第 15 章

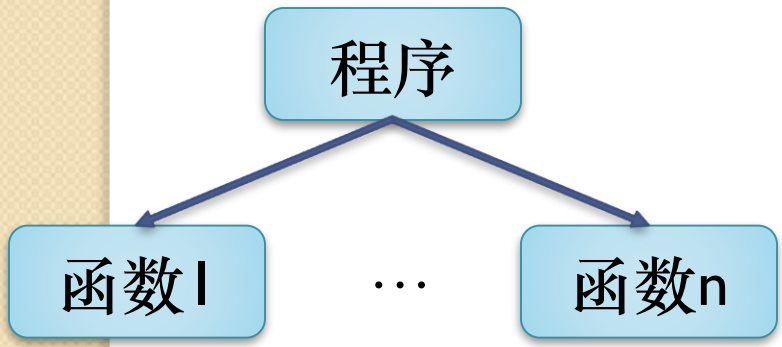
编写大型程序

程序结构

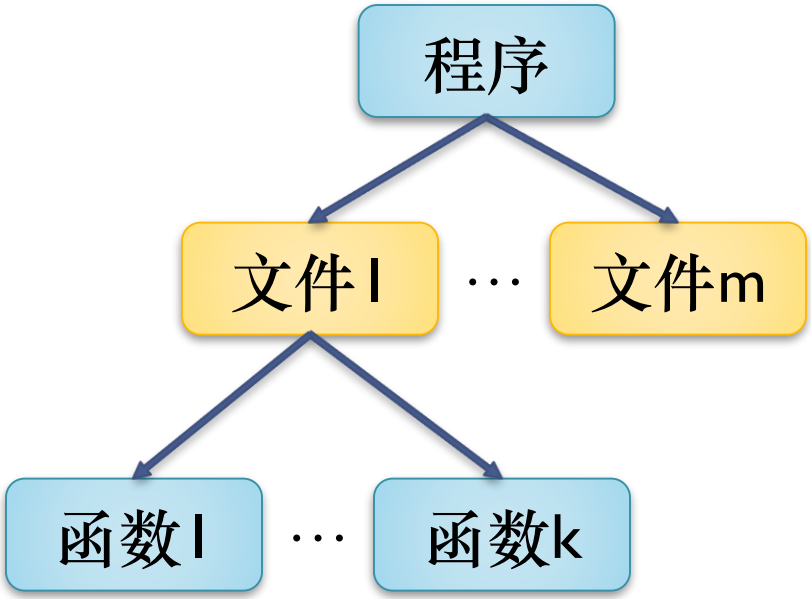
- 小程序——一个文件构成
- 常见程序由多个文件构成
  - 源文件（.c，.cpp）：函数定义和外部变量
  - 头文件（.h)：包含可在源文件之间共享的信息

程序结构化

- 函数分隔



- 文件+函数分隔



类似：磁盘、文件组织

源文件

- C程序可分成任意数量的源文件。
- 每一个源文件包含部分程序，主要是函数和变量的定义。
- 必须有一个源文件包含一个名为main的函数，它是程序的入口。

源文件

- 将一个程序分为多个源文件有重大好处：
  - 相关函数和变量放入一个文件，有助于澄清程序结构.
  - 每个源文件可以单独编译，节省时间.
  - 函数能够更利于复用到其他程序中.
  - 分工协作

头文件

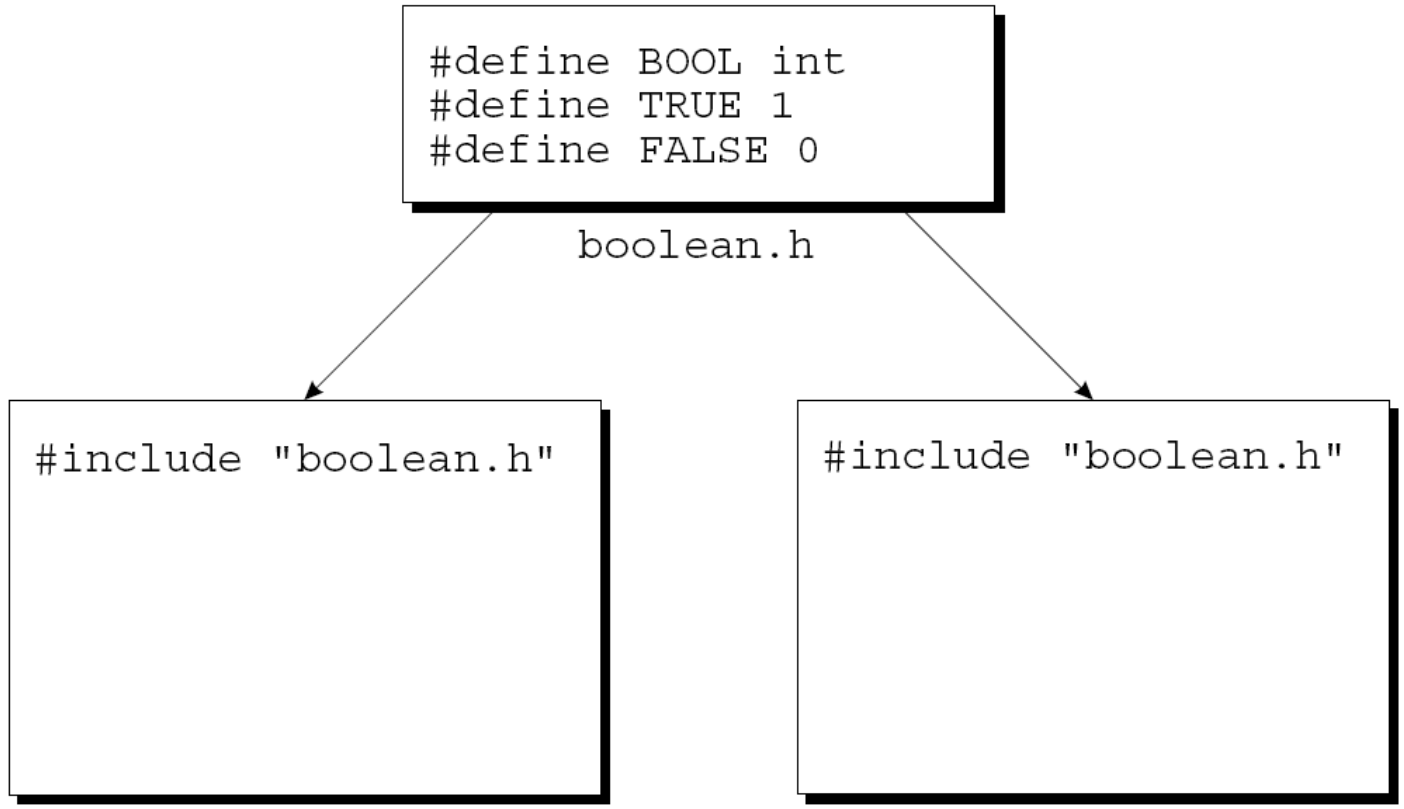
- 把程序分为几个源文件需解决如下问题：
  - 一个文件中的函数怎样调用在另一个文件中定义的函数?
  - 函数怎样访问其他文件中的外部变量?
  - 多个文件怎样共享宏定义和类型定义?
- **#include**，使多个文件共享信息
  - 多个文件需共享的信息放入头文件
    - 函数原型、宏定义、类型定义
  - 每一个源文件**#include**头文件

#include

- **#include** 有两种主要形式.
- 用于C库中的头文件:
  - **#include <filename>**
  - 查找系统头文件所在的路径.
- 用于所有的其他头文件:
  - **#include “filename”**
  - 查找当前路径，然后查找系统头文件所在的路径.

共享宏定义和类型定义

- 多个源文件共享的宏定义和类型定义应放入头文件中
- 共享的源文件**#include**之



共享函数原型

- 一个源文件调用函数f，而f定义在另一个文件foo.c中

/* h.c */使用
#include “foo.h”

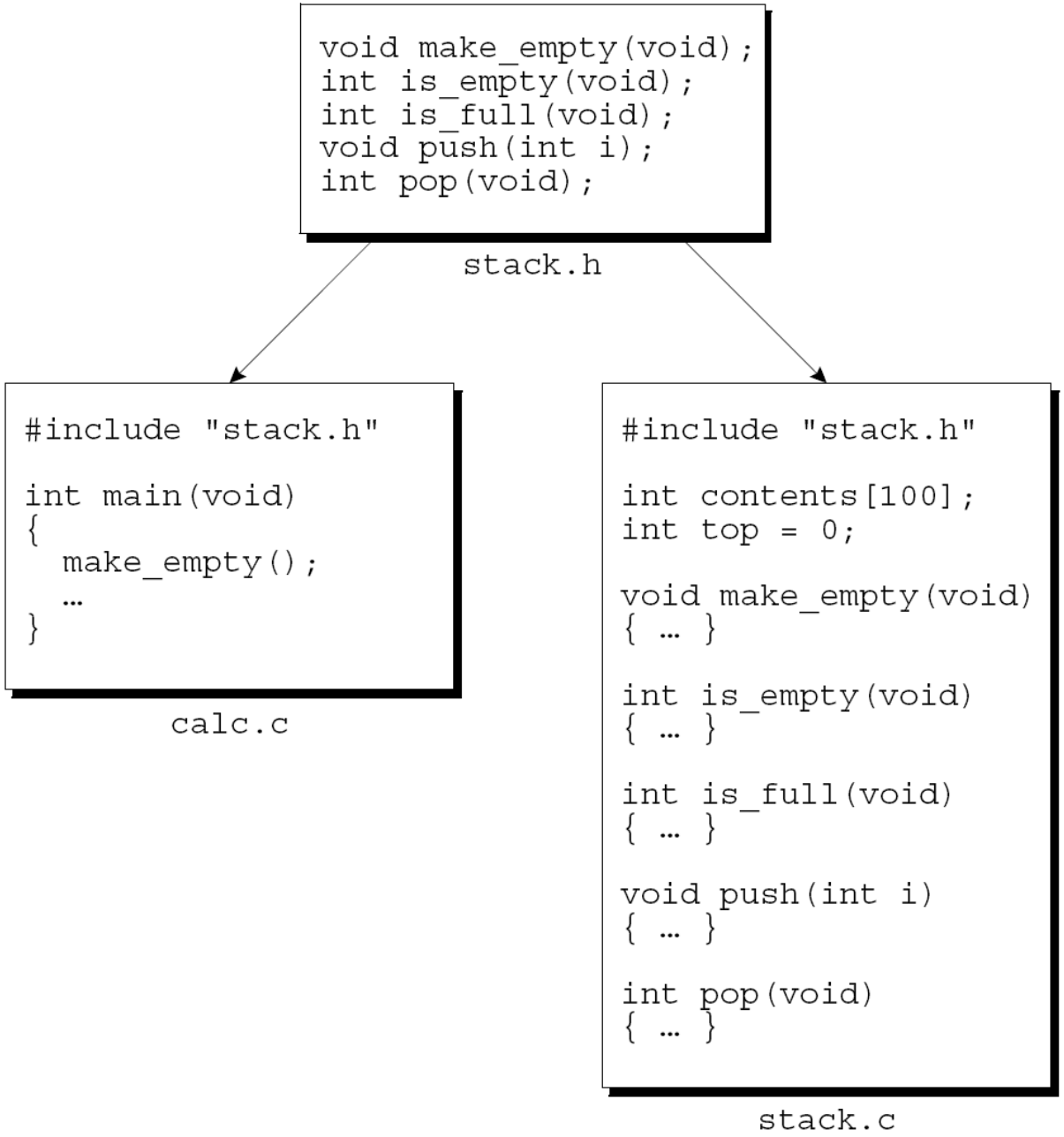
……
f(…);
……

/* foo.h */声明
void f(…);

/* foo.c */定义
#include “foo.h”

……
void f(…)
{……}

共享函数原形


void make_empty(void);
int is_empty(void);
int is_full(void);
void push(int i);
int pop(void);
stack.h
#include "stack.h"
int main(void)
{
make_empty();
…
}
calc.c
#include "stack.h"
int contents[100];
int top = 0;
void make_empty(void)
{ … }
int is_empty(void)
{ … }
int is_full(void)
{ … }
void push(int i)
{ … }
int pop(void)
{ … }
stack.c

共享变量声明

- 在几个源文件中共享外部变量，首先在一个文件里定义：
  - `int i;//声明并分配空间`
- 声明通常放在头文件里
  - `extern int i;`
  - `extern`告诉编译器变量在其他文件中定义，此处不必分配空间
- 需要访问该共享变量的文件`#include`之
- 与共享函数类似，定义变量的源文件需包含声明的头文件，便于编译器检查是否匹配
  - `#include<*.h>`
  - `int i;`
  - ……
  - `i=x;`

嵌套包含

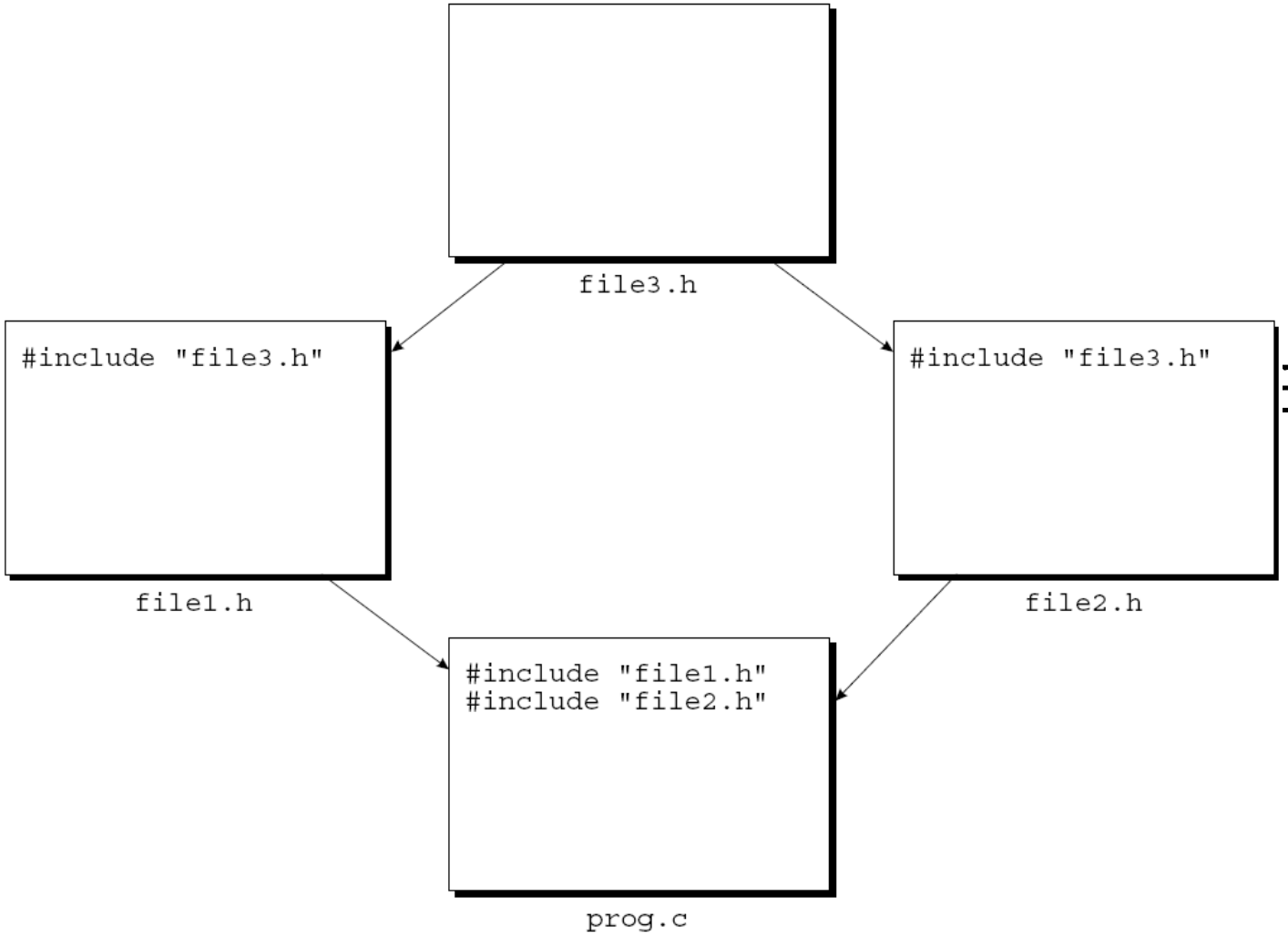
- 头文件可#include其他头文件.
- stack.h 含有下面的原型:

```
Bool is_empty(void);
Bool is_full(void);
```

- Bool定义在boolean.h中
- 需在stack.h中包含boolean.h

重复包含

- 一个源文件包含同一个头文件两次会
- 问题



保护头文件

- 把文件内容放入#ifndef-#endif中
- 比如boolean.h:

```
#ifndef BOOLEAN_H
#define BOOLEAN_H//开关宏定义
//被include一次就设置该开关
#define TRUE 1
#define FALSE 0
typedef int Bool;

#endif
```

把程序分为文件

- 设计程序
  - 确定需要哪些函数
  - 把这些函数组织成逻辑相关的组
- 将设计好的程序分成文件

把程序分为文件

- 函数归集，分别放入单独源文件，eg foo.c、stack.c
- 有共享函数的源文件，创建同名头文件(foo.h)
  - 声明要共享函数的原型，及共享变量声明、类型定义等

```
/*  foo.h */声明
void f(…);
```

```
/*  h.c */使用
#include “foo.h”

……
f(…);
……
```

```
/*  foo.c */定义
#include “foo.h”

……
void f(…)
{……}
```

把程序分为文件

- 主函数放在一个文件中，文件名与程序名相配.
- 主函数所在的文件也可能含有其他函数，只要它们不被其他文件调用.

程序分文件、共享变量声明及嵌套示例

```
//main.c
#include "main.h"
#include <stdio.h>
#include “find.h"

int main(int argc, char *argv[])
{
        int i;
        puts("Enter:");
        for(i=0;i<N;i++)
                scanf("%d",&a[i]);

        printf("max:
%d",findmax(N));
        return 0;
}
```

```
//find.h
#define N 5
extern int a[];
int findmax(int n);
```

```
//find.c
#include "find.h"
int a[N];

int findmax(int n)
{
        int i, max;
        max = a[0];
        for (i = 1; i < n; i++)
        if (a[i] > max)
                max = a[i];
        return max;
}
```