编程环境与基本技能

1. 编译、链接

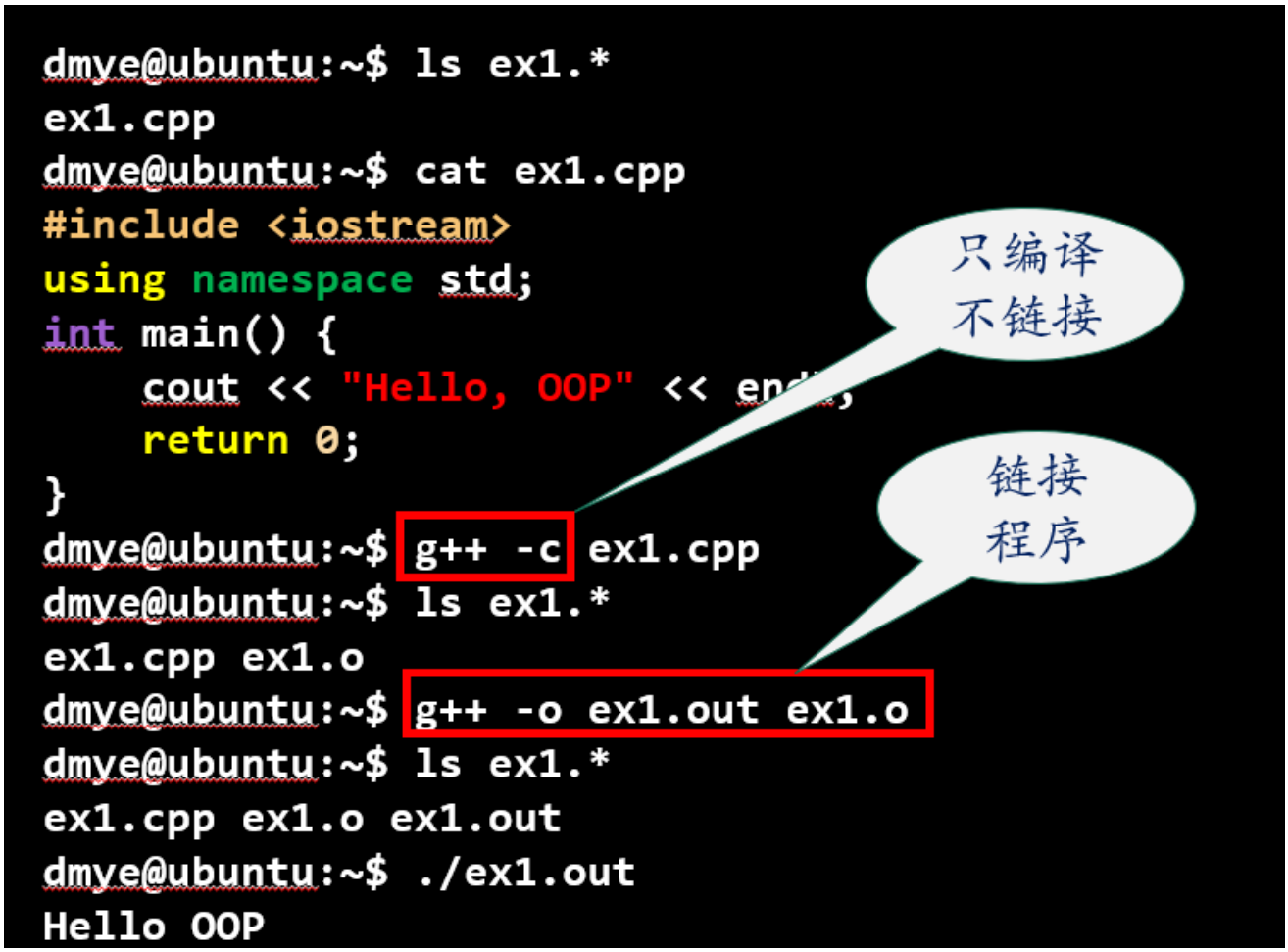
编译：

- **第一遍**执行语法分析和静态类型检查，将源代码解析为**语法分析树**的结构
- **第二遍**由代码生成器遍历语法分析树，把树的每个节点转换为**汇编语言**或机器代码，生成目标模块(.o或.obj文件)

链接：

- 把一组目标模块连接为**可执行程序**，使得操作系统可以执行它
- 处理目标模块中的函数或变量引用，必要时搜索**库文件**处理所有的**引用**

C语言的编译链接过程要把我们编写的一个c程序（源代码）转换成可以在硬件上运行的程序（可执行代码），需要进行编译和链接。编译就是把文本形式源代码翻译为机器语言形式的目标文件的过程。链接是把目标文件、操作系统的启动代码和用到的库文件进行组织形成最终生成可加载、可执行代码的过程。



【知识点】C++的argv和argc

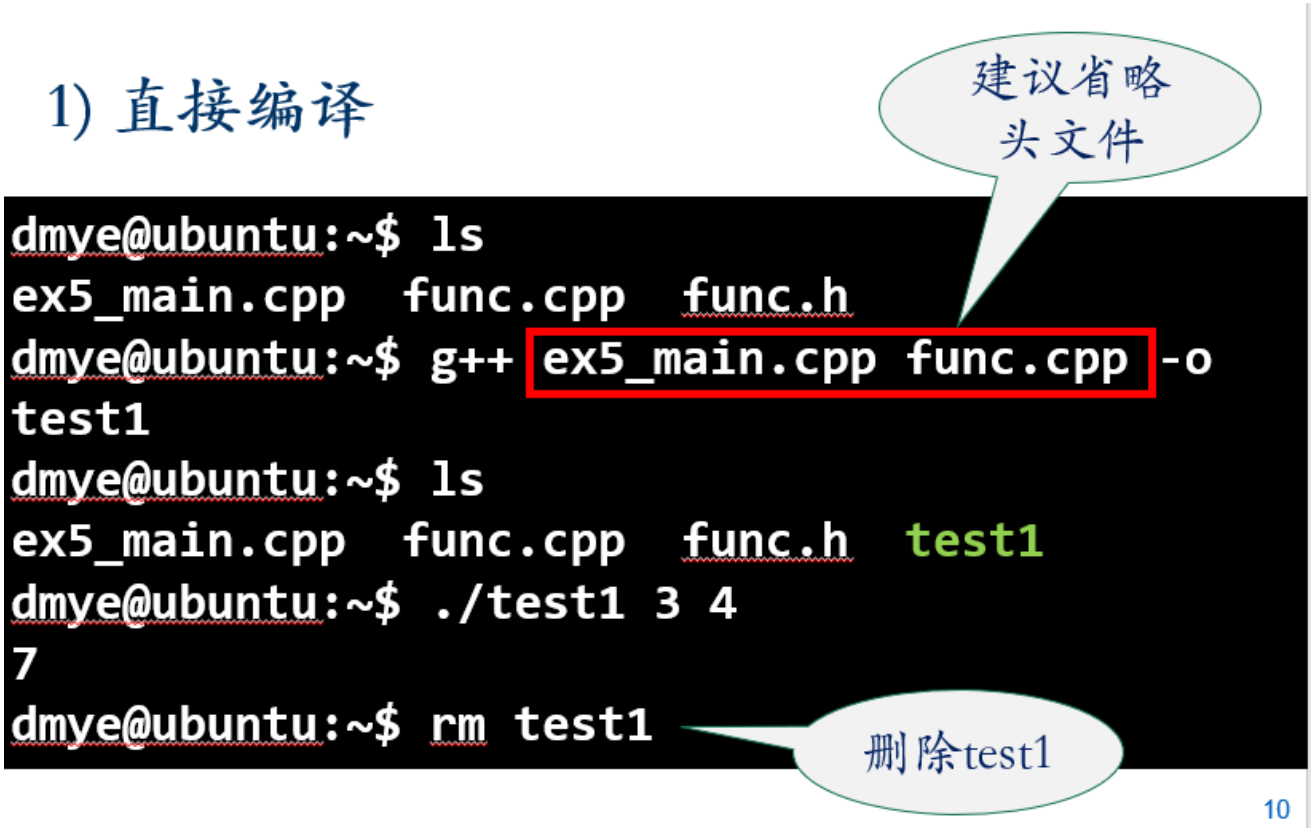
- argc 是 argument count的缩写，表示传入main函数的参数个数；

- argv 是 argument vector的缩写，表示传入main函数的参数序列或指针，并且第一个参数argv[0]一定是**程序的名称**，并且包含了程序所在的完整路径，所以确切的说需要我们输入的main函数的参数个数应该是argc-1个

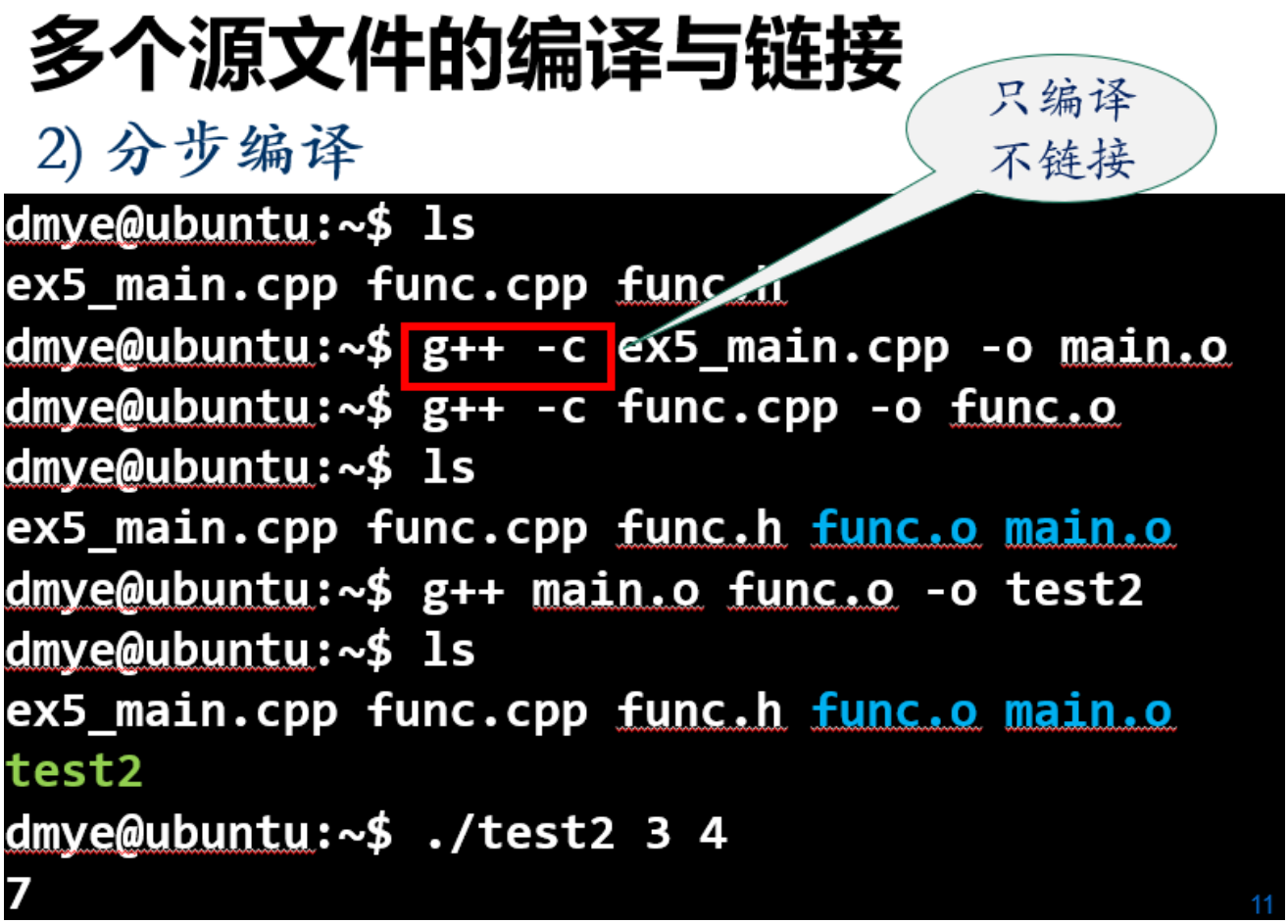
```
int main(int argc, char** argv) {
    if (argc != 3)  {
        std::cout << "Usage: " << argv[0]  << " op1 op2" << std::endl;
        return 1;
    }

    int a, b;
    a = atoi(argv[1]);  b = atoi(argv[2]);
    std::cout << ADD(a, b) << std::endl;
    return 0;
}
```

多个源文件的编译与链接：



直接生成了可运行文件test1. gcc -o



先编译出main.o和func.o, 然后链接两者，生成可执行文件test2. "gcc -o"

2. 宏定义

防止头文件被重复包含的方法:

（1）#ifndef

```
#ifndef __BODYDEF_H__
#define __BODYDEF_H__
// 头文件内容
#endif
```

（2）pragma once

```
#pragma once
// 头文件内容
```

#pragma once 保证物理上的同一个文件不会被编译多次

用于Debug输出:

```
#ifdef 标识符
  程序段1
#else
  程序段2
#endif
```

例:

```
// #define DEBUG
#ifdef DEBUG
    cout << "val:" << val << endl;
#endif
```

3. 编译器

•**MinGW**: Minimalist GNU For Windows，是个精简的Windows平台C/C++、ADA及Fortran编译器

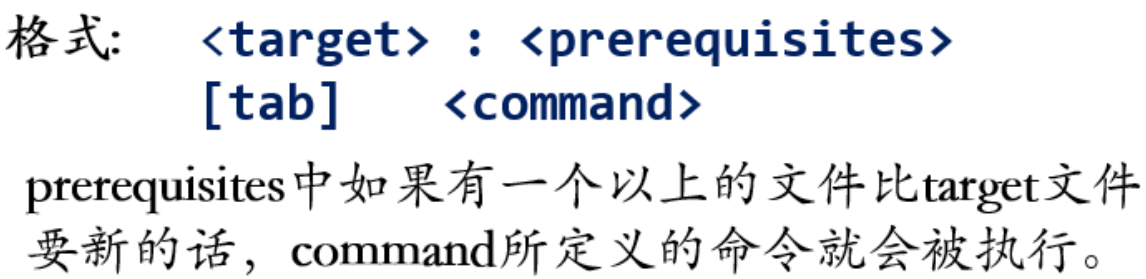
•TDM-**GCC**: Windows版的编译器套件，结合了 GCC 工具集中最新的稳定发行版本

4. Makefile

•如果工程没有编译过，那么我们的所有cpp文件都要编译并被链接。

•如果工程的某几个cpp文件被修改，那么我们只编译**被修改**的cpp文件，并链接目标程序。

•如果工程的头文件被改变了，那么我们需要编译引用了这几个头文件的cpp文件，并链接目标程序。



```
# Yao HaiLong @ 20180130
# C++ Course for THU2018
#
all: test.exe

test.exe: product.o sum.o main.o functions.h
    g++ product.o sum.o main.o -o test.exe

product.o: product.cpp functions.h
    g++ -c product.cpp -o product.o

sum.o: sum.cpp functions.h
    g++ -c sum.cpp -o sum.o

main.o: main.cpp functions.h
    g++ -c main.cpp -o main.o

clean:
    del *.o *.exe
```

```
F:\TeachingExamples\ARGCV\Ex3>make
g++ -c product.cpp -o product.o
g++ -c sum.cpp -o sum.o
g++ -c main.cpp -o main.o
g++ product.o sum.o main.o -o test.exe
```

- g++ -o：指定生成文件名称
- g++ -c：要求只编译不链接