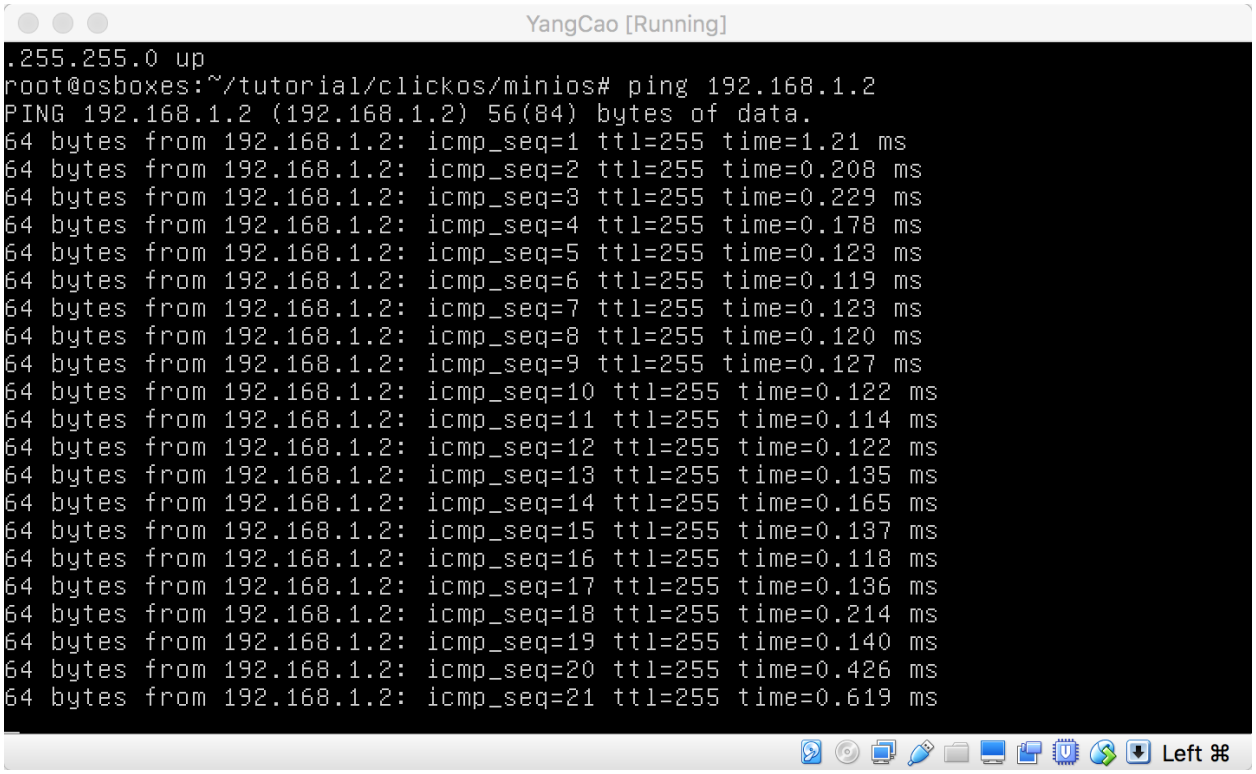
YangCao [Running]

```
c[0] -> ARPPrint -> arpr -> sink;
c[1] -> [1]arpq;
Idle -> [0]arpq;
arpq -> ARPPrint -> sink;
c[2] -> CheckIPHeader(14) -> ICMPPingResponder() ->Print('YangCao') -> EtherMir$
c[3] -> Discard;
                              [ Wrote 20 lines ]

root@osboxes:~/tutorial/clickos/minios# xl create config.xen
Parsing config from config.xen
libxl: error: libxl.c:319:libxl__domain_rename: domain with name "click0" alread
y exists.
libxl: error: libxl_create.c:743:initiate_domain_create: cannot make domain: -6
root@osboxes:~/tutorial/clickos/minios# xl destroy click0
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 click
0 icmp.click
Domain ID for click0: 3
Location of click script: icmp.click
root@osboxes:~/tutorial/clickos/minios# ifconfig eth0 0
root@osboxes:~/tutorial/clickos/minios# ovs-vsctl add-port ovs-lan eth0
root@osboxes:~/tutorial/clickos/minios# ifconfig ovs-lan 192.168.1.1 netmask 255
.255.255.0 up
root@osboxes:~/tutorial/clickos/minios# ping 192.168.1.2_
```

Left ⌘

Then ping the ClickOS instance from my VM. I received the responds from the ClickOS.



I used the following command to see the outputs of the ClickOS instance. My name was printed.

```
root@osboxes:~/tutorial/clickos/minios# xl console click0
```