扬州大学网络安全研究中心

编译原理的一些操作

- 汇报人：李敬

2018/06/08

misc

flag格式的认知
10

鬼子电报员
50

妹子自拍照找不到了！
100

crypto

据说是经典密码
10

小明的数学题
10

Challenge 1 Solve

神奇火把

150

军事小队5名士兵执行夺宝任务，来到了狭窄到最多仅容两人同时通过的王者峡谷。通过峡谷必须使用神奇火把，否则就会受到峡谷主宰攻击身亡。不幸的是，该军团一共只有一只神奇火把。

如果各自通过的话，5人所需要的时间如下单位是：
（10011 20023 50045 70229 80112）

而如果两人同时通过，所需要的时间就是走得比较慢的那个人单独行动时所需的时间。请计算小队通过峡谷最快需要多少时间。

用该时间的16位md5值作为密钥利用Twofish算法解密如下hash可以得到明文

i87rRETUnUiFY+KNC6tdmg==

Flag

Submit

图灵测试
30

我们不一样！
100

OoK!
10

艾泽拉斯的召唤
150

提 纲 Synopsis

（1） 编译原理的一些基本概念

（2） 程序员必备的编译原理的姿势

（3） 代码是如何执行的

（4） 构造一个TINY编译器的操作

编译器是将源语言翻译为目标语言的程序

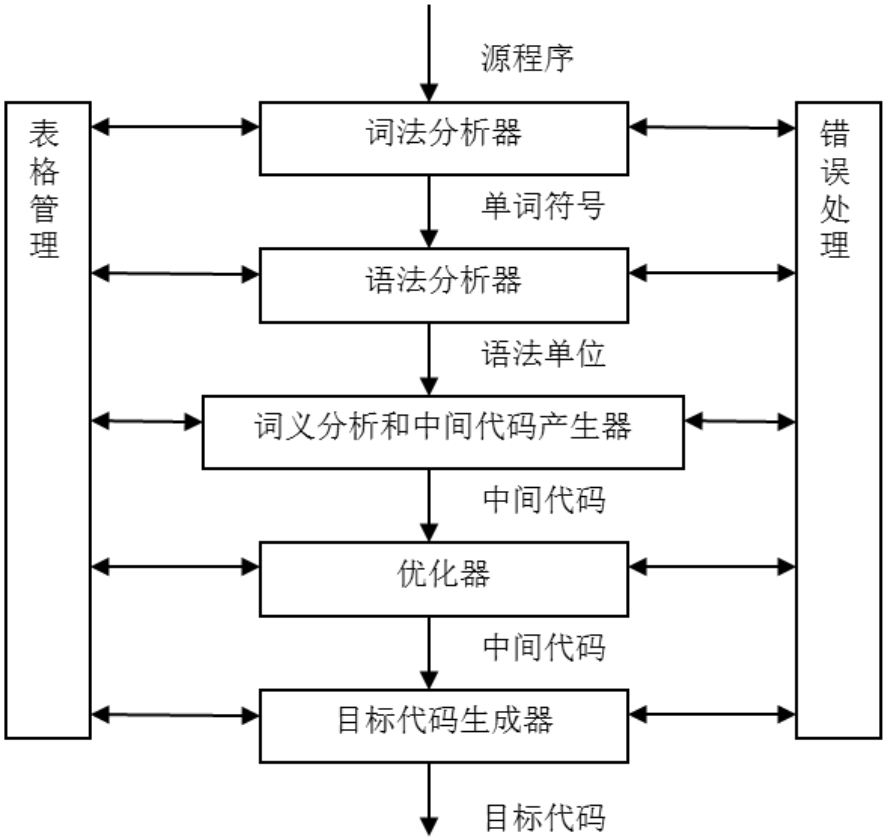
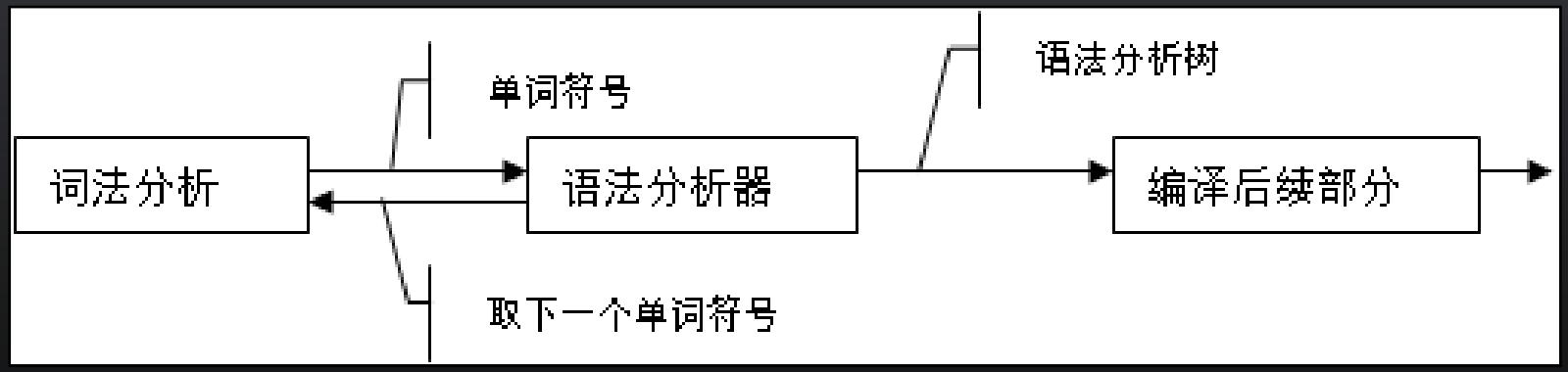


图1-1编译程序框

概念 Concept

语法分析（syntax analysis）的任务：在词法分析识别出单词符号串的基础上，分析并判定程序的语法结构是否符合语法规则。



实现语法分析器步骤：

a写出文法

b根据文法选择合适的分析算法。

C实现算法

概念 Concept

语言的语法结构是用上下文无关文法描叙的。

因此，

语法分析器的工作本质上就是：

按文法的产生式，识别输入符号串（指由单词符号组成的有限序列）是否为一个句子。

对于一个文法，当给出一串符号时，怎么知道它是不是该文法的一个句子（“程序”）？

就要从文法的开始符号出发推导出这个输入串

也就是要建立一个与输入串相匹配的语法分析树。

建立语法分析树的方法：

（1） 自上而下分析

从文法的开始符号出发，向下推导，推出句子。

a.递归下降分析算法　b.预测分析算法

（2）自下而上分析

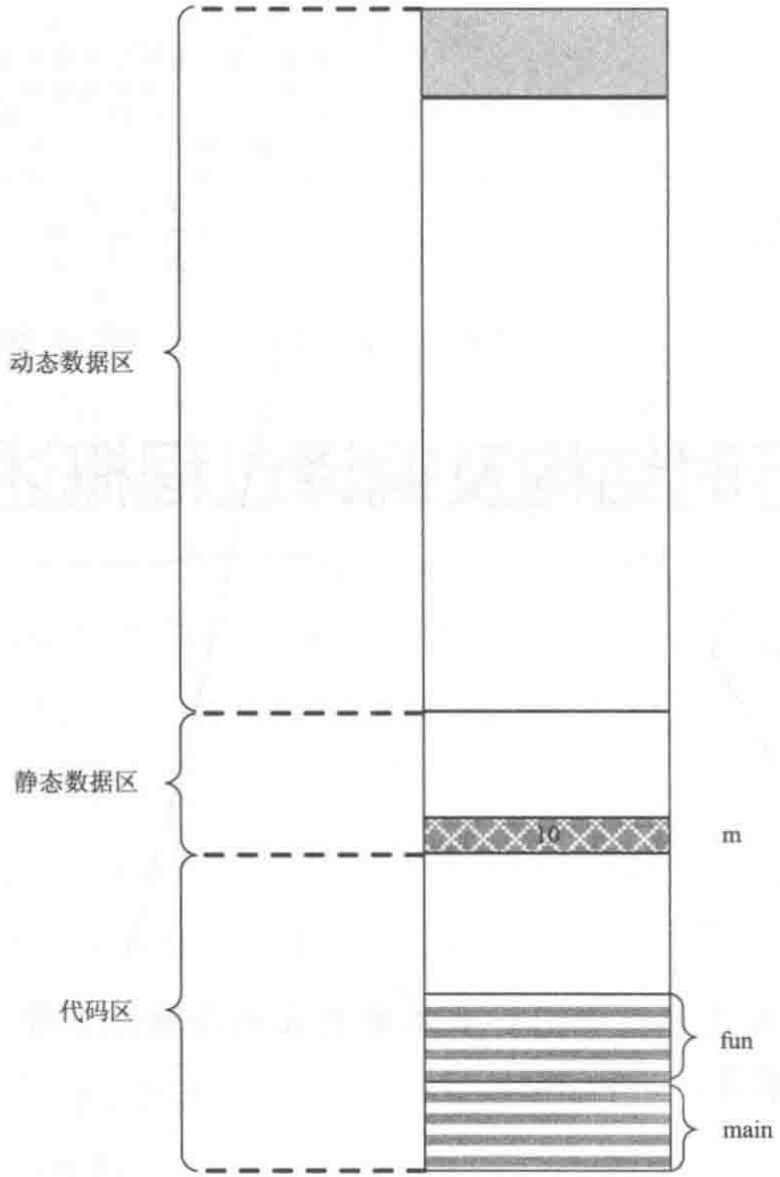
从输入串开始，逐步规约，直至规约到文法开始符号

a.算符优先分析算法　b.LR算法

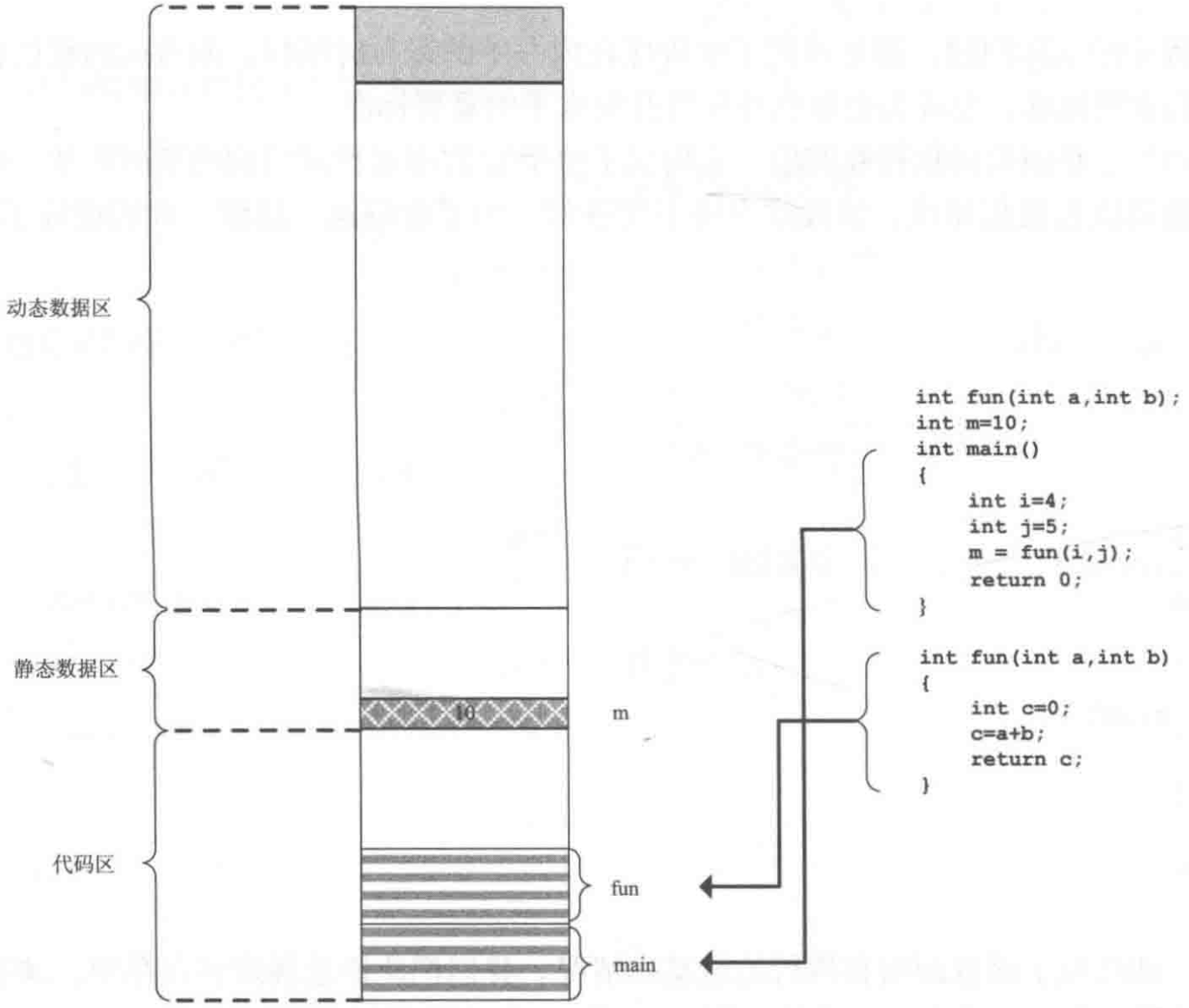
程序运行时结构：

- *核心：函数的执行和调用*
- 指令驱动、数据压栈、出栈

程 序 Program




```
int fun(int a,int b)
int m=10;
int main()
{
    int i=4;
    int j=5;
    m = fun(i,j);
    return 0;
}

int fun(int a,int b)
{
    int c=0;
    c=a+b;
    return c;
}
```


动态数据区
静态数据区
代码区
10
m
fun
main
int fun(int a,int b);
int m=10;
int main()
{
int i=4;
int j=5;
m = fun(i,j);
return 0;
}
int fun(int a,int b)
{
int c=0;
c=a+b;
return c;
}

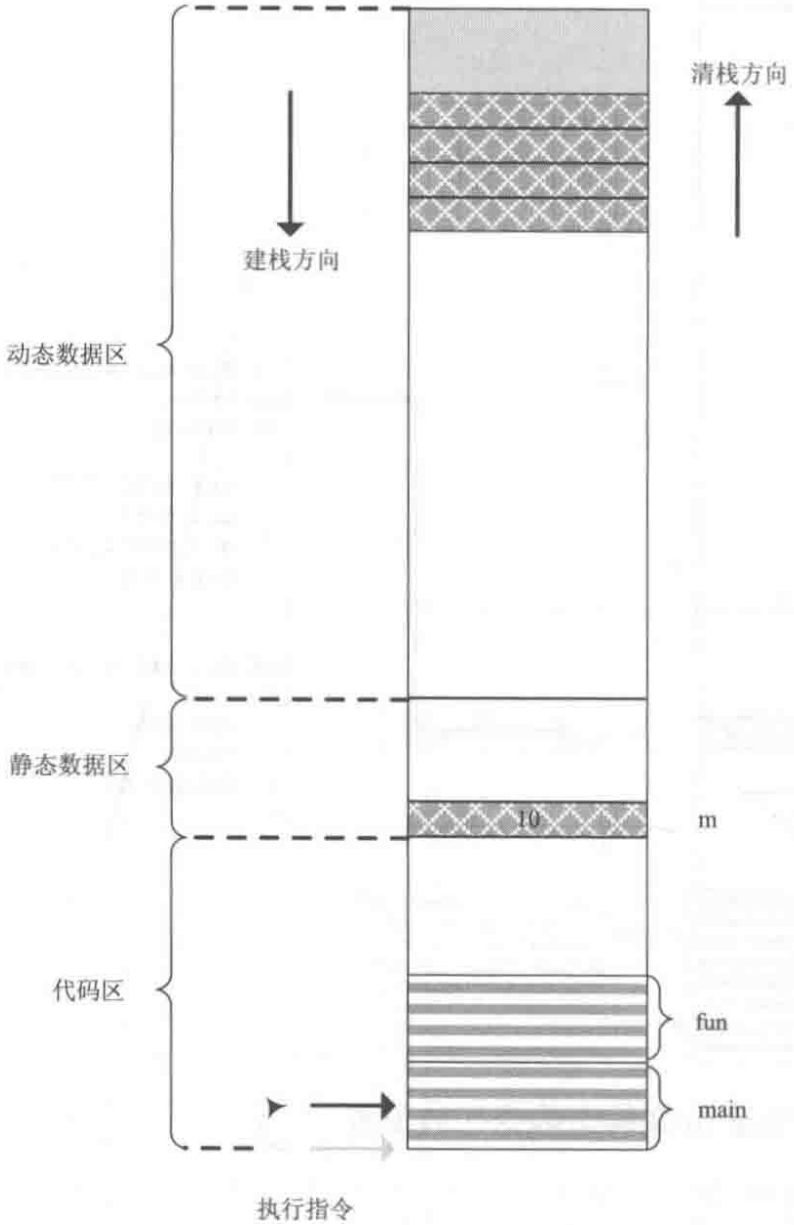
程序运行的本质就是代码区指令不断执行

驱动动态数据区和静态数据区产生数据变化

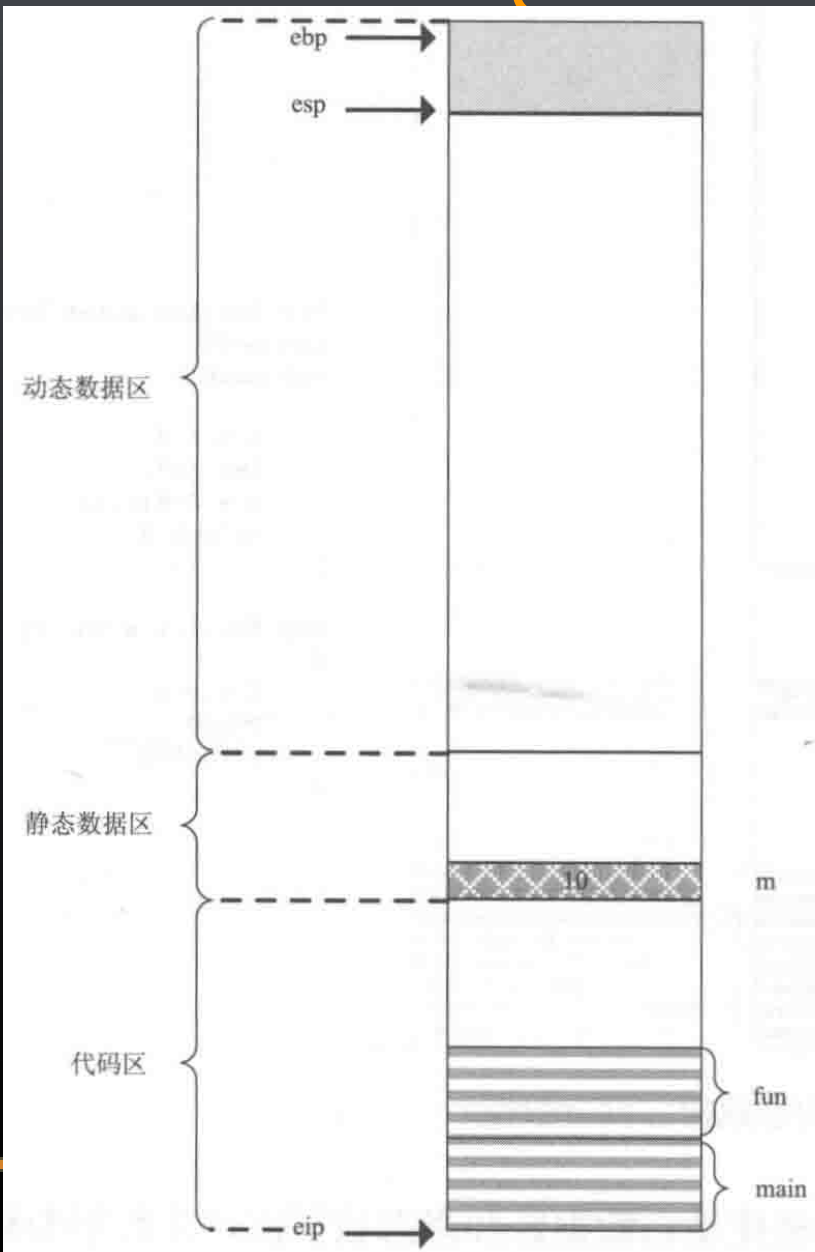
TIP：CPU有3个寄存器： eip ebp esp

eip 永远指向下一条被执行的指令

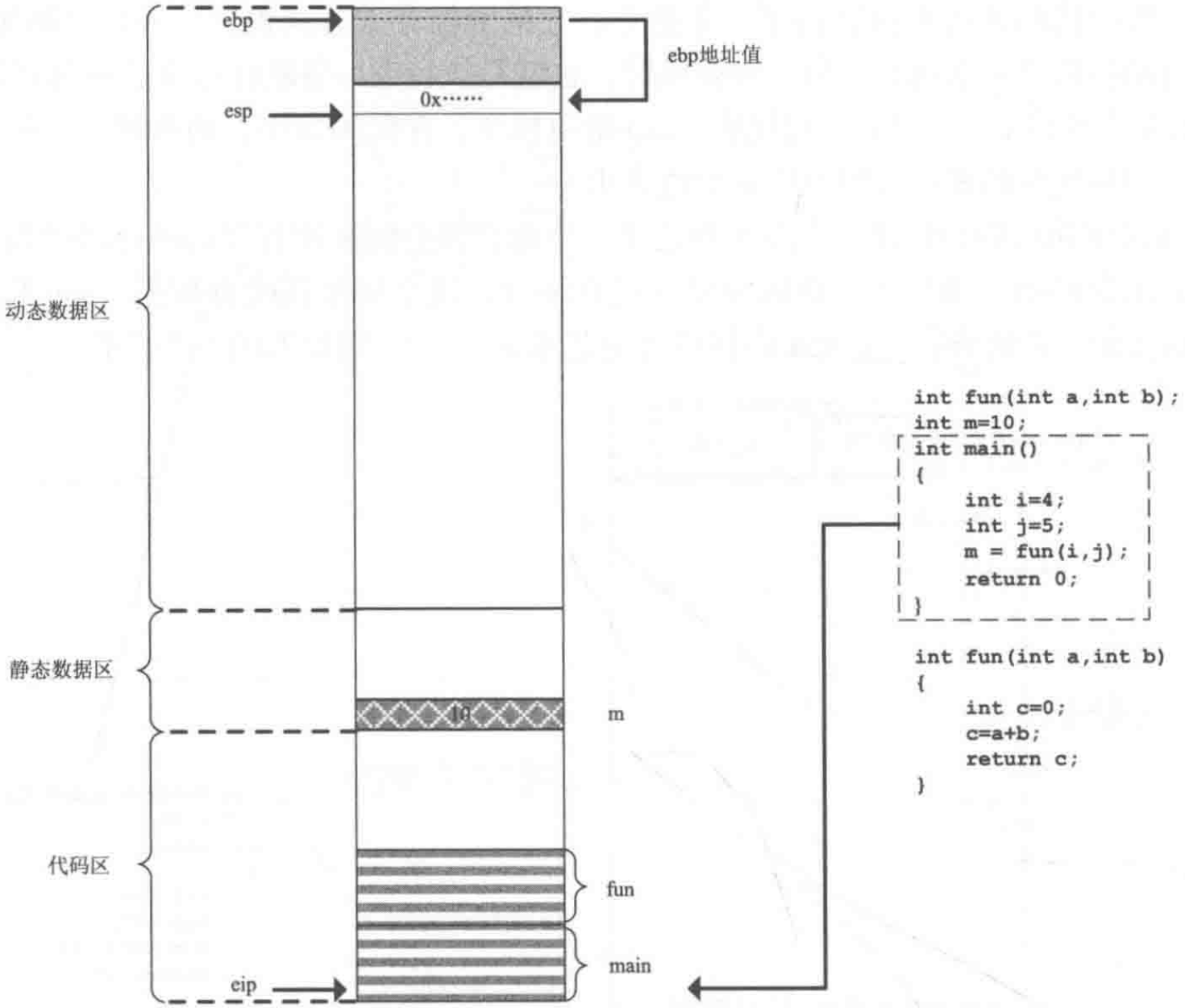
ebp esp 管控栈

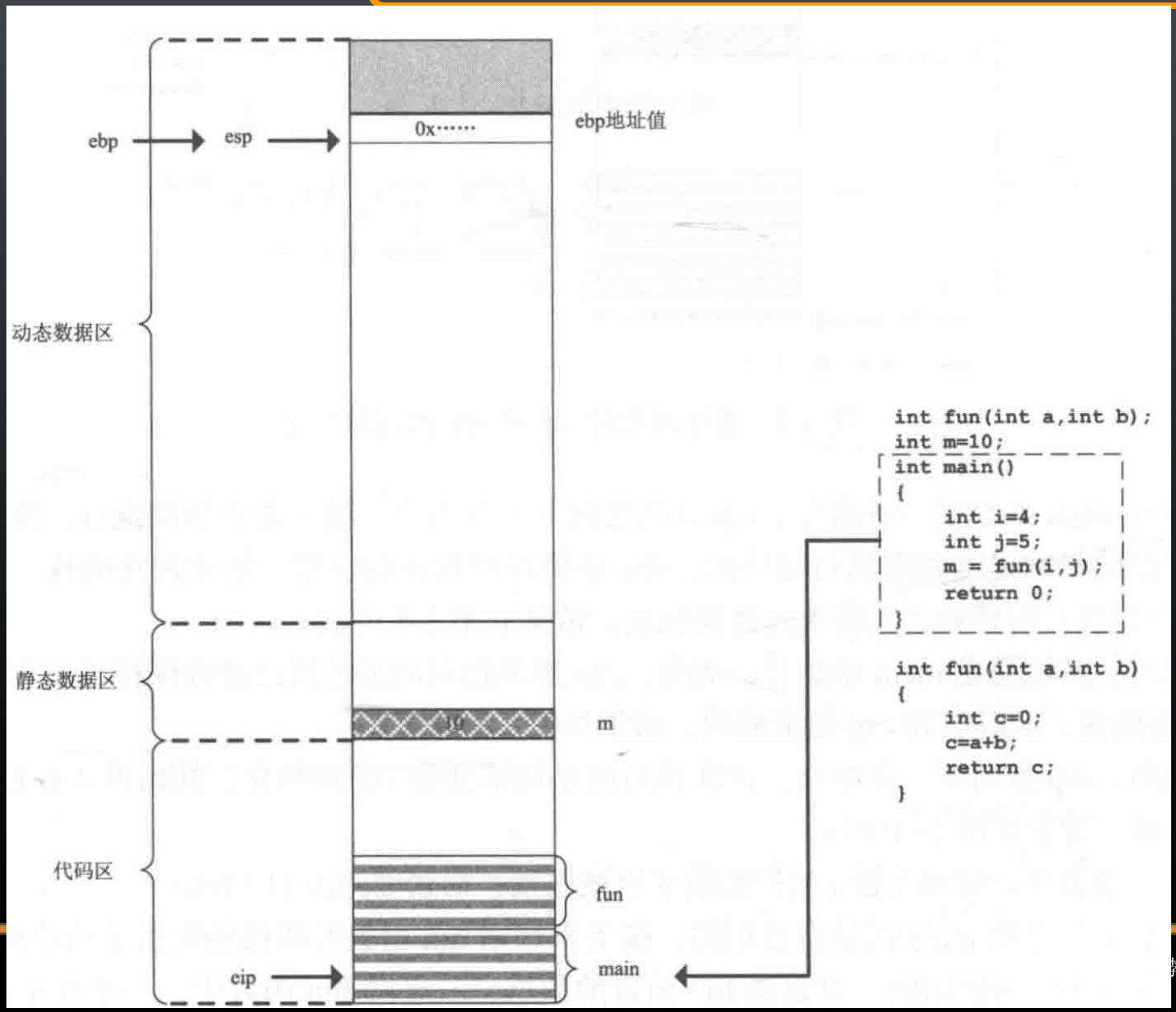


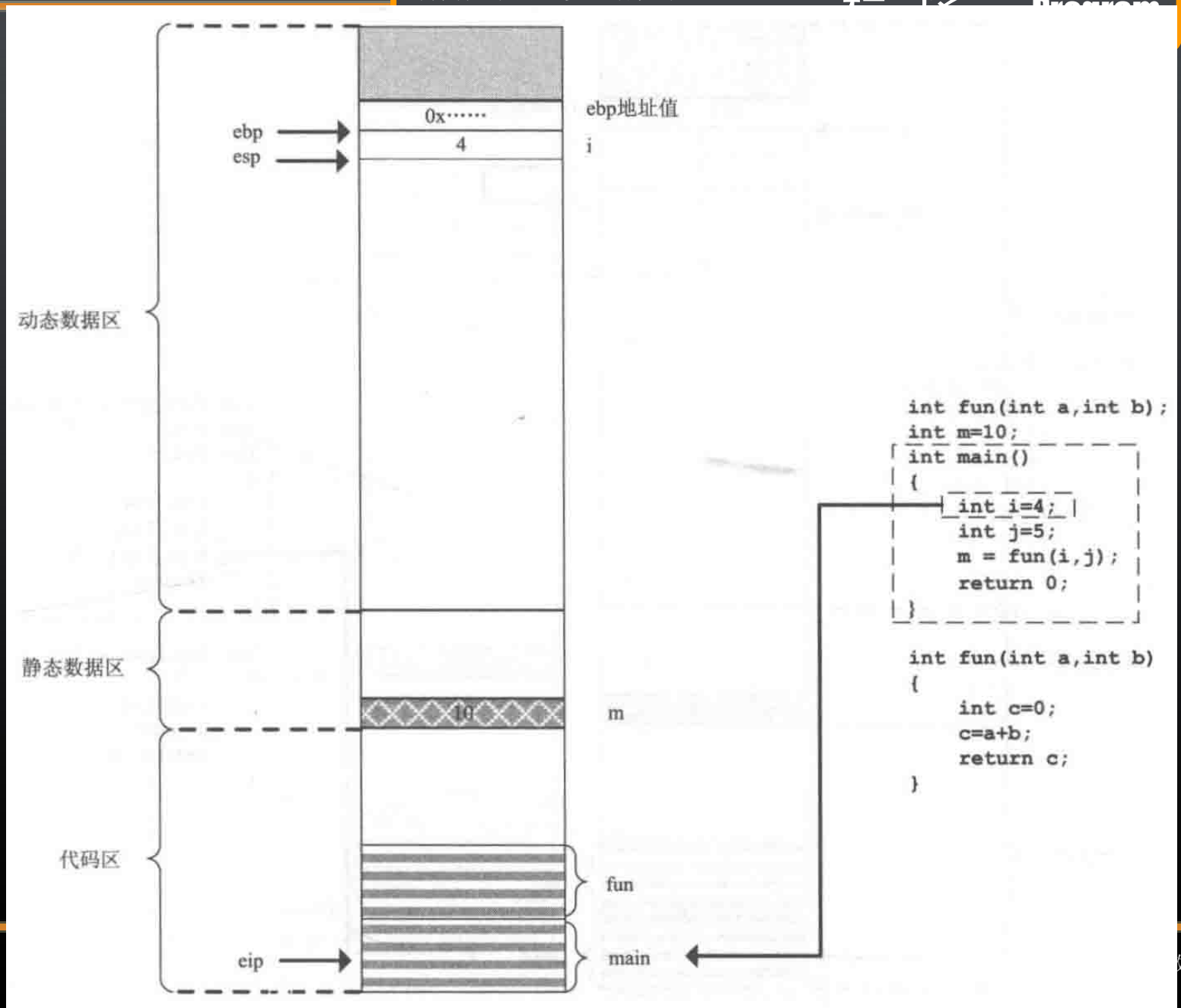

```
int fun(int a,int b);
int m=10;
int main()
{
    int i=4;
    int j=5;
    m = fun(i,j);
    return 0;
}

int fun(int a,int b)
{
    int c=0;
    c=a+b;
    return c;
}
```

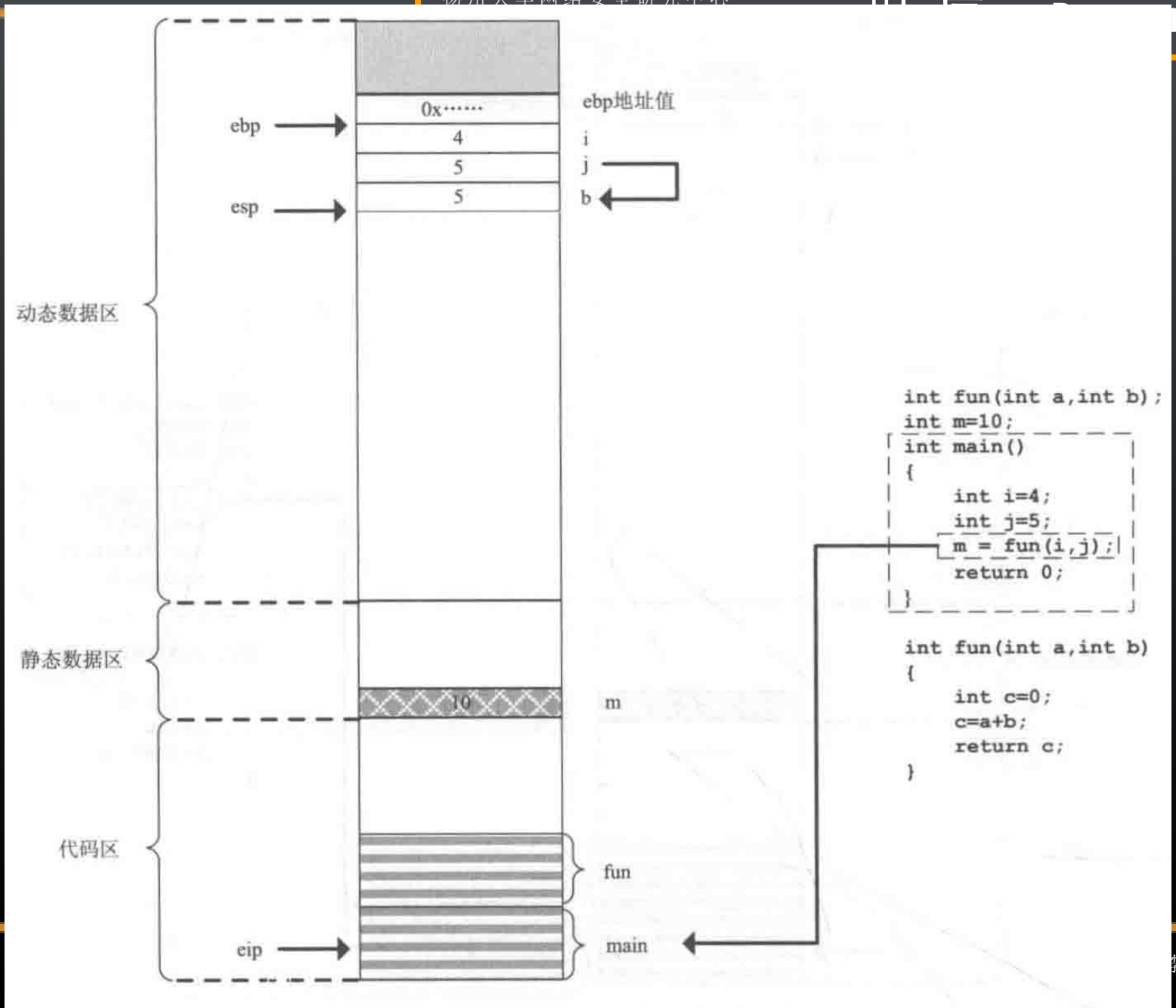


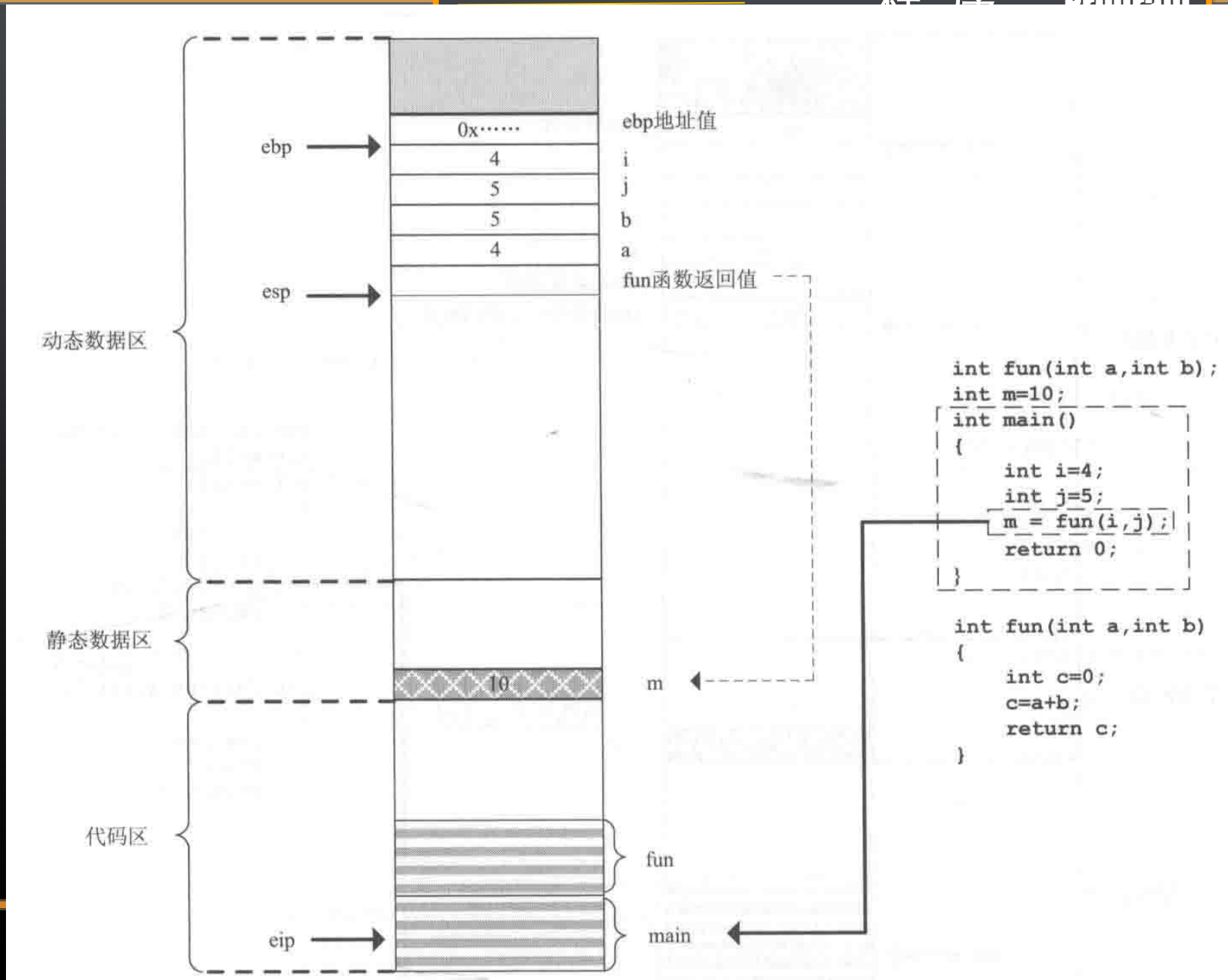

```
int fun(int a,int b);
int m=10;
int main()
{
    int i=4;
    int j=5;
    m = fun(i,j);
    return 0;
}

int fun(int a,int b)
{
    int c=0;
    c=a+b;
    return c;
}
```

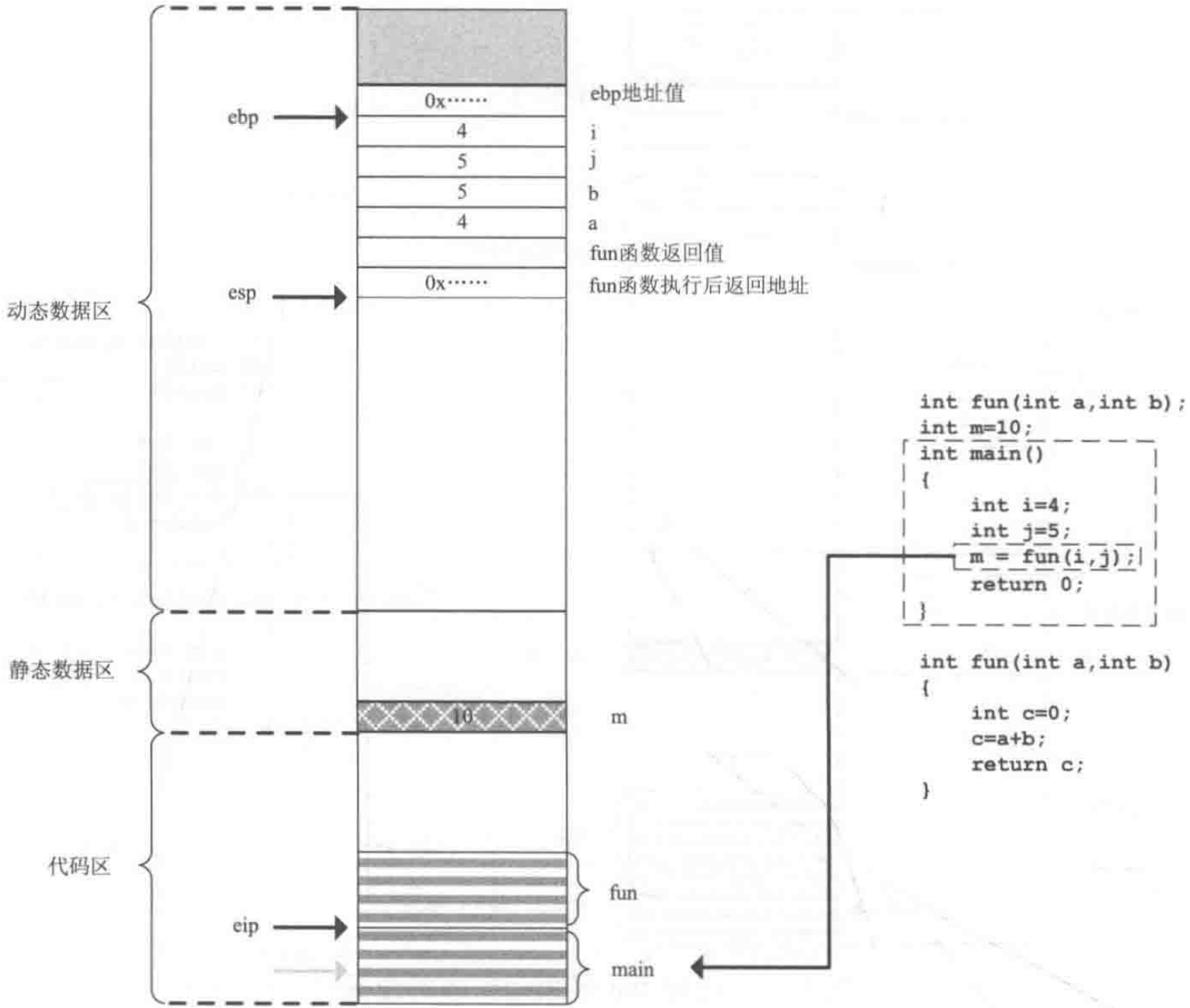

ebp
esp
ebp地址值
0x……
动态数据区
静态数据区
代码区
m
fun
main
eip
int fun(int a,int b);
int m=10;
int main()
{
int i=4;
int j=5;
m = fun(i,j);
return 0;
}
int fun(int a,int b)
{
int c=0;
c=a+b;
return c;
}

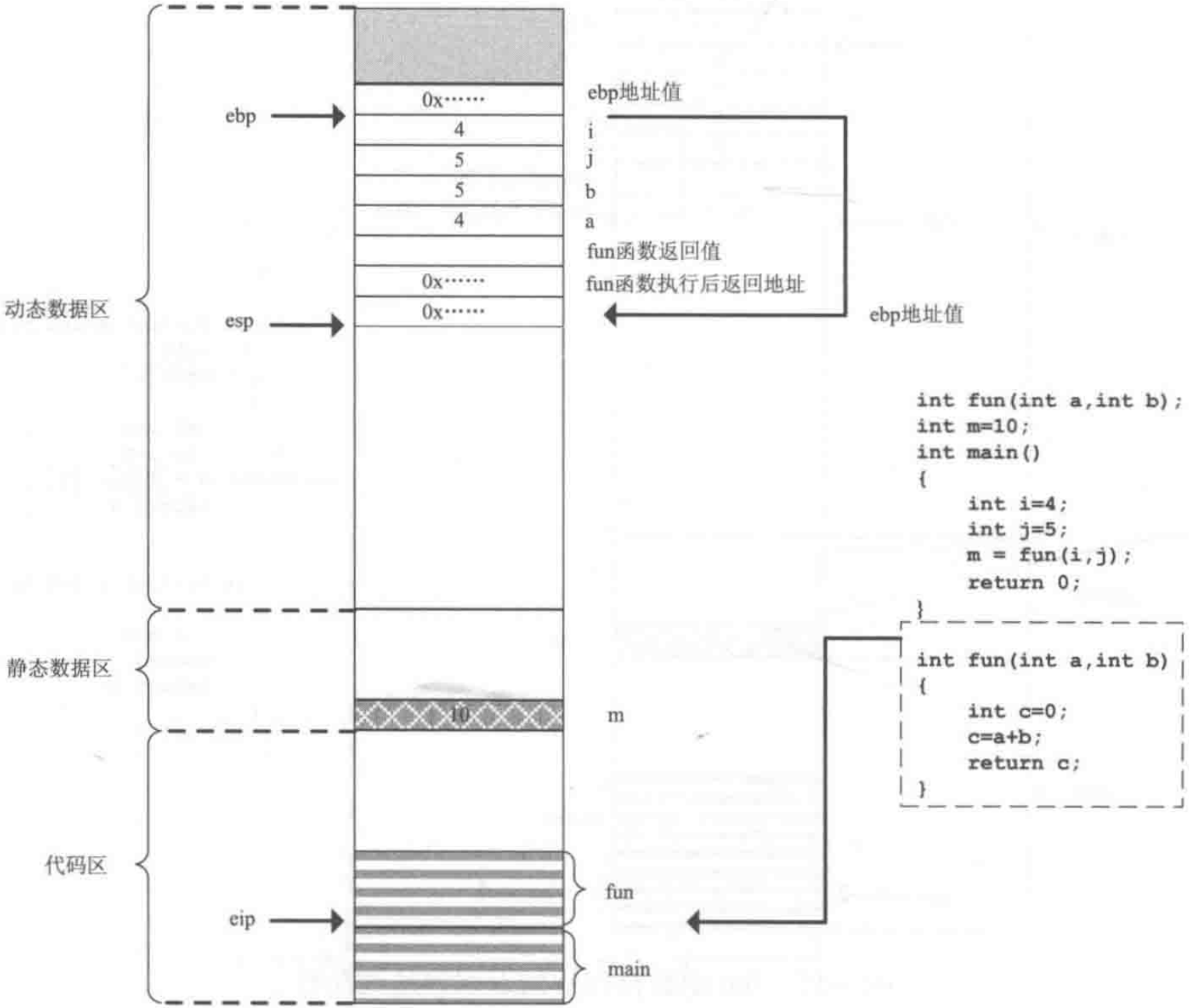

ebp
esp
0x……
ebp地址值
动态数据区
静态数据区
m
代码区
fun
eip
main
int fun(int a,int b);
int m=10;
int main()
{
int i=4;
int j=5;
m = fun(i,j);
return 0;
}
int fun(int a,int b)
{
int c=0;
c=a+b;
return c;
}

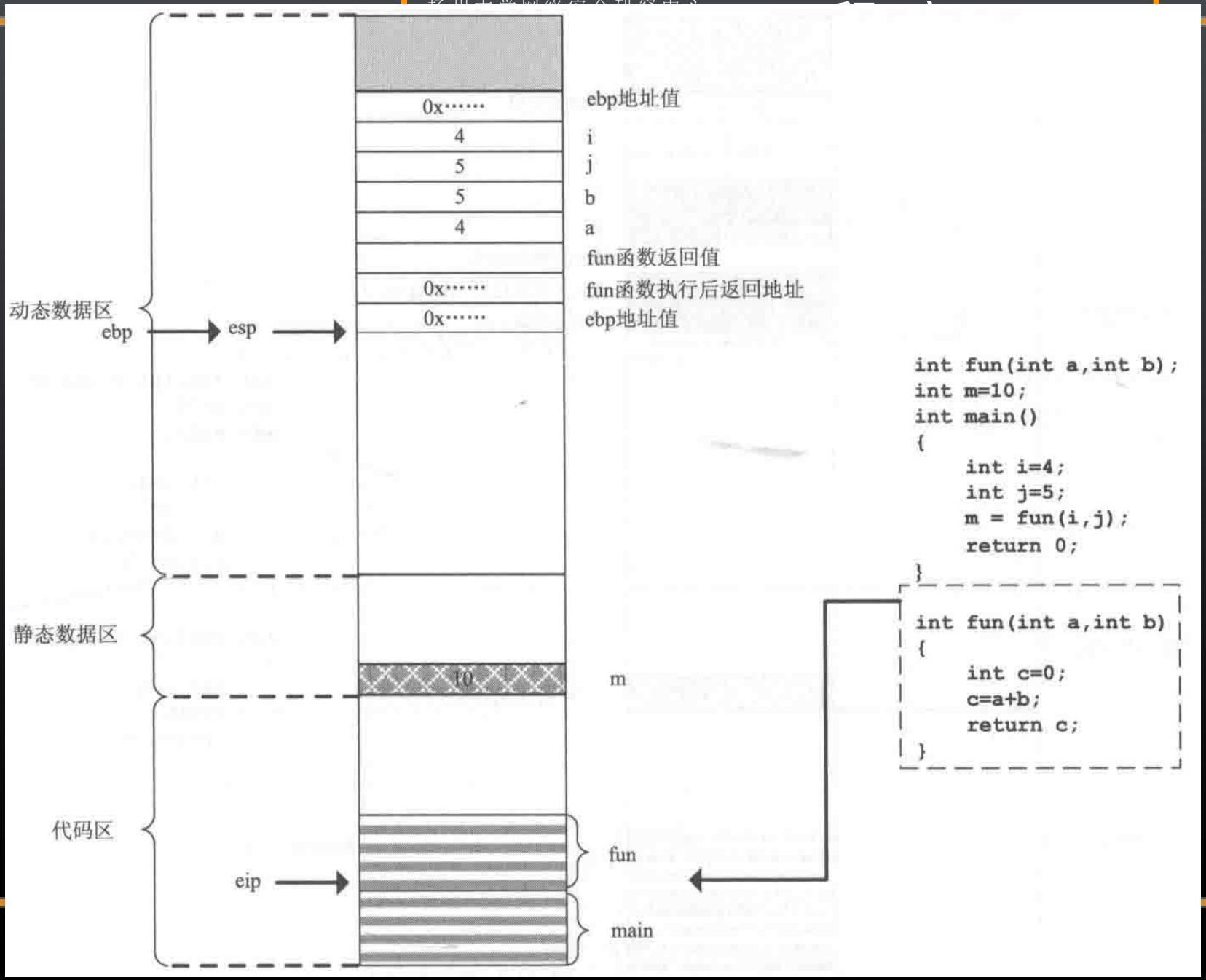

ebp
esp
0x……
4
ebp地址值
i
动态数据区
静态数据区
10
m
代码区
fun
eip
main
int fun(int a,int b);
int m=10;
int main()
{
int i=4;
int j=5;
m = fun(i,j);
return 0;
}
int fun(int a,int b)
{
int c=0;
c=a+b;
return c;
}

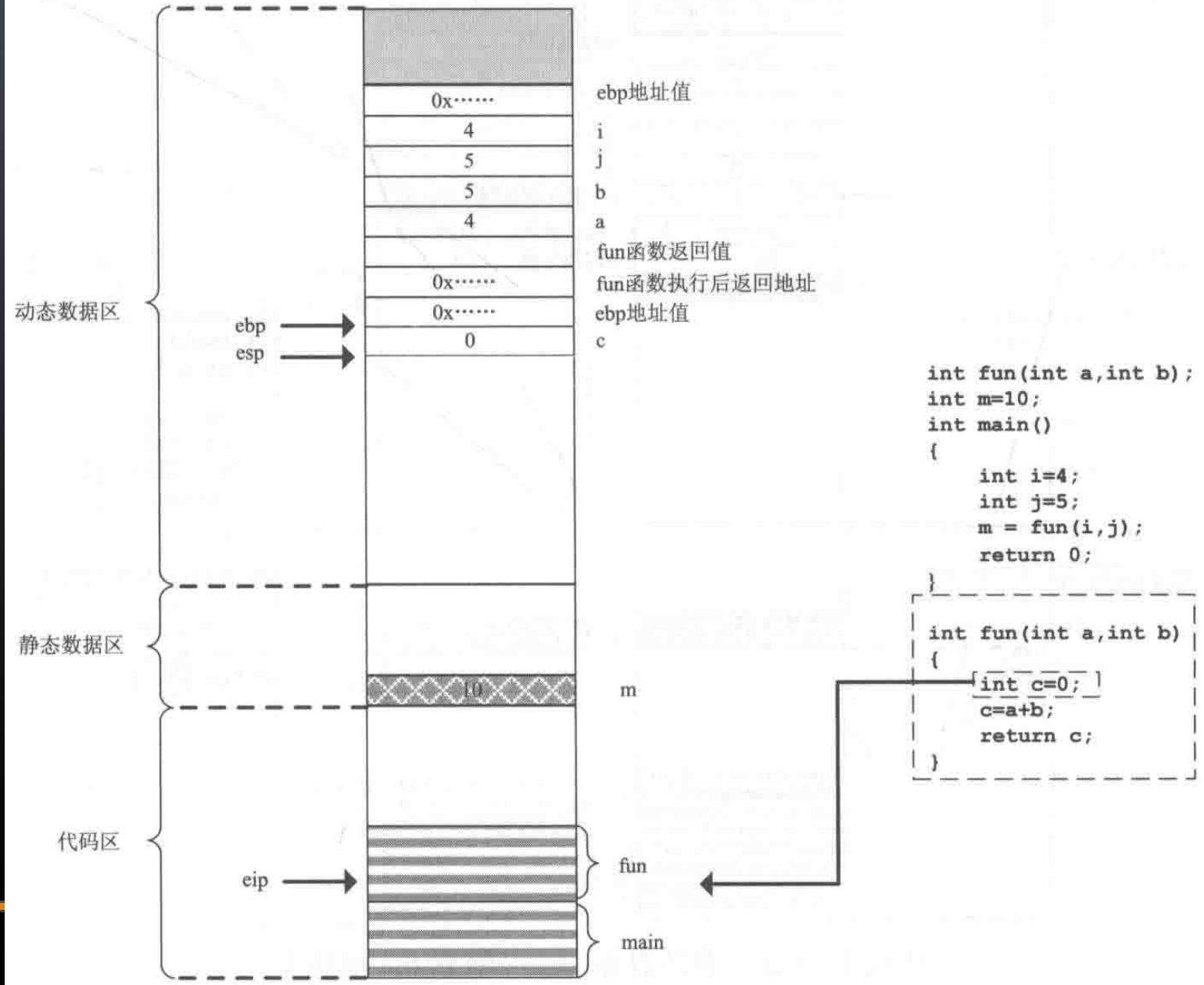

ebp地址值
0x……
ebp
4
i
5
j
esp
5
b
动态数据区
静态数据区
10
m
代码区
fun
eip
main
int fun(int a,int b);
int m=10;
int main()
{
int i=4;
int j=5;
m = fun(i,j);
return 0;
}
int fun(int a,int b)
{
int c=0;
c=a+b;
return c;
}

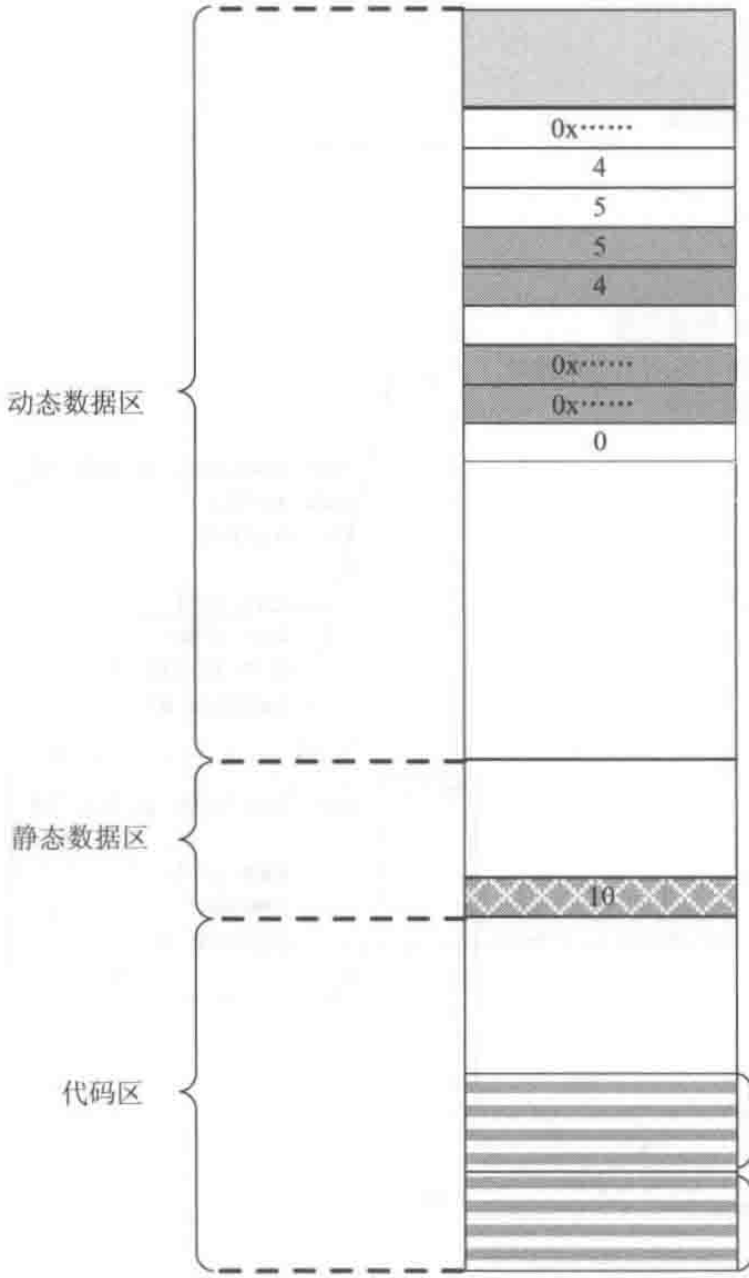

ebp
esp
eip
0x……
4
5
5
4
10
ebp地址值
i
j
b
a
fun函数返回值
m
fun
main
动态数据区
静态数据区
代码区
int fun(int a,int b);
int m=10;
int main()
{
int i=4;
int j=5;
m = fun(i,j);
return 0;
}
int fun(int a,int b)
{
int c=0;
c=a+b;
return c;
}


ebp
0x……
ebp地址值
4
i
5
j
5
b
4
a
fun函数返回值
esp
0x……
fun函数执行后返回地址
动态数据区
静态数据区
10
m
代码区
fun
eip
main
int fun(int a,int b);
int m=10;
int main()
{
int i=4;
int j=5;
m = fun(i,j);
return 0;
}
int fun(int a,int b)
{
int c=0;
c=a+b;
return c;
}

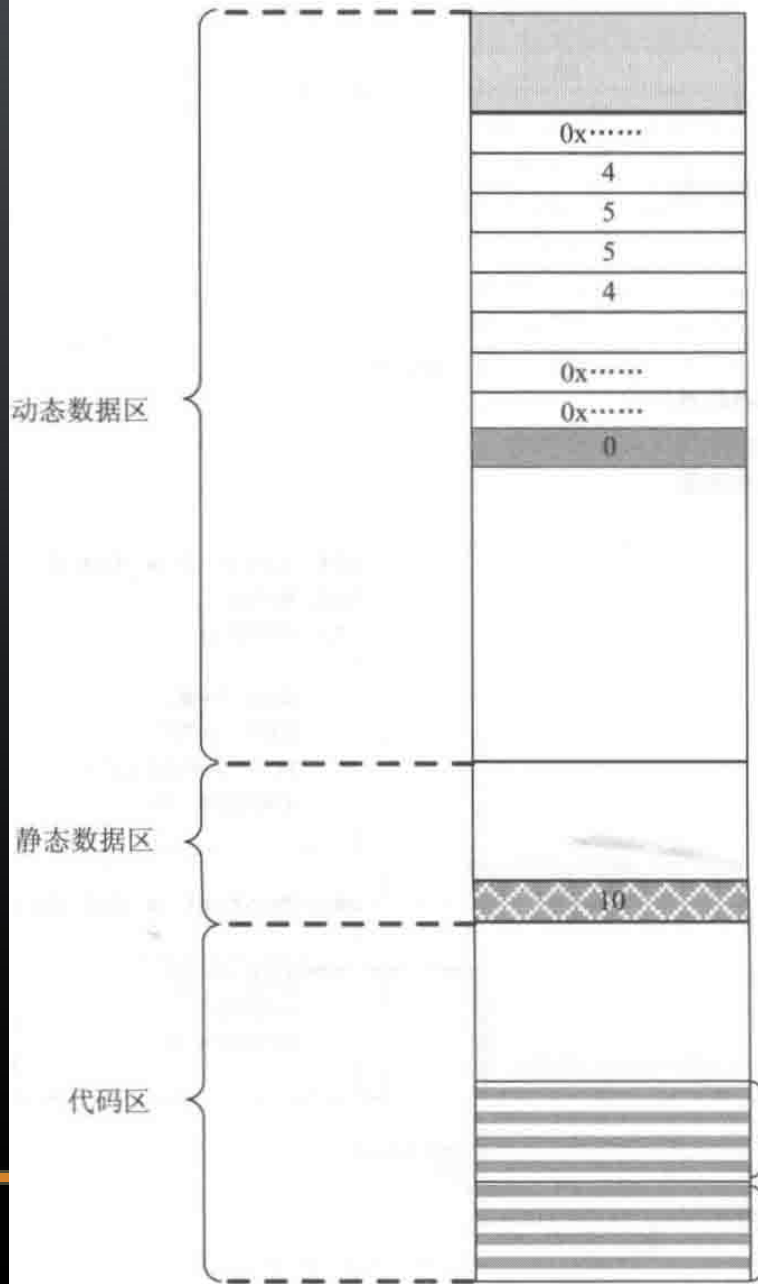

ebp
0x……
4
5
5
4
0x……
esp
0x……
ebp地址值
i
j
b
a
fun函数返回值
fun函数执行后返回地址
ebp地址值
动态数据区
静态数据区
10
m
代码区
eip
fun
main
int fun(int a,int b);
int m=10;
int main()
{
int i=4;
int j=5;
m = fun(i,j);
return 0;
}
int fun(int a,int b)
{
int c=0;
c=a+b;
return c;
}


0x……
ebp地址值
4
i
5
j
5
b
4
a
fun函数返回值
0x……
fun函数执行后返回地址
0x……
ebp地址值
动态数据区
ebp
esp
静态数据区
10
m
代码区
eip
fun
main
int fun(int a,int b);
int m=10;
int main()
{
int i=4;
int j=5;
m = fun(i,j);
return 0;
}
int fun(int a,int b)
{
int c=0;
c=a+b;
return c;
}

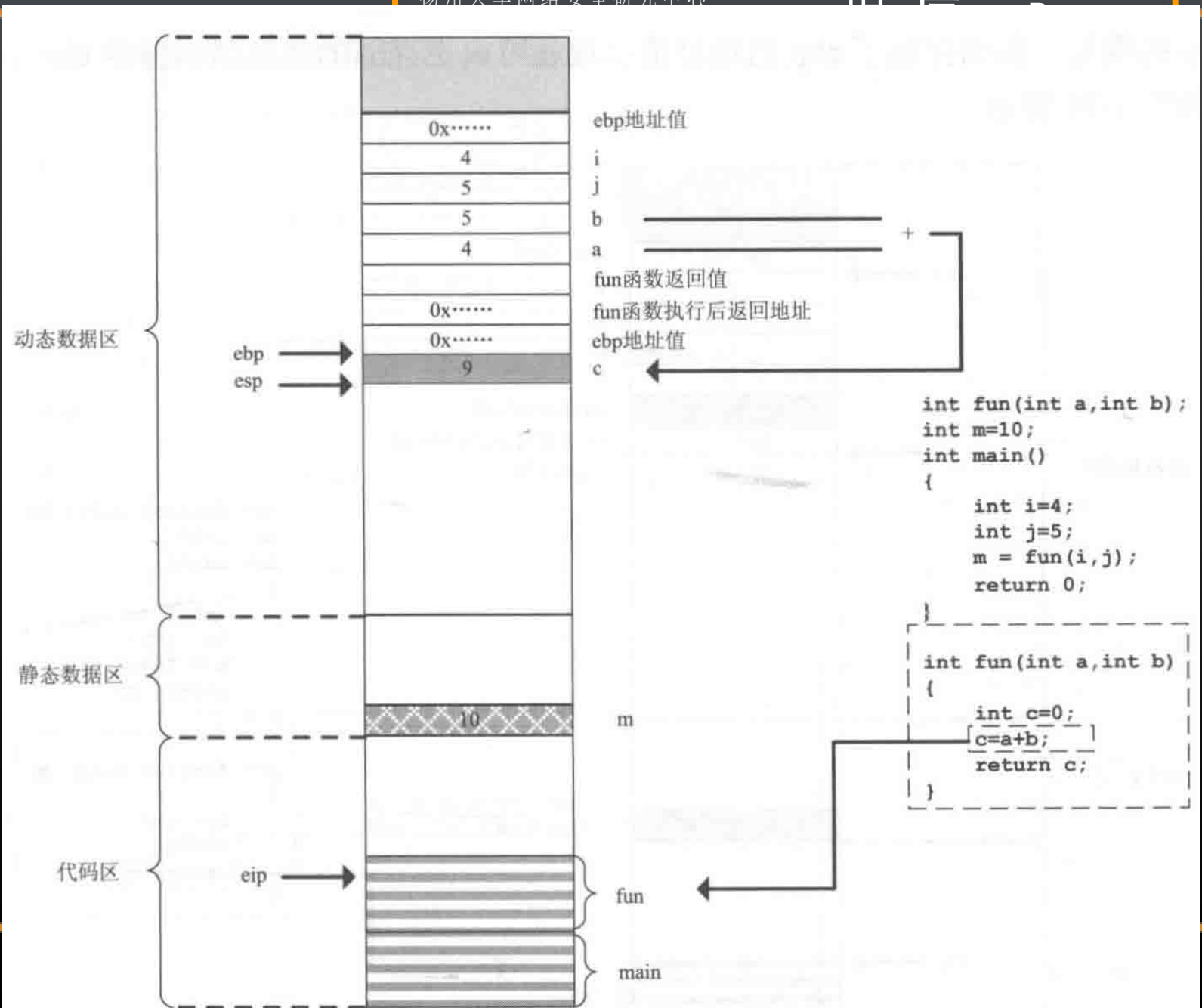

动态数据区
静态数据区
代码区
ebp
esp
eip
0x……
4
5
5
4
0x……
0x……
0
10
ebp地址值
i
j
b
a
fun函数返回值
fun函数执行后返回地址
ebp地址值
c
m
fun
main
int fun(int a,int b);
int m=10;
int main()
{
int i=4;
int j=5;
m = fun(i,j);
return 0;
}
int fun(int a,int b)
{
int c=0;
c=a+b;
return c;
}

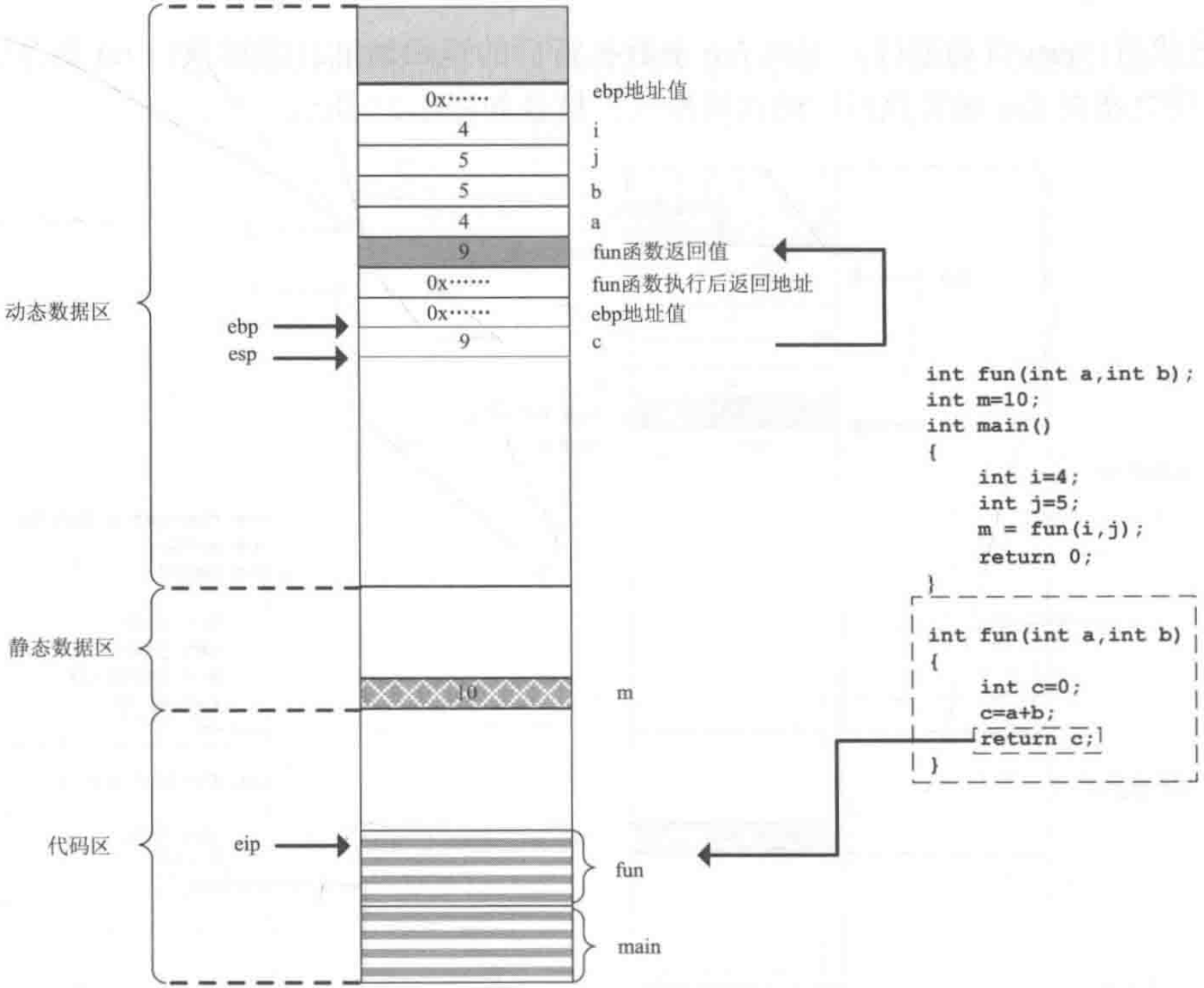


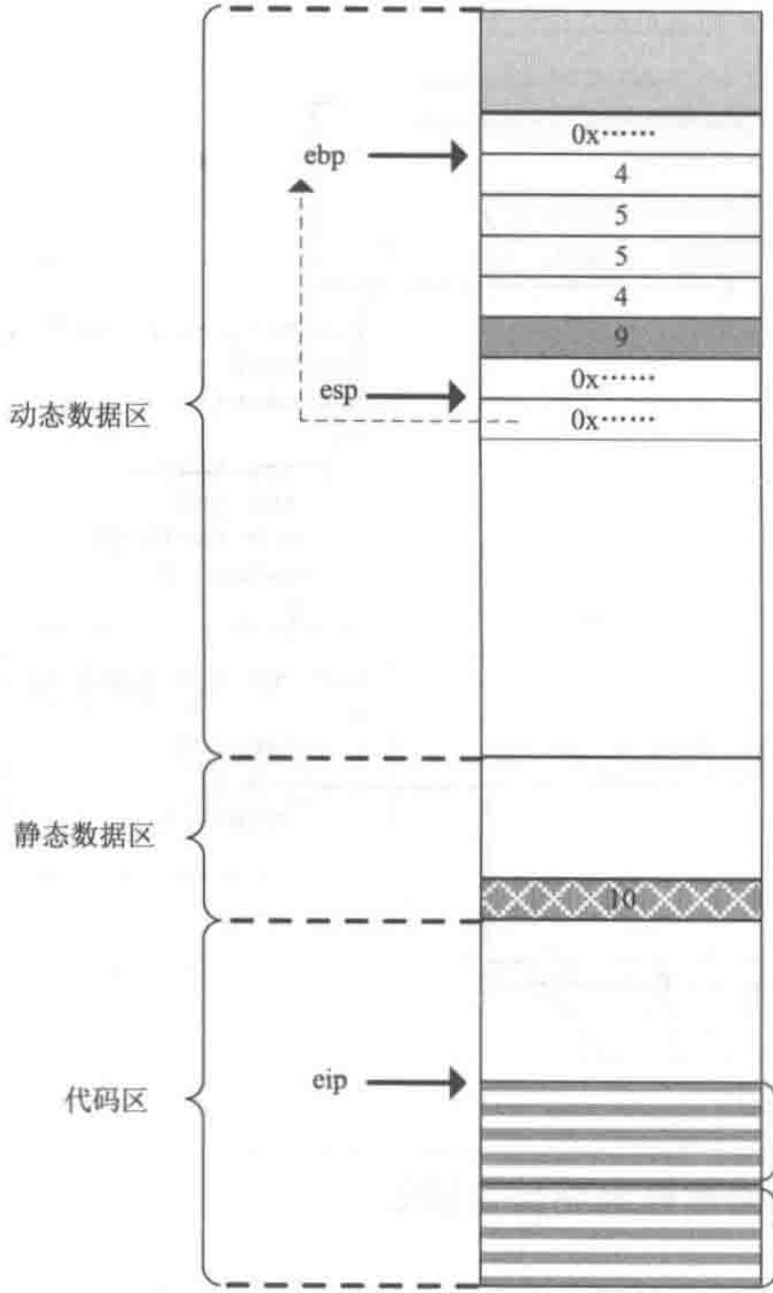

```
int fun(int a,int b);
int m=10;
int main()
{
    int i=4;
    int j=5;
    m = fun(i,j);
    return 0;
}

int fun(int a,int b)
{
    int c=0;
    c=a+b;
    return c;
}
```




```
int fun(int a,int b);
int m=10;
int main()
{
    int i=4;
    int j=5;
    m = fun(i,j);
    return 0;
}

int fun(int a,int b)
{
    int c=0;
    c=a+b;
    return c;
}
```

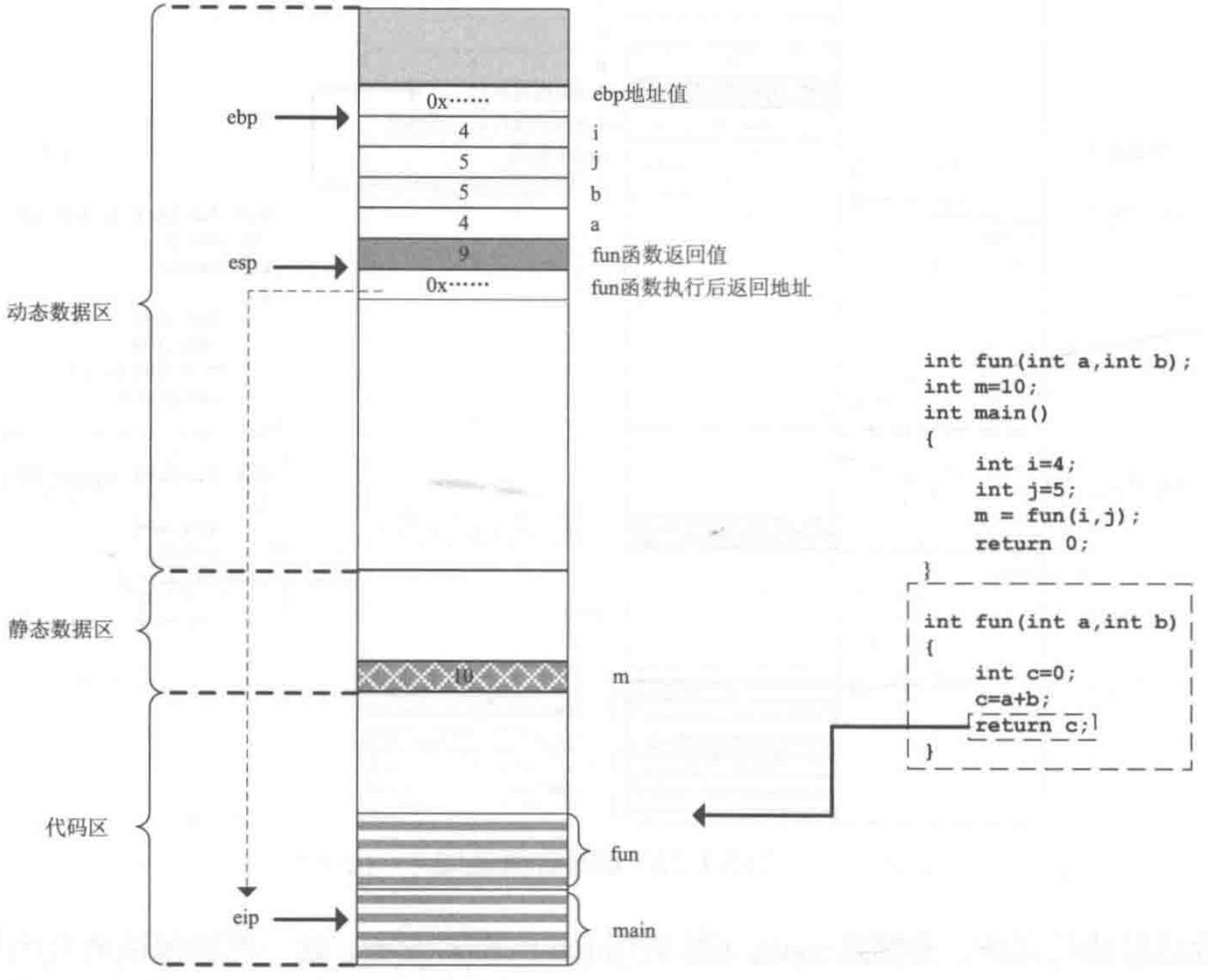

动态数据区
静态数据区
代码区
ebp
esp
eip
0x……
4
5
5
4
0x……
0x……
9
10
ebp地址值
i
j
b
a
fun函数返回值
fun函数执行后返回地址
ebp地址值
c
m
fun
main
+
int fun(int a,int b);
int m=10;
int main()
{
int i=4;
int j=5;
m = fun(i,j);
return 0;
}
int fun(int a,int b)
{
int c=0;
c=a+b;
return c;
}

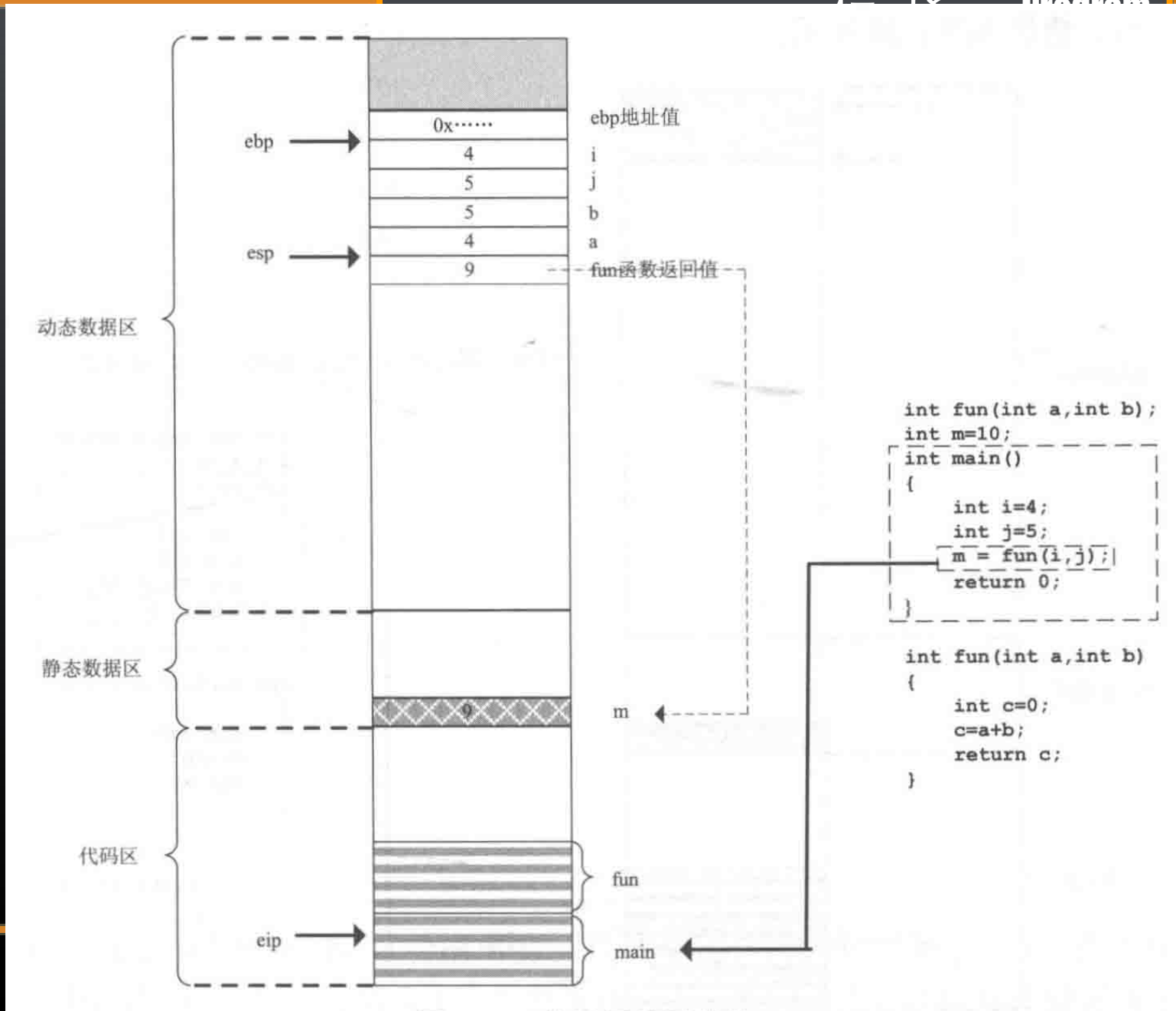

动态数据区
静态数据区
代码区
ebp
esp
eip
0x……
4
5
5
4
9
0x……
0x……
9
10
ebp地址值
i
j
b
a
fun函数返回值
fun函数执行后返回地址
ebp地址值
c
m
fun
main
int fun(int a,int b);
int m=10;
int main()
{
int i=4;
int j=5;
m = fun(i,j);
return 0;
}
int fun(int a,int b)
{
int c=0;
c=a+b;
return c;
}

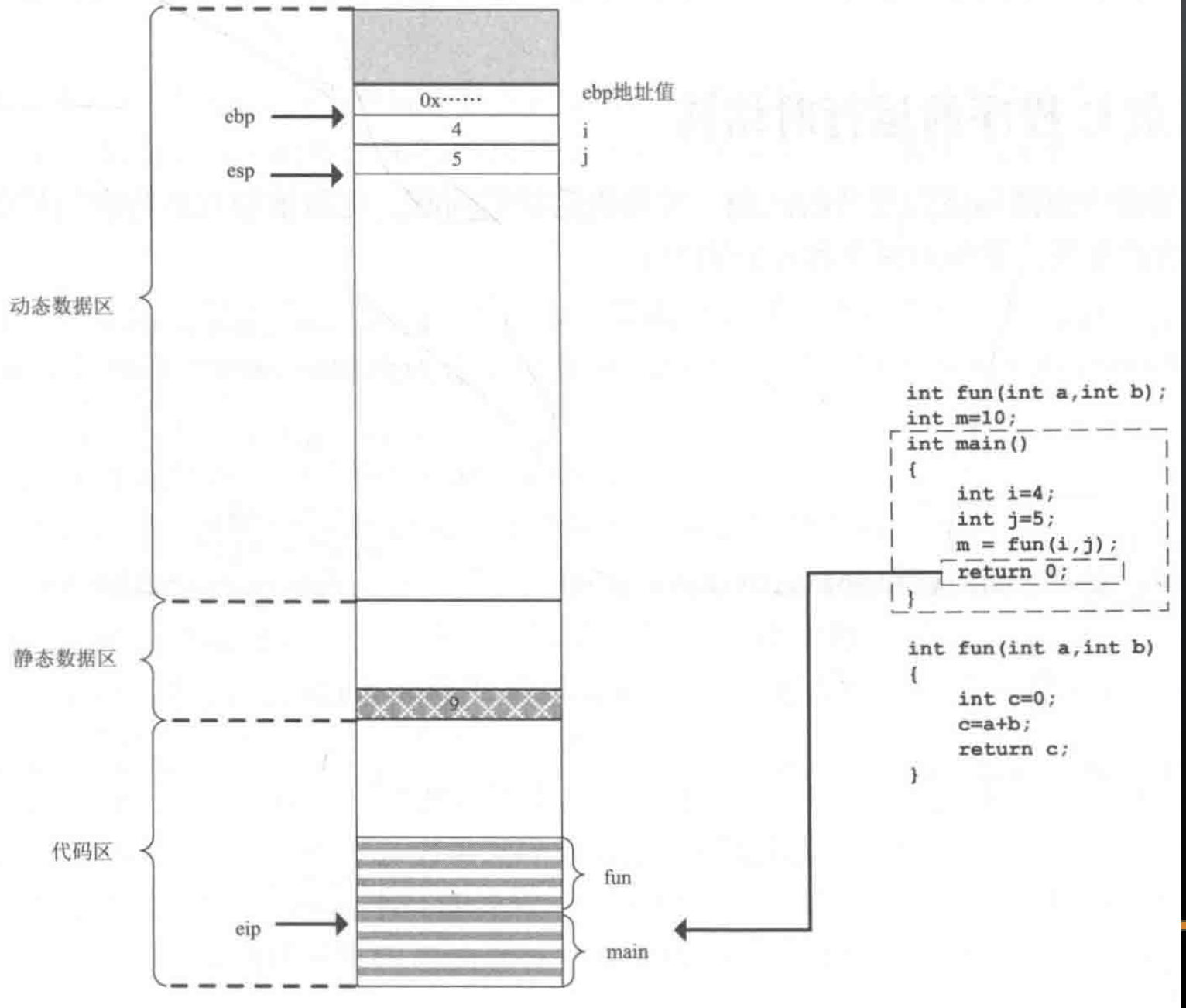


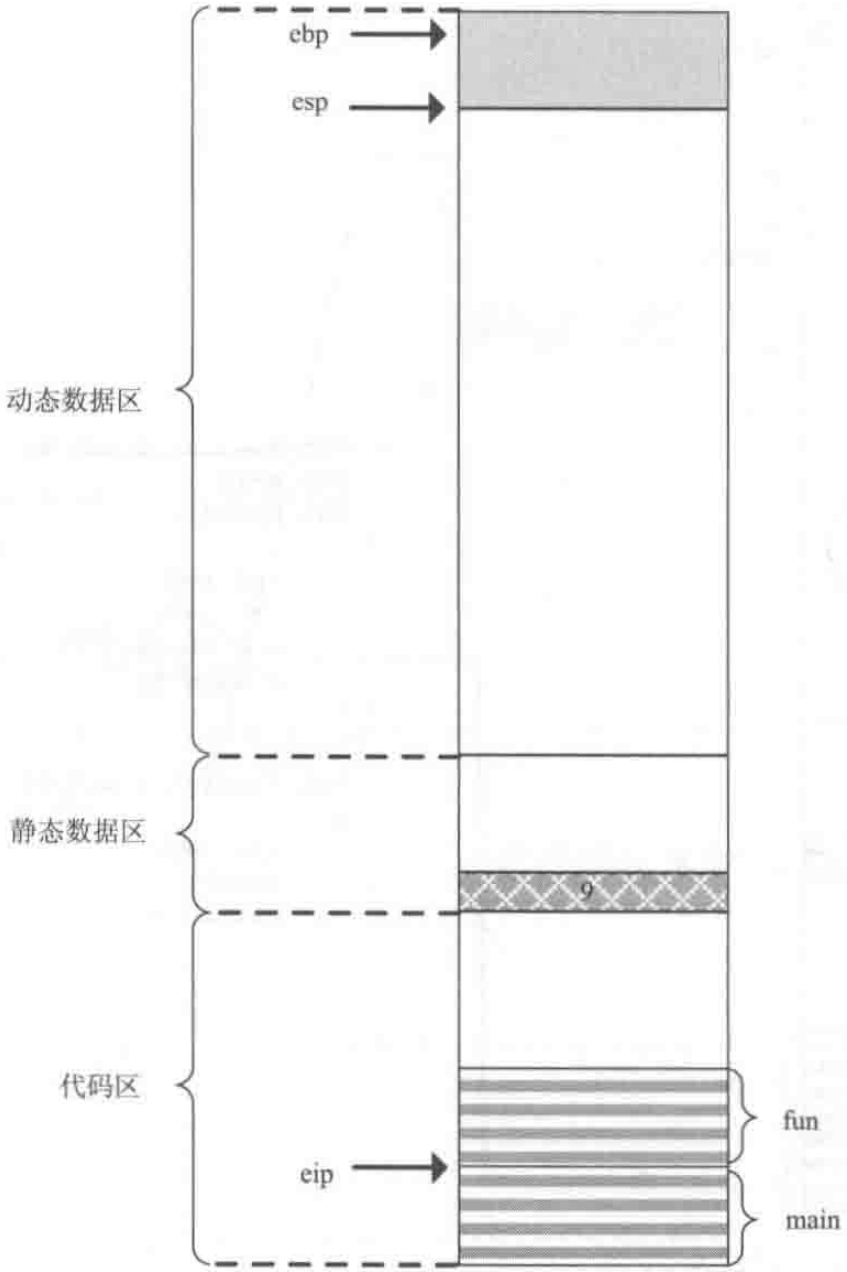

```
int fun(int a,int b);
int m=10;
int main()
{
    int i=4;
    int j=5;
    m = fun(i,j);
    return 0;
}
```

```
int fun(int a,int b)
{
    int c=0;
    c=a+b;
    return c;
}
```


ebp
esp
eip
0x……
4
5
5
4
9
0x……
ebp地址值
i
j
b
a
fun函数返回值
fun函数执行后返回地址
10
m
fun
main
动态数据区
静态数据区
代码区
int fun(int a,int b);
int m=10;
int main()
{
int i=4;
int j=5;
m = fun(i,j);
return 0;
}
int fun(int a,int b)
{
int c=0;
c=a+b;
return c;
}


ebp
0x……
ebp地址值
4
i
5
j
5
b
esp
4
a
9
fun函数返回值
动态数据区
静态数据区
9
m
代码区
fun
eip
main
int fun(int a,int b);
int m=10;
int main()
{
int i=4;
int j=5;
m = fun(i,j);
return 0;
}
int fun(int a,int b)
{
int c=0;
c=a+b;
return c;
}


ebp
esp
0x……
4
5
ebp地址值
i
j
动态数据区
静态数据区
9
代码区
eip
fun
main
int fun(int a,int b);
int m=10;
int main()
{
int i=4;
int j=5;
m = fun(i,j);
return 0;
}
int fun(int a,int b)
{
int c=0;
c=a+b;
return c;
}




```
int fun(int a,int b);
int m=10;
int main()
{
    int i=4;
    int j=5;
    m = fun(i,j);
    return 0;
}

int fun(int a,int b)
{
    int c=0;
    c=a+b;
    return c;
}
```

Q&A

E-mail: Lixeon.lij@gmail.com

Blog: lixeon.site