

1. 习题说明

2. 熟悉 Linux (2pt, 2h)

2.1 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？

- 安装软件
 - apt 安装 (Ubuntu v16+) : `sudo apt install xxx`
 - snap 安装 (载到虚拟环境) : `sudo snap install xxx`
 - dpkg 安装: `sudo dpkg -i 文件名.deb`
 - 参考: <https://zhuanlan.zhihu.com/p/270908077>
- 安装到位置
 - apt 安装 -> `/usr/bin`
 - dpkg 安装 -> `/opt`
- 查找软件安装的位置
 - 用 `find` 或 `whereis/which` 命令查找文件位置

2.2 linux 的环境变量是什么？我如何定义新的环境变量？

- 系统环境变量详解: <https://blog.csdn.net/netwalk/article/details/9455893>
- 环境变量
 - 用来指定系统运行环境的一些参数
- 定义新的环境变量
 - `sudo gedit ~/.bashrc`
 - 输入用户密码
 - 末尾添加 `export PATH=/opt/mysoft/bin:$PATH`
 - 其中 `/opt/mysoft/bin` 为你自己需要设置的环境变量路径
- 使其立即生效，在终端执行 `source ~/.bashrc`

2.3 假设我要给 a.sh 加上可执行权限，该输入什么命令？

- 输入命令: `sudo chmod +x a.sh`

```
shiyancelou:~/ $ sudo chmod +x a.sh
shiyancelou:~/ $ ls -alh a.sh
-rwxrwxr-x 1 xiangxiang xiangxiang 0 10月 8 14:28 a.sh
```

注释一些内容

- 列出文件 `ls -l`

```
shianlou:~/ $ ls -l [14:32:39]
总用量 8
lrwxrwxrwx 1 root      root      14 10月  8 14:27 anaconda3 -> /opt/anaconda3
-rw-rw-r-- 1 shianlou shianlou   0 10月  8 14:28 a.sh
drwxr-xr-x 2 shianlou shianlou 4096 1月  6 2021 Code
drwxr-xr-x 2 shianlou shianlou 4096 1月  6 2021 Desktop
```

- 上图的解释

文件类型和权限: drwxr-xr-x 链接数: 2 所有者: shianlou 所属用户组: shianlou 文件大小: 4096 最后修改时间: 11月 13 15:27 文件名: Documents

- 关于权限

'r' 表示允许读权限
'w' 表示允许写权限
'x' 表示允许执行权限

表示文件类型
'd', 目录
'l', 软链接
'b', 块设备
'c', 字符设备
's', socket
'p', 管道
'-', 普通文件

拥有者权限 其他用户权限
drwxr-xr-x
所属用户组权限

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

- 首先创建 a.sh 这个文件, 并查看当前的权限信息以及拥有者和所属用户, 这里是 shianlou

```
shianlou:~/ $ touch a.sh
shianlou:~/ $ ls
anaconda3 a.sh Code Desktop
shianlou:~/ $ ls -al a.sh
-rw-rw-r-- 1 shianlou shianlou 0 10月  8 14:28 a.sh
shianlou:~/ $ ls -alh a.sh
-rw-rw-r-- 1 shianlou shianlou 0 10月  8 14:28 a.sh
```

- 创建一个 xiangxiang 用户

```
shianlou:~/ $ ls /home
labex project shianlou
shianlou:~/ $ sudo adduser xiangxiang
正在添加用户"xiangxiang"...
正在添加新组"xiangxiang" (1000)...
正在添加新用户"xiangxiang" (1000) 到组"xiangxiang"...
创建主目录"/home/xiangxiang"...
正在从"/etc/skel"复制文件...
输入新的 UNIX 密码：
重新输入新的 UNIX 密码：
passwd：已成功更新密码
正在改变 xiangxiang 的用户信息
请输入新值，或直接敲回车键以使用默认值
    全名 []:
    房间号码 []:
    工作电话 []:
    家庭电话 []:
    其它 []:
这些信息是否正确？ [Y/n]
shianlou:~/ $ ls /home
labex project shianlou xiangxiang
shianlou:~/ $
```

- 更改文件拥有者 `chown`

```
shianlou:~/ $ sudo chown xiangxiang a.sh
shianlou:~/ $ ls -alh a.sh
-rw-rw-r-- 1 xiangxiang shianlou 0 10月 8 14:28 a.sh
```

- 更改文件的用户组 `chgrp`

```
shianlou:~/ $ sudo chgrp xiangxiang a.sh
shianlou:~/ $ ls -alh a.sh
-rw-rw-r-- 1 xiangxiang xiangxiang 0 10月 8 14:28 a.sh
```

3. SLAM 综述文献阅读 (3pt, 3h)

3.1 SLAM 会在哪些场合中用到？至少列举三个方向。

- 增强现实

- 自动驾驶
- 救援任务

3.2 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？

- For being precisely localized in an environment, a correct map is necessary, but in order to construct a good map it is necessary to be properly localized when elements are added to the map.
- 为了在环境中精确定位，需要一个正确的建图，但为了构建一个好的地图，在向地图添加元素时，需要正确的定位。

3.3 SLAM 发展历史如何？我们可以将它划分成哪几个阶段？

- classical age (1986-2004)
 - the introduction of the main probabilistic formulations for SLAM
- algorithmic-analysis age (2004-2015)
 - the study of fundamental properties of SLAM
- nowadays

3.4 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] L. Haomin, Z. Guofeng, and B. Hujun, "A survey of monocular simultaneous localization and mapping," Journal of Computer-Aided Design and Compute Graphics, vol. 28, no. 6, pp. 855–868, 2016. in Chinese.

4. CMake 练习 (2pt, 1.5h)

4.1 include/hello.h 和 src/hello.c 构成了 libhello.so 库。hello.c 中提供一个函数 sayHello(), 调用此函数时往屏幕输出一行"HelloSLAM"。我们已经为你准备了 hello.h 和 hello.c 这两个文件，见"code/"目录下。

4.2 文件 useHello.c 中含有一个 main 函数，它可以编译成一个可执行文件，名为"sayhello"。

4.3 默认用 Release 模式编译这个工程。

4.4 如果用户使用 sudo make install，那么将 hello.h 放至/usr/local/include/下，将 libhello.so 放至/usr/local/lib/下。

请按照上述要求组织源代码文件，并书写 CMakeLists.txt。

文件树

更改了文件后缀名为 .cpp

```

├─ CMakeLists.txt
├─ include
│   └─ hello.h
├─ Release
├─ src
│   └─ hello.cpp
└─ useHello.cpp

```

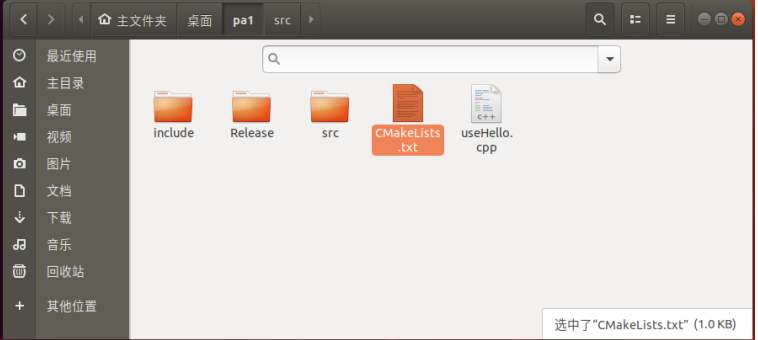
```

#CMakeLists.txt
cmake_minimum_required(VERSION 2.7)
project(hello_slam)
#####
# Create a library
#####
#根据hello.cpp生成动态库
add_library(libhello SHARED
    src/hello.cpp
)
#为这个库目标，添加头文件路径，PUBLIC表示包含了这个库的目标也会包含这个路径
target_include_directories(libhello
    PUBLIC
        ${PROJECT_SOURCE_DIR}/include
)
#####
# Create an executable
#####
#根据main.cpp生成可执行文件
add_executable(sayhello
    useHello.cpp
)
#链接库和可执行文件，使用的是这个库的别名。PRIVATE 表示
target_link_libraries(sayhello
    PRIVATE
        libhello
)
#####
#set release
#####
SET(CMAKE_BUILD_TYPE "Release")

```

执行截图

```
pzlu@pzlu-virtual-machine:~/桌面/pa1/Release$ ls
pzlu@pzlu-virtual-machine:~/桌面/pa1/Release$ cmake ..
-- The C compiler identification is GNU 7.5.0
-- The CXX compiler identification is GNU 7.5.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /home/pzlu/桌面/pa1/Release
pzlu@pzlu-virtual-machine:~/桌面/pa1/Release$ make
Scanning dependencies of target libhello
[ 25%] Building CXX object CMakeFiles/libhello.dir/src/hello.cpp.o
[ 50%] Linking CXX shared library liblibhello.so
[ 50%] Built target libhello
Scanning dependencies of target sayhello
[ 75%] Building CXX object CMakeFiles/sayhello.dir/useHello.cpp.o
[100%] Linking CXX executable sayhello
[100%] Built target sayhello
pzlu@pzlu-virtual-machine:~/桌面/pa1/Release$ ./sayhello
Hello SLAM
pzlu@pzlu-virtual-machine:~/桌面/pa1/Release$
```



5. 理解 ORB-SLAM2 框架 (3pt, 2h)

6. *使用摄像头或视频运行 ORB-SLAM2 (3pt, 1h)
