

Due Thursday, January 24, 2019

To develop a better sense of how an operating system works, you will also do a few projects inside a real OS kernel. The kernel we'll be using is a port of the original Unix (version 6), and is runnable on modern x86 processors. It was developed at MIT and is a small and relatively understandable OS and thus an excellent focus for simple projects. In this project, we'll learn how to build a xv6 kernel.

FORMAT

You should work in a group of two students (or work individually). Every student needs to sign up for one of the project groups on canvas **no later than 1/20**.

Preparation: Linux Virtual Machine

The easiest way to build xv6 is to install a modern Linux distribution on your computer. With platform virtualization, Linux can cohabitate with your normal computing environment.

Download VirtualBox

Download VirtualBox to run Linux.

[VirtualBox download](#)

Follow the instructions at the website to install the VirtualBox.

Download Ubuntu Linux Virtual Machine Image

I have created a Linux Virtual Machine Image, [OSVM.ova](#), with xv6 and other related tools pre-installed. You can click the link and download it.

Open and configure VirtualBox

1. Open VirtualBox from "Applications"
2. Click "Important Appliance" under the File menu and select the OSVM appliance "OSVM.ova"
3. Linux virtual machine will start. You will be prompted for a username and password:
 - Default username: **osstudent**
 - Default password: **student2019**

Building and Running xv6

Inside the Linux VM, open a terminal, run the following commands:

```
cp -r xv6-public xv6
cd xv6
make
make qemu
```

An original source code copy of `xv6` is stored in the folder `xv6-public`. The first line in the above is to make a copy of the source code to the `xv6` folder. So, you can use the folder `xv6` for any modification while keeping a backup of the original codes under `xv6-public`. The last two lines are for building and running `xv6`, respectively.

Then you shall see something like this:

```
xv6...
cpu1: starting
cpu0: starting
init: starting sh
$
```

You can try the command `ls` in `xv6` as follows:

```
$ls
```

`xv6` doesn't provide a function to shutdown itself. You can just close the pop-up terminal to shutdown the `xv6`.

Adding a new user program

Add into `xv6` the first user program called `hello.c`, which is implemented as follows:

```
#include "types.h"
#include "user.h"

int
main(int argc, char *argv[])
{
    printf(1, "Hello, my name is %s %s\n", argv[1], argv[2]);
    exit();
}
```

Add into `Makefile` this user program under `UPROGS` as

```
UPROGS=\
```

```
_hello\  
_cat\  
...
```

Make sure to use tab space instead of blank space in front of `_hello`.

Then rerun the xv6 with the command, “make qemu”

Then you can test the new user program `hello` in the xv6 shell as follows:

```
$hello yourFirstName yourLastName
```

Resources

Even though in this project you are not required to write any new C code, it might be better for you to get familiar with it now. In next project, we will hack the xv6 and modify the C source codes. You can find C Programming Online Tutorial here:

<https://www.tutorialspoint.com/cprogramming>.

The online textbook also provides a very brief document to familiarize you with the basics of the C programming environment on UNIX systems: <http://pages.cs.wisc.edu/~remzi/OSTEP/lab-tutorial.pdf>

Report

Your report should include two screenshots showing that xv6 is running and `hello` is working (showing your own names) and a brief reflection on what you learned (> 100 words).

Please submit your report to Canvas under the “P1” assignment folder. Each group just submits one report.