

2023秋操作系统 实验部分环境配置流程

1、安装VLAB

建议大家使用学校的VLAB虚拟机平台，登录网站 <https://vlab.ustc.edu.cn/> 选择**虚拟机管理**，利用统一身份认证系统就可以登录。选择新建一个虚拟机，**镜像选择Ubuntu-20.04版本**，如下图：

创建虚拟机 - Vlab VM

← → ↻ vlab.ustc.edu.cn/vm/create

Vlab

虚拟机名称*

OS

镜像选择* [我该选择哪个镜像?](#)

vlab02-ubuntu-desktop-mate-20.04.tar.gz

☐ 我已阅读并接受[服务条款](#)和[使用限制](#)*

创建

2、获取实验环境

实验所需的文件通过克隆下述地址可以得到，在终端中输入：

```
git clone git://g.csail.mit.edu/xv6-labs-2020
```

下面进入所下载的文件夹，并且切换到每次实验所需的分支，例如第一次实验为util实验，则需要在终端中输入：

```
cd xv6-labs-2020  
git checkout util
```

每次依据实验名称xx的不同，切换到不同的分支：**git checkout xx**

3、qemu以及打分

本次实验所采用的系统是xv6操作系统，我们需要一个可以运行xv6操作系统的模拟器，也就是qemu模拟器。因此我们需要学会如何构建以及启动qemu模拟器，并且在其上面进行一系列操作。

首先我们需要安装qemu系统，本次实验只需要misc版即可完成全部实验，在终端输入下面的指令来安装qemu系统：

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

接下来需要安装qemu中编译文件的gcc编译器，在终端输入：

```
sudo apt install gcc-riscv64-unknown-elf
```

接下来启动qemu模拟器，在终端输入：

```
sudo make qemu
```

进入qemu模拟器后，输入ls可以查看目前已有的命令如下：

```
ubuntu@VM7186-05:/home/ubuntu/桌面/xv6-labs-2020$ sudo make qemu
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

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
sleep      2 20 22760
console    3 21 0
MQQ8sysz0  2 22 0
x5oMkLU6   1 23 64
NmfaKLGK   1 27 48
a          1 29 48
```

同时输入Ctrl+P可以打印当前进程的信息

```
$
1 sleep  init
2 sleep  sh
```

若需要退出qemu模拟器：同时输入Ctrl+A，松手后按X键即可。

QEMU: Terminated

每次实验的最终完成依据是需要进行一个打分操作，大家需要提交打分操作后的界面截图，分数达标即可完成每次实验。只需要在终端执行文件夹中已经写好的打分脚本即可，例如第一次util实验中的打分脚本为 grade-lab-util (**不是在qemu模拟器中!**)则需要输入 python3 grade-lab-util 即可进行打分：

```

ubuntu@VM7186-OS:/home/ubuntu/桌面/xv6-labs-2020$ sudo python3 grade-lab-util
make: "kernel/kernel"已是最新。
== Test sleep, no arguments == sleep, no arguments: OK (7.5s)
== Test sleep, returns == sleep, returns: OK (2.4s)
== Test sleep, makes syscall == sleep, makes syscall: OK (2.6s)
== Test pingpong == pingpong: FAIL (2.0s)
...
    $ pingpong
    exec pingpong failed
    $ echo OK
GOOD OK
    $ qemu-system-riscv64: terminating on signal 15 from pid 133998 (make)
MISSING '^d+: received ping$'
MISSING '^d+: received pong$'
QEMU output saved to xv6.out.pingpong
== Test primes == primes: FAIL (2.2s)
...
    $ primes
    exec primes failed
    $ echo OK
GOOD OK
    $ qemu-system-riscv64: terminating on signal 15 from pid 134030 (make)
MISSING 'prime 2'
MISSING 'prime 3'
MISSING 'prime 5'
MISSING 'prime 7'
MISSING 'prime 11'
MISSING 'prime 13'
MISSING 'prime 17'
MISSING 'prime 19'
MISSING 'prime 23'
MISSING 'prime 29'
MISSING 'prime 31'
QEMU output saved to xv6.out.primes
== Test find. in current directory == find. in current directory: FAIL (2.2s)

```

若需要对一次实验中的单个任务打分，例如第一次实验中的sleep任务，可以使用 `python3 grade-lab-util sleep`

```

ubuntu@VM7186-OS:/home/ubuntu/桌面/xv6-labs-2020$ sudo python3 grade-lab-util sleep
make: "kernel/kernel"已是最新。
== Test sleep, no arguments == sleep, no arguments: OK (3.3s)
== Test sleep, returns == sleep, returns: OK (2.1s)
== Test sleep, makes syscall == sleep, makes syscall: OK (2.2s)

```