

# 实验 - Linux 实验环境初探

## 实验简介

在第一节课的收尾部分，我们简单学习了一下 Linux 的一些 shell 命令，以及基础文件权限和访问控制，此次实验即简单地回顾一下内容，并且对课堂上遗留的几个点做拓展。

在此次实验完成之后，你或将拥有自己的首个 Linux 环境，做好环境备份，可能这个虚拟机会陪伴你一长时间 :)

## 基础部分

### 1. 安装 Linux 虚拟机 (50 points)

注: 如果你已经有了其他更称手的 Linux 环境，如 WSL 等，也请完成这一步骤进行拿分，但后续的 3、4 操作可以在自己熟悉的环境下进行

#### 下载虚拟机管理器

要使用虚拟机，首先需要下载一个虚拟机管理程序，这里我们推荐

- [Virtual Box](#): 开源正版，windows 和 macOS 都可以使用
- [VMware Workstation Player](#): windows 下的 vmware 社区版，注册 account 下载
- [VMware Fusion](#): macOS 下的 vmware 社区版，注册 account 下载

其他的如 Parallels Desktop 的也可以按需使用

#### 下载 Linux 发行版镜像

发行版 (distribution) 可以理解成为使用者已经预装了 Linux 内核以及各种方便使用软件的整体系统，如库代码、包管理器、GUI 工具、以及一些办公套件等，发行版为我们使用 Linux 环境减少了很多工作，常见的有商业发行版 Ubuntu, Red Hat, SUSE Linux 以及社区发行版 Debian, Fedora, Arch，以及 hacker prefers 的 Kali Linux 等

实验中你可以下载自己期待使用的发行版镜像，考虑到更新速度、受众比例等，手册中的安装以 Ubuntu 为例。

可以前去发行版的官网或者国内的一些源镜像下载 ISO 文件，这里以[浙大源](#)为例子

#### Ubuntu Releases

```
ubuntu-14.04.6-desktop-amd64.iso
ubuntu-14.04.6-desktop-i386.iso
ubuntu-14.04.6-server-amd64.iso
ubuntu-14.04.6-server-i386.iso
ubuntu-16.04.6-desktop-i386.iso
ubuntu-16.04.6-server-i386.iso
ubuntu-16.04.7-desktop-amd64.iso
ubuntu-16.04.7-server-amd64.iso
ubuntu-18.04.6-desktop-amd64.iso
ubuntu-18.04.6-live-server-amd64.iso
ubuntu-21.10-desktop-amd64.iso
```

```
ubuntu-21.10-live-server-amd64.iso
ubuntu-18.04.6-desktop-amd64.iso
ubuntu-18.04.6-live-server-amd64.iso
ubuntu-21.10-desktop-amd64.iso
ubuntu-21.10-live-server-amd64.iso
ubuntu-14.04.6-desktop-amd64.iso
ubuntu-14.04.6-desktop-i386.iso
ubuntu-14.04.6-server-amd64.iso
ubuntu-14.04.6-server-i386.iso
ubuntu-16.04.6-desktop-i386.iso
ubuntu-16.04.6-server-i386.iso
ubuntu-16.04.7-desktop-amd64.iso
ubuntu-16.04.7-server-amd64.iso
ubuntu-20.04.4-desktop-amd64.iso
ubuntu-20.04.4-live-server-amd64.iso
ubuntu-22.04-desktop-amd64.iso
ubuntu-22.04-live-server-amd64.iso
ubuntu-20.04.4-desktop-amd64.iso
ubuntu-20.04.4-live-server-amd64.iso
ubuntu-22.04-desktop-amd64.iso
ubuntu-22.04-live-server-amd64.iso
```

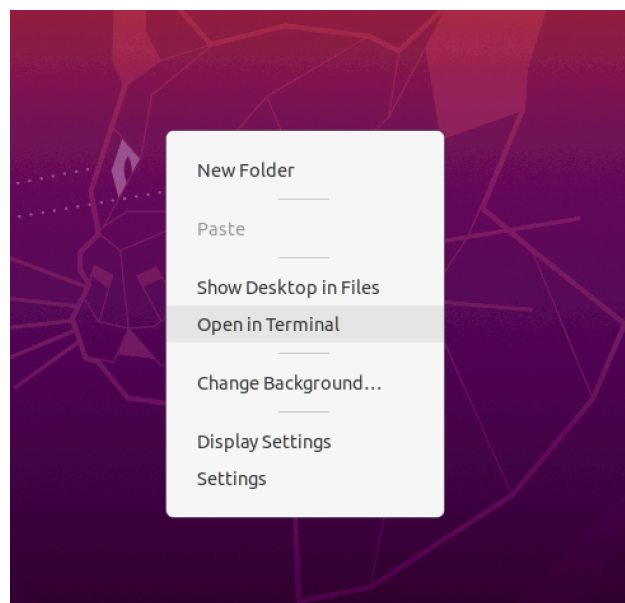
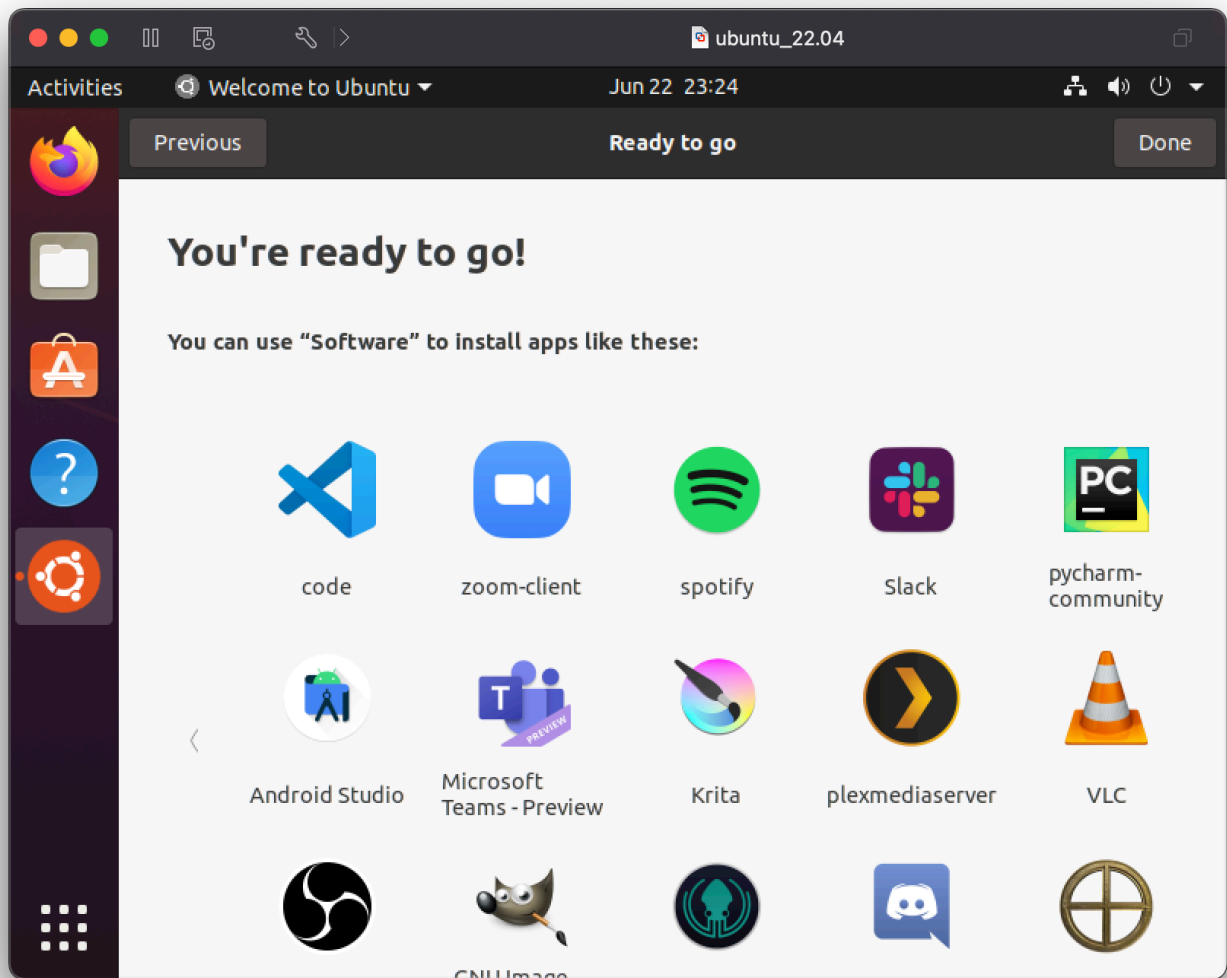
可以看到大量选项，其中命名是 `ubuntu-<发行版版本号>-<发行版类别>-<架构>.iso`，推荐使用 `20.04` 版本的桌面端，如 `ubuntu-20.04.4-desktop-amd64.iso`

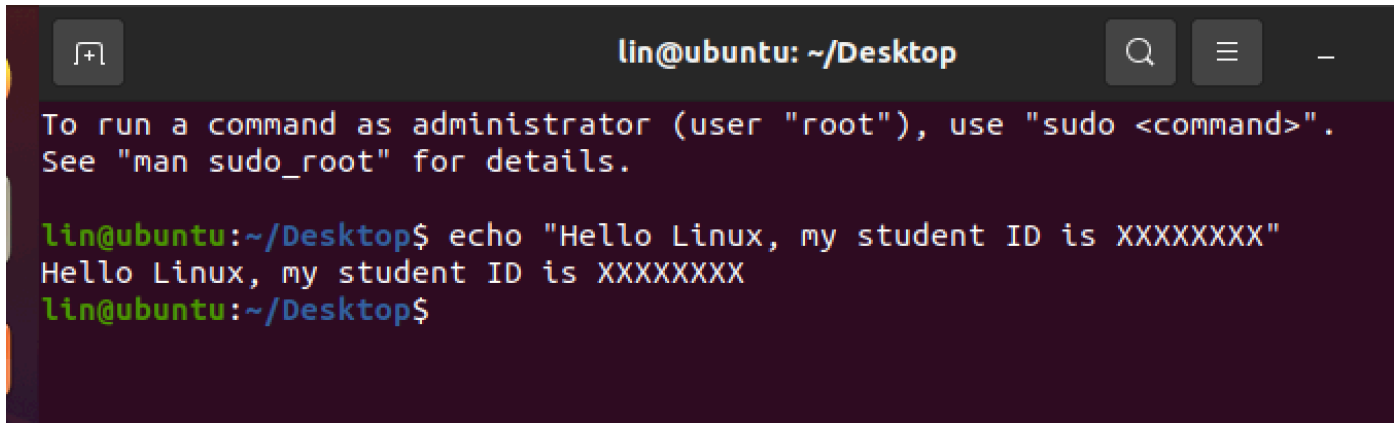
下载好的 ISO 丢给虚拟机管理软件往往就可以开始安装了，这里考察一下「独立」能力，请自行根据下载的虚拟机管理软件和刚下载的 ISO 完成虚拟机的安装，配置和启动

P.S. 建议硬盘大小40G以上，不然之后的工具安装多了要resize反而麻烦

另外安装过程还是挺久的，保持网络状态良好以及电源充足，可以一边安装的时候一边看课上的PPT拿来复习复习

安装过程结束后，你应该可以看到这样的桌面端，在递交的实验报告中请截图 `open terminal` 后的图片，并在 shell 上完成自己学号的 `echo`





```
lin@ubuntu: ~/Desktop

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

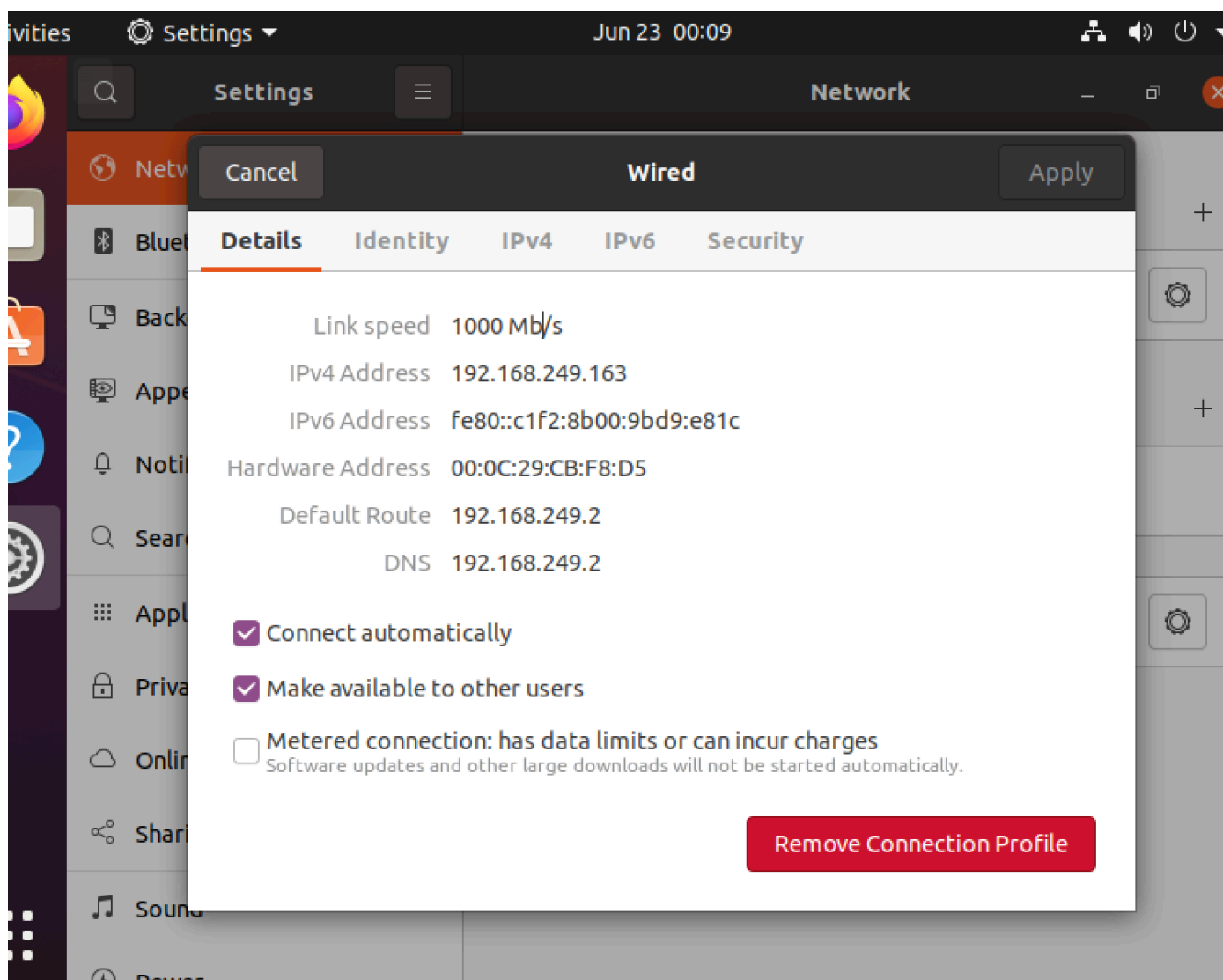
lin@ubuntu:~/Desktop$ echo "Hello Linux, my student ID is XXXXXXXXX"
Hello Linux, my student ID is XXXXXXXXX
lin@ubuntu:~/Desktop$
```

## 2. 通过 ssh 连接虚拟机 shell (10 points)

当然，Linux 目前作为虚拟机（Guest），虽然现在的虚拟机管理软件做的越来越方便，如共享文件夹、共享粘贴板等功能做的越来越齐全，但是在使用体验上，Guest还是没有宿主机（Host），即目前正在使用的操作系统那样方便的。尤其是到了日后的开发和科研，Linux 环境往往是以在服务器上的形式呈现给使用者。因此，这一小节我们将通过最常用的远程交互方式 —— SSH 来完成从宿主机到虚拟机的操作，

- 如果使用 windows，可以下载ssh客户端 putty

首先，我们需要保证宿主机能访问到虚拟机的网络，我们先查看虚拟机的网络地址，打开 settings 选择到 network，点击小齿轮，可以看到如下图的 connection profile（例子中虚拟机使用的是 NAT 网络模式）



如果宿主机可以通过 ping 命令访问到虚拟机地址，则说明网络可通

```
~ ping (ping)
{} ~ ping 192.168.249.163
PING 192.168.249.163 (192.168.249.163): 56 data bytes
64 bytes from 192.168.249.163: icmp_seq=0 ttl=64 time=0.936 ms
64 bytes from 192.168.249.163: icmp_seq=1 ttl=64 time=0.412 ms
64 bytes from 192.168.249.163: icmp_seq=2 ttl=64 time=1.047 ms
64 bytes from 192.168.249.163: icmp_seq=3 ttl=64 time=1.192 ms
64 bytes from 192.168.249.163: icmp_seq=4 ttl=64 time=1.123 ms
64 bytes from 192.168.249.163: icmp_seq=5 ttl=64 time=0.979 ms
```

在确定网络可通之后，前往虚拟机的命令行安装 ssh 服务

```
sudo apt install -y ssh
```

这里的 `sudo` 是以 root 权限来进行执行，`apt` 或者 `apt-get` 是包管理器的命令程序，这里使用 `install` command，`-y` 表示接受后续的条款，安装对象是 `ssh`

成功安装之后，就可以开始通过如下命令打开 ssh 服务

```
sudo service ssh start
```

当打开之后，宿主机就可以通过 ssh 客户端程序连接到宿主机上了

```
$ ssh lin@192.168.249.163
lin@192.168.249.163's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.13.0-51-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

171 updates can be applied immediately.
115 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

lin@ubuntu:~$
```

注意，要把这里的 `lin` 替换成你虚拟机中的用户名，将 `192.168.249.163` 替换成你虚拟机中的 IPv4 地址，并在过程中输入你预设的用户密码才可以成功连上。

如果在这一段操作中遇到困难请及时联系助教，当然，我们鼓励通过自行搜索的方式解决问题)

请在作业报告中给出成功连接到ssh的客户端程序窗口

你可以选择在虚拟机的图形界面，或者通过ssh工具连上虚拟机完成后续的操作

### 3. Linux 基础命令 (20 points)

我们在这里执行课堂上讨论的，以及更多的几个命令，请在作业报告中截图每个命令完成后的显示结果

1. `cd` 前往 home 目录

```
cd
```

`cd` 可以接受一个绝对路径或者相对路径，当不给参数或者给特殊参数 `~` 时，会前往默认的用户 home 目录

2. `mkdir` 创建一个工作文件夹 `lab1`

```
mkdir lab1
```

3. 通过 `cd` 进入工作目录
4. 通过 `pwd` 查看所在目录路径
5. 通过 `touch` 创建几个空文件

```
touch empty1 empty2 empty3
```

6. 通过 `mkdir` 创建几个空文件夹
7. 通过 `ls` 命令查看创建出来的文件和文件夹
8. 通过如下命令输出内容到文件 `info.txt` 中

```
echo "What a nice day" > info.txt
```

9. 通过 `cat` 命令查看 `info.txt` 文件内容

```
cat info.txt
```

10. 通过 `mv` 命令将 `info.txt` 更名为 `information.txt`

```
mv info.txt information.txt
```

所有步骤完成即可拿满20分 :) 每个步骤的结果 2 分

## 4. Linux 文件访问控制 (20 points)

课堂结尾时候提到了 `chmod` 命令和 `chown` / `chgrp` 命令，虽然课堂上没展开，但是其使用是非常简单的

1. 通过 `man` 查看 `chmod` 和 `chown` 以及 `chgrp` 的手册，并在报告中阐述命令的含义

```
man chmod
man chown
man chgrp
```

对了，BTW，man命令是最好的老师

2. 通过如下命令得到一个简单的脚本程序 `hello.sh`

```
echo -e "#! /bin/sh\n\nnecho 'this is a shell script program'\n" > hello.sh
```

通过 `cat` 命令确定文件内容

```
cat hello.sh
```

3. 尝试执行 `hello.sh`

```
./hello.sh
```

这里其实相当于直接将一个相对路径作为参数给了目前正在运行的 shell，因此，传绝对路径也是可以的  
此时理应得到类似如下的结果

```
-bash: ./hello.sh: Permission denied
```

4. 通过 `ls -l` 查看文件的 permissions

```
ls -l hello.sh
```

请在报告中解释 `ls -l` 的输出，并分析为什么前面执行会报 `Permission denied` 错误

5. 通过 `chmod` 命令为 `hello.sh` 添加可执行权限

```
chmod +x hello.sh
or
chmod 775 hello.sh
```

添加完成后请执行该脚本

6. 通过 `chown` 更改文件的 owner user 与 group



首先，我们添加一个新用户 `bob`

```
sudo adduser bob
```

注意，只有超级用户才能添加用户或者用户组，添加过程中会有一些琐碎的信息可以快速通过（但是请牢记 bob 的密码

然后我们使用如下命令更改所有权

```
sudo chown bob:bob hello.sh
```

更改完后请使用 `ls -l` 查看结果

并且说明，当前用户对于文件 `hello.sh` 拥有哪些权限

7. 请通过 `chmod` 和 `chown` 命令的组合将 `hello.sh` 的权限变得满足如下要求（请新建用户 `ctf`）

- 文件拥有者是 `bob`
- 文件所属的组是 `ctf`
- 文件 `bob` 可读写但是不可执行，文件 `ctf` 组内用户可读，但不可写和执行，其他外的用户不能对文件进行读、写、或者执行

请在报告中给出过程，以及通过 `ls -l` 展现最后的结果

## 挑战部分

### 1. shell编程

shell当然不止简单几个命令，其本身作为脚本语言就可以完成复杂的逻辑。请通过网上搜索sh/bash脚本的格式，然后编写一个循环累加从0到100的目标脚本程序（当然，欢迎更复杂的程序

### 2. setuid

文件权限除了用户、组、其他；以及读写和执行外，有一个特殊的，名为 `setuid` 的情形。请以命令 `passwd` 为对象，阐述 `setuid` 的含义，以及使用场景

## 拓展问题和阅读

可以思考，在基础题 4.3 中，如果通过路径直接执行 `hello.sh` 会报错权限问题，但如果此时使用 `sh hello.sh` 或者 `bash hello.sh` 则可以成功完成脚本的执行。诶，此时 `hello.sh` 不是不具备执行权限么，那这里是否有矛盾呢？

可以继续阅读

[An introduction to Linux Access Control Lists \(ACLs\)](#)

如果汇编需要快速入门建议

1. 小白的课程资料 (<http://10.71.45.100/bhh/>)
2. CTF wiki (<https://ctf-wiki.org/assembly/x86-x64/readme/>)

推荐的额外资源

- 校巴 [zjusec.com](http://zjusec.com)