

# 狼牙算法组第一次培训

@ powered by 邹佳吕

## 一. Linux基础

### 1. ubuntu

- 为什么使用Ubuntu?

1. 社区环境好: Ubuntu是一个Debian系分支的第一大系统, 是用户量最大的linux发行版。他的社区环境也非常好, 遇到任何问题都可以在用户社区[askubuntu](https://askubuntu.com)中寻找解答。
2. 内核精简: Linux的内核和系统比Windows更加精简, 故在**运行时占用的各类资源都要小于Windows**。

在不打开任何应用的情况下, 笔者的电脑在运行Windows10时占用的内存为4.2G, cpu占用率在10-20%左右, 而运行Ubuntu20.04LTS的时候, 只使用了2.2G的内存, cpu占用率只有10%不到。这样, 在运行我们的视觉算法程序时, 可以更充分地利用系统资源, 最大程度压榨电脑的性能。(甚至可以在测试结束后实际运行时关闭图形界面, 只保留终端! 这样, 系统内核作为唯一需要运行的基础程序, 大概能将cpu占用率缩小到1-2%)

3. Linux配置环境比较方便, 特别是对深度学习的支持比较友好, 经常有sh脚本能够一键配置开发环境; 并且ROS也是在Ubuntu下开发的, 其对ROS的支持也是最大的

### 2. Linux常用命令:

cat	cd	chmod	cp	export	find	grep
kill	man	mkdir	mv	ps	pwd	rm
su	tar	top	ls	ln		

一些常用的Linux命令如上, 请大家自行百度具体含义及用法

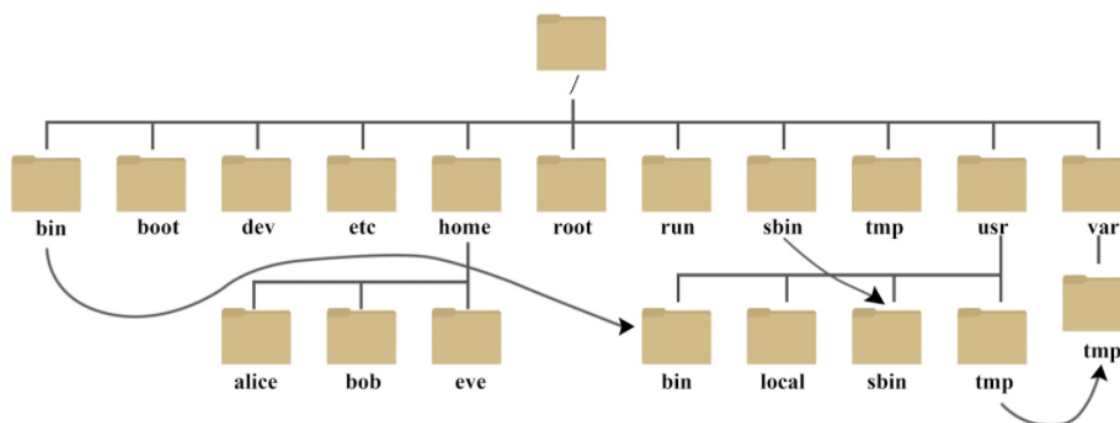
### 3. Vim编译器

Vim是一种文本编辑器。代码补完、编译及错误跳转等方便编程的功能特别丰富, 在一些没有图形化界面的系统中运用比较广泛, 大家需要掌握一些基础用法

- 了解使用vim编译器打开, 保存, 关闭文件的方法
- 了解vim编译器的三种模式含义
- 了解上下左右移动光标对应的按键
- 了解搜索, 删除, 替换, 复制等操作
- 了解进入和退出插入模式快捷键

详细教程见[vim.pdf](#)

## 4. Linux目录结构（了解即可）



- **/bin:**  
bin 是 Binaries (二进制文件) 的缩写, 这个目录存放着最经常使用的命令。
- **/boot:**  
这里存放的是启动 Linux 时使用的一些核心文件, 包括一些连接文件以及镜像文件。
- **/dev :**  
dev 是 Device(设备) 的缩写, 该目录下存放的是 Linux 的外部设备, 在 Linux 中访问设备的方式和访问文件的方式是相同的。
- **/etc:**  
etc 是 Etcetera(等等) 的缩写,这个目录用来存放所有的系统管理所需要的配置文件和子目录。
- **/home:**  
用户的主目录, 在 Linux 中, 每个用户都有一个自己的目录, 一般该目录名是以用户的账号命名的, 如上图中的 alice、bob 和 eve。
- **/lib:**  
lib 是 Library(库) 的缩写这个目录里存放着系统最基本的动态连接共享库, 其作用类似于 Windows 里的 DLL 文件。几乎所有的应用程序都需要用到这些共享库。
- **/lost+found:**  
这个目录一般情况下是空的, 当系统非法关机后, 这里就存放了一些文件。
- **/media:**  
linux 系统会自动识别一些设备, 例如U盘、光驱等等, 当识别后, Linux 会把识别的设备挂载到这个目录下。
- **/mnt:**  
系统提供该目录是让用户临时挂载别的文件系统的, 我们可以将光驱挂载在 /mnt/ 上, 然后进入该目录就可以查看光驱里的内容了。
- **/opt:**  
opt 是 optional(可选) 的缩写, 这是给主机额外安装软件所摆放的目录。比如你安装一个ORACLE 数据库则就可以放到这个目录下。默认是空的。
- **/proc:**  
proc 是 Processes(进程) 的缩写, /proc 是一种伪文件系统 (也即虚拟文件系统), 存储的是当前内核运行状态的一系列特殊文件, 这个目录是一个虚拟的目录, 它是系统内存的映射, 我们可以通过直接访问这个目录来获取系统信息。  
这个目录的内容不在硬盘上而是在内存里, 我们也可以直接修改里面的某些文件, 比如可以通过下面的命令来屏蔽主机的ping命令, 使别人无法ping你的机器:

```
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

- **/root:**  
该目录为系统管理员，也称作超级权限者的用户主目录。
- **/sbin:**  
s 就是 Super User 的意思，是 Superuser Binaries (超级用户的二进制文件) 的缩写，这里存放的是系统管理员使用的系统管理程序。
- **/selinux:**  
这个目录是 Redhat/CentOS 所特有的目录，Selinux 是一个安全机制，类似于 windows 的防火墙，但是这套机制比较复杂，这个目录就是存放selinux相关的文件的。
- **/srv:**  
该目录存放一些服务启动之后需要提取的数据。
- **/sys:**  
这是 Linux2.6 内核的一个很大的变化。该目录下安装了 2.6 内核中新出现的一个文件系统 sysfs 。  
sysfs 文件系统集成了下面3种文件系统的信息：针对进程信息的 proc 文件系统、针对设备的 devfs 文件系统以及针对伪终端的 devpts 文件系统。  
该文件系统是内核设备树的一个直观反映。  
当一个内核对象被创建的时候，对应的文件和目录也在内核对象子系统中被创建。
- **/tmp:**  
tmp 是 temporary(临时) 的缩写这个目录是用来存放一些临时文件的。
- **/usr:**  
usr 是 unix shared resources(共享资源) 的缩写，这是一个非常重要的目录，用户的很多应用程序和文件都放在这个目录下，类似于 windows 下的 program files 目录。
- **/usr/bin:**  
系统用户使用的应用程序。
- **/usr/sbin:**  
超级用户使用的比较高级的管理程序和系统守护程序。
- **/usr/src:**  
内核源代码默认的放置目录。
- **/var:**  
var 是 variable(变量) 的缩写，这个目录中存放着在不断扩充着的东西，我们习惯将那些经常被修改的目录放在这个目录下。包括各种日志文件。
- **/run:**  
是一个临时文件系统，存储系统启动以来的信息。当系统重启时，这个目录下的文件应该被删掉或清除。如果你的系统上有 /var/run 目录，应该让它指向 run。

详细学习可见：[Linux文件目录结构一览表\(biancheng.net\)](http://biancheng.net)

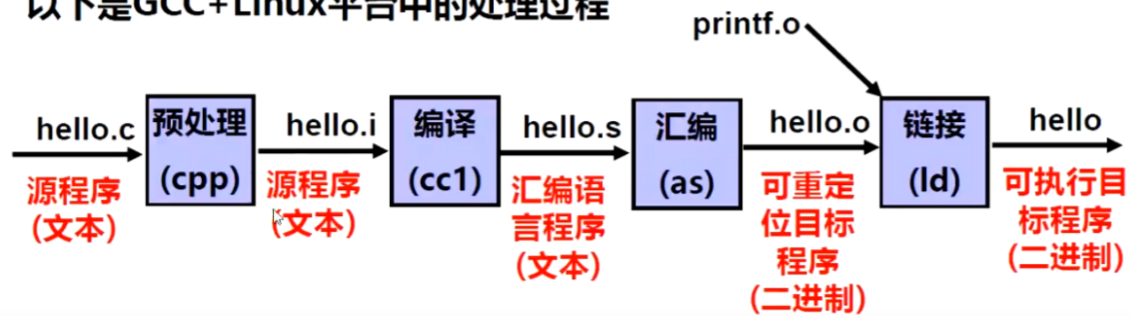
## 二 . CMake && Makefile

### 1. 可执行文件

cpp文件是我们的代码文件，是不能直接执行的，需要编译源码生成可执行文件

```
g++ -o demo demo.cpp
./demo
```

以下是GCC+Linux平台中的处理过程



## 2. 链接第三方库

编译代码时如果用到了第三方库（非c++标准库），我们需要告诉编译器第三方库的头文件（.h .hpp）和库文件（.so .a）的位置

安装第三方库一般有两种方式

1. apt

```
sudo apt-get install xxxx
```

直接在终端输入以上命令即可安装，apt默认情况下，头文件放在/usr/include下，库文件放在/usr/lib下，可执行文件放在/usr/bin下

2. 从源码编译安装

在对应库的官网下载源码安装，如opencv等，默认情况下头文件放在/usr/local/include下，库文件放在/usr/local/lib下，可执行文件放在/usr/local/bin下

## 3. Makefile

Makefile 可以简单的认为是一个**工程文件的编译规则**，描述了整个工程的编译和链接等规则。其中包含了那些文件需要编译，那些文件不需要编译，那些文件需要先编译，那些文件需要后编译，那些文件需要重建等等

以下是一段makefile的代码

```
hello:hello.c
gcc -I${HOME}/incl -c hello.c
gcc -o hello hello.o
rm -f hello.o
mv hello ${HOME}/bin
```

有了makefile文件，我们只需要在终端输入

```
make
```

即可自动编译整个工程

不同的系统（linux, mac, windows）cpp文件生成可执行文件的过程可能不通，makefile文件也可能不同。不同的Make工具遵循着不同的规范和标准，makefile的格式也不同，其可移植性比较差

## 4.CMake

CMake 使你在构建项目时不再郁闷的想自杀了

--一位 KDE 开发者

CMake是一个 **跨平台的安装（编译）工具**，可以用简单的语句来描述所有平台的安装（编译过程）

在工程文件夹中编写***CMakeLists.txt***文件，执行

```
cmake
```

即可自动生成***Makefile***文件，再执行

```
make
```

即可自动编译整个工程

以下是一段链接***opencv***的***CMakeLists.txt***文件

```
cmake_minimum_required(VERSION 3.5)
project(opencv_test)
find_package(OpenCV REQUIRED)
include_directories(${OpenCV_INCLUDE_DIRS})
add_executable(opencv_test main.cpp)
target_link_libraries( opencv_test ${OpenCV_LIBS})
```

详细教程见***CMake Practice.pdf***

## 三 . Git && Github

Git是一个开源的分布式版本控制系统，可以有效、高速地处理从很小到非常大的项目版本管理

GitHub是一个面向开源及私有软件项目的托管平台，因为只支持Git作为唯一的版本库格式进行托管，故名GitHub

git的版本管理非常强大，希望大家熟练使用。

详细教程见：[GitHub教程](#) [Git Bash详细教程](#) [LolitaSian-CSDN博客](#)

## 四 . Markdown

### 1. Markdown是什么

**Markdown** 是一种**轻量级**标记语言，创始人是**约翰·格鲁伯**(John Gruber)。它允许人们“使用易读易写的纯文本格式编写文档，然后转换成有效的 HTML 文档。

来自维基百科

### 2. Markdown语法

**1. 标题类**，#数量决定几级标题

# 一级标题

## 二级标题

#### 四级标题

## 2. 列表类

\* 无需列表1- 生活

\* 无需列表2- 态度

\1. 有序列表- 起床

\2. 有序列表- 刷牙

## 3. 分割线

---

## 4. 引用

> 是我引用

## 5 文字格式

**粗体**

*斜体*

~~被删除~~

## 6. 代码

```
print('hello world')
```

详细教程见[为什么要学Markdown? 有什么用? - 知乎\(zhihu.com\)](https://www.zhihu.com/question/20132836)

推荐使用[Typora](https://typora.io/)作为编辑器，好用好看且比较轻量



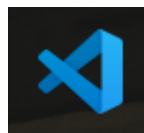
# 五 . IDE

工欲善其事，必先利其器，下面推荐几款Linux下编写C++程序使用的IDE

1. [Clion](https://www.jetbrains.com/clion/): JetBrains家的IDE，审美很戳我，并且功能强大，会自动配置Cmake, gcc, g++, 并且自带git, 支持markdown, 远程部署代码功能。缺点是打开速度较慢，并且有些尴尬的一点是目前仿佛似乎好像我科的教育邮箱不支持白嫖了...



2. [VScode](https://code.visualstudio.com/): 微软的小儿子，本质上是一个文本编辑器，安装各种插件之后可以用来写C++, python, java等多种语言代码。插件生态非常丰富，可定制性强，配置完之后用起来非常舒心。



# 六 . 任务要求

1. 熟悉Linux命令，制作表格如下：

命令	含义	选项	选项含义
----	----	----	------

类型		命令	含义	选项	选项一含义	选项二	选项二含义	选项三	选项三含义
文件操作		ls	查看当前文件夹内容	- l	列表显示	- h	配合- l显示文件大小	- a	显示隐藏文件
		pwd	查看当前位置						
		cd	切换文件（无其它则返	-	在最近两次目录切换				
		touch	创建文件						

2. 使用vim编译器对***vim.txt***进行指定操作
3. 任选自己招新考核题其中的一道题，使用CMake重新构建（大二的同学必须选择4 5 6其中一个）
4. 完成stl库中常用数据结构练习（vector，list，queue），完成使用的demo代码，要求代码：注释== 1：1（环境为linux下cmake编译）
5. 所有任务都要写报告，要求使用markdown
6. 本次及以后所有的作业均用github提交和管理，我会建立一个organization，会邀请大家加入，每个人都建立一个以自己名字命名的仓库，然后把代码和报告push到上面。