

What is Vagrant?

You can think of vagrant as a virtual machine manager for building and maintaining vm's. It is compatible with Windows, macOS, Linux. We will be using vagrant to

Prerequisite

- Download the version of vagrant that is for you system (<https://www.vagrantup.com/downloads.html>)
 - Note: I am on 2.0.3
 - To verify that vagrant was installed properly, run the command `vagrant version`. You should get an output similar to the one below.

```
C:\Users\mugra_000>vagrant version
Installed Version: 2.0.3
Latest Version: 2.0.3
```

- Install an updated version of virtual box (<https://www.virtualbox.org/wiki/Downloads>)
 - Note: I am on 5.2.8 (Make sure that the version of vagrant you install supports the version of virtual box you install)

Instructions

Setting up the vm with vagrant

- Fork my repository (https://github.com/LuvneeshM/os3224_vagrant) and then *git clone* a local version of the freshly forked repository. You will be using this repository to submit homework assignments, via tagged released on github (<https://help.github.com/articles/creating-releases/>).
 - The release must be made before the deadline. Once you've made the release, submit the URL of the release so that we know that you got it in before your deadline. (Make a private repo and add give permissions to all the TAs)
- Using terminal/command prompt `cd` into the cloned repo
- To start up the vm run the command: `vagrant up`
 - Note: This might take a few minutes since it will be running for the first time and thus setting up a few things
 - If you open VirtualBox, you should notice a new named xv6_box has been made

- Once it finishes, the vm will have booted and you should see something similar to below

```
C:\WINDOWS\system32\cmd.exe
==> default: Forwarding ports...
      default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Running 'pre-boot' VM customizations...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
      default: SSH address: 127.0.0.1:2222
      default: SSH username: vagrant
      default: SSH auth method: private key
      default: Warning: Connection aborted. Retrying...
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
      default: The guest additions on this VM do not match the installed version
of
      default: VirtualBox! In most cases this is fine, but in rare cases it can
      default: prevent things such as shared folders from working properly. If y
ou see
      default: shared folder errors, please make sure the guest additions within
the
      default: virtual machine match the version of VirtualBox you have installe
d on
      default: your host and reload your VM.
      default:
      default: Guest Additions Version: 4.1.44
      default: VirtualBox Version: 5.2
==> default: Mounting shared folders...
      default: /vagrant => C:/Users/mugra_000/Desktop/os_vagrant_material/os3224
_vagrant
      default: /tmp/vagrant-puppet/manifests-a11d1078b1b1f2e3bdea27312f6ba513 =>
C:/Users/mugra_000/Desktop/os_vagrant_material/os3224_vagrant/manifests
==> default: Machine already provisioned. Run `vagrant provision` or use the `
--provision`
==> default: flag to force provisioning. Provisioners marked to run always wil
l still run.

C:\Users\mugra_000\Desktop\os_vagrant_material\os3224_vagrant>
```

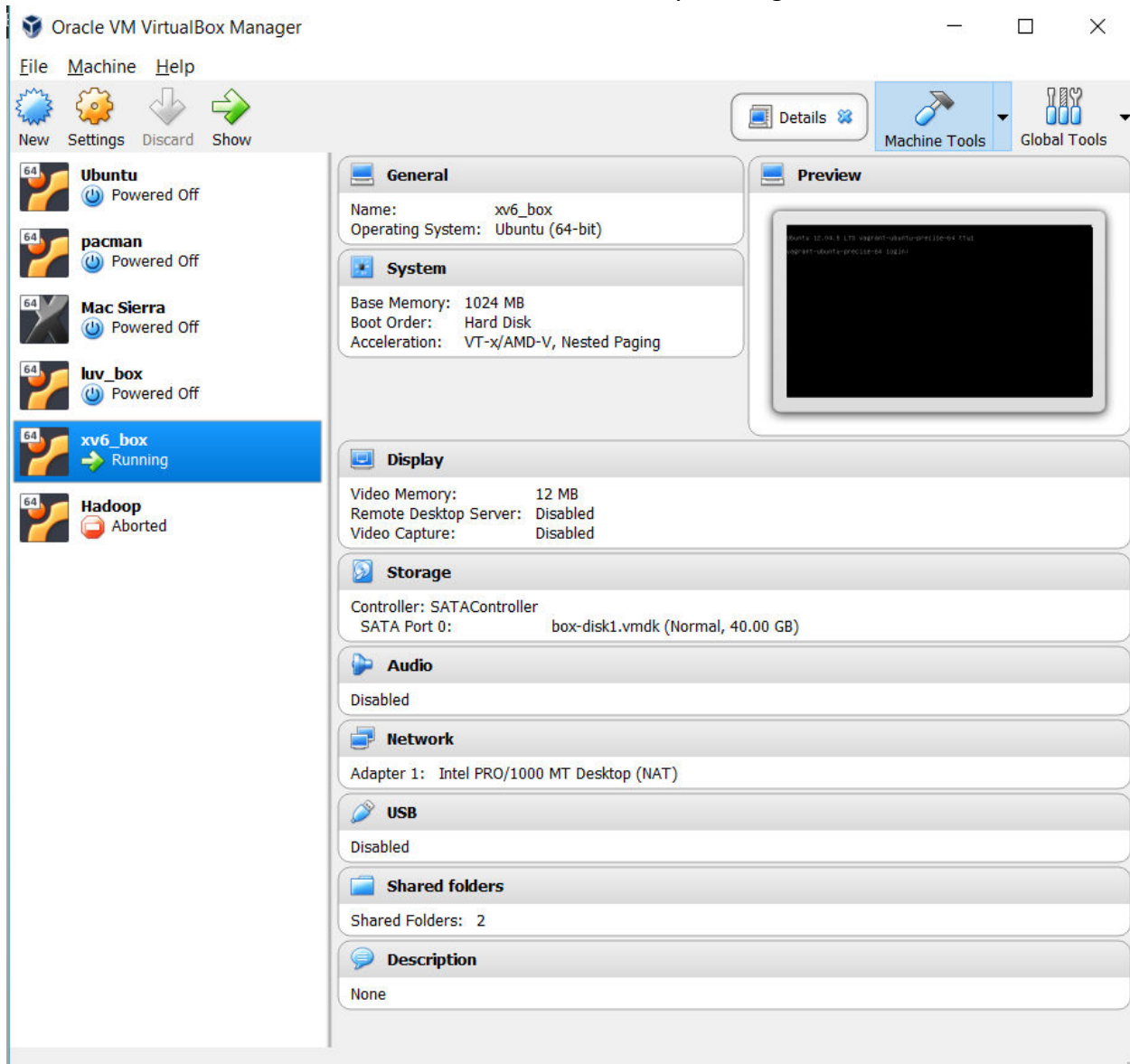
- Run the command *vagrant status*, to verify that the vm is up and running

```
C:\Users\mugra_000\Desktop\os_vagrant_material\os3224_vagrant>vagrant status
Current machine states:

default                                running (virtualbox)

The VM is running. To stop this VM, you can run `vagrant halt` to
shut it down forcefully, or you can run `vagrant suspend` to simply
suspend the virtual machine. In either case, to restart it again,
simply run `vagrant up`.
```

- Note that the vm in virtual box will also say running



- We will now ssh into the machine by running the command: *vagrant ssh*

```
C:\Users\mugra_000\Desktop\os_vagrant_material\os3224_vagrant>vagrant ssh
Welcome to Ubuntu 12.04.5 LTS (GNU/Linux 3.2.0-126-virtual x86_64)

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

System information as of Sun Apr  8 09:23:44 UTC 2018

System load:  0.0               Processes:           66
Usage of /:   3.3% of 39.37GB   Users logged in:    0
Memory usage: 5%               IP address for eth0: 10.0.2.15
Swap usage:   0%

Graph this data and manage this system at:
  https://landscape.canonical.com/

Get cloud support with Ubuntu Advantage Cloud Guest:
  http://www.ubuntu.com/business/services/cloud

New release '14.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

This Ubuntu 12.04 LTS system is past its End of Life, and is no longer
receiving security updates. To protect the integrity of this system, it's
critical that you enable Extended Security Maintenance updates:
* https://www.ubuntu.com/esm

Last login: Sun Apr  1 00:46:52 2018 from 10.0.2.2
vagrant@vagrant-ubuntu-precise-64:~$
```

- If the *vagrant ssh* command does not work, we will connect to the machine manually:
 - Run the command: *vagrant ssh-config*
 - From what shows up in terminal, you can run the following command (make sure to fill in the variables with those from what shows up after running the *vagrant ssh-config* command)
`ssh <USER>@<HOSTNAME> -p <PORT> -i <IDENTITYFILE>`

```

Luvneesh@Luvneesh-Laptop MINGW64 ~
$ cd Desktop/os_vagrant_material/os3224_vagrant/

Luvneesh@Luvneesh-Laptop MINGW64 ~/Desktop/os_vagrant_material/os3224_vagrant (master)
$ vagrant ssh-config
Host default
  HostName 127.0.0.1
  User vagrant
  Port 2222
  UserKnownHostsFile /dev/null
  StrictHostKeyChecking no
  PasswordAuthentication no
  IdentityFile C:/Users/mugra_000/Desktop/os_vagrant_material/os3224_vagrant/.vagrant/machines/default/virtualbox/private_key
  IdentitiesOnly yes
  LogLevel FATAL

Luvneesh@Luvneesh-Laptop MINGW64 ~/Desktop/os_vagrant_material/os3224_vagrant (master)
$ ssh vagrant@127.0.0.1 -p 2222 -i C:/Users/mugra_000/Desktop/os_vagrant_material/os3224_vagrant/.vagrant/machines/default/virtualbox/private_key
Welcome to Ubuntu 12.04.5 LTS (GNU/Linux 3.2.0-126-virtual x86_64)

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

System information as of Sun Apr  8 09:34:06 UTC 2018

System load:  0.0               Processes:    66
Usage of /:   3.3% of 39.37GB    Users logged in: 0
Memory usage: 5%               IP address for eth0: 10.0.2.15
Swap usage:  0%

Graph this data and manage this system at:
https://landscape.canonical.com/

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

New release '14.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

This Ubuntu 12.04 LTS system is past its End of Life, and is no longer
receiving security updates. To protect the integrity of this system, it's
critical that you enable Extended Security Maintenance updates:
 * https://www.ubuntu.com/esm

Last login: Sun Apr  8 09:31:36 2018 from 10.0.2.2
vagrant@vagrant-ubuntu-precise-64:~$ |

```

(Note: I had to swap over to my Git Bash terminal since Windows does not allow for you to directly perform an ssh).

- To exit the vm, enter the command *exit*
- To shut down the vm, once you have exited the vm enter command *vagrant halt*. This will attempt to gracefully shut down the vm.

Using vagrant with xv6

- Now that you have successfully ssh into the vm, we now need to navigate to the directory with the vagrant file and the xv6 folder (I will refer to this directory as the root directory later on). Follow these commands:
 - `cd /vagrant` (Note: The `/` is necessary, as you can see from the screenshot, you will get an error without it.)

- cd xv6

```
vagrant@vagrant-ubuntu-precise-64:~$ cd vagrant
-bash: cd: vagrant: No such file or directory
vagrant@vagrant-ubuntu-precise-64:~$ cd /vagrant
vagrant@vagrant-ubuntu-precise-64:/vagrant$ ls
manifests  README.md  Vagrantfile  xv6
vagrant@vagrant-ubuntu-precise-64:/vagrant$ cd xv6/
vagrant@vagrant-ubuntu-precise-64:/vagrant/xv6$
```

- We are now inside the xv6 directory. To start up xv6 run the following commands:
 - make clean

```
vagrant@vagrant-ubuntu-precise-64:/vagrant/xv6$ make clean
rm -f *.tex *.dvi *.idx *.aux *.log *.ind *.ilg \
    *.o *.d *.asm *.sym vectors.S bootblock entryother \
    initcode initcode.out kernel xv6.img fs.img kernelmemfs mkfs \
    .gdbinit \
    _cat _echo _forktest _grep _init _kill _ln _ls _mkdir _rm _sh _stressfs
    _usertests _wc _zombie
vagrant@vagrant-ubuntu-precise-64:/vagrant/xv6$
```

- make

```
objdump -S _zombie > zombie.asm
objdump -t _zombie | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > zombie.sym
./mkfs fs.img README _cat _echo _forktest _grep _init _kill _ln _ls _mkdir _rm
_sh _stressfs _usertests _wc _zombie
used 29 (bit 1 ninode 26) free 29 log 30 total 1024
ballocc: first 468 blocks have been allocated
ballocc: write bitmap block at sector 28
dd if=/dev/zero of=xv6.img count=10000
10000+0 records in
10000+0 records out
5120000 bytes (5.1 MB) copied, 0.710337 s, 7.2 MB/s
dd if=bootblock of=xv6.img conv=notrunc
1+0 records in
1+0 records out
512 bytes (512 B) copied, 0.000917215 s, 558 kB/s
dd if=kernel of=xv6.img seek=1 conv=notrunc
295+1 records in
295+1 records out
151512 bytes (152 kB) copied, 0.0390436 s, 3.9 MB/s
vagrant@vagrant-ubuntu-precise-64:/vagrant/xv6$
```

- make qemu

```
vagrant@vagrant-ubuntu-precise-64:/vagrant/xv6$ make qemu
qemu-system-i386 -serial mon:stdio -hdb fs.img xv6.img -smp 1 -m 512 -display none
xv6...
cpu0: starting
init: starting sh
$
```

- Now that we are inside xv6, you can run the normal xv6 commands or any system calls that you may write yourself.

```
cpu0: starting
init: starting sh
$ ls
.          1 1 512
..         1 1 512
README    2 2 2021
cat        2 3 11435
echo       2 4 10836
forktest   2 5 6881
grep       2 6 12923
init       2 7 11229
kill       2 8 10836
ln         2 9 10790
ls         2 10 12633
mkdir      2 11 10897
rm         2 12 10882
sh         2 13 20461
stressfs   2 14 11400
usertests  2 15 46451
wc         2 16 11945
zombie     2 17 10582
console    3 18 0
$
```

(Note: To exit xv6 you can use the shortcut Ctrl A, X)

How To Do My Homework...

Now that we have the environment and xv6 set up and running, you can proceed to doing any homework that involved using xv6. You will make any changes/add new files to the local repository you git cloned at the start. Then to run these changes you will need to start up the vm using vagrant, ssh into it, change to the right directory once inside the vm then run the commands to compile and boot xv6. You may stay ssh'ed in vagrant when doing homework as any changes you make inside the folder will be reflected inside the vm. For example, add a txt file, named name.txt, to the root of the local repository in your normal laptop with your Name and Email. Then inside the vm, if you cd into the root directory and type ls you will see the newly made txt file. If you still do not believe they are one and the same, open the file in the vm with vim: *vim name.txt*