

Linux下C/C++开发

教师: 李军辉

第一个C程序

```
#include <stdio.h>

int main() {
    printf("Hello , world!\n");
    return 0;
}
```

第一个C程序

```
#include <stdio.h>

int main() {
    printf("Hello , world!\n");
    return 0;
}
```

```
gcc -o helloworld helloworld.c
```

其中，-o 选项指定可执行程序名，helloworld.c 是源码文件
将生成helloworld的可执行程序。

第一个C程序

```
#include <stdio.h>

int main() {
    printf("Hello , world!\n");
    return 0;
}
```

```
gcc -o helloworld helloworld.c
```

其中，-o 选项指定可执行程序名，helloworld.c 是源码文件
将生成helloworld的可执行程序。

```
./helloworld
```

第一个C++程序

```
#include <iostream>

using std::cout;
using std::endl;

int main() {
    cout << "Hello , world!" << endl;
    return 0;
}
```

第一个C++程序

```
#include <iostream>

using std::cout;
using std::endl;

int main() {
    cout << "Hello , world!" << endl;
    return 0;
}
```

```
g++ -o helloworld helloworld.cpp
```

其中，-o 选项指定可执行程序名， helloworld.cpp 是源码文件
将生成helloworld的可执行程序.

第一个C++程序

```
#include <iostream>

using std::cout;
using std::endl;

int main() {
    cout << "Hello , world!" << endl;
    return 0;
}
```

```
g++ -o helloworld helloworld.cpp
```

其中，-o 选项指定可执行程序名，helloworld.cpp 是源码文件
将生成helloworld的可执行程序.

```
./helloworld
```

多个文件

say.c

```
#include <stdio.h>

void say_hello() {
    printf("Hello , world!\n");
}
```

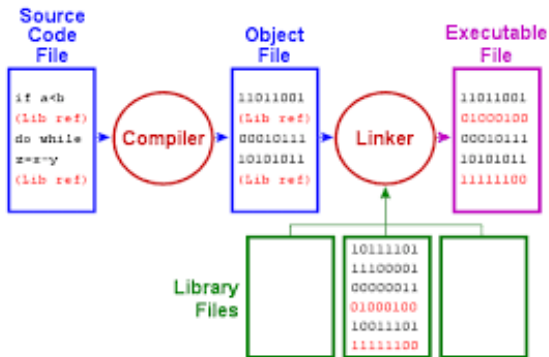
helloworld.c

```
void say_hello();

int main() {
    say_hello();
    return 0;
}
```


多个文件

```
gcc -o helloworld helloworld.c say.c
```



多个文件

say.h

```
void say_hello();
```

say.c

```
#include <stdio.h>

void say_hello() {
    printf("Hello , world!\n");
}
```

多个文件

helloworld.c

```
#include "say.h"

int main() {
    say_hello();
    return 0;
}
```

多个文件

```
gcc -c say.c
```

编译say.c文件, 将生成say.o (目标文件)

多个文件

```
gcc -c say.c
```

编译say.c文件, 将生成say.o (目标文件)

```
gcc -c helloworld.c
```

编译helloworld.c文件, 将生成helloworld.o (目标文件)

多个文件

```
gcc -c say.c
```

编译say.c文件, 将生成say.o (目标文件)

```
gcc -c helloworld.c
```

编译helloworld.c文件, 将生成helloworld.o (目标文件)

```
gcc -o helloworld helloworld.o say.o
```

链接俩对象文件, 生成可执行程序hellowrld

分阶段编译的好处

进行部分编译 — 只编译有变更部分 (大型程序项目中效果特别明显)

- https://learn-linux.readthedocs.io/zh_CN/latest/development-c-cpp.html
- <https://wiki.ubuntu.org.cn/>跟我一起写 Makefile:MakeFile 介绍
- <https://github.com/yuanzhaoming/osc/tree/master/OSC-C>