

XV6 Toolchain Installation Instructions

These are the toolchain installation instructions for all 3 OS Versions: Mac/Windows/Linux, use the appropriate one for you. We have tried to make the instructions available natively on every OS, however as Operating Systems get upgrades, things invariably break.

NOTES (Please read first)

- If you are running Mac OS X 10.11 (El Capitan) or above you might have trouble installing, because of some OS changes. If you are adventurous, you might try this [stackoverflow answer](#), otherwise **just use the Virtual Machine**.
- If you are running Windows I mentioned that you could use the Linux Subsystem for Windows (LSW), this only works if you have the 64 bit version of Windows 10, if you don't, **just use the Virtual Machine**. Also LSW is still beta so it will have some bugs.
- If you are using Mac/Windows and it's taking you a long time (more than 60 minutes) to get it sorted out you should strongly consider **just using the Virtual Machine** or come to office hours to see if we can help you. Depending on the problem you might still have to **just use the Virtual Machine**.
- After installing your compiler and all the tools, you will need to get the code for **XV6** please see the git instructions.

Virtual Machine Instructions

- Download VirtualBox from <https://www.virtualbox.org/> and install. Make sure to run the install program as an admin see this [video](#) if you need to understand what I mean.
Download the Virtual Machine image (CS3224.ova) from http://bit.ly/CS3224_VM
- Double click the .ova file. Click
Import and leave the defaults.
Once the VM is imported turn it on
and you are done.
- It might ask you for a password an it's **cs3224**
- Open the the VM and launch the command line. Everything is installed already.

Linux Instructions

- If you're using the Linux VM from above, everything should be installed already.
- If you are using another version of Ubuntu or a Debian-based version of Linux, installing the tools you'll need is very simple. At a terminal type:

```
$ sudo apt-get update && sudo apt-get install git nasm build-essential qemu gdb
```

Windows Instructions

As mentioned before this will only work if you have 64bit windows 10.

- Install the Linux Subsystem for Windows (LSW) following the instructions [here](#)
- When you are done installing open a "Bash on Ubuntu on Windows" terminal console and type the following:

```
$ sudo apt-get update && sudo apt-get install git nasm build-essential qemu gdb
```

- Download xv6 from github (instructions are on homework1), then once you have the code, make the modifications below.
- Modify the Makefile. Find the following line:

```
QEMUOPTS = -hdb fs.img xv6.img -smp $(CPUS) -m 512 $(QEMUEXTRA)
```

- Modify it to this:

```
QEMUOPTS = -hdb fs.img xv6.img -smp $(CPUS) -m 512 $(QEMUEXTRA) -display none
```

Mac OS X Instructions

Getting things setup on Mac OS X is a bit trickier. It's sufficiently tricky that Prof. Dolan-Gavitt created a shell script that should set things up for you.

Download the shell script: **install_cross.sh**. You may have to use File->Save Page As in order to download it; on Safari, make sure to save as "Page Source" rather than "Web Archive". Now, open up OS X's Terminal app (it's in the Utilities folder under Applications), and type the following to run it:

```
$bash install_cross.sh
```

Note: Do not type the \$ sign in the above command -- it's just there to indicate you're typing something in at the command line prompt. This will take a long time (roughly 30 minutes) and you will see lots of output. At the end you should see:

```
fi ; \
fi
make[11]: Nothing to be done for `install-data-am'.
make[1]: Nothing to be done for `install-target'.
Installation report:
✓ qemu-system-i386
✓ gcc-5
✓ nasm
✓ git
✓ i386-jos-elf-gcc
✓ i386-jos-elf-gdb
✓ i386-jos-elf-objdump
Looks like everything installed successfully, you are good to go.
Happy hacking!
```

Getting the XV6 Source Code

As a prerequisite, make sure that you have followed the **install instructions from NYU classes** to get your build environment set up.

A common theme of the homework assignments is that we'll start off with xv6, and then add something or modify it in some way. This assignment is no exception. Start by getting a copy of xv6 using `git` (commands typed at the terminal, and their output, will be shown using a monospace font; the commands type will be indicated by a `$`):

```
$ git clone https://github.com/gussand/xv6-public.git
Cloning into 'xv6-public'...
remote: Counting objects: 4475, done.
remote: Compressing objects: 100% (2679/2679), done.
remote: Total 4475 (delta 1792), reused 4475 (delta 1792), pack-reused 0
Receiving objects: 100% (4475/4475), 11.66 MiB | 954.00 KiB/s, done.
Resolving deltas: 100% (1792/1792), done.
Checking connectivity... done.
```

Make sure you can build and run xv6. To build the OS, use `cd` to change to the xv6 directory, and then run `make` to compile xv6:

```
$ cd xv6-public
$ make
```

Then, to run it inside of QEMU, you can do:

```
$ make qemu
```

QEMU should appear and show the xv6 command prompt, where you can run programs inside xv6. It will look something like:

```
QEMU
SeaBIOS (version rel-1.7.5-0-ge51488c-20140602_164612-nilsson.home.kraxel.org)

iPXE (http://ipxe.org) 00:03.0 C980 PCI2.10 PnP PMM+1FF93BB0+1FEF3BB0 C980

Booting from Hard Disk...

cpu0: starting xv6
cpu1: starting
cpu0: starting
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap star
t 58
init: starting sh
$ _
```

You can play around with running commands such as `ls`, `cat`, etc. by typing them into the QEMU window; for example, this is what it looks like when you run `ls` in `xv6`:

```
QEMU
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap star
t 58
init: starting sh
$ ls
.          1 1 512
..         1 1 512
README    2 2 1973
cat       2 3 13320
echo      2 4 12425
forktest  2 5 8153
grep      2 6 14988
init      2 7 13086
kill      2 8 12573
ln        2 9 12431
ls        2 10 14795
mkdir     2 11 12578
rm        2 12 12555
sh        2 13 23539
stressfs  2 14 13301
usertests 2 15 58588
wc        2 16 13838
zombie    2 17 12195
console   3 18 0
$ _
```

To exit out of `xv6`, type `Ctrl-A-X`.