

## python多文件编程



非我梦想

关注他

14 人赞同了该文章

平时我们练习编程都是在一个文件里面写代码，但是随着代码量的增加，在一个文件里面查看、修改代码非常不容易，我们查看那些经典的源代码，可以发现都是写在不同的文件里面，可以说，学会多文件编程，就可以开始编写大工程了，编程开始上升一个台阶。

下面我们以python语言为例，开始写一个多文件的例子：

我们在一个文件里面新建2个文件：test.py,main.py

其中test.py里面定义一个class类，然后在main.py里面调用test.py里面定义的方法和类

```
#test.py文件
class student:
    def __init__(self,name,age):
        self.name=name
        self.age=age
    def print1(self):
        print("我的名字是%s,我的年龄是%d"%(self.name,self.age))

# main1.py文件
import test      #首行必须要先导入test,即文件名
a = test.student("李华",18) #调用test文件里面的方法和类，前面必须要先加上那个文件名
a.print1()
```

然后用命令python main1.py 就可以看到程序输出的结果。

上面两个文件都在同一个文件夹里面，如果两个文件不在同一个文件夹里面，那么一个Python程序调用另一个Python里面的方法和类，必须要先导入那个类，即找到那个文件的路劲。

还是以上面两个文件为例，假设其中test.py在python的文件夹里面，mian.py在另一个文件夹里面。例子如下：

```
# main.py 文件
import sys
sys.path.append('/home/ming/python') #导入test文件的绝对路径
import test
a= test.student("李华",18)
a.print1()
```

这样就可以调用不同文件夹里面的python代码。

这些就是python的多文件编程

作为对比，下面还写了有关c的多文件编程：

```
//test.h文件

typedef struct student
{
    char* name;
    int age;
}student;
```



```
//test.c文件

#include<stdio.h>
#include<stdlib.h>
#include "test.h"
void print1(student* a){
    printf("我的名字是%s,我的年龄是%d",a->name,a->age)

}

//main1.c

#include<stdio.h>
#include<stdlib.h>
#include "test.h"
int main(){
    student* a=(student*)malloc(sizeof(student));
    a->name="李华";
    a->age=18;
}
```

编写好上面三个文件，然后用命令 gcc main1.c test.c

就会生成一个a.out可执行文件，然后执行./a.out就会输出程序结果。

说明: gcc main1.c test.c 其实是先分别编译main1.c和test.c文件，然后将两个文件链接成一个可执行文件，这里面涉及到编译原理知识：即预处理，编译，汇编，链接等过程。

其中链接的作用是为了main方法找到其他文件里面定义的函数位置（即函数的内存地址，一开始编译main里面的函数方法时，编译器不知道其他函数里面的函数地址，于是随意给一个地址，链接后，就可以找到函数的正确地址，这时候将之前随意给的地址替换为正确地址，这样就可以执行其他文件里面的函数）。

可以发现c语言的多文件编程与python大同小异,c语言的多文件编程还涉及到编译原理等复杂知识点，还是有一定难度的。

不知不觉就写了这么多，暂时就写这么多，拜拜~

发布于 2020-12-16 19:45

Python



评论千万条，友善第一条



还没有评论，发表第一个评论吧



文章被以下专栏收录



编程语言学习

写一些有关编程方面的知识，大家一起交流学习。

推荐阅读



错过这15个顶级Python库，你就不算Python程序员

数据黑客



一文讲透：如何用Python编写带界面的HFSS小工具，并打...

况泽灵

发表于天线设计与...



只需一行代码，就能导入所有的Python库？

Fan

发表于Pytho...



Pytho与文件

Jerry...