

# Linux 开发基础

## Linux Essentials

### —《机器视觉技术实训》课程笔记

杨青青

2020 年 2 月 25 日

本次课程介绍机器视觉技术课程中所需要的一些基础技能，用于机器视觉算法实验环境的部署。我们使用免费开源的 Linux 操作系统（以 Ubuntu Linux 发行版为例）。

## Linux 操作系统

平时，我们经常接触的电脑一般是个人电脑 (Personal Computer)，除了 Apple 公司的 Macintosh（简称 Mac）电脑外，其它的品牌电脑、或者组装电脑都统称为 PC 机 (Personal Computer 的简称)。Apple 公司为自己家的 Mac 电脑打造了自己的操作系统：macOS，而 PC 机一般都会安装 Microsoft 公司的 Windows 操作系统。这两个操作系统各有特点。macOS 系统源于 Unix 系统，为 Mac 电脑的硬件进行了优化，在购买 Mac 电脑时便可以免费获得，而且可以终身免费升级；而 Windows 系统则拥有丰富的驱动程序，可以兼容市场上主流的硬件系统，一般需要购买许可证才能进行安装。目前，Windows 的大版本更新<sup>1</sup>需要单独购买。

实际上，除了我们常用的这两种操作系统外，我们平时上网所访问的服务器大多数都运行另外一种开源、免费的操作系统：Linux 操作系统<sup>2</sup>。各个盈利/非盈利组织在开源的 Linux 内核基础上结合开源的程序包和图形用户界面，包装了不同的 Linux 发行版<sup>3</sup>。可以说，Linux 操作系统的生态圈云集了世界上最优秀的程序员，其开源、免费的特点也更加适合进行软硬件开发。本课程选择 Ubuntu Linux 发行版作为开发平台，Ubuntu 也是目前学术界和工业界在软硬件开发中使用最多的 Linux 发行版之一。一般，Ubuntu 在每年的 4 月都会发行一个新版本，目前，Ubuntu 19.04 是最新的发行版。每隔 2 年会发行一个长期支持版，Ubuntu 18.04 LTS 是目前最新的稳定的长期支持版本<sup>4</sup>，可以在 ubuntu 官网下载：<https://ubuntu.com/download>。

注意：本课程推荐下载安装 Ubuntu 18.04 LTS Desktop 英文版。英文环境对软件开发更加友好，中文的问题可以通过安装软件包解决。课程实践所用的 Jetson TX2 搭载的是基于 Ubuntu 16.04 剪裁的操作系统 NVIDIA L4T (Linux for Tegra)，是 NVIDIA JetPack SDK<sup>5</sup>的一个组件，可以通过官方提供的 SDK Manager

Copyright © 2020 杨青青  
Email : qqyang@nit.zju.edu.cn

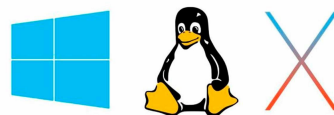


图 1: 三大桌面操作系统图标。

<sup>1</sup> 比如从 Windows 8 升级到 Windows 10。微软曾经在 Windows 10 刚推出的那一年，实行了 1 年的正版免费升级计划。

<sup>2</sup> 准确来说，我们常说的 Linux，是指 Linux 内核程序，由 Linus Torvalds 最早开发、全世界程序员共同贡献维护。

<sup>3</sup> 常见的 Linux 发行版有 Debian, Ubuntu, Centos/Red Hat Enterprise Linux, Fedora, openSUSE/SUSE Linux Enterprise 等等，所有的 Linux 发行版都是免费的，有些企业用途的发行版会提供收费的技术支持，这也是开源项目的重要盈利来源。

<sup>4</sup> LTS 代表 Long Term Support，推荐安装。有兴趣的同学可以去了解 Linux 各个发行版的特点。

<sup>5</sup> <https://developer.nvidia.com/embedded/jetpack>

进行系统升级<sup>6</sup>。

<sup>6</sup> 刷机有风险

## Ubuntu Linux 安装指南

我不会在课程笔记中详细介绍 Ubuntu Linux 的安装步骤，这个留给同学们课余自己去研究<sup>7</sup>。这里，我介绍几种推荐的安装方式：

<sup>7</sup> 不同的安装方法可以搜索网络教程，课程视频里演示了一种通过虚拟机安装的方法，可以作为参考。

1. 在 Windows 10 上，可以安装 Linux 子系统。这种安装方式对原操作系统的影响最小，推荐尝试 Linux 系统的用户安装。  
参考：<https://docs.microsoft.com/en-us/windows/wsl/install-win10>。
2. 通过虚拟机安装，适用于所有操作系统。虚拟机是独立于操作系统的一个软件系统，可以调用系统硬件资源。通过虚拟机安装 Linux 操作系统，体验和真实的操作系统更为接近。但是，虚拟机运行时会占用原有系统内存，建议较高配置的电脑安装。推荐虚拟机包括：Oracle VirtualBox(<https://www.virtualbox.org/>) 和 VMware Workstation Player(<https://www.vmware.com/cn/products/workstation-player/workstation-player-evaluation.html>)<sup>8</sup>。
3. 安装双系统。通过光盘或者 U 盘安装介质可以在电脑原有的操作系统上同时安装 Linux 操作系统，但是需要额外的独立硬盘分区。这种安装方式可以体验 Linux 操作系统的所有特性，但硬盘分区等操作会有一定风险，建议希望体验高性能或者原先电脑硬件配置不高的同学安装。

<sup>8</sup> VirtualBox 是免费软件，VMware Workstation Player 可免费用于非商业、个人和家用目的，而 VMware Workstation 是需要付费的。

## Shell

Ubuntu Linux 会默认安装一个图形用户界面 (Graphical User Interface, GUI)，而且随着发行版本的更新，其图形用户界面的体验已经逐渐接近 Windows 和 macOS。对于一般的用户，图形用户界面可以满足所有的应用需求。但是，图形用户界面的操作方便性也带来了其限制性。我们只能按照软件提供的菜单、按钮和窗口等进行交互。对于普通用户，这种操作方式可以在一定程度上减少无法预知的交互错误，而对于开发者或者系统管理员来说，这种限制是无法接受的。所以，一般的操作系统都会提供一种文本交互界面 (Text Interface)，在 Unix/Linux 下，就是 Shell 程序<sup>9</sup>。在这一小节，我们通过对 bash(Bourne Again SHell)<sup>10</sup>的学习来介绍 Linux 环境下开发所需要的常用命令，同时对 Linux 的一些基本特性进行介绍。

在图形界面下使用 Shell<sup>11</sup>，首先需要有一个终端模拟器 (terminal emulator)。尽管不同的 Linux 发行版会使用不同的桌面系统，但一般都会将其提供的终端模拟器命名为“Terminal”。当然，除了

<sup>9</sup> 在 Windows 中，有命令行工具 CMD 或者 Power Shell

<sup>10</sup> bash 是很多 Linux 的默认 Shell 程序，除了 bash 外，还有 csh, zsh 等 Shell。Bourne Again SHell 的命名又充分体现了 Linux 的命名逻辑，Bourne(born) Again SHell，即 bash，是 sh (Bourne Shell，以作者 Steve Bourne 命名) 的增强版本，而 sh 是 Unix 下的默认 Shell。

<sup>11</sup> 这里需要提及，Linux 操作系统的桌面程序是一个单独的应用包，我们可以不安装桌面程序，这样，我们会直接在终端使用 Shell。

系统自带的 Terminal，我们也可以安装其它的终端模拟器。这些模拟器的功能或者界面不同，但最终的目的都是提供给用户操作 Shell 的环境。

在 Ubuntu 下，可以在系统应用菜单栏，选择 Terminal 打开终端，或者在桌面/文件夹的任意处右击鼠标，在上下文菜单栏中选择“Open Terminal”打开终端。右击鼠标打开终端的方便之处在于打开终端后，当前所在的位置就是右击鼠标时所在的文件夹。而从系统菜单栏打开菜单，默认会进入用户的“家目录” (home directory, Linux 下用符号“~”表示家目录)。

现在在桌面任意空白处右击鼠标，打开一个终端，在默认情况下会显示一个黑色背景的窗口，在窗口界面的左上方会显示一串提示符，后面是一个闪烁的光标，提示用户可以用键盘输入。提示符的默认显示格式是这样的：

```
[用户名@计算机名 当前文件夹名]$
```

符号“\$”前的文本会根据不同的用户环境和所在目录改变。如果提示字符串最后不是“\$”，而是“#”的话，就要注意了，因为你现在是在 root 权限下，就是超级管理员，可以操作任意系统文件。我们可以通过 `su` 命令从普通用户权限升级到系统管理员权限<sup>12</sup>，但是不建议这么做。现在，一种更安全的方式是使用 `sudo` 命令来行使管理员权限。

在后面的笔记内容中，为了简洁，我们会将“\$”前的内容省略，有时为了明确目录名，会在偶尔的情况下在“\$”符号前加上当前文件夹名。<sup>13</sup>

<sup>12</sup> 前提是具有升级的权限。一般自己安装的 Linux 操作系统只有一个用户，是默认赋予升级权限的。

<sup>13</sup> 事实上，提示字符串的格式是在配置文件里修改的，有兴趣的同学可以在网上搜索修改的方法。

## Linux Shell 常用命令

### 浏览文件夹

树形结构是操作系统文件组织的基本结构。和 Windows 系统的物理分区方式不同，Linux 系统下的分区对于文件系统是透明的。例如，在 Windows 下，我们可能会将一块硬盘划分成 C 盘和 D 盘两个分区，这两个分区是各自独立的，用户很容易就区分这两个不同的分区。在 Linux 下，物理分区在文件系统中都是不可见的，所有的物理分区都挂载 (mount) 在以根目录所在的文件系统树上<sup>14</sup>。因此，一个 Linux 操作系统必需划分一个根目录分区。

目前，大家不需要关心挂载分区或者外接存储的问题。只要有这样的概念：Linux 的文件系统的底层目录是根目录，用“/”表示，根目录包含了一些文件和子目录，在这些子目录中又可以包含其它的文件和子目录<sup>15</sup>。在图形界面的文件浏览器中，我们可以展开文件系统树形结构，当前系统的家目录在文件系统树中的位置如图所示。

<sup>14</sup> 一种常见的分区方式是，将 home 目录单独进行物理分区，对于用户而言，该分区依然挂载在“/home”目录下，在浏览文件系统时，和没有单独分区的 home 目录没有任何分别。

<sup>15</sup> 大家是否还记得数据结构课程中介绍的树结构？与文件系统树对应，根目录可以看作是根节点，文件是叶子节点，子目录是普通节点。

在浏览文件夹时，最常用的几个命令如下：

1. **pwd**: **p**rint **w**orking **d**irectory, 打印当前工作目录；
2. **cd**: **c**hange **d**irectory, 改变目录；
3. **ls**: **l**ist, 列出目录内容。

在浏览文件夹时，我们有时需要确定当前目录。可以使用**pwd**<sup>16</sup>命令显示当前所在的目录名，以绝对路径名（absolute pathname）方式显示。绝对路径是从根目录开始到目标目录（pwd 命令中，目标目录即当前目录）的一条路径，用“/”分隔。例如：

<sup>16</sup> print working directory 的首字母

```
$ pwd
/home/mv
```

此处示例中，显示的是当前用户 home 目录的绝对路径名。与绝对路径名对应的另外一种路径名形式是**相对路径名**（relative pathname），相对路径名指的是目标目录相对与当前目录的路径，常常和转换目录命令**cd**一起使用，在临近的目录层次进行切换。在相对路径中，有两个特殊的符号需要了解：

1. “.”：单独一个句点，表示当前目录；
2. “..”：连续两个句点，表示当前目录的上一层目录，即父目录。

例如，我们可以使用绝对路径名转换到/bin目录：

```
$ cd /bin
```

在/bin目录下一般是 Linux 系统自带的程序，我们使用的 bash 程序也在该目录下。为了找到它，可以使用**ls**指令列出当前目录的所有内容。试一下，在 Shell 中输入**ls**，找一下 bash 在哪里吧。

我们也可以使用**cd**指令转换到用户安装程序路径：

```
$ cd /usr/local/bin
```

现在尝试一下使用相对路径名：

```
$ cd ..
```

这个命令会转换到当前工作目录的上一层目录，即/usr/local目录。所有用户自己安装的内容默认都会放在这里。用**ls**命令看一下这个目录下有什么内容吧。

我们也可以给**ls**指定需要列出的目录内容：

```
$ ls ./lib
```

这个命令就会列出当前目录下的lib目录中的所有内容。这里，“./”并不是必需的，可以直接输入

```
$ ls lib
```

**小技巧：**有些目录会包含很多文件或者子目录，为了确定我们需要寻找的文件是否在该目录下，可以使用这样的命令：

```
$ ls | grep bash
```

这个命令用到将 **ls** 列出的所有内容传送给 **grep** 命令，**grep** 会查找是否存在 **bash**，如果存在，就会仅打印出和 **bash** 有关的内容。

**小技巧：**几个有用的捷径。我们可以直接使用 **cd** 命令而不输入任何路径名：

```
$ cd
```

可以将工作目录转换到用户的 home 目录；通过输入

```
$ cd -
```

可以跳转到前一次所在的目录。

会得到一样的结果。Shell 默认会以当前目录为相对路径的起始点。如果当前目录下没有lib目录，就会显示错误信息。因为我们当前在/usr/local目录下，所以这个命令不会报错，在usr/local/lib中存放了用户安装的库文件。以后我们编译安装 OpenCV，编译结果产生的动态连接库文件就会放在这个目录中，可以通过查看该目录内容检查 OpenCV 是否正确编译安装了。

通过使用pwd、cd和ls这三个命令，结合绝对/相对目录名，可以浏览系统中的大部分目录内容<sup>17</sup>。关于指令ls的用法，我们在下一小节中再展开介绍。

<sup>17</sup> 有些目录是当前普通用户无法查看的，例如这台机器上的其他用户的 home 目录中的内容。这涉及到权限问题，在后面的内容中再详细讲解。

## 查看内容

在上一节的基础上，我们可以进一步探索 Linux 系统。为了探索系统内容，仅仅切换工作目录显然是不够的，我们需要一些工具可以查看目录或者文件的内容。本小节介绍以下命令：

1. ls：列出目录内容；
2. file：查看文件的类型属性；
3. less：查看文本文件内容。

上一小节中的ls程序可以列出目录内容，除了简单地直接使用ls命令外，还可以在使用时加入一些命令选项。

一般，一个 Linux 命令后可以加入若干个选项（options）和若干个参数（arguments），所以大部分命令以下面的形式出现：

`command -options arguments`

这里，第一个command是所执行的命令名，即命令本身；-options用连字符（dash）后紧接字母的形式，可以为该命令指定执行选项；最后的arguments是命令参数，指定命令所操作的对象。一般而言，选项可选的，可以不加，也可以加多个选项；而参数则是由命令本身的操作属性决定，有些命令没有指定的参数个数，而有些命令必需输入规定个数的参数。

以ls程序为例，常用的选项有-l，-a等。在 Linux Shell 中，一般使用“-”，即连字符<sup>18</sup>后接一个字符来输入命令参数，例如：

<sup>18</sup> 在键盘中和减号运算符是同一个按键。

```
$ ls -a /usr/local/include
```

这个指令会打印出目录/usr/local/include下的所有内容（-a参数的作用，包括以“.”开头的内容<sup>19</sup>）。而选项-l则可以将目录输出以详细列表的方式呈现：

<sup>19</sup> 在 Linux 下，一般以“.”开头的文件或者目录是隐藏不显示的。

```
$ ls -l /usr/local/include
```

多个参数可以接在一个“-”后合并输入，例如：

```
$ ls -la ~
```

这个指令结合了两个选项，会以详细的列表方式打印出用户 home 目录下的所有内容<sup>20</sup>。

关于ls命令的详细用法，可以查找系统手册来查看：

```
$ man ls
```

注意，这里的man命令不是英文单词 man 的含义，而是 **manual** 的缩写。一般系统提供的程序，都会有详细的 manual 可以查看如何使用。在使用 Linux Shell 时，对于不熟悉的命令或者参数，请多翻阅手册。

如果需要查看一个文件的类型或者文件属性，可以使用file命令。例如：

```
$ file ~/.bashrc
```

该命令会显示用户目录中 bash 配置文件的属性。如果把file命令用于文件夹，那么它仅仅会输出：directory。

当我们使用man命令查看文档的时候，大部分的 Linux Shell 会使用less程序来打开文档。less程序是 Shell 中功能比较强大的文本查看器，可以用分页的格式输出文本文档。但是，该程序仅限于查看文本文件，并没有编辑文本的功能<sup>21</sup>。该程序提供了灵活的文件内导航按键支持，完整的使用方法请man less。一般，常用的按键是“空格键”或者“f键”向下滚动一个屏幕的内容；“b键”向上滚动一个屏幕的内容；“q键”退出。

如果文本文档内容很少，在一个窗口页面内可以显示完整，也可以使用轻量级的程序：cat。

<sup>20</sup> 这里，波浪号“~”是 Shell 中用户 home 目录的捷径写法

<sup>21</sup> 可以按 v 键，less 会调用 vi 程序编辑文档。

## 操作文件和目录

有了前面的基础，就可以在 Shell 中进行文件/目录的操作了。本节介绍以下 Linux 常用命令：

1. mkdir：创建目录；
2. cp：copy，复制文件/目录；
3. mv：move，移动/重命名文件/目录；
4. rm：remove：删除文件/目录；

首先介绍mkdir命令，该命令可以创建一个指定的文件夹。为了统一安排工作内容，我习惯在自己的用户目录创建一个名字为 Workspace 的文件夹。可以直接使用绝对路径名创建：

```
$ mkdir ~/Workspace
```

回忆一下，“~”是用户家目录的快捷方式。也可以先用cd命令导航到需要创建目录的位置，然后用相对路径名进行创建。现在我们在 Workspace 下创建一个沙盒（sandbox）目录。

```
$ cd ~/Workspace
$ mkdir sandbox
```

如果需要创建的目录已经存在，系统会输出提示：File exists。

创建了一个新目录后，我们可以创建一个文本文件，然后往里面写一些内容：

```
$ touch text1
$ echo "Hello, Shell" > text1
```

这里使用了touch命令创建了一个空的文本文件<sup>22</sup>，命名为text1。在Linux中并没有后缀名的概念，但是对于一些源程序或者文本文档，我们会可以加上惯用的后缀名加以区分。echo命令的功能非常简单，就是回显。这里，我们用echo命令将字符串Hello, Shell进行回显，并使用输出重定向“>”将其写入新建的text1文件中<sup>23</sup>。

有了一个文本以后，我们可以使用cp命令进行复制。

```
$ cp text1 text2
```

上面的命令创建了一个text1的副本，取名为text2。现在再新建一个新目录：

```
$ mkdir dir1
```

现在，我们可以使用cp命令将文本文件 text2 拷贝到新建的目录dir1中：

```
$ cp text2 dir1
```

注意，这里的第二个参数是目录名，因此，会将text2这个文件以同样的名字拷贝到dir1目录中。此外，cp命令还可以复制整个文件夹的内容，只不过，这时需要-r选项的帮助：

```
$ cp -r dir1 dir2
```

这个命令会把dir1目录中的内容完整复制到dir2目录。如果dir2目录不存在，就会新建一个。

命令mv提供了移动文件和重命名文件的功能。下面的命令将text1文件移动到dir1中：

```
$ mv text1 dir1
```

如果第二个参数是文件名，那就是执行重命名功能。例如，将仍在当前目录下的 text02文件重命名为text：

```
$ mv text02 text
```

命令mv也可以重命目录，例如：

```
$ mv dir2 dir3
```

<sup>22</sup> 事实上，touch 命令是用于修改文件时间的，如果这个文件不存在，它就会新建一个，并将这个文件的修改时间设置为当前时间。

<sup>23</sup> 这里用了简单的方式创建了一个文本，常规的方法应当使用文本编辑器来创建，这个我们在下一个小节进行简介。

该命令将目录`dir2`重命名为`dir3`。这里需要注意的是，如果`dir3`已经存在，那么会将`dir2`目录移动到`dir3`中作为其子目录。事实上，`mv`命令是借助`cp`和`rm`命令来实现重命名的。命令`rm`可以删除文件/目录。例如：

```
$ rm dir1/text1
```

该命令会删除`dir1`目录下的`text1`文件。如果需要删除目录，需要`-R` 或`-r`选项：

```
$ rm -R dir3
```

上面的命令，不管`dir3`目录是否为空，都会删除整个目录。需要注意的是，在 Shell 中是没有文件回收站的，所有的命令执行了就会立即生效。所以，一个好的建议是，除非确定需要删除文件/目录，否则，将其重命名成一个 `back` 文件/目录是一个比较好的选择，在确定需要删除之后再进行删除。一个常见的做法是：

```
$ mv dir1 dir1.back
```

即将`dir1`目录重命名为一个带`.back`后缀的目录，用来区分其它正常文件/目录。

### 命令为何物？

这一节中分了几小节介绍了常用的 Linux 命令，但一个 Linux 命令究竟是什么？和程序有什么区别？这里，回答这个疑惑。

在 shell 中，一个命令可能是以下四种情况中的一种，可以通过`type`命令来找到该命令的真实类型：

- Shell 的内建命令。这个相当于 Shell 程序内部提供的特殊指令，比如`cd`命令。使用`type cd`，会显示`cd is a shell builtin`。说明，这个命令是 shell 的一个内建命令；
- 一个可执行程序。可以是编译的二进制程序，例如用 C/C++ 开发的可执行二进制文件；也可以是可执行的脚本文件，例如 shell、python、perl 脚本等；例如，`less`就是一个可执行程序，通过`type less`可以打印出该程序的绝对路径名。
- 一个 shell 函数。Shell 可以配置环境变量，还支持编程。因此，一个 shell 函数也可以作为一个命令。
- 一个别名。在 shell 中自定义的一个别名。这个别名可以是上面三种情况的别名，也可以是加了选项命令的别名。比如常用的`ls`命令。

**动手试一下：**用 `type ls` 命令看看 `ls` 是什么属性的命令。



## 小结

在本次笔记中，我们介绍了 Linux 操作系统和安装 Ubuntu Linux 系统的建议。此外，我们介绍了 shell 和 Linux shell 中常用的命令。通过这些常用命令，基本上可以实现大部分的文件/目录操作。每一个命令是什么用途，需要同学们自己多实践练习，在遇到不明白的地方通过查找查看手册或者查找网上相关的资料进行了解。进行工程开发没有捷径，唯有保持平常的心态在不断的思考、提问和总结中才能不断提升自己的能力。网络提供了开发者无尽的资源，希望大家能够充分利用，培养搜索解决问题方法的技术。

## 阅读和参考资料

[1] W. Shotts, “The Linux Command Line.”<sup>24</sup> [Online]. Available: <http://www.linuxcommand.org/tlcl.php>.

<sup>24</sup> 该书中文版名为《Linux 命令行大全》

## 实践

- 参考演示视频，安装一个 Ubuntu Linux 操作系统；可以选择虚拟机或者双系统方式。
- 熟悉 Linux 命令行。

Virtual Box 网盘分享链接：

<https://pan.nit.zju.edu.cn/share/8c7cdc6de03580509529d7cc96>

Ubuntu 16.04 64-Bit 镜像文件网盘分享链接：<https://pan.nit.zju.edu.cn/share/b0e499c71f2ef7d2895e3d77c2>