

# 04 构建和链接动态库

构建和链接动态链接库

准备工作

创建Account类和main.cpp文件:

account.h 文件:

```
#ifndef __ACCOUNT__
#define __ACCOUNT__

#include <iostream>
#include <string>

using namespace std;

class Account
{
private:
    string name;
    unsigned long nr;
    double balance;
public:
    bool init(const string&, unsigned long, double);
    void display();
};

#endif
```

account.cpp 文件:

```
#include "account.h"

#include <iomanip>

bool Account::init(const string& i_name, unsigned long i_nr, double
i_balance)

{
    if (i_name.size() < 1)
    {
        return false;
    }

    name = i_name;

    nr = i_nr;

    balance = i_balance;

    return true;
}

void Account::display()

{
    cout << fixed << setprecision(2)

        << "-----\n"

        << "Account holder:      " << name << '\n'

        << "Account number:      " << nr << '\n'

        << "Account balance:      " << balance << '\n'

        << "-----\n"

        << endl;
}
```

main.cpp 文件:

```
#include "account.h"

int main()
```

```

{
    Account current1, current2;

    current1.init("Cheers, Mary", 1234567, -1200.99);

    current1.display();

    current2 = current1;

    current2.display();

    current2.init("Jones, Tom", 3512347, 199.40);

    current2.display();

    Account& mtr = current1;

    mtr.display();

    return 0;
}

```

具体实施

在根目录下创建CMakeLists.txt文件：

CMakeLists.txt：

```

# 所需的最低cmake版本

cmake_minimum_required(

    VERSION

    3.5

    FATAL_ERROR

)

# 定义工程名为Account

project(

    Account

    LANGUAGES

    CXX

)

# 使用account.h, account.cpp源文件, 创建一个对象库account-objs

```

```
add_library(
    account-objs
    OBJECT
    account.h
    account.cpp
)

# 设置编译的对象库account-objs与生成位置无关。（兼容老编译器）

set_target_properties(
    account-objs
    PROPERTIES
    POSITION_INDEPENDENT_CODE 1
)

# 使用account-objs对象库，创建一个动态链接库目标account-shared

add_library(
    account-shared
    SHARED
    ${TARGET_OBJECTS:account-objs} #对象库生成器表达式
)

# 使用account-objs对象库，创建一个静态链接库目标account-static

add_library(
    account-static
    STATIC
    ${TARGET_OBJECTS:account-objs} #对象库生成器表达式
)

# 通过编译和链接main.cpp源文件，来生一个可执行文件Account

add_executable(
    Account
```

```
main.cpp

)

# 将静态连接库account-static链接到可执行目标Account上

target_link_libraries(

    Account

    account-static

)
```

### 对项目进行配置和构建

1. 打开终端cd到项目所在的根目录。
2. 输入：cmake -H. -Bbuild 并执行。
3. 输入：cd ./build并执行。
4. 输入：cmake --build .并执行。
5. 在build目录中即可看到生成的可执行文件Account
6. 输入：./Account并执行即可在终端看到程序处理的结果。