

2. 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?

- install: using the command `sudo apt install XYZ`, it might be however necessary to run `sudo apt update` firstly to synchronize the package index with their sources.
- Where the packages are installed are commonly sepecified by the developer. Commonly, the system software are installed under `/user/share`; the excutable are installed under `/user/bin`; the libraries are installed under `/user/lib`; the config files are installed under `/etc`.

2. linux 的环境变量是什么?我如何定义新的环境变量?

- As linux is an os for multi users, for every single user, the os has set up a bunch of variables to run his/her programs in a specified context. Those parameters are called **environment variables**.
- To set up environment parameters, one can use the `export VAR_NAME=VAR_VALUE` command in shell to do that.
- If this variable should be set up permanantly, this command can be written to a file so that it will be excuted everytime when a terminal is opened.
To make the variable valid globally, i.e. for all users, one can write this command to `etc/profile`.
To make this variable only valid for current user, one can write this command to `~/.bashrc`.

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。

- /
 - ├── bin
 - ├── boot
 - ├── cdrom
 - ├── dev
 - ├── etc
 - ├── home
 - ├── lib
 - ├── lib64
 - ├── lost+found
 - ├── media
 - ├── mnt
 - ├── opt
 - ├── proc
 - ├── root
 - ├── run
 - ├── sbin
 - ├── snap
 - ├── srv
 - ├── sys
 - ├── tmp
 - ├── usr
 - └── var

22 directories

Above is the directories under my root.

- Usage:
 - the `/bin` contains the commands and executables that are commonly avoked in ther terminal;
 - the `/etc` contains some config file for the system;
 - the `/home` contains directories for every user.

4. 假设我要给 a.sh 加上可执行权限,该输入什么命令?

- It depends, for example, if everyone should have the executive privilege, the following command can be used:

```
sudo chmod a+x a.sh;
```

if only user of this file should be granted executive privilege, one can user:

```
sudo chmod u+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?

- `sudo chown xiang:xiang a.sh`

3 SLAM 综述文献阅读

1. SLAM 会在哪些场合中用到?至少列举三个方向。

- Autonomous Driving, expecially in the perception and localization tasks;
- Telepresence;
- AR / VR

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?

- **Localization** and **Mapping** are accomplished at same time, i.e. simultaneously.
- In the localization, it is important to know the *relative* distance of a robot to obscales in its envrionment. Therefore, during the localization, the enrionment should also be built to make sense.

3. SLAM 发展历史如何?我们可以将它划分成哪几个阶段?

- At first beginning, the slam is mainly accomplished by radar/lidar sensor.
- Currently, visual slam is very popular as the computaiton power increased and a lot of camera sensors can be well used in slam tasks.

4. 列举三篇在 SLAM 领域的经典文献。

1. C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," IEEE Transactions on Robotics, vol. 32, no. 6, pp. 1309–1332, 2016.
2. J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: a survey," Artificial Intelligence Review, vol. 43, no. 1, pp. 55–81, 2015.
3. H. F. Durrant-Whyte and T. Bailey. Simultaneous Localisation and Mapping (SLAM): Part I. IEEE Robotics and Automation Magazine, 13(2):99–110, 2006.

4 CMake 练习

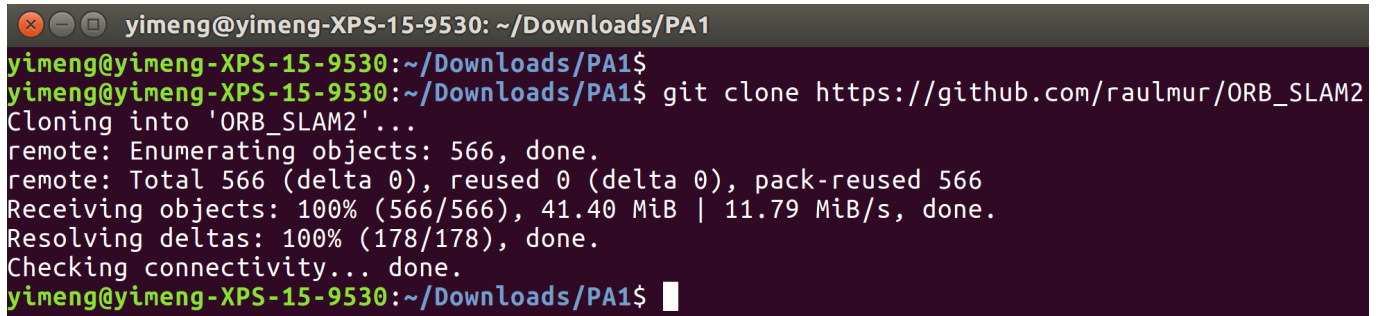
see zip files.

5 理解 ORB-SLAM2 框架

ORB-SLAM2[4] 是一个非常经典的视觉 SLAM 开源方案,它可以作为你学习 SLAM 的范本。但是现 在我们还没有讲解很多关于视觉 SLAM 的知识,所以仅从代码工程角度上来了解 ORB-SLAM2。请按照提示完成以下工作。

1.

从 github.com 下载 ORB-SLAM2 的代码。地址在:https://github.com/raulmur/ORB_SLAM2 提示:在安装 git 之后,可以用 `git clone https://github.com/raulmur/ORB_SLAM2` 命令下载ORB-SLAM2。下载完成后,请给出终端截图。



```
yimeng@yimeng-XPS-15-9530: ~/Downloads/PA1
yimeng@yimeng-XPS-15-9530:~/Downloads/PA1$
yimeng@yimeng-XPS-15-9530:~/Downloads/PA1$ git clone https://github.com/raulmur/ORB_SLAM2
Cloning into 'ORB_SLAM2'...
remote: Enumerating objects: 566, done.
remote: Total 566 (delta 0), reused 0 (delta 0), pack-reused 566
Receiving objects: 100% (566/566), 41.40 MiB | 11.79 MiB/s, done.
Resolving deltas: 100% (178/178), done.
Checking connectivity... done.
yimeng@yimeng-XPS-15-9530:~/Downloads/PA1$
```

2.

此时我们不着急直接运行 ORB-SLAM2,让我们首先来看它的代码结构。ORB-SLAM2 是一个 cmake 工程,所以可以从 CMakeLists.txt 上面来了解它的组织方式。阅读 ORB-SLAM2 代码目录下的 CMakeLists.txt,回答问题:

(a) ORB-SLAM2 将编译出什么结果?有几个库文件和可执行文件?

- Library: libORB_SLAM2.so
- Executable: rgb_d_tum, stereo_kitti, stereo_euroc, mono_tum, mono_kitti, mono_euroc.

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

- include: header files;
- src: source code for the libORB_SLAM2.so;
- Examples: example source code for executables.

(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?

The executables all have dependency on libORB_SLAM2.so, which has dependencies on OpenCV, Eigen3, Pangolin, libDBow2.so and libg2o.so.

6 使用摄像头或视频运行 ORB-SLAM2

1. 为了实际运行 ORB-SLAM2,你需要安装它的依赖项,并通过它本身的编译。请给出它编译完成的截图。

```

yimeng@yimeng-XPS-15-9530: ~/Downloads/PA1/ORB_SLAM2
-- Configuring done
-- Generating done
-- Build files have been written to: /home/yimeng/Downloads/PA1/ORB_SLAM2/build
Scanning dependencies of target ORB_SLAM2
[ 3%] Building CXX object CMakeFiles/ORB_SLAM2.dir/src/LocalMapping.cc.o
[ 9%] Building CXX object CMakeFiles/ORB_SLAM2.dir/src/Tracking.cc.o
[ 9%] Building CXX object CMakeFiles/ORB_SLAM2.dir/src/System.cc.o
[12%] Building CXX object CMakeFiles/ORB_SLAM2.dir/src/LoopClosing.cc.o
[15%] Building CXX object CMakeFiles/ORB_SLAM2.dir/src/ORBExtractor.cc.o
[18%] Building CXX object CMakeFiles/ORB_SLAM2.dir/src/FrameDrawer.cc.o
[21%] Building CXX object CMakeFiles/ORB_SLAM2.dir/src/ORBMatcher.cc.o
[25%] Building CXX object CMakeFiles/ORB_SLAM2.dir/src/KeyFrameDatabase.cc.o
[28%] Building CXX object CMakeFiles/ORB_SLAM2.dir/src/MapPoint.cc.o
[31%] Building CXX object CMakeFiles/ORB_SLAM2.dir/src/KeyFrame.cc.o
[34%] Building CXX object CMakeFiles/ORB_SLAM2.dir/src/Converter.cc.o
[37%] Building CXX object CMakeFiles/ORB_SLAM2.dir/src/Optimizer.cc.o
[43%] Building CXX object CMakeFiles/ORB_SLAM2.dir/src/Map.cc.o
[40%] Building CXX object CMakeFiles/ORB_SLAM2.dir/src/Frame.cc.o
[46%] Building CXX object CMakeFiles/ORB_SLAM2.dir/src/MapDrawer.cc.o
[50%] Building CXX object CMakeFiles/ORB_SLAM2.dir/src/Sin3Solver.cc.o
[53%] Building CXX object CMakeFiles/ORB_SLAM2.dir/src/PnP solver.cc.o
[56%] Building CXX object CMakeFiles/ORB_SLAM2.dir/src/Initializer.cc.o
[59%] Building CXX object CMakeFiles/ORB_SLAM2.dir/src/Viewer.cc.o
[62%] Linking CXX shared library ../libORB_SLAM2.so
[62%] Built target ORB_SLAM2
Scanning dependencies of target mono_euroc
Scanning dependencies of target stereo_kitti
Scanning dependencies of target mono_kitti
Scanning dependencies of target stereo_euroc
Scanning dependencies of target mono_tum
Scanning dependencies of target rgbd_tum
[ 65%] Building CXX object CMakeFiles/stereo_kitti.dir/Examples/Stereo/stereo_kitti.cc.o
[ 68%] Building CXX object CMakeFiles/mono_euroc.dir/Examples/Monocular/mono_euroc.cc.o
[ 71%] Building CXX object CMakeFiles/rgbd_tum.dir/Examples/RGB-D/rgbd_tum.cc.o
[ 75%] Building CXX object CMakeFiles/mono_kitti.dir/Examples/Monocular/mono_kitti.cc.o
[ 78%] Building CXX object CMakeFiles/mono_tum.dir/Examples/Monocular/mono_tum.cc.o
[ 81%] Building CXX object CMakeFiles/stereo_euroc.dir/Examples/Stereo/stereo_euroc.cc.o
[ 84%] Linking CXX executable ../Examples/Monocular/mono_tum
[ 87%] Linking CXX executable ../Examples/Monocular/mono_kitti
[ 87%] Built target mono_tum
[ 87%] Built target mono_kitti
[ 90%] Linking CXX executable ../Examples/Monocular/mono_euroc
[ 93%] Linking CXX executable ../Examples/RGB-D/rgbd_tum
[ 96%] Linking CXX executable ../Examples/Stereo/stereo_kitti
[100%] Linking CXX executable ../Examples/Stereo/stereo_euroc
[100%] Built target mono_euroc
[100%] Built target rgbd_tum
[100%] Built target stereo_kitti
[100%] Built target stereo_euroc
yimeng@yimeng-XPS-15-9530:~/Downloads/PA1/ORB_SLAM2$

```

2. 如何将 myslam.cpp 或 myvideo.cpp 加入到 ORB-SLAM2 工程中?请给出你的 CMakeLists.txt 修改方案。

```

set(CMAKE_RUNTIME_OUTPUT_DIRECTORY ${PROJECT_SOURCE_DIR}/Examples/MyVideo)

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

set(CMAKE_RUNTIME_OUTPUT_DIRECTORY ${PROJECT_SOURCE_DIR}/Examples/MySlam)

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

```

3. 现在你的程序应该可以编译出结果了。请给出运行截图,并谈谈你在运行过程中的体会。

```
ORB-SLAM2: Current Frame
yimeng@yimeng-XPS-15-9530:~/Downloads/PA1/ORB_SLAM2/Examples/MyVideo$ ./myvideo
```

```
ORB-SLAM2 Copyright (C) 2014-2016 Raul Mur-Artal, University of Zaragoza.
This program comes with ABSOLUTELY NO WARRANTY;
This is free software, and you are welcome to redistribute it
under certain conditions. See LICENSE.txt.
```

```
Input sensor was set to: Monocular
```

```
Loading ORB Vocabulary. This could take a while...
Vocabulary loaded!
```

```
Camera Parameters:
```

```
- fx: 500
- fy: 500
- cx: 320
- cy: 180
- k1: 0
- k2: 0
- p1: 0
- p2: 0
- fps: 30
- color order: BGR (ignored if grayscale)
```

```
ORB Extractor Parameters:
```

```
- Number of Features: 2000
- Scale Levels: 8
- Scale Factor: 1.2
- Initial Fast Threshold: 20
- Minimum Fast Threshold: 7
```

```
QObject::startTimer: Timers cannot be started from another thread
QObject::startTimer: Timers cannot be started from another thread
QObject::startTimer: Timers cannot be started from another thread
QObject::startTimer: Timers cannot be started from another thread
```

