

# Xv6

## Overview

Xv6 is an instructional OS consisting of a stripped down version of unix. To get ready to work within xv6, please start reading the [xv6 book](#) as well as the other xv6 resources listed below.

## Resources

- [xv6 book](#)
- [xv6 indexed/cross referenced code](#) (all 9299 lines of it :-)
- [Official website](#)

Sledge is school cluster. We can either work on sledge or use a VM for these labs.  
Connect to sledge: `ssh apati027@sledge.cs.ucr.edu`

## Getting Started (sledge)

In sledge, most software is pre-installed, so you only need to download xv6.

```
[sledge] $ cd ~
[sledge] $ git clone https://github.com/mit-pdos/xv6-public.git xv6
...
[sledge] $ cd xv6
[sledge] $ make
[sledge] $ echo "add-auto-load-safe-path $HOME/xv6/.gdbinit" > ~/.gdbinit
```

## Getting Started (VM) working on your own machine

Get familiar with [development environment](#). You can use course machine (sledge) or setup your own VM as follows.

### 1. Download and install Virtualbox/Vagrant/SSH client

- Download and install the latest version of virtualbox at <https://www.virtualbox.org/wiki/Downloads>
- Download and install the latest version of vagrant at <http://www.vagrantup.com/downloads.html>
- Windows: SSH client [PuTTY](#) or [Cygwin](#).

Note: Ubuntu users may want to use the following commands to install Virtualbox and Vagrant

```
[host] $ sudo apt-get install virtualbox
[host] $ sudo apt-get install vagrant
```

## 2. Add guest OS and run the VM

```
# download a 64-bit VM
[host] $ vagrant box add ubuntu/xenial64
==> box: Loading metadata for box 'ubuntu/xenial64'
      box: URL: https://atlas.hashicorp.com/ubuntu/xenial64
==> box: Adding box 'ubuntu/xenial64' (v20170331.0.0) for provider: virtualbox
      box: Downloading:
https://atlas.hashicorp.com/ubuntu/boxes/xenial64/versions/20170331.0.0/providers/virtualbox

==> box: Successfully added box 'ubuntu/xenial64' (v20170331.0.0) for 'virtualbox'!

# move to your working directory
[host] $ mkdir cs153
[host] $ cd cs153

# initialize the VM
[host] $ vagrant init ubuntu/xenial64
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.

# launch!
[host] $ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'ubuntu/xenial64'...
...

[host] $ vagrant ssh
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-71-generic x86_64)
...
ubuntu@ubuntu-xenial:~$
```

## 3. Once you have the VM up and running, let's initialize your VM for this course:

```
# in the VM, install toolchain
[vm] $ sudo apt-get update
[vm] $ sudo apt-get install -y build-essential gdb git gcc-multilib

# install the patched QEMU
#
[vm] $ cd ~
[vm] $ git clone http://web.mit.edu/ccutler/www/qemu.git -b 6.828-2.3.0
[vm] $ sudo apt-get install -y libstdl1.2-dev libtool-bin libglib2.0-dev libz-dev
libpixman-1-dev
[vm] $ cd qemu
[vm] $ ./configure --disable-kvm --target-list="i386-softhmmu x86_64-softhmmu"
[vm] $ make
[vm] $ sudo make install

# finally it's time for setting up xv6
#
[vm] $ cd ~
[vm] $ git clone https://github.com/mit-pdos/xv6-public.git xv6
...
[vm] $ cd xv6
[vm] $ make

# NOTE: allow local gdbinit to be loaded (only done once)
#
[vm] $ echo "add-auto-load-safe-path $HOME/xv6/.gdbinit" > ~/.gdbinit
```

## Debugging

Open two terminal windows (and enter the VM if not on sledge). Alternatively, you can also try [screen](#) (shortcut) or [byobu](#) (not on sledge).

```

# in window one
[vm|sledge] $ cd ~/xv6
[vm|sledge] $ make qemu-nox-gdb

# this starts up QEMU, but QEMU stops just before the processor
# executes the first instruction and waits for a debugging
# connection from GDB.

# in window two
#
# NOTE: -q removes the annoying init message
#
[vm|sledge] $ cd xv6
[vm|sledge] $ gdb -q
+ target remote localhost:26000
warning: A handler for the OS ABI "GNU/Linux" is not built into this configuration
of GDB. Attempting to continue with the default i8086 settings.

The target architecture is assumed to be i8086
[f000:fff0] 0xffff0: ljmp $0xf000,$0xe05b
0x0000fff0 in ?? ()
+ symbol-file kernel

# to terminate, for now, in window 2
[vm] $ killall qemu-system-i386

# for sledge
[sledge] kill $(pgrep qemu)

```

Read more on [how to use gdb with QEMU/JOS](#).

## Labs