AP1400-2作业环境配置

配置教程基于Linux 或者 是 WSL

获取作业源码，作业1为例

打开终端，使用git clone下载代码

```
git clone https://github.com/courseworks/AP1400-2-HW1
```

安装gtest

gtest 使用cmake管理项目，首先需要安装cmake，可以根据自己的Linux 发行版安装

ubuntu/debian：

```
sudo apt intall cmake make
```

fedora/centos：

```
sudo dnf in cmake make
```

archlinux/manjaro：

```
sudo pacman -S cmake make
```

opensuse：

```
sudo zypper in cmake make
```

下一步安装googletest：

> 参考链接 https://github.com/google/googletest/blob/main/googletest/README.md

clone下来gtest的仓库

```
git clone https://github.com/google/googletest.git -b release-1.12.1
cd googletest        # Main directory of the cloned repository.
mkdir build          # Create a directory to hold the build output.
cd build
cmake ..             # Generate native build scripts for GoogleTest.
```

```
make
sudo make install
```

最后安装完成：

```
> sudo make install
[sudo] root 的密码：
[ 25%] Built target gtest
[ 50%] Built target gmock
[ 75%] Built target gmock_main
[100%] Built target gtest_main
Install the project...
-- Install configuration: ""
-- Up-to-date: /usr/local/include
-- Installing: /usr/local/include/gmock
-- Installing: /usr/local/include/gmock/gmock-spec-builders.h
-- Installing: /usr/local/include/gmock/gmock-cardinalities.h
-- Installing: /usr/local/include/gmock/gmock-function-mocker.h
-- Installing: /usr/local/include/gmock/gmock-more-actions.h
-- Installing: /usr/local/include/gmock/gmock.h
-- Installing: /usr/local/include/gmock/gmock-more-matchers.h
-- Installing: /usr/local/include/gmock/gmock-nice-strict.h
-- Installing: /usr/local/include/gmock/gmock-matchers.h
-- Installing: /usr/local/include/gmock/internal
-- Installing: /usr/local/include/gmock/internal/custom
```

完成代码并测试

安装好gtest之后，用vscode打开应该就不会报找不到头文件的错误了

```
src > main.cpp > main(int, char **)

#include <iostream>
#include <gtest/gtest.h>
#include "hw1.h"
```

当写好代码之后，或者是写好部分函数想要测试的时候，可以在main.cpp中将测试模式打开

```
7    7 {
8          if (true) // make false to run unit-tests
       8   if (false) // make false to run unit-tests
9    9     {
```

然后在`unit_test.cpp`中取消注释掉你想测试的部分

接着对源码进行编译和测试

编译：

```
mkdir build
cd build
```

```
cmake ..
make
```

编译成功：

```
AP1400-2-HW1/build on main [!?] via v3.25.1
❯ make
[ 25%] Building CXX object CMakeFiles/main.dir/src/main.cpp.o
[ 50%] Building CXX object CMakeFiles/main.dir/src/hw1.cpp.o
[ 75%] Building CXX object CMakeFiles/main.dir/src/unit_test.cpp.o
[100%] Linking CXX executable main
[100%] Built target main
```

运行测试：在`build`文件夹内执行

```
./main
```

测试结果如下：

```
[----------] Global test environment tear-down
[==========] 24 tests from 1 test suite ran. (1 ms total)
[  PASSED  ] 24 tests.
<<<SUCCESS>>>
```