



§ 6. 指针基础 – 画内存图并分析

要求:

- 1、模仿第06模块PDF课件中 (P. 15-18/P. 26-29) 的样式，画出下列每小题每一步执行的内存分配及指向图示，分析为什么得到最后的结果。
 - ★ PDF课件的P. 30（如何同时得到周长和面积）
 - ★ PDF课件的P. 31（为什么无法进行交换）
 - ★ PDF课件的P. 32（为什么会出现错误，导致错误的关键词句是哪一句）
- 2、每个语句要画一张内存状态图，每小题都是4张图
 - ★ 第1张初始内存分配图附件已给出
- 3、不允许手写、手写后贴图
- 4、转换为pdf后在“文档作业”中提交（11.17前）



§ 6. 指针基础 – 画内存图并分析

★ PDF课件的P. 30（如何同时得到周长和面积）

```
#include <iostream>
using namespace std;
```

```
#define PI 3.14159
```

```
double SL(double R, double *L)
```

```
{ double S;
```

```
  S = PI*R*R;
```

```
  *L = 2*PI*R;
```

```
  return S;
```

```
}
```

```
int main()
```

```
{ double s, l, r=3;
```

```
  s=SL(r, &l);
```

```
  cout << "s=" << s << endl;
```

```
  cout << "l=" << l << endl;
```

```
}
```

函数执行后同时得到周长及面积

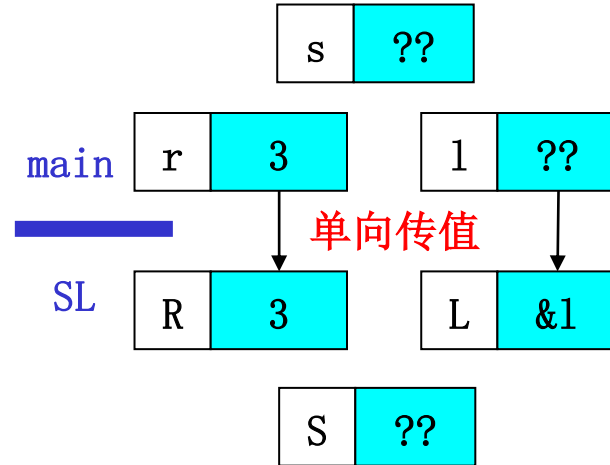
周长：指针变量做形参方式

面积：函数返回值方式

注：函数的return只能带一个返回值!!

s=28.2743

l=18.8495



初始内存分配如图所示
请自行画出SL中三句话
执行时内存的变化
理解最后的输出结果



§ 6. 指针基础 – 画内存图并分析

★ PDF课件的P. 30（如何同时得到周长和面积）

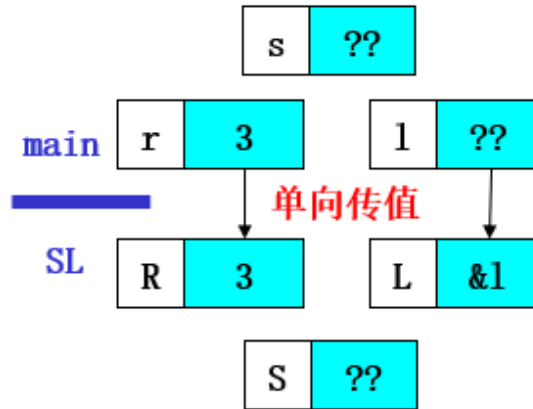
```
#include <iostream>
using namespace std;

#define PI 3.14159

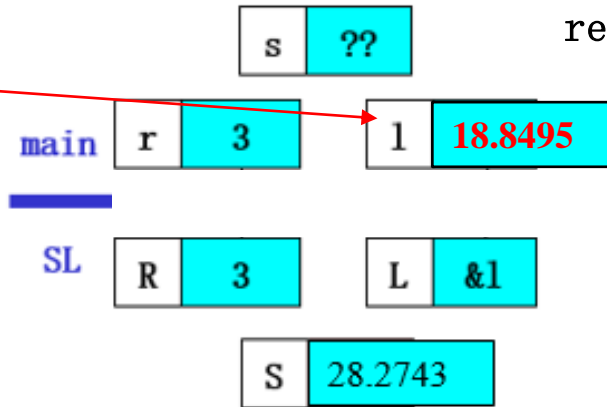
double SL(double R, double *L)
{
    double S;
    S = PI*R*R;
    *L = 2*PI*R;
    return S;
}

int main()
{
    double s, l, r=3;
    s=SL(r, &l);
    cout << "s=" << s << endl;
    cout << "l=" << l << endl;
}
```

初始:

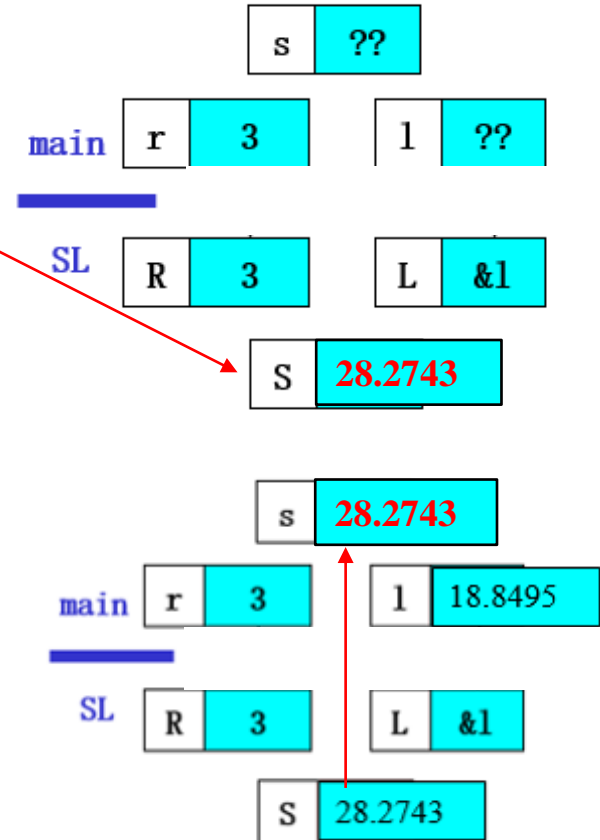


第二句:
`*L = 2*PI*R;`



第一句:
`S = PI*R*R;`

第三句:
`return S;`



如何同时得到周长和面积?

将存储周长的元素`l`地址传给`SL`函数, 就可以实现在`SL`函数内改变`l`的值 (将周长值赋给`l`)

`SL`函数设置成返回值为`double`类型, 返回面积值, 将函数返回值赋给`s`即可

如此便可以同时得到周长和面积

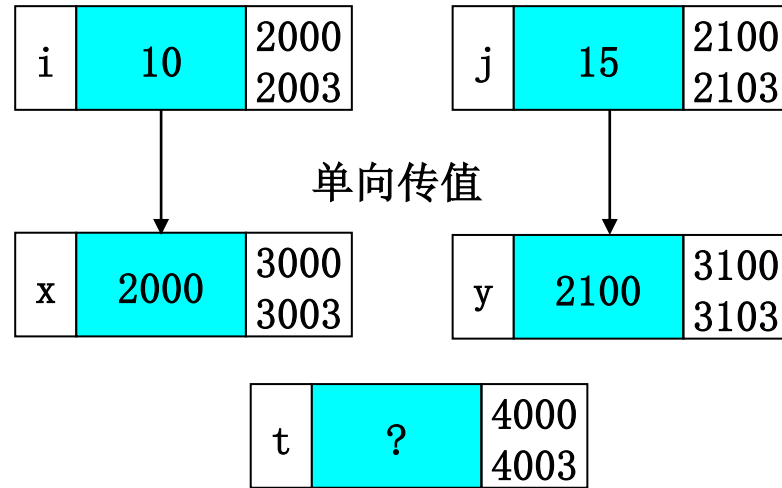


§ 6. 指针基础 – 画内存图并分析

★ PDF课件的P. 31 (为什么无法进行交换)

```
void swap(int *x, int *y)
{
    int *t;
    t = x;
    x = y;
    y = t;
}
```

```
int main()
{
    int i=10, j=15;
    cout << "i=" << i << " j=" << j << endl;
    swap(&i, &j);
    cout << "i=" << i << " j=" << j << endl;
}
```



初始内存分配如图所示
请自行画出swap中三句话
执行时内存的变化
理解为什么无法交换



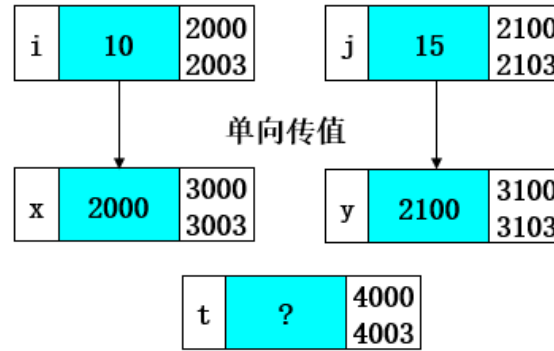
§ 6. 指针基础 – 画内存图并分析

★ PDF课件的P. 31 (为什么无法进行交换)

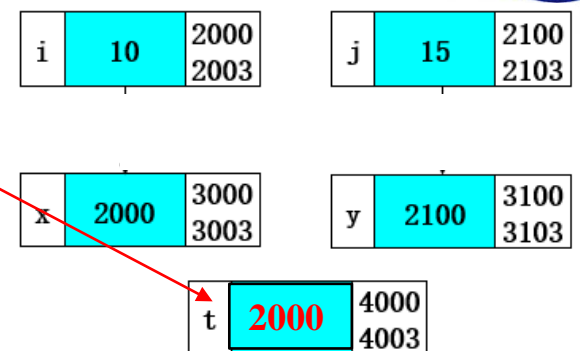
```
void swap(int *x, int *y)
{
    int *t;
    t = x;
    x = y;
    y = t;
}
```

```
int main()
{
    int i=10, j=15;
    cout << "i=" << i << " j=" << j << endl;
    swap(&i, &j);
    cout << "i=" << i << " j=" << j << endl;
}
```

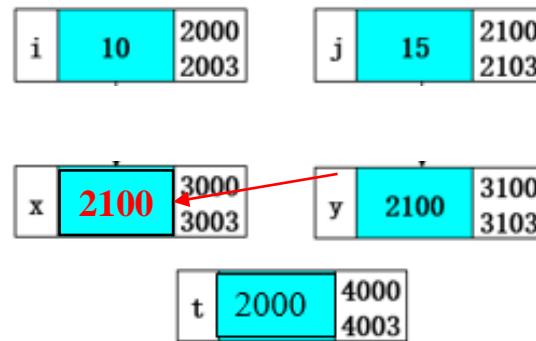
初始:



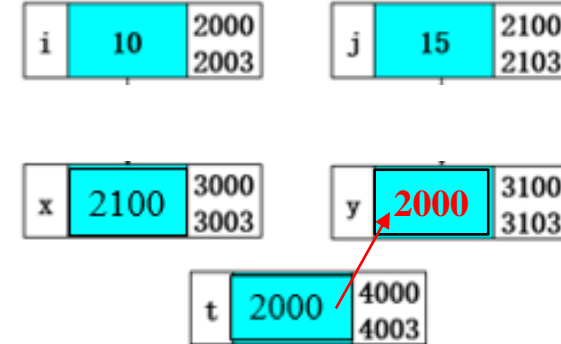
第一句:
`t = x;`



第二句:
`x = y;`



第三句:
`y = t;`



为什么无法进行交换

swap函数中交换的是指针x与指针y中存储的内容 (即i、j的地址)
并没有改变指针x与指针y指向地址所存的内容 (即i、j的值没有被改变)
所以无法交换i、j的值



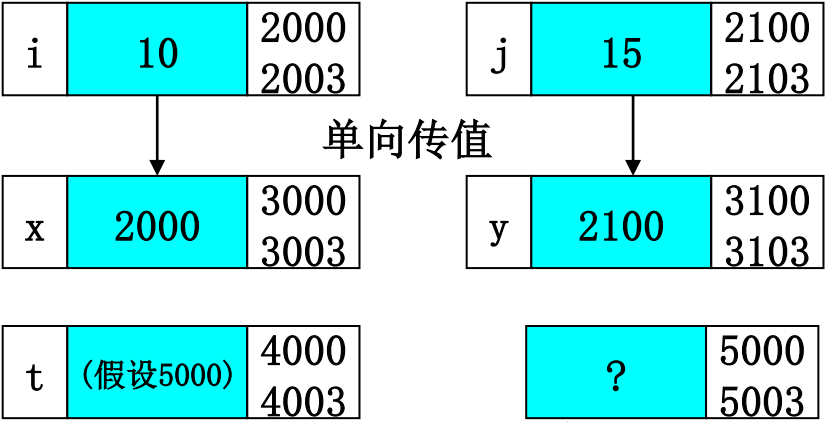
§ 6. 指针基础 – 画内存图并分析

★ PDF课件的P. 32（为什么会出现错误，导致错误的关键词是哪一句）

```
void swap(int *x, int *y)
{
    int *t;
    *t = *x;
    *x = *y;
    *y = *t;
}
```

VS编译报错
-使用了未初始化的局部变量t
其它编译器可能可以运行
初始内存分配如图所示，请自行画出
swap中三句话执行时内存的变化，理解为什么出现严重错误
另1：哪句是错误的关键词？
另2：int *t 改为 int tt, *t;
t = &tt;
为什么就正确了？

```
int main()
{
    int i=10, j=15;
    cout << "i=" << i << " j=" << j << endl;
    swap(&i, &j);
    cout << "i=" << i << " j=" << j << endl;
}
```



单向传值

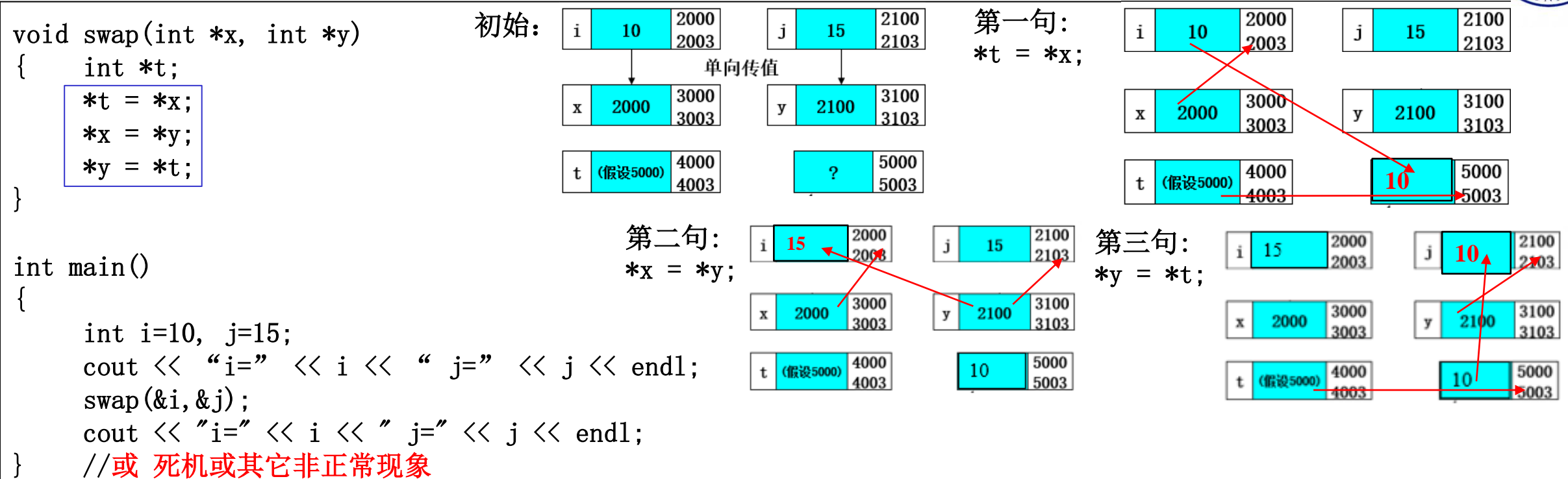
提示：5000-5003系统是否分配给了程序？

i=10 j=15
i=15 j=10
或 死机或其它非正常现象



§ 6. 指针基础 – 画内存图并分析

★ PDF课件的P.32（为什么会出现错误，导致错误的关键词句是哪一句）



(1) 为什么会出现错误, 导致错误的关键词句是哪一句

没有初始化指针t, 没有确定t所指向的地址而直接对其地址内容进行赋值导致错误。

导致错误的关键词句: `int *t;` (定义时未初始化, 危险) `*t = *x;` (使用未给指针t分配的空间, 错误)

(2) `int *t` 改为 `int tt, *t; t = &tt;` 为什么就正确了?

因为确定了指针t所指向的地址 (分配了空间): 即 `int` 型变量 `tt` 的地址。

(3) 5000-5003系统是否分配给了程序?

没有。程序没有为指针t分配空间。