二、熟悉Linux

1. 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？

1) 使用命令行工具apt安装。

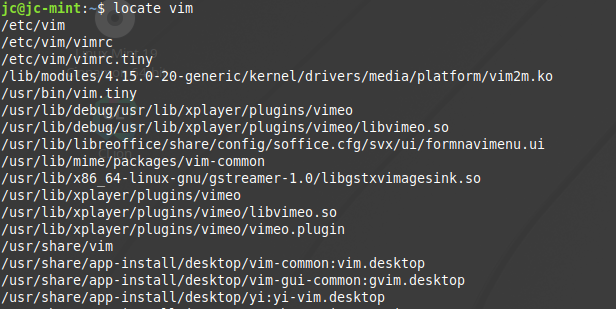
例如安装vim编辑器：sudo apt install vim

2) 使用Deb包安装

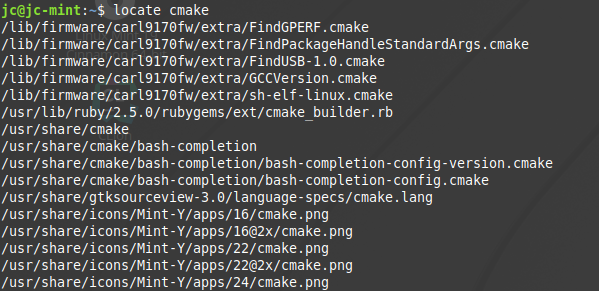
例如：sudo dpkg -i <package.deb>

通常安装在/usr目录下，但也有部分文件在/var、/lib目录下。

如vim(部分)：



Cmake(部分)：



2. linux 的环境变量是什么？我如何定义新的环境变量？

环境变量是$PATH变量的值（一系列目录），决定了shell到哪个目录中寻找命令或者程序。

1. 查看环境变量值：echo $PATH



1. 添加PATH环境变量(临时)，可用：

export PATH=/xxx/xxx/xxx:$PATH **但是这种方法在关闭终端后失效**。

1. 永久添加环境变量(影响当前用户):

编辑 ~/.bashrc 文件，gedit ~/.bashrc

export PATH="/xxx/xxx/xxx:$PATH"

1. 永久添加环境变量(影响所有用户):

编辑 /etc/profile 文件，gedit /etc/profile

export PATH="/xxx/xxx/xxx:$PATH"

3. 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。

目录结构：



部分目录作用：

|  |
| --- |
|  |
| **目录** | **说明** | **备注** |
| bin | 存放普通用户可执行的指令 | 即使在单用户模式下也能够执行处理 |
| boot | 开机引导目录 | 包括Linux内核文件与开机所需要的文件 |
| dev | 设备目录 | 所有的硬件设备及周边均放置在这个设备目录中 |
| etc | 各种配置文件目录 | 大部分配置属性均存放在这里 |
| opt | 第三方软件安装目录 | 现在习惯性的放置在/usr/local中 |
| root | 系统管理员主目录 | 除root之外,其他用户均放置在/home目录下 |
| tmp | 存放临时文件目录 | 所有用户对该目录均可读写 |
| usr | 应用程序放置目录 |  |
| var | 存放系统执行过程经常改变的文件 |  |

4. 假设我要给a.sh 加上可执⾏权限，该输⼊什么命令？

Chmod –x a.sh

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

xang:xiang的意思是xiang组下xiang用户吗？如果是，那么：

改变所属组群：sudo chgrp xiang a.sh

改变拥有者：sudo chown xiang a.sh

三、SLAM 综述文献阅

1. SLAM 会在哪些场合中用到？至少列举三个方向。

1) 增强现实

2) 自主机器人

3) 计算机视觉与信号处理

4) 无人驾驶

2. 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？

定位与建图的关系是相互关联相互制约的，准确的定位需要精确的地图，精确的地图来自准确的定位。定位的同时需要建图，因为如果没有地图，主体的定位会很快发生漂移（draft），从而产生很大的累计误差。如果存在地图，可以通过回环检测及时纠正（reset）定位误差，很大程度上限制（limit）了估计错误。而地图构建的信息来自每一次定位，所以在定位的同时需要建图。

3. SLAM 发展历史如何？我们可以将它划分成哪几个阶段？

SLAM从出现至今有30多年的发展历史，可以划分为**三**个阶段：

最初的20年称之为经典阶段（classical age，1986-2004），这个阶段的主要思想是基于概率学的方法（probabilistic formulations），出现了扩展卡尔曼滤波（EKF）、粒子滤波（RBPF）、最大似然估计等方法。

接下来的10年称之为算法分析阶段（algorithmic-analysis age，2004-2015），这个阶段重点研究了SLAM的基础属性，包括可观测性，收敛性和一致性。同时，对于高效SLAM求解很关键的的稀疏性理论也被理解，各种主要的开源SLAM库得以发展。

2015年至今，我们正在进入第三个阶段，称之为鲁棒性理解阶段（robust-perception age）。在这个阶段，SLAM研究的重点在于4个方面：鲁棒性，高级语义理解，资源限制下的调整，具体任务驱动下的SLAM。

4. 列举三篇在SLAM 领域的经典⽂献。

2006 《Probabilistic approaches and data association》 Durrant-Whyte and Bailey

2008 《Visual SLAM》 Neira et al. (special issue)

2016 《Visual place recognition》 Lowry et al.

四、CMake练习

三个CMakeLists文件：

CMakeLists:

PROJECT(USECMAKE)

ADD\_SUBDIRECTORY(include)

ADD\_SUBDIRECTORY(src)

include/CMakeLists:

INSTALL(FILES hello.h DESTINATION include)

src/CMakeLists:

ADD\_LIBRARY(hello SHARED hello.c)

INSTALL(TARGETS hello

LIBRARY DESTINATION lib)

INCLUDE\_DIRECTORIES(/home/jc/SLAM/ch1/cmake/include)

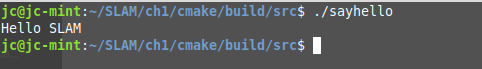
ADD\_EXECUTABLE(sayhello useHello.c)

TARGET\_LINK\_LIBRARIES(sayhello hello

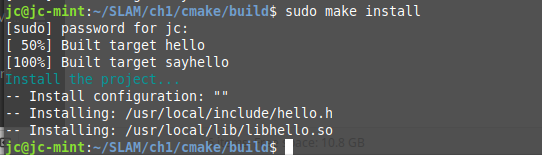
生成库文件：



可执行文件：

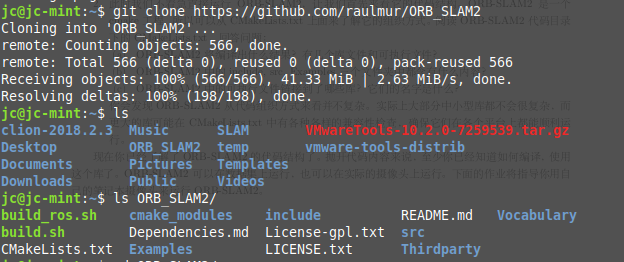


安装至相应位置：



五、理解ORB-SLAM2 框架

1. 下载截图



2. 代码结构

(a) 将编译出一个名为ORB\_SLAM2的工程，包括1个库文件和6个可执行文件

(b) include 含有用来编译库的头文件，src含有用来编译库的源文件，example含有用 来编译可执行文件的源文件

(c) 链接到的库有：

${OpenCV\_LIBS}

${EIGEN3\_LIBS}

${Pangolin\_LIBRARIES}

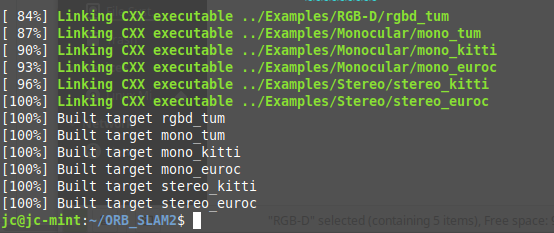
${PROJECT\_SOURCE\_DIR}/Thirdparty/DBoW2/lib/libDBoW2.so

${PROJECT\_SOURCE\_DIR}/Thirdparty/g2o/lib/libg2o.so

它们的名字分别是OpenCV、Eigen3、Pangolin、DBoW2、g2o

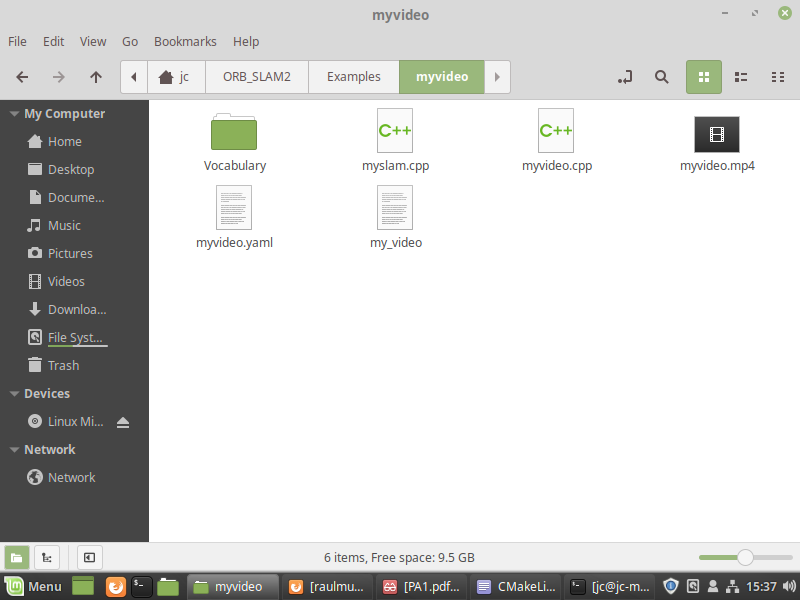
六、使用摄像头或视频运行ORB-SLAM2

1.编译完成



2.使用myvideo.cpp

1) 在Example目录下新建myvideo文件夹，放入相关文件如图：



2)在CMakeLists.txt末尾添加如下代码：

set(CMAKE\_RUNTIME\_OUTPUT\_DIRECTORY ${PROJECT\_SOURCE\_DIR}/Examples/myvideo)

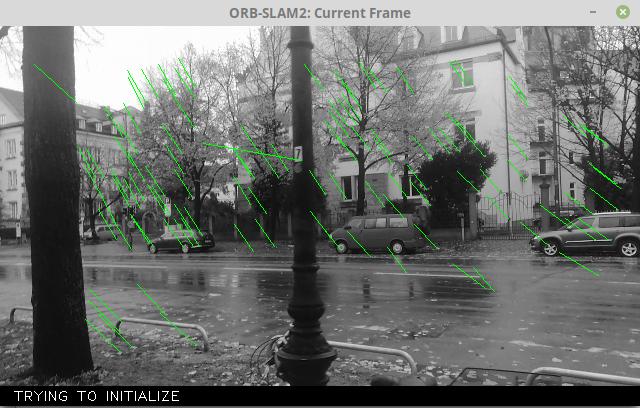
add\_executable(my\_video

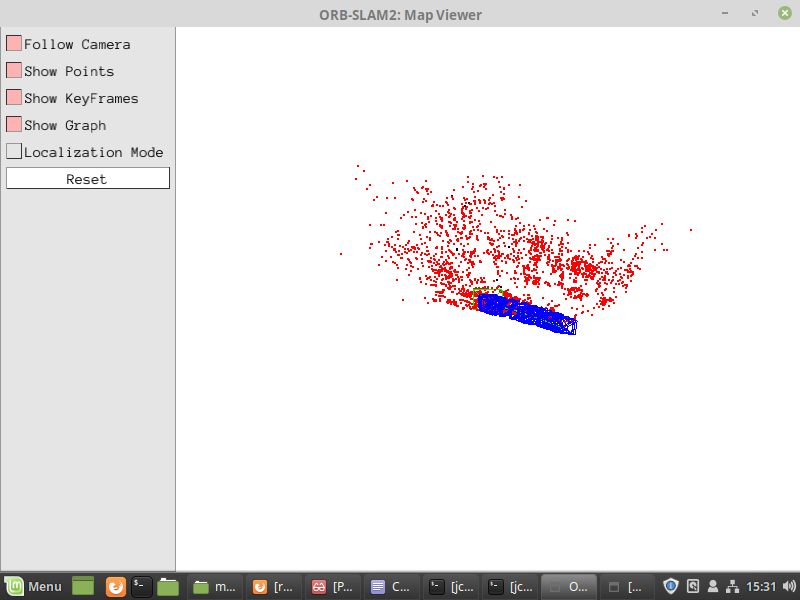
Examples/myvideo/myvideo.cpp)

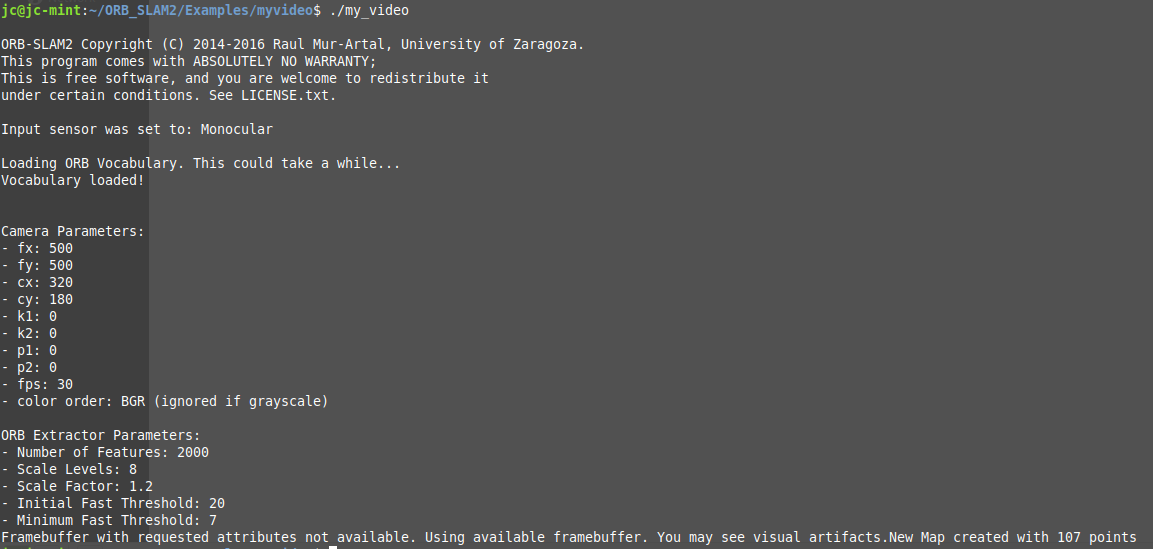
target\_link\_libraries(my\_video ${PROJECT\_NAME})

编译完成后在video目录下生成my\_video目标文件

3.运行截图







由于环境为虚拟机，所以使用了myvideo.mp4来运行ORB-SLAM2。首先加载ORB Vocabulary需要等待几秒钟，这个字典文件是什么还不清楚。运行过程中，帧窗口总是运行几秒后就停止（或者是卡住？尝试了很多次都是在一个位置停止），停止的同时终端输出最后一行信息，给出了新地图中关键点的数量。但是Map viewer窗口仍然在运行，在不断推进建图过程，很奇怪是否视频流也应该继续运行？大概一分钟后停止，所有窗口关闭。一旦开始运行，机器运转的声音就会变大，说明计算资源较多。