2.熟悉 Linux

Q1. 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.apt-get

sudo apt-get install + packagename

2.dpkg

sudo dpkg -i packagename.deb

/usr/share/

Q2. linux 的环境变量是什么？我如何定义新的环境变量？

环境变量指操作系统中用来指定操作系统运行环境的一些参数 eg.程序所在的完整路径. 系统需要一些变量来提供他的数据存储 或者是一些环境的设定参数值 通过环境变量来帮助 Linux 系统构建起用户的工作环境

1.print the environment variable

echo $PATH

2.open the file

sudo getdit /etc/profile

(/etc/profile/ is effective for all the users)

3.add needed environment variable

export PATH="…"

4.Update

source etc/profile

Q3. 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。

-首先 FHS 针对目录树框架仅定义出三层目录下应该存放的数据 分别为

/ 根目录 与系统开机有关

/urs 与软件的安装执行有关

/var 与系统运作过程有关

-其次 对于根目录,所有的目录都是由根目录衍生出来的,根目录与开机/还原/系统修复等劢作有关。根目录(/)所在分割槽应该越小越好, 且应用程序所安装的软件最好不要与根目录放在同一个分割槽内。 如此不但效能较佳,根目录所在的文件系统也较不容易发生问题。

-根目录下有如下次目录

1.FHS 定义出根目录下应该有的目录

/bin /boot /dev /etc /home /lib /media / mnt /opt /root /sbin /srv /temp

2.linux 下其他一些重要的目录

/lost+found /proc /sys

Linux 中所有的目录都有根目录开始 再一个一个分支下来 所以是树状结构来储存 目录树的起点为根目录

"/etc":所有的系统配置文件

"/opt":第三方软件的存放目录

"/boot":内核和加载内核所需要的文件 存放开机会使用要的档案

Q4. 假设我要给 a.sh 加上可执行权限，该输入什么命令？

修改文件权限应使用 chmod 命令

chmod u+x a.sh

PS:

u for user

g for group

o for others

a for all

r for read

w for write

x for executable

Q5. 假设我要将 a.sh 文件的所有者改成 xiang:xiang，该输入什么命令？

chown xiang:xiang a.sh

3.SLAM 综述文献阅读

Q1. SLAM 会在哪些场合中用到?至少列举三个方向。

1.Autonomous navigation of mobile robots which enables the robot to accomplish tasks like rescue task for high-risk environment and planetary exporation.

2.Automatic car piloting on unrehearsed off-road terrains

3.AR (Argumented Reality) applications where virtual objects are included in real-world scenes

4.Biological use for medical treatment eg. Body-SLAM

Reference: Visual simultaneous localization and mapping: a survey

Q2. 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?

Localization and mapping are dependent. SLAM should do the task of localization and mapping simultaneously.

For being precisely localized in an environment, a correct map is necessary. In order to construct a good map, it is also necessary to be properly localized when elements are added to the map [1].

Furthermore, the map is often required to support other tasks such as path planning or providing an visualization for a human operator. The map allows limiting the error committed in estimating the state of the robot. The robot can reset its localization error by re-visiting known areas [2].

Ref:

[1]Visual simultaneous localization and mapping: a survey

[2]Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age

Q3. SLAM 发展历史如何?我们可以将它划分成哪几个阶段?

1.Mapping and localization were studied independently [1].

2.SLAM problem was raised. The localization problem and the mapping problem were carried out in a current manner.

The probabilistic approaches, including approaches based on Extended Kalman Filters, Rao-Balckwellised Particle Filters and maximum likelihood estimation, were proposed to handle the SLAM problem[2].

3.Different sensors have been introduced to perceive and obtain information from the surrounding environment. Cameras which is inexpensive, lighter and have lower power comsumption are introduced to sense the environment. Approaches based on geometry and computer vision are proposed [1].

Ref:

[1]Visual simultaneous localization and mapping: a survey

[2]Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age

Q4. 列举三篇在 SLAM 领域的经典文献

R. Mur-Artal, J. M. M. Montiel, and J. D. Tardo´s. ORB-SLAM: a Versatile and Accurate Monocular SLAM System. IEEE Transactions on Robotics (TRO), 31(5):1147–1163, 2015.

R. Mur-Artal and J. D. Tardo´s. ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras . IEEE Transactions on Robotics, 33(5):1255-1262, 2016

G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR), pages 225–234. IEEE, 2007.

J. Engel, J. Scho¨ps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In European Confonference on Computer Vision (ECCV), pages 834–849. Springer, 2014.

4.CMake 练习

```
#声明一个 project/cmake 工程
project(Hello)

#set 'Release' mode
SET(CMAKE_BUILD_TYPE "Release")

#set g++ compiler for c++
SET(CMAKE_C_COMPILER "g++")

#generate a dynamic library called hellolib.so
add_library(libhello SHARED hello.c)
```

#添加可执行程序

add_executable(sayhello useHello.c)

#链接相应的库文件

target_link_libraries(sayhello libhello)

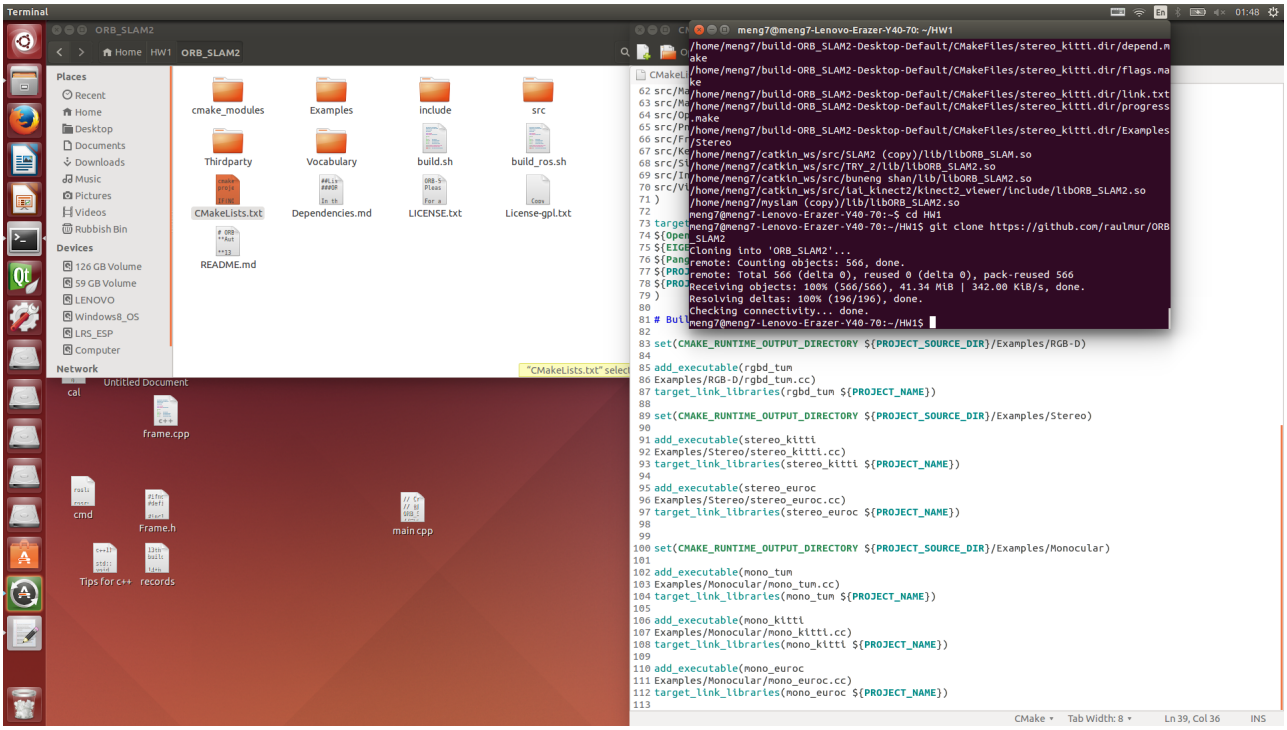
#INSTALL

INSTALL(TARGETS libhello LIBRARY DESTINATION /usr/local/lib/)

INSTALL(FILES include/hello.h DESTINATION /usr/local/include)

5.理解 ORB-SLAM2 框架

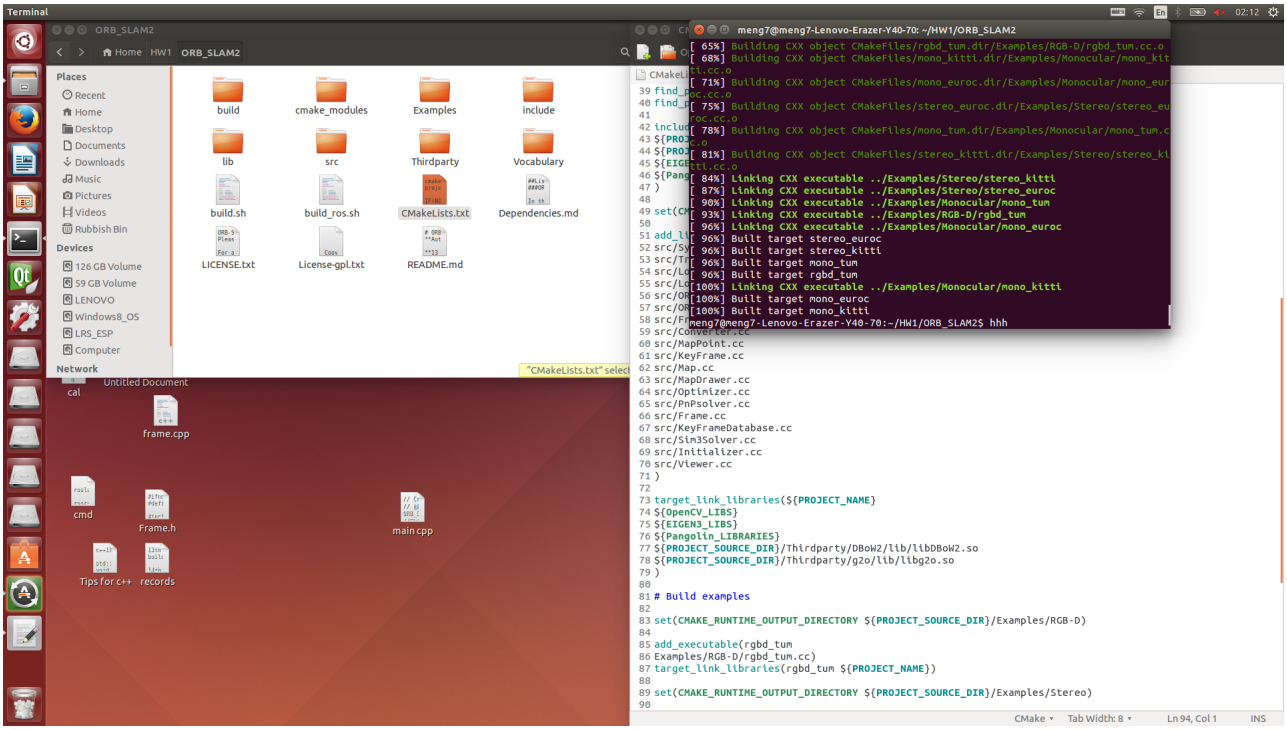
1.下载完截图



2.

(a) ORB-SLAM2 将编译出什么结果?有几个库文件和可执行文件?

结果如下



生成了六个可执行文件

rgbd_tum

stereo_euroc

stereo_kitti

mono_tum

mono_kitti

mono_euroc

生成了一个库文件

libORB_SLAM2.so

(b) ORB-SLAM2 中的 include, src, Examples 三个文件夹中都含有什么内容

include 里面存放的是头文件

src 里面存放的是源文件

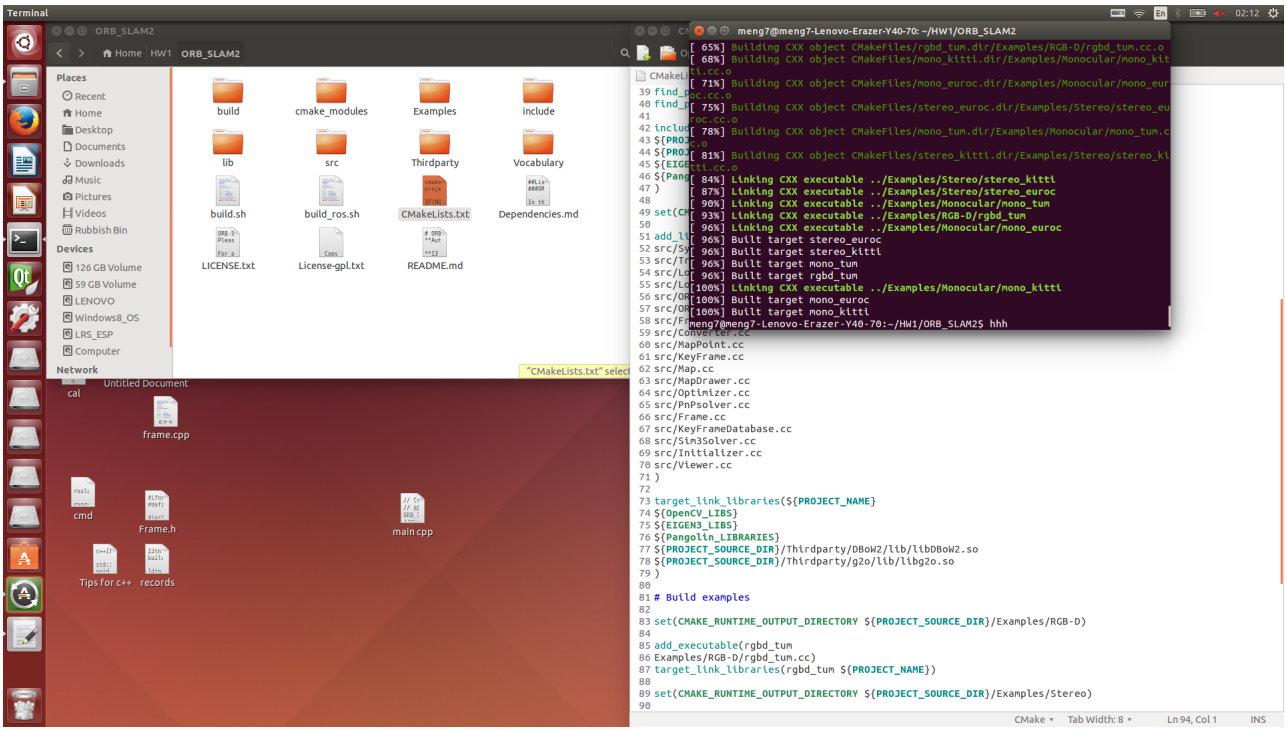
Example 里面存放的是基于单目、双目、RGBD 以及 ROS 的实例程序

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

链接到了
自身 ORB_SLAM2 相关代码生成的库
OpenCV_LIBS（OpenCV 库）
EIGEN3_LIBS （eigen 库）
Pangolin_LIBRARIES (Pangolin 库)
libDBow2.so (DBoW2 的动态库)
libg2o.so (g2o 的动态库)

6.使用摄像头或视频运行 ORB-SLAM2

Q1 编译完成的截图



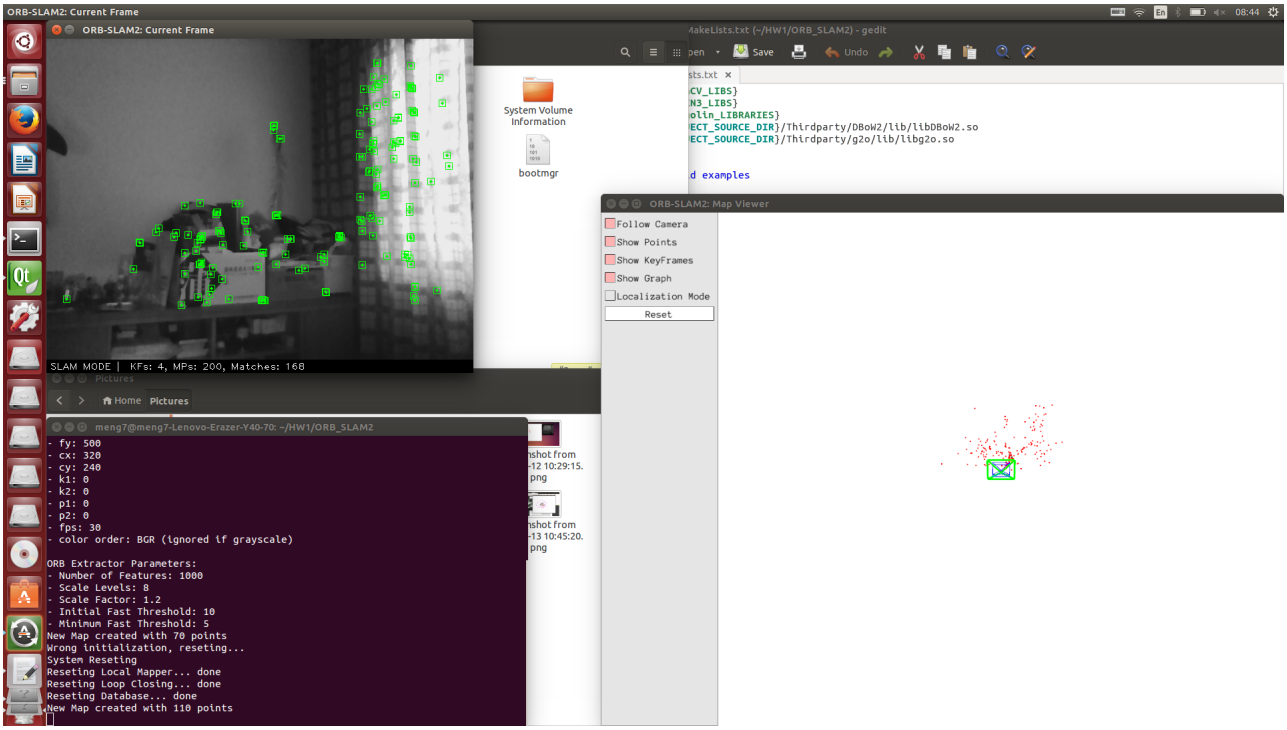
Q2 修改 CMakeLists.txt

添加上
set(CMAKE_RUNTIME_OUTPUT_DIRECTORY ${PROJECT_SOURCE_DIR}/Examples/mySLAM)

add_executable(mySLAM
Examples/mySLAM/myslam.cpp)
target_link_libraries(mySLAM ${PROJECT_NAME})

add_executable(myVIDEO
Examples/mySLAM/myvideo.cpp)
target_link_libraries(myVIDEO ${PROJECT_NAME})

Q3 运行截图



相机不能移动太快 否则会跟丢

但是跟丢之后可以使相机对着已有的 scene 可使程序找回并且继续 tracking