## 2熟悉 Linux (2 分,约 2 小时)

#### 如何在Ubuntu 中安装软件（命令⾏界⾯）？它们通常被安装在什么地⽅？

sudo apt-get install 软件名

或者sudo dpkg -i 软件名.deb，还有编译安装二进制文件安装啊之类的

一般被安装在/usr和/usr/local下，也有在/opt下的

#### linux 的环境变量是什么？我如何定义新的环境变量？

当我们使用指令的时候，比如说ls，系统会依照环境变量PATH的设定去每一个PATH定义的目录下面寻找名为ls的可执行文件。

可以直接在terminal中使用export命令增加新的环境变量，比如export PATH=$PATH:/myexecutable/。

PATH中的路径通过：相隔，即这条命令的意思是在当前PATH的末尾加上myexecutable这个目录。这样通过terminal的修改立即生效但仅对本次本账户有效果，关闭terminal之后会失效。

另外可以修改当前用户目录下的~/.bashrc文件，在文件末尾加上export PATH=$PATH:/myexecutable/可以使得这次修改永久对当前用户有效。这种修改需要当前用户再打开terminal，或者手动执行source ~/.bashrc使之生效。

同样可以修改etc/bashrc etc/profile等文件，这样的修改是系统级别的。

#### linux 根⽬录下⾯的⽬录结构是什么样的？⾄少说出3 个⽬录的⽤途。

/为根目录，根目录下面包括/bin,/sbin,/etc,/dev,/proc,/var,/tmp,/usr,/home,/boot,/lib,/opt,/mnt,/media,/src这些目录。

其中bin是用户们使用的二进制可执行文件目录，比如ls就在这里面。Sbin是系统管理员用的二进制可执行文件所在目录。Boot是开机相关的引导文件目录。Etc是配置文件目录。Dev是设备文件目录。Proc是进程相关。Var是变量文件。Tmp是临时文件。Usr是用户的二进制啊库文件等的所在，它有自己的/usr/bin,/usr/sbin/user/lib,/usr/local。Home目录是默认是放每个用户个人文件的地方，也就是~/。Lib是系统库。Opt是可选的附加程序目录。Mnt是挂在目录。Media是光碟啊软盘啊之类的设备目录。Src是服务目录。

#### 假设我要给a.sh 加上可执⾏权限，该输⼊什么命令？

以“给用户加可执行权限，而组和其他人只保留read权限”为例

sudo chmod u+x,g=r,o=r a.sh

另外也可以用数值421的组合来修改权限

#### 假设我要将a.sh ⽂件的所有者改成xiang:xiang，该输⼊什么命令？

只有root权限可以更改文件所有者和所在组，sudo chown xiang:xiang a.sh

3 SLAM 综述文献阅读 (3 分,约 3 小时)

1. *SLAM 会在哪些场合中⽤到？⾄少列举三个⽅向。*

无人驾驶，无人车，无人机，无人船，AR，机器人，三维重建啊，总之需要理解环境并做出交互的场景都需要SLAM。

1. *SLAM 中定位与建图是什么关系？为什么在定位的同时需要建图？*

定位和建图是相辅相成的，毕竟定位是需要参考系的，而地图就是我们建的参考系。很多情况下，机器人或者人进入到一个陌生空间，是没有办法提前获得地图的。比如“好奇号”到了火星，我们现在没有其他技术可以提前测量好整个火星的地图，只能让火星车一边探索一边建图。

1. *SLAM 发展历史如何？我们可以将它划分成哪⼏个阶段？*

最早的是80年代到00年代的基于概率论的SLAM，比EKF啊粒子滤波啊这些基于贝叶斯滤波的SLAM。

然后00年代后期SLAM的架构因为一些经典算法的涌现而确定下来了。 可以把SLAM分为前后端，前端主要负责处理传感器数据，比如里程计相机激光雷达IMU乃至磁力计声纳等等；后端要负责建图，回环检测，重定位和优化等等。比如基于graph的SLAM就具备回环检测和全局优化的能力。

1. *列举三篇在 SLAM 领域的经典⽂献*

Gmapping – Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective sampling.

Cartographer – real-time loop closure in 2d lidar slam

PATM – parallel tracking and mapping for small AR workspace

ORB-SLAM – ORB-SLAM: a versatile and accurate monocular SLAM system

4 CMake 练习 (2 分,约 1.5 小时)

1. *请按照上述要求组织源代码文件,并书写 CMakeLists.txt。*

文件结构

cmake\_pratice/build

/CMakeLists.txt

/include

/hello.h

/src

/CMakeLists.txt

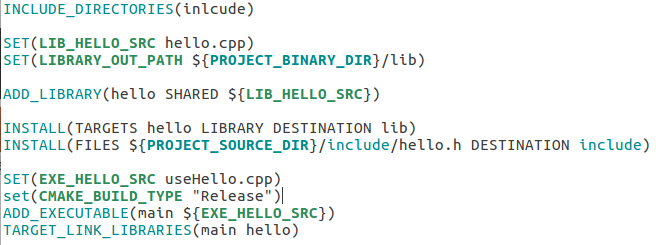
/hello.cpp

/useHello.cpp

其中cmake\_pratice/CMakeLists.txt内容:



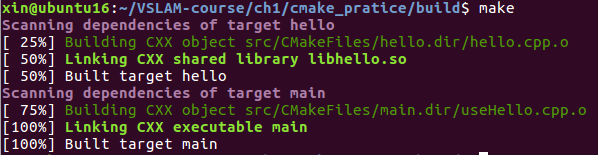
而cmake\_pratice/src/CMakeLists.txt内容:



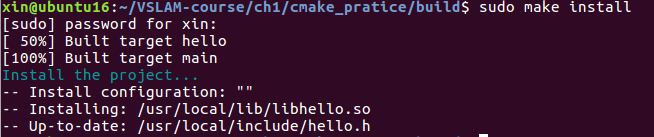
运行结果:

cmake ..

make

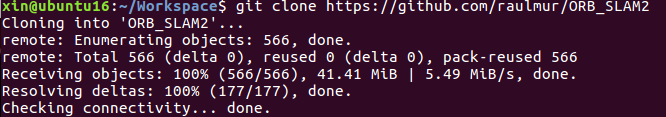


sudo make install



5 理解ORB-SLAM2 框架

1. *下载完成后,请给出终端截图。*



* 1. *ORB-SLAM2 将编译出什么结果?有几个库文件和可执行文件?*

编译结果包含一个libORB\_SLAM2.so库文件和 6个可执行文件: mono\_tum,rgbd\_tum,stereo\_kitti,stereo\_euroc,mono\_euroc,mono\_kitti

* 1. *ORB-SLAM2 中的 include, src, Examples 三个文件夹中都含有什么内容?*

Include 下是各种头文件

src下是源码，用来编译出orbslam库的

example包含各个例子的源代码，各个例子编译出来的可执行文件也在这里

* 1. *ORB-SLAM2 中的可执行文件链接到了哪些库?它们的名字是什么?*

编译出来的可执行文件都link的是liborb\_slam2.so库

但是liborb\_slam2.so这个库在被编译的时候又link了

opencv\_libs

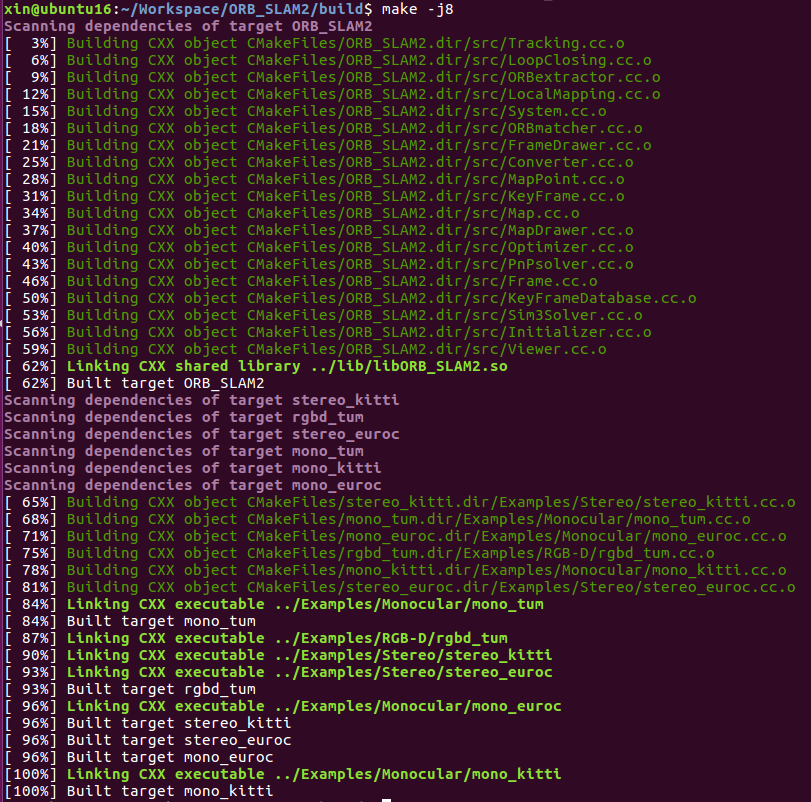
eigen3\_libs

libDBow2.so

libg2o.so

6\* 使用摄像头或视频运行 ORB-SLAM2(3 分,约 1 小时)

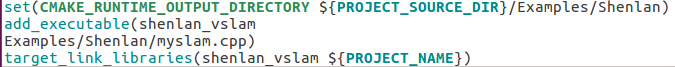
1. *请给出它编译完成的截图。*



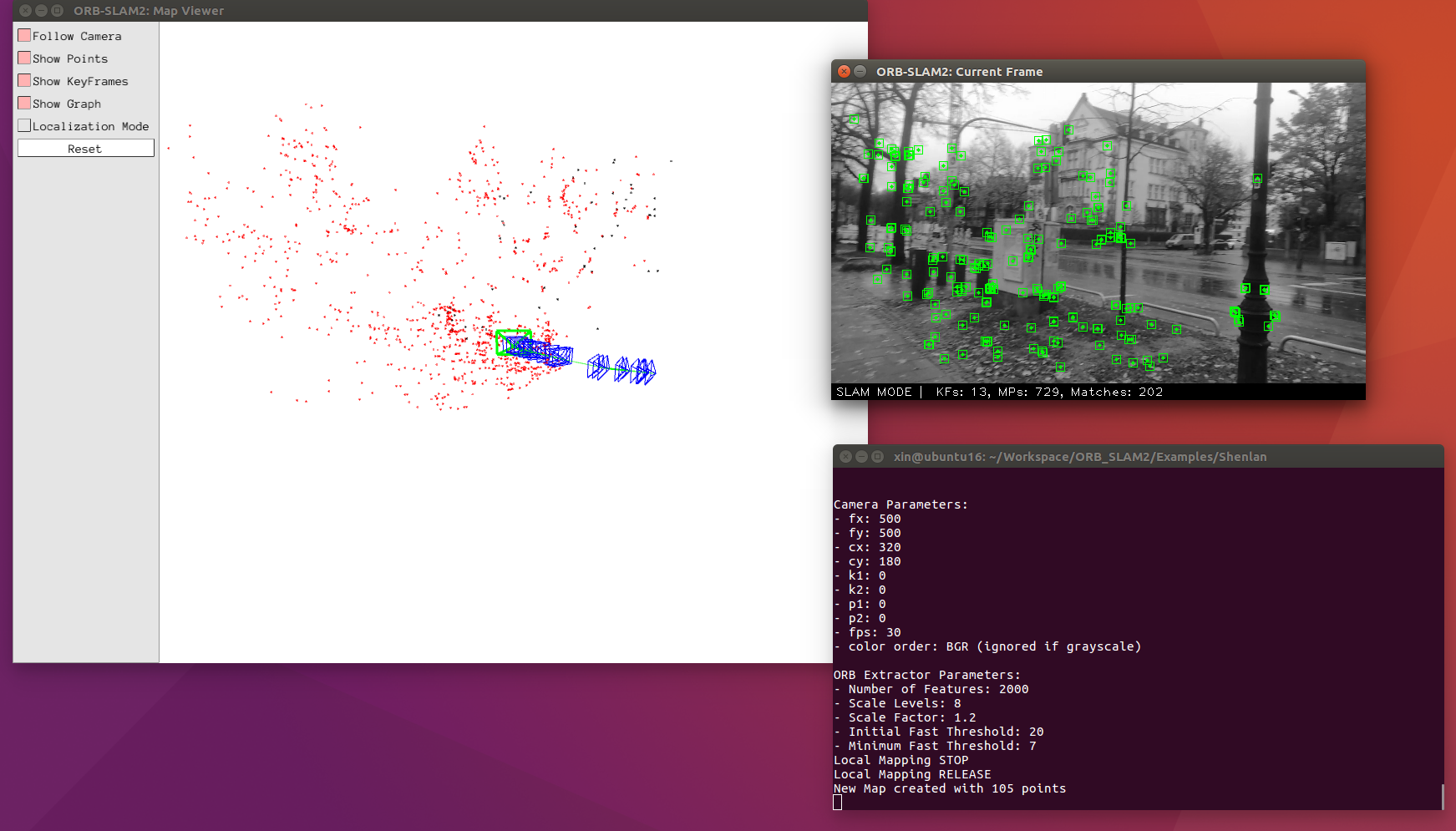
1. *请给出你的 CMakeLists.txt 修改方案。*

在example中创建文件夹“shenlan\_vslam”，把myslam.cpp进去。

Cmakelist.txt最下面加入



1. *请给出运行截图,并谈谈你在运行过程中的体会。*

每次都要加载词库不是很方便。