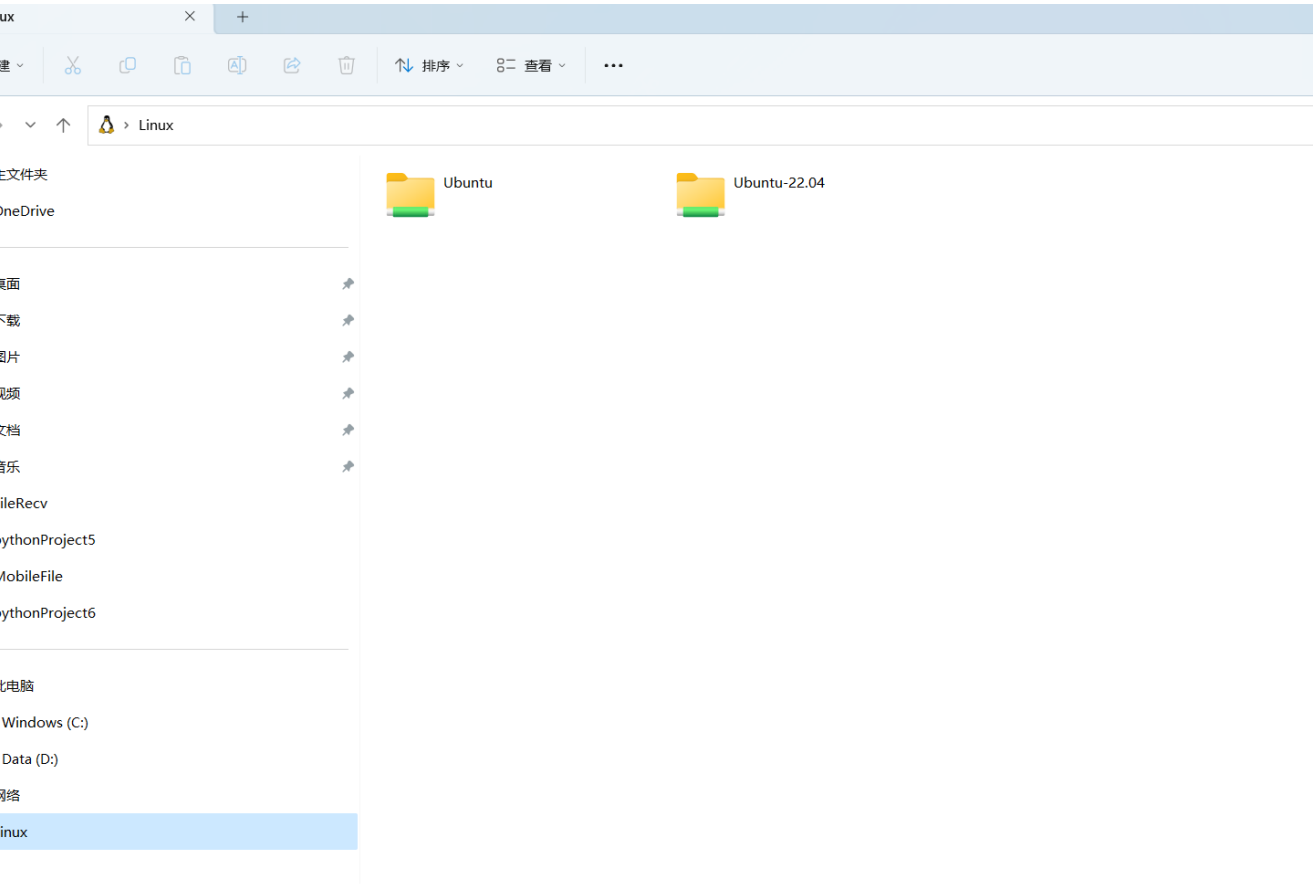


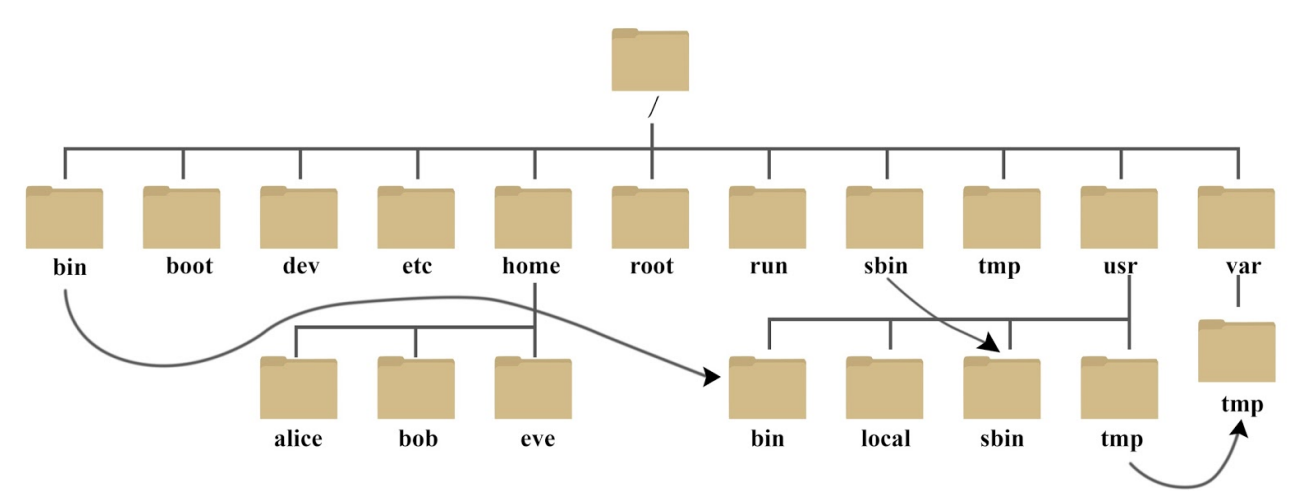
Lesson1-Linux环境与CMake

任务一：



我使用的是wsl2，拥有了自己的Linux系统

任务二：了解了解linux文件系统



解释：

- **/bin:**
bin 是 Binaries (二进制文件) 的缩写, 这个目录存放着最经常使用的命令。

最常用的命令位于/bin目录下

任务三：linux命令

进入主目录--创建test文件夹--cd进入该文件夹

```
root@honor:/home/qmy# cd /.
root@honor:/# mkdir test
root@honor:/# cd test
```

使用touch创建cpp--vim编辑cpp

```
include<iostream>
using namespace std;

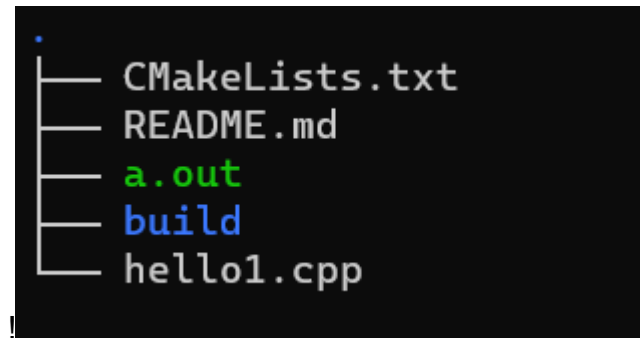
int main()
{
    cout<<"hello,world!"<<endl;
    return 0;
}
```

g++编译生成out文件--输出

```
root@honor:/test# vim hello1.cpp
root@honor:/test# g++ hello1.cpp
root@honor:/test# ls
a.out hello1.cpp
root@honor:/test# ./a.out
bash: ./a.: No such file or directory
root@honor:/test# ./a.out
hello,world!
root@honor:/test# |
```

任务四：CMake 简单实践

文件树如下：



操作步骤：

1：创建目录

2：编写CMakeList.txt：

```
cmake_minimum_required(VERSION 2.8)
project(test)
add_executable(helloworld hello1.cpp)
```

3：构建项目

进入build目录--运行CMake：cmake..(指向上层的CMakeLists.txt)--make命令来编译项目

4：执行项目

```
./helloworld
```

```
root@honor:/test/build# make
[ 50%] Building CXX object CMakeFiles/helloworld/hello1.o
[100%] Linking CXX executable helloworld
[100%] Built target helloworld
root@honor:/test/build# ./helloworld
hello,world!
```

任务五：使用编译安装OpenCV

```
root@honor: /digital/build x + v
CMake.

Update the VERSION argument <min> value or use a ...<max> suffix to tell
CMake that the project does not need compatibility with older versions.

-- Configuring done
-- Generating done
-- Build files have been
Consolidate compiler generated dependencies of target demo
[100%] Built target demo
root@honor: /digital/build
cmake ..
make
./demo # 执行
bash: cd: build: No such file or directory
CMake Deprecation Warning at CMakeLists.txt:12 (cmake_minimum_required):
Compatibility with CMake < 2.8.12 will be removed from a future version of
CMake.

Update the VERSION argument <min> value or use a ...<max> suffix to tell
CMake that the project does not need compatibility with older versions.

-- Configuring done
-- Generating done
-- Build files have been written to: /digital/build
Consolidate compiler generated dependencies of target demo
[100%] Built target demo
```

测试---安装成功

任务六：CMake实践

补全如下：

```
cmake_minimum_required(VERSION 2.8)
project(segmentation)
set(CMAKE_CXX_STANDARD 11)
find_package(OpenCV REQUIRED)
include_directories(${CMAKE_SOURCE_DIR}/include)
include_directories( ${OpenCV_INCLUDE_DIRS} )
set(SOURCE_FILES src/main.cpp src/segment.cpp )
add_executable(main src/main.cpp src/segment.cpp)
target_link_libraries(main ${OpenCV_LIBS})
```

The screenshot displays a Linux terminal window with the following content:

```

root@honor:/ceshi/segment# cd build
root@honor:/ceshi/segment/build# cmake ..
CMake Deprecation Warning at CMakeLists.txt:1 (cmake_minimum_required):
Compatibility with CMake < 2.8.12 will be removed from a future version of
this software.
-- The CXX compiler identification is GNU 7.5.0
-- Detecting CXX compiler ABI info
-- Detecting CXX compile features
-- Detecting CXX compile features
-- Configuring done
-- Generating done
-- Build files have been written to: /home/qmy/build/build
root@honor:/ceshi/segment/build# make
[100%] Linking CXX executable main
[100%] Built target main
root@honor:/ceshi/segment/build# ./main
[1920 x 1080]
global /home/qmy/build/build/opencv-4.6.0/modules/highgui/src/window.cpp (697) createTrackbar UI/Trackba
r(max: @contours): Using 'value' pointer is unsafe and deprecated. Use NULL as value pointer. To fetch trackbar value se
e
[WARN:0@0.030] global /home/qmy/build/build/opencv-4.6.0/modules/highgui/src/window.cpp (697) createTrackbar UI/Trackba
r(max: @contours): Using 'value' pointer is unsafe and deprecated. Use NULL as value pointer. To fetch trackbar value se
e
tup callback.

```

The terminal output shows the successful compilation and execution of the program. The program displays a window titled 'origin' showing a video frame of a document with a yellow bounding box. The terminal also shows the output of 'ls -ls' and 'cat' commands, and the program's output, which includes a warning about deprecated CMake features and the program's execution path.

