

cmake(二)构建Hello World简单工程

原创wzj_110于 2021-04-14 00:37:42 发布阅读量943收藏3点赞数2

分类专栏:cmake DSL语言

版权

cmake DSL语言 专栏收录该内容

38 篇文章

已订阅

摘要 本文是一篇关于CMake的基础教程，涵盖了CMakeLists.txt的使用，包括project、set和add_executable函数的解读。通过实例解析，指导读者如何在子项目下配置和构建CMake项目，并在make后进行测试。同时，文章还涉及了遇到问题时的解决策略。

摘要由CSDN通过智能技术生成

一 简单结构

① 项目宏观查看

```
kiosk@k8s test $ tree CmakeProjects/
CmakeProjects/
├── HelloCmake
│   ├── CMakeLists.txt
│   └── HelloCmake.cpp
```

1 directory, 2 files

编辑CMakeLists.txt文件，这个文件是CMake构建定义文件，名字必须是这个名字，大小写也要一致，如果项目有多个目录，则每个目录下都要有一个CMakeLists.txt文件

② HelloCmake.cpp

```
1 #include <iostream>
2 #include <stdlib.h>
3 int main(int argc, char** argv) {
4     std::cout << "Hello CMake!" << std::endl;
5     return EXIT_SUCCESS;
6 }
```

"HelloCmake.cpp" 6 lines --33%--

③ CMakeLists.txt

```
1 project(HelloCmake)
2 set(SRC_LIST HelloCmake.cpp)
3 add_executable(hello_cmake ${SRC_LIST})
```

"CMakeLists.txt" 3L, 89C

1) project函数解读

project函数，用于定义项目名称，
这里定义项目名称为HelloCMake，
CMake里函数是不分大小写的，
写成PROJECT也是可以的

https://blog.csdn.net/wzj_110

2) set函数解读

set函数，用于定义变量并赋值，
这里定义一个SRC_LIST变量，
并且赋值为HelloCMake.cpp

3) add_executable函数解读

add_executable函数，
用于添加一个可执行的目标，
第一个参数是生成的目标文件名
(即可执行的程序名)，
第二个参数是依赖的源文件，
这里传入了SRC_LIST，
引用变量的内容要用\${VAR_NAME}这样的语法

https://blog.csdn.net/wzj_110

④ cmake下

备注：注意当前'所在'位置 --> '子项目'下

```
kiosk@k8s HelloCmake $ ls
CMakeLists.txt  HelloCmake.cpp
kiosk@k8s HelloCmake $ cmake3
-- The C compiler identification is GNU 4.8.5
-- The CXX compiler identification is GNU 4.8.5
-- 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: /var/ftp/pub/pub/cmake/test/CmakeProjects/HelloCmake
kiosk@k8s HelloCmake $ echo $?
0
kiosk@k8s HelloCmake $ ls
CMakeCache.txt  CMakeFiles  cmake_install.cmake  CMakeLists.txt  HelloCmake.cpp  Makefile
```

执行完cmake命令后，没有报错的情况下，
除了生成了Makefile，还生成了一些其它的文件，
其它的文件可以先不管它

⑤ make后进行测试

make是一个'命令'工具，是一个'解释makefile'中指令的'命令工具'

有了Makefile，就可以执行make命令来生成程序了，最终生成的程序名称为在CMakeLists.txt里指定的名称，即hello_cmake

```
kiosk@k8s HelloCmake $ make
Scanning dependencies of target hello_cmake
[100%] Building CXX object CMakeFiles/hello_cmake.dir/HelloCmake.cpp.o
Linking CXX executable hello_cmake
[100%] Built target hello_cmake
kiosk@k8s HelloCmake $ echo $?
0
kiosk@k8s HelloCmake $ ls
CMakeCache.txt  CMakeFiles  cmake_install.cmake  CMakeLists.txt  hello_cmake  HelloCmake.cpp  Makefile
kiosk@k8s HelloCmake $ ./hello_cmake
Hello CMake!
kiosk@k8s HelloCmake $
```

make之后才会生成可执行的二进制文件

可执行文件的名字是在CMakeLists中定义的

⑥ 报错解决

```
kiosk@k8s HelloCmake $ make
Scanning dependencies of target hello_cmake
[ 50%] Building CXX object CMakeFiles/hello_cmake.dir/HelloCmake.cpp.o
/var/ftp/pub/pub/cmake/test/CmakeProjects/HelloCmake/HelloCmake.cpp: In function 'int main(int, char**)':
/var/ftp/pub/pub/cmake/test/CmakeProjects/HelloCmake/HelloCmake.cpp:4:12: error: 'EXIT_SUCCESS' was not declared in this scope
    return EXIT_SUCCESS;
           ^
make[2]: *** [CMakeFiles/hello_cmake.dir/HelloCmake.cpp.o] Error 1
make[1]: *** [CMakeFiles/hello_cmake.dir/all] Error 2
make: *** [all] Error 2
```

显示推荐内容