

Assignment 1: 环境搭建与使用

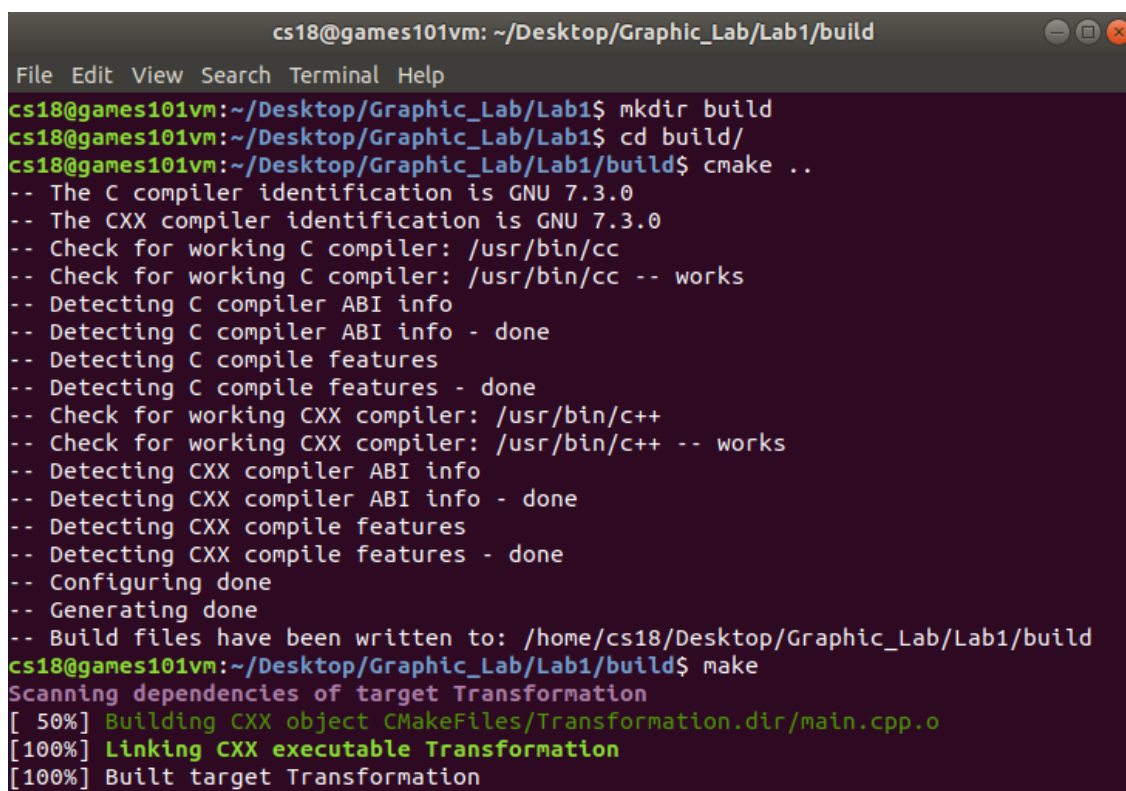
18340040 冯大纬

实验要求

为了免去配置作业所需环境的麻烦，本次课程使用虚拟机，学生在虚拟机内编写，编译和运行代码。我们提供的文件为虚拟硬盘文件，使用虚拟机挂载该文件后，就可以保证所有人的环境是统一并且完善的，不需要再手动配置环境。在安装完虚拟机后，我们需要手动安装 Guest Additions 来增强虚拟机的功能。

实验结果

1. 按照前面的教程搭建好虚拟机，并编译运行框架代码，在报告中贴出程序运行的结果。

A terminal window titled 'cs18@games101vm: ~/Desktop/Graphic_Lab/Lab1/build' showing the CMake compilation process. The user runs 'mkdir build', 'cd build/', and 'cmake ..'. The terminal output shows CMake detecting the C and CXX compilers (GNU 7.3.0), checking for working compilers, detecting compiler ABI info and features, and finally configuring and generating the build files. The user then runs 'make', and the terminal shows the progress of building the CXX object and linking the executable 'Transformation'.

```
cs18@games101vm: ~/Desktop/Graphic_Lab/Lab1/build
File Edit View Search Terminal Help
cs18@games101vm:~/Desktop/Graphic_Lab/Lab1$ mkdir build
cs18@games101vm:~/Desktop/Graphic_Lab/Lab1$ cd build/
cs18@games101vm:~/Desktop/Graphic_Lab/Lab1/build$ cmake ..
-- The C compiler identification is GNU 7.3.0
-- The CXX compiler identification is GNU 7.3.0
-- 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/cs18/Desktop/Graphic_Lab/Lab1/build
cs18@games101vm:~/Desktop/Graphic_Lab/Lab1/build$ make
Scanning dependencies of target Transformation
[ 50%] Building CXX object CMakeFiles/Transformation.dir/main.cpp.o
[100%] Linking CXX executable Transformation
[100%] Built target Transformation
```

图1 CMAKE编译过程

```

cs18@games101vm:~/Desktop/Graphic_Lab/Lab1/build$ ./Transformation
Example of cpp
1
0.5
1.41421
3.14159
0.5
Example of vector
Example of output
1
2
3
Example of add
2
2
3
Example of scalar multiply
3
6
9
2
4
6
Example of matrix
Example of output
1 2 3
4 5 6
7 8 9
cs18@games101vm:~/Desktop/Graphic_Lab/Lab1/build$

```

图2 Transformation运行结果

2. 在框架代码的基础上，了解Eigen库的向量的使用，并在代码中实现v 和w 向量点乘并输出结果，在报告中贴上结果截图。

因为原本的代码输出太多，为了避免影响观感，这里把其他的输出先注释掉了，只留下了要求输出的结果。

```

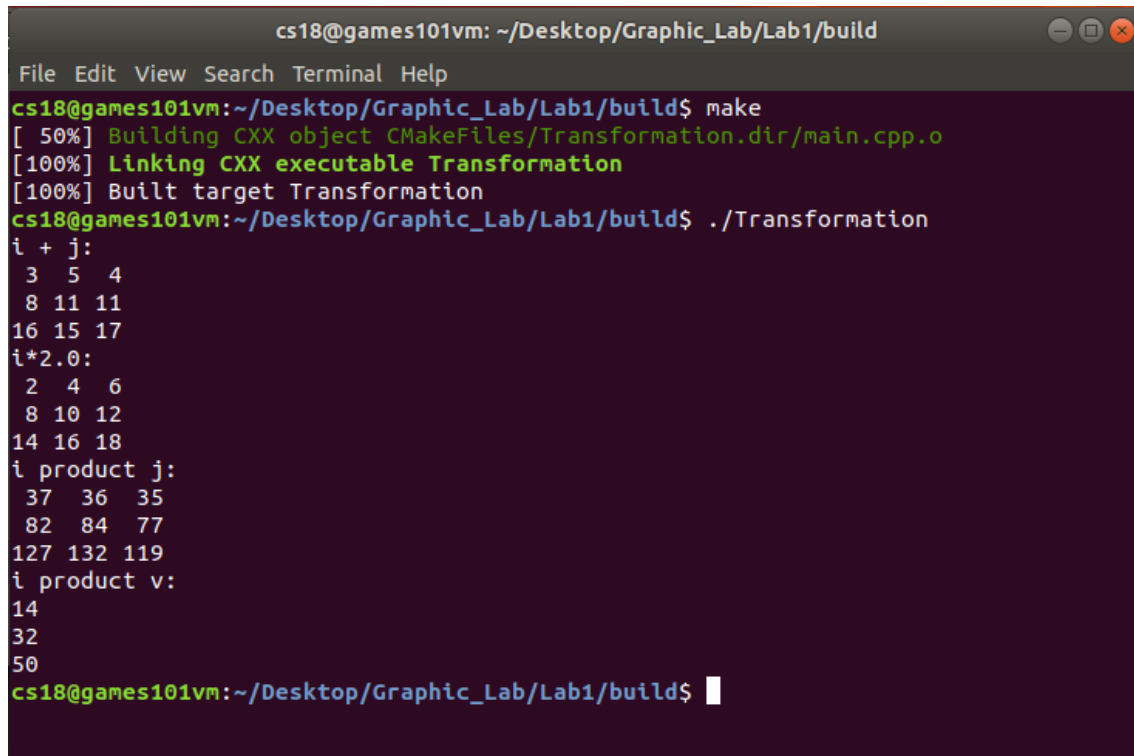
cs18@games101vm: ~/Desktop/Graphic_Lab/Lab1/build
File Edit View Search Terminal Help
cs18@games101vm:~/Desktop/Graphic_Lab/Lab1/build$ make
Scanning dependencies of target Transformation
[ 50%] Building CXX object CMakeFiles/Transformation.dir/main.cpp.o
[100%] Linking CXX executable Transformation
[100%] Built target Transformation
cs18@games101vm:~/Desktop/Graphic_Lab/Lab1/build$ ./Transformation
vector v:
1
2
3
vector w:
1
0
0
u dot product v is:
1
cs18@games101vm:~/Desktop/Graphic_Lab/Lab1/build$

```

图3 v点乘w运行结果

3. 在框架代码的基础上，了解Eigen库的矩阵的使用，并在代码中实现i 与j 的矩阵相加、i 与2.0的数乘、i 与j 的矩阵相乘、i 与v 的矩阵乘向量，并输出相应的结果，在报告中贴上结果截图。

同样为了避免冗余输出过多，将不必要的输出先全部注释掉了



```
cs18@games101vm: ~/Desktop/Graphic_Lab/Lab1/build
File Edit View Search Terminal Help
cs18@games101vm:~/Desktop/Graphic_Lab/Lab1/build$ make
[ 50%] Building CXX object CMakeFiles/Transformation.dir/main.cpp.o
[100%] Linking CXX executable Transformation
[100%] Built target Transformation
cs18@games101vm:~/Desktop/Graphic_Lab/Lab1/build$ ./Transformation
i + j:
 3  5  4
 8 11 11
16 15 17
i*2.0:
 2  4  6
 8 10 12
14 16 18
i product j:
37 36 35
82 84 77
127 132 119
i product v:
14
32
50
cs18@games101vm:~/Desktop/Graphic_Lab/Lab1/build$
```

图4 job2运行结果

实验过程

跟着pdf的教程和Eigen的文档一步步走就完成了

谈谈学习了这些教程章节感想

感觉Assignment1比Assignment0要简单好多，可能是TA把大部分工作都做掉了，只给我们留了为数不多的代码需要编写(笑)，而且代码也都很简单，矩阵的运算符Eigen库都有重载，写起来有一种在调用numpy包写Python的感觉，总而言之就是写起来很舒服。

话说清华镜像属实8太行，校园网下个文件断线四次才下载下来，建议以后改用蓝奏云。

感觉其实如果用VS把环境搭好的话好像更方便，而且VS环境配置起来也不难，就把Eigen的头文件放到项目的include文件夹里然后在解决方案的属性里把包含路径设置一下就行了，经过我一番尝试也成功运行了。

话说又回到看不到三角形的命令行界面了，不知道什么时候才能再见到和蔼的三角形。