Linux 系统

本学期的实验需要在 Linux 系统上进行，在后续的实验中，我们统一使用 Ubuntu 20.04 发行版。如果

没有适合的环境，则需要使用虚拟机安装 Linux 系统。

下面介绍两种虚拟机方案。

VMware Workstation

VMWare 在 Windows/MacOS 系统上的虚拟化软件 Workstation Pro 以及 Fusion Pro 于2024年5

月13日起可供个人免费使用。

这两个虚拟化软件的图形界面设计比较清晰，我们只需要安装对应的软件，以及需要的Linux发行版镜像即可：

- Vmware: https://www.vmware.com/

  学校的正版下载入口：http://mvls.fudan.edu.cn/

- 发行版镜像：
  - https://mirrors.ustc.edu.cn/ubuntu-releases/

  VMWare Desktop Hypervisor 系列使用的系统镜像可以从镜像站下载，如上面科大镜像站的Ubuntu镜像链接。我们将使用的 Ubuntu 20.04 LTS 发行版，从上面的链接进去的话，可以在 `20.04/ubuntu-20.04.6-desktop-amd64.iso` 找到对应的镜像文件。

WSL 2

WSL(Windows Subsystem for Linux)是 Microsoft 推出的虚拟机方案，仅支持 Windows 系统，使用体验极佳。对于 Windows10 2004以上版本或 Windows 11系统，可以参考 WSL 官方文档（https://learn.microsoft.com/zh-cn/windows/wsl/install）。

- 优点：方便，且占用资源比虚拟机少
- 缺点：可能有一些文件互操作的问题
- 视频参考：https://www.youtube.com/watch?v=qYlgUDKKK5A&t=79s

Linux Shell

使用命令行与Linux系统进行交互对于程序员来说是十分有必要的，毕竟在很多情况下，图形界面对于这些系统来说是一个相当不必要的东西。

> 书籍推荐：Beginning Linux Programming, 4 Ed

但不用真的看大部头的书，通常只要掌握几个常见的指令和一些简单的知识就足够了：

- 文件系统是一个树状结构，根目录用/表示，用户目录用~表示。在类Unix系统中，你需要使用指令去操作文件。
- 一些常见的操作命令：`ls,pwd,cd,mkdir,rm,mv,cp,touch,cat`。具体的细节直接百度或者google搜索或者去问GPT老师或者去问助教。也可以直接在终端使用 `man [指令]` 查看。

- 有时候你会遇到权限的问题，你需要在命令前面加上 `sudo` ，然后输入用户密码，获得根用户的权限。谨慎使用，避免危险操作。
- 善用一些指令可以事半功倍，比如 `find,locate,grep` 查找文件和代码或者用diff简单高效比较两个文本是否一致。

> 课程推荐：MIT Missing Semester 前面两节 https://missing.csail.mit.edu/
>
> 不需要去非常详尽的掌握各种细枝末节，尽管了解他们可能对于理解Linux系统有一定的帮助，但是大多数时刻一些基本的命令对于我们来说是完全足够了的。
>
> 用过的一些比较好用的东西：
>
> - tmux（终端多路复用）同一个终端可以打开多个窗口，不必每个任务都开一个如果要使用服务器的话，进程会在在后台运行，不会因为关机/网卡导致程序中断
> - alias（别名）linux用户在.bashrc，macos用户在.zshrc可以设置。比如我不希望我每次都输入命令ls -alh（这个命令相当于ls加上all,list,humanreadble这些参数）我可以alias ll="ls -alh"。以后就只要输入ll。

包管理器

以 Ubuntu 系统的 apt 包管理器为例，其它发行版可以自行查找有关资料。

包管理器全称是软件包管理器，顾名思义是用来管理软件包的软件。在大家熟悉的 Windows 系统中，通常下载软件就是去软件的官网上下载。而在 Linux 系统中，最常见的安装软件的方式是使用软件包管理器从“软件仓库”中下载。包管理器会负责一个软件的

全生命周期，包括下载、安装、依赖关系、卸载、更新等等。

Ubuntu 发行版中带有 apt 和 dpkg 包管理器，我们一般使用 apt，基本用法可以参考 Ubuntu包管理器文档中的 apt 一节。完整的官方文档可以运行 man apt 查阅。

在后续课程中，如果遇到命令行提示说 xxx not found ，可以尝试使用 apt 安装相应的软件包，

```
如 sudo apt install xxx 。。
```

Tips:

apt 默认的软件源服务器在国外，可能被 the Great Fire Wall 直接拦下。建议将其更换为科大镜像或者清华镜像等

Ubuntu 的更换方法如下:

```
sudo cp /etc/apt/sources.list /etc/apt/sources.list.bak

sudo sed -i 's@//.*archive.ubuntu.com@//mirrors.ustc.edu.cn@g'

/etc/apt/sources.list

sudo apt update
```

其中第一行是将原来的文件进行备份，这只是一个好习惯而已。

上面几行的详细说明可以参考这个链接（https://mirrors.ustc.edu.cn/help/ubuntu.html），如果你使用其他发行版，也可以去这个链接中寻找相关说明。如需要在命令行下使用代理，可以使用环境变量，或者可以了解一下 proxychains 这个工具。

一些常用软件

Vim

Vim与其说是一个软件，更像是一种编辑文本的模式，它可以使我们编写代码的过程变得很cool...或者很快？除去这些原因，学习Vim的基本编辑方式，对于我们在linux系统下操作文本还是挺重要的，至少应该知道怎么编辑和保存，使用s/i/o进入编辑模式，使用esc+q!/wq退出。。

vim不需要特别多的教程：直接在终端输入vimtutor然后通关就够了。

vim的插件非常强大，.vimrc里面可以设置非常多的东西。

视频参考：https://www.youtube.com/watch?v=jXud3JybsG4&t=1788s

VS Code

强大的Editor：轻量快速 + 支持很多插件

需要配置一下环境：具体根据自己 os ，以及编程所使用的语言，参考网络上的教程。

如果你使用的是 VMWare，可以将 VS Code 安装在虚拟机里，也可以通过本机上的 VSCode SSH 连接到虚拟机中进行开发。 如果你使用的是 WSL 2，直接将 VSCode 装在本机上即可使用本机上的VS Code编辑和运行虚拟机中的代码。

WSL2 常用用法：`code some.txt` 即可用 Windows 上的 VSCode 打开文件。`code .` 即可用 Windows 上的 VSCode 打开当前文件夹。

VS Code 会提示你安装常用插件，大家也可以自行在网上寻找好用的插件。关于 VS Code 安装与配置的问题，可参考 VS Code官方文档（https://code.visualstudio.com/docs）。

C/C++编译工具链

GCC

官方网站：https://gcc.gnu.org/Check

- the installation of GCC `gcc --version`
- How to install: https://gcc.gnu.org/install/

使用GCC编译一个C语言程序：

- `gcc <filename> -o <name_of_executable>`

对于更多的使用，可以参考 CSAPP: chapter 5 and chapter 7

调试工具：GDB

官方网站：https://www.gnu.org/software/gdb/

如何安装（Ubuntu/Debian）：`sudo apt install gdb`。在终端输入 `gdb --version` 可以查看 GDB 是否已经安装。

使用 GDB 调试程序：
GDB 是 GCC 配套的调试工具，可以帮助开发者单步执行代码、查看变量值、设置断点等。调试一个 C/C++ 程序的步骤如下：

1. 首先，使用 `-g` 选项编译程序以生成调试信息：`gcc -g <filename>.c -o <name_of_executable>`
2. 使用 GDB 启动调试器：`gdb <name_of_executable>`
3. 一些常用的命令：
    - `break <line_number>`：设置断点（例如，`break 10` 在第 10 行设置断点）。
    - `run`：启动程序。
    - `next` / `step`：执行下一行代码，`step` 会进入函数内部。
    - `print <variable>`：打印变量的值。
    - `quit`：退出调试。