

Linux使用培训课程大纲

介绍

通过本课程，学员将学习基本的Linux操作和使用技巧，掌握在RoboMaster比赛视觉组工作中所需的Linux基础知识，能够有效地使用Linux系统进行开发和调试。

课程时间

- 课程时间：现在这一节课

课程内容

一：安装Linux系统

下载系统安装光盘文件，制作启动盘

- 下载系统光盘文件：[Ubuntu 22.04](#)
- 下载U盘刻录工具：[Rufus](#)

当然你可以直接问组长要一个安装U盘，这样就不用自己下载了。

安装系统

1. 网上查你电脑的进入U盘引导的方法，可以是开机狂按某个键(如Esc)进入bios然后选择U盘作为启动盘，也可以是按某个键(如F6)，然后选择U盘作为启动盘，然后就会读取U盘里的系统文件，进入安装界面。（现在的电脑很可能需要关闭Secure Boot才能顺利进入安装引导盘，有独显的还需要关闭独显直连，具体方法需要上网查自己的电脑如何设置）
2. 按照安装界面提示安装Ubuntu
3. 安装过程选择不联网，安装完成后，拔出U盘，重启电脑
4. 进入安装好的系统，按下 `Ctrl+Alt+T`，打开一个终端，输入 `wget http://fishros.com/install -O fishros && . fishros`，这是一个自动脚本，你可以等待终端里显示可选功能。应该1是安装ROS，按照提示，选择替换默认源，然后安装就好了。（这一步也就包括了换源这个重要步骤。默认的ubuntu源比较慢，换成清华的源下载速度就会快很多）
5. 利用上面说过的自动脚本装上code
6. 随便点点，到处看看，熟悉一下

本节任务: 安装Ubuntu 22.04

二：Linux入门与基础知识

什么是Linux系统

Linux是在嵌入式系统、服务器、开发等领域的广泛应用的操作系统。\\

相比于Windows，Linux系统具有更高的自由度和可定制性，更适合开发和调试工作。\\

Linux系统的核心是Linux内核，而Linux发行版则是在Linux内核的基础上添加了一些其他软件，形成了一个完整的操作系统。\\

常见的Linux发行版有Ubuntu、Debian、CentOS等。

我们FYT战队的Robots上的计算平台都使用 `Ubuntu22.04`

为什么使用Linux而不是Windows？因为Linux系统更适合开发和调试工作。在Windows配置编程环境都是很麻烦的。在Linux，可以使用Docker一行代码完成环境配置，也可以用apt自动下载安装软件包和第三方库。Linux系统开销小，能节约一些计算资源，使我们的算法能够更快地运行。

Linux系统的特点

- Linux系统的理念是：**一切都是文件**，包括硬件设备、网络接口、进程等
- 各种操作都是使用**指令**实现，有点像Windows下的cmd
- 安装软件使用**包管理器 apt**，或者下载源代码自行编译安装
- 很多工具都没有图形化界面(GUI)，都是一个黑框框里面输入指令来使用(CUI)，如gcc,vim,htop
- 使用文件系统ext4，不同于Windows的NTFS，ext4很容易损坏，所以安装Linux系统的电脑一定不能随意断电，要使用指令 `shutdown now` 或者 `halt` 来关机，否则下次你开机开不了，就得重装系统和一大堆环境了

Linux的根目录结构

Linux文件系统采用了一种层次化的结构，由多个目录（文件夹）组成，形成一个树状的体系。在Linux中，根目录用 `/` 表示，其他目录通过路径来定位。例如：

- `/bin`：系统命令的存放目录
- `/sbin`：系统管理员命令的存放目录
- `/lib`：系统库文件的存放目录
- `/dev`：设备文件的存放目录
- `/proc`：虚拟文件系统，存放系统信息
- `/boot`：系统启动目录
- `/mnt`：挂载目录
- `/home`：用户主目录的根目录
- `/usr`：用户软件的存放目录
- `/etc`：系统配置文件的存放目录
- `/var`：变化数据的存放目录，如日志文件

通常我们的文件都会存放在 `/home` 目录下，例如 `/home/fyt/fyt-2024`，这个目录就是我们的主目录，我们可以在这个目录下创建其他目录，存放我们的文件。（源代码，文档等）

`/home` 下会有很多应用程序的配置文件，例如 `/home/fyt/.ssh`，这个目录存放的是我们的ssh公钥，用于接受远程登录。

三：基本命令与文件操作

一些文件的简单操作

基本命令的进一步学习

- `touch` : 创建空白文件
- `cat` : 查看文件内容
- `echo` : 输出文本内容
- `grep` : 在文件中搜索关键词
- `pwd` : 显示当前目录
- `cd` : 切换目录
- `ls` : 列出目录下的文件
- `cp` : 复制文件
- `mv` : 移动文件
- `rm` : 删除文件
- `mkdir` : 创建目录
- `chmod` : 设置文件权限
- `systemctl` : 管理系统服务
- `dpkg -i a.deb` : 安装a.deb这个软件包

通配符的应用

- `*` : 匹配任意字符
- `?` : 匹配单个字符
- `[]` : 匹配指定范围的字符
- 通配符可以在文件操作中极大地提高效率, 例如 `*.txt` 可以匹配所有的txt文件。

以下是一些基本命令的示范代码, 用于演示更多基本命令的使用:

```
#进入/home
cd

# 创建一个文件夹
mkdir test

# 进入文件夹
cd test

# 创建一个空白文件"example.txt"
touch a.txt

# 查看文件内容
cat a.txt

# 输出文本内容
echo "Hello, Linux!" > a.txt
```

```
# 在文件中搜索关键词
grep "Linux" a.txt

#重命名文件
mv a.txt bbb.txt

# 设置文件权限为可读可写可执行
# 777表示所有用户都有读写执行权限
# 可执行程序需要有执行权限才可被执行
chmod 777 bbb.txt

# 删除当前目录的所有文件
rm *
```

文本编辑器

- `nano`：简单易用的文本编辑器，比较简单
- `vim`：功能强大的文本编辑器，比较复杂，但是记住一些快捷键后会变得好用高效

apt安装软件

apt是Linux系统的包管理器，用于安装软件

- `apt update`：更新软件源
- `apt upgrade`：升级软件
- `apt install`：安装软件
- `apt remove`：卸载软件
- `apt search`：搜索软件
- `apt list`：列出已安装的软件
- `aptitude`：apt的图形化界面

一般在安装软件之前，需要先更新软件源，然后再安装软件。

也就是说，安装软件的步骤是：`apt update -> apt install [软件名]`

本节任务:

1. 更换apt源后，用apt安装 `fzf`
2. 用纯指令连上WiFi

四：上手体验一下linux环境下开发

这节，我们随便写一个简单的程序，然后编译运行，体验一下。

首先，创建一个名为 "my_cpp_project" 的文件夹，然后在其中创建以下文件。

1. `main.cpp`：放置你的C++源代码文件。

```
#include <iostream>

int main() {
    std::cout << "Hello, World!" << std::endl;
    return 0;
}
```

2. CMakeLists.txt：用于描述如何构建你的项目。

```
cmake_minimum_required(VERSION 3.10)

# 设置项目名称
project(my_cpp_project)

# 添加可执行文件
add_executable(my_cpp_project main.cpp)
```

这是一个简单的CMakeLists.txt文件，它定义了一个名为 "my_cpp_project" 的项目，并将 "main.cpp" 源文件与之关联。请确保将这两个文件放在相同的文件夹中。

接下来，打开终端并导航到项目文件夹。假设你的项目文件夹位于 "/home/my_cpp_project"，执行以下命令来构建项目：

```
cd /home/my_cpp_project
mkdir build
cd build
cmake ..
make
```

这将在 "build" 文件夹中生成可执行文件 "my_cpp_project"。运行它：

```
./my_cpp_project
```

这会输出 "Hello, World!"。

未来实际开发通常是在ROS2框架下举行的，将会使用另外一些编译工具如colcon，但是基本的编译流程是一样的。

基本流程就是：

1. 写源代码.cpp，.h，*.hpp等等
2. 写好CMakeLists.txt告诉编译器该如何编译你的项目，比如，需要包含哪些库，链接哪些库，需要编译哪些文件等等。当然这个txt往往是在写源代码的时候同步编写的。
3. 执行编译命令 `make`，编译器会根据CMakeLists.txt的描述编译你的项目，生成可执行文件。

本节任务: 学会使用code写代码，然后make一个cpp程序

五：最后的一些话

上面说的都是一些基本操作，常识。毕竟大家第一次使用linux，过程中肯定会发现很多问题，都需要大家去查阅解决方案，询问同学。

这是一个很方便的查询linux使用方法的网站，有各种常用指令和软件的使用说明 <https://dunwu.github.io/linux-tutorial/>

其实Ubuntu桌面版的使用跟Windows也是很相似的，在平时的使用也就是打开浏览器，写文档，写代码。安装好后不用一周就能熟练掌握实现日常使用。遇到抽象的问题时候，稍微上网查查，问问AI，就能轻松解决。还有很多视觉组的同学可以交流。所以大家不需要担心学不会linux,也不需要觉得学linux很麻烦。我们只需要会用，用到什么就查相应的资料。

一些常用的工具：

- `git`：版本控制工具，用于代码管理
- `code`：vscode，代码编辑器
- `cmake`：编译工具
- `make`：编译工具
- `firefox`：web浏览器
- `htop`：查看系统资源占用情况
- `ssh`：远程登录工具
- `nomachine`：远程控制工具
- `foxglove`：ROS数据可视化工具

code安装好后不要再用code官方提供的c++扩展了，网上找找clangd的安装方法，这好用多了。

然后就是熟悉一下使用终端，有很多快捷键

- `Ctrl+Alt+T`：打开终端
- `Ctrl+C`：终止当前进程
- `Ctrl+D`：退出终端
- `Ctrl+L`：清屏
- `Ctrl+Z`：挂起当前进程
- `Ctrl+R`：搜索历史命令

这里我强烈推荐大家去了解使用ChatGPT，Google Bard，Claude，bing chat等等的诸多大语言模型，使用这些AI能轻松解决这些linux使用上的问题。

这些大多是国外的项目，受到中国法律影响，国内使用需要一些手段。

手段一般称为 科学上网，梯子，机场，VPN，代理等等。

Github上有很多免费的，但是不太稳定[Github搜索免费clash](#)

网上有免费的有收费的，收费的月费一般是25~80元，大家根据自己的需求选择。

Github，Google Colab等都是优秀的工具网站，但是需要科学上网才能流畅使用。

本节任务:

1. 学会使用 `clash` 科学上网

2. 下载安装 `nomachine`
3. 注册使用 `poe.com` 中提供的AI
4. 学会关机指令 `shutdown now`

这一段是可能用得上的一段clash配置文件，如果现场条件不允许，则这段配置文件可以不用理会

在 clash 配置文件中添加以下内容

proxies:

- name: "temporary-socks5"

type: socks5

server: 0.0.0.0 # 这里需要改成我现场提供给你的ip地址

port: 15858

上面描述的是一个局域网内的socks5节点

这个节点搭建在我的电脑里，你们连接上我的热点后暂时通过这个节点科学上网

结束语

最后，祝大家学习愉快，遇到困难精良克服一下，保持良好心态，坚持下去，早日成为一名合格的视觉工程师。

Copyright: 秦胜林

2023/9/22