实验环境搭建

MIT 6.S081课程官网

1.安装实验所需软件

1.VMware Workstation的安装

> - 对于该课程实验，需要一个虚拟机，可以使用`VMware Workstation`，也可以使用开源的`Virtual Box`。
> - 对于`VMware Workstation`，可以下载17 PRO版本，直接在官网下载即可。破解可以到知乎或者是B站，可以找到可用的激活码。
> - 安装过程比较简单，没有需要特别注意的点。
> - VMware下载地址

2.Ubuntu的安装

> - 在本实验环境中，使用Ubuntu 20.04这个版本，该版本可以省去安装、编译RISCSV工具链的过程。
> - Ubuntu下载地址（清华镜像下载网站）
> - 安装过程和虚拟机的安装没有太大的差别。

2.更换源

1.修改`/etc/apt/sources.list`文件中的源

- Ubuntu中的默认软件更新源是国外的节点，在国内下载速度慢，需要更换为国内的源，安装和更新软件的速度更快。
- 打开sources.list文件

  ```
  sudo gedit /etc/apt/sources.list
  ```

- 编辑文件，在文件最前面添加阿里云镜像源：

  ```
  #中科大源
  deb https://mirrors.ustc.edu.cn/ubuntu/ focal main restricted universe multiverse
  deb-src https://mirrors.ustc.edu.cn/ubuntu/ focal main restricted universe multiverse
  deb https://mirrors.ustc.edu.cn/ubuntu/ focal-updates main restricted universe multiverse
  ```

```
deb-src https://mirrors.ustc.edu.cn/ubuntu/ focal-updates main restricted universe multiverse
deb https://mirrors.ustc.edu.cn/ubuntu/ focal-backports main restricted universe multiverse
deb-src https://mirrors.ustc.edu.cn/ubuntu/ focal-backports main restricted universe multiverse
deb https://mirrors.ustc.edu.cn/ubuntu/ focal-security main restricted universe multiverse
deb-src https://mirrors.ustc.edu.cn/ubuntu/ focal-security main restricted universe multiverse
deb https://mirrors.ustc.edu.cn/ubuntu/ focal-proposed main restricted universe multiverse
deb-src https://mirrors.ustc.edu.cn/ubuntu/ focal-proposed main restricted universe multiverse

#添加阿里源
deb http://mirrors.aliyun.com/ubuntu/ focal main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ focal main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu/ focal-security main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ focal-security main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu/ focal-updates main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ focal-updates main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu/ focal-proposed main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ focal-proposed main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu/ focal-backports main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ focal-backports main restricted universe multiverse

#添加清华源
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-updates main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-updates main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-backports main restricted universe multiverse
```

```
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-backports main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-security main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-security main restricted universe multiverse multiverse
```

- 刷新列表

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install build-essential
```

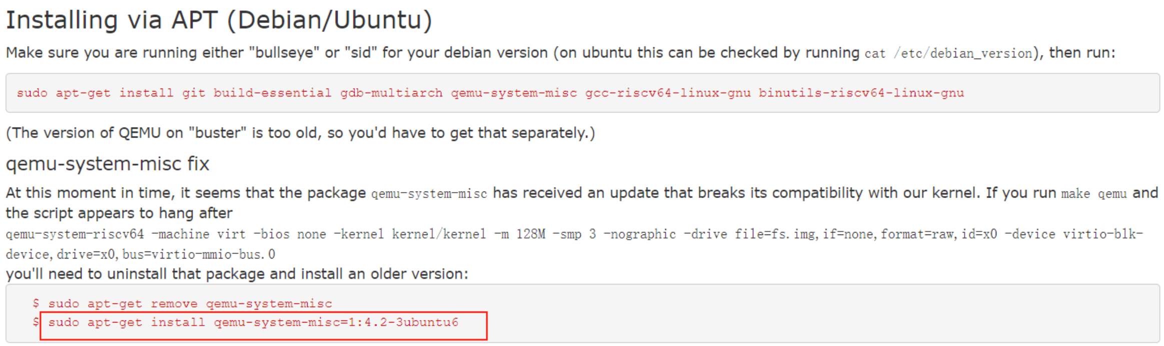
3.安装SSH

- 默认情况下，首次安装Ubuntu时，不允许通过SSH进行远程访问。
- 在Ubuntu上启用SSH非常简单。以root 用户或具有sudo特权的用户执行以下步骤：

> - 打开终端并安装`openssh-server`软件包：
>
> ```
> sudo apt update
> sudo apt install openssh-serve
> ```
>
> - 安装完成后，SSH服务将自动启动。输入下列命令验证SSH是否正在运行：
>
> ```
> sudo systemctl status ssh
> ```
>
> - 输出应为：
>
> ```
> ● ssh.service - OpenBSD Secure Shell server
>      Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
>      Active: active (running) since Sun 2021-08-15 07:13:19 PDT; 23s ago
>        Docs: man:sshd(8)
>              man:sshd_config(5)
>    Main PID: 46470 (sshd)
>       Tasks: 1 (limit: 2275)
>      Memory: 1.3M
>      CGroup: /system.slice/ssh.service
>              └─46470 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
> ```
>
> - 按`q`即可返回至命令行。
> - 若启用了防火墙，使用如下命令打开SSH端口：

```
sudo ufw allow ssh
```

4.安装RISC-V交叉编译工具



Installing via APT (Debian/Ubuntu)

Make sure you are running either "bullseye" or "sid" for your debian version (on ubuntu this can be checked by running `cat /etc/debian_version`), then run:

```
sudo apt-get install git build-essential gdb-multiarch qemu-system-misc gcc-riscv64-linux-gnu binutils-riscv64-linux-gnu
```

(The version of QEMU on "buster" is too old, so you'd have to get that separately.)

qemu-system-misc fix

At this moment in time, it seems that the package `qemu-system-misc` has received an update that breaks its compatibility with our kernel. If you run `make qemu` and the script appears to hang after `qemu-system-riscv64 -machine virt -bios none -kernel kernel/kernel -m 128M -smp 3 -nographic -drive file=fs.img,if=none,format=raw,id=x0 -device virtio-blk-device,drive=x0,bus=virtio-mmio-bus.0` you'll need to uninstall that package and install an older version:

```
$ sudo apt-get remove qemu-system-misc
$ sudo apt-get install qemu-system-misc=1:4.2-3ubuntu6
```

```
sudo apt install git build-essential gdb-multiarch qemu-system-misc gcc-riscv64-linux-gnu binutils-riscv64-linux-gnu libglib2.0-dev libpixman-1-dev gcc-riscv64-unknown-elf
```

5.安装QEMU

1.安装QEMU

QEMU用于在我们机器上(X86)模拟RISC-V架构的CPU，编译生成的risc-v平台的机器码，需要通过模拟cpu执行。

```
# 下载qemu
wget https://download.qemu.org/qemu-5.1.0.tar.xz
# 对下载的文件进行解压
tar xvf qemu-5.1.0.tar.xz
cd qemu-5.1.0
# 编译
./configure --disable-kvm --disable-werror --prefix=/usr/local --target-list=riscv64-softmmu
make
sudo make install
```

- 在下载`qemu-5.1.0`这步时，可能会导致下载速度十分慢，可以在搜索引擎中直接搜索下载，将其复制到虚拟机中即可。

6.测试

1.下载xv6源码

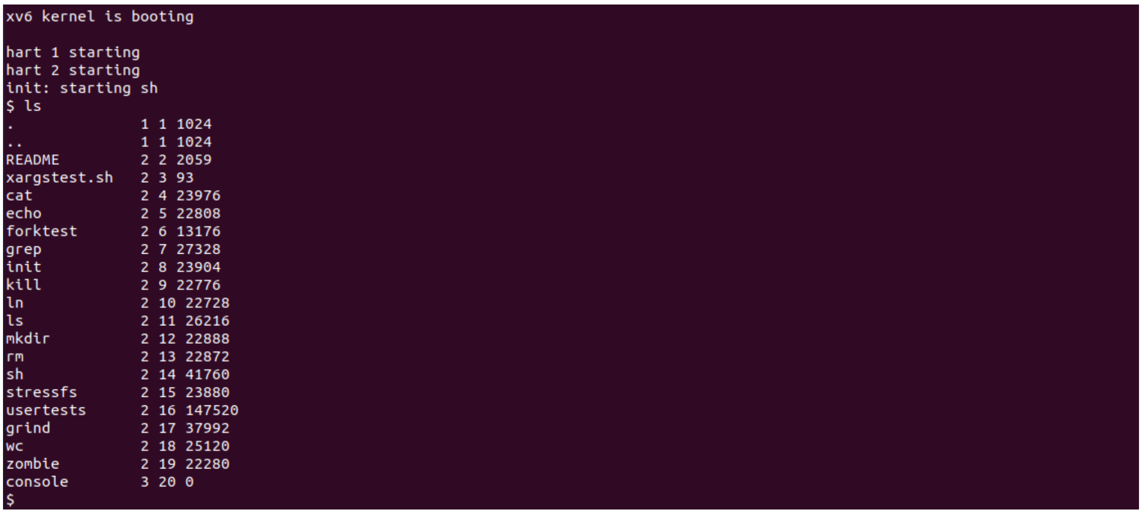
- 从github中下载xv6的源码，切入源码的主目录，**将分支切换到util**

```
git clone git://g.csail.mit.edu/xv6-labs-2020
cd xv6-labs-2020
git checkout util.
# 拉取特定分支到本地
git clone -b pgtbl  git://g.csail.mit.edu/xv6-labs-2020
```

- 在项目目录下编译，使用如下命令：

```
make
make qemu
```

- 输出如下则说明环境搭建成功：



```
xv6 kernel is booting

hart 1 starting
hart 2 starting
init: starting sh
$ ls
.              1 1 1024
..             1 1 1024
README         2 2 2059
xargstest.sh   2 3 93
cat            2 4 23976
echo           2 5 22808
forktest       2 6 13176
grep           2 7 27328
init           2 8 23904
kill           2 9 22776
ln             2 10 22728
ls             2 11 26216
mkdir          2 12 22888
rm             2 13 22872
sh             2 14 41760
stressfs       2 15 23880
usertests      2 16 147520
grind          2 17 37992
wc             2 18 25120
zombie         2 19 22280
console        3 20 0
$
```