交换指针

通常情况下，我们只是对普通数据进行交换，交换指针的问题很少涉及，今天看书时候想到了指针交换问题，这里总结下，也方便我以后查阅。

首先看下整型两个数据的交换（这个比较简单，就不多介绍了），代码如下：

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

void m_swap(int *a,int *b)
{
	int tmp = *a;
	*a = *b;
	*b = tmp;
}

int main()
{
	int a=10,b=20;
	cout<<"a = "<<a<<" , b = "<<b<<endl;
	m_swap(&a,&b);
	cout<<"a = "<<a<<" , b = "<<b<<endl;
}
```

输出结果：

a = 10 , b = 20

a = 20 , b = 10

指针是内存地址，应该也算是整型变量，交换两个指针和交换两个整型变量类似。下面以两种方式进行。

传统C方式

可以通过传递指向指针的指针来进行交换，核心代码如下：

```
void m_swap(int **a,int **b)
{
	int *tmp = *a;
```

```
	*a = *b;
	*b = tmp;
}
```

demo 如下：

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

void m_swap(int **a,int **b)
{
	int *tmp = *a;
	*a = *b;
	*b = tmp;
}

int main()
{
	int a=10,b=20;
	int *pa = &a,*pb = &b;
	cout<<"a = "<<a<<" , b = "<<b<<endl;
	cout<<"*pa = "<<*pa<<" , *pb = "<<*pb<<endl;
	cout<<"pa = "<<pa<<" , pb = "<<pb<<endl<<endl;
	m_swap(&pa,&pb);
	cout<<"a = "<<a<<" , b = "<<b<<endl;
	cout<<"*pa = "<<*pa<<" , *pb = "<<*pb<<endl;
	cout<<"pa = "<<pa<<" , pb = "<<pb<<endl;
}
```

输出结果：

```
a = 10 , b = 20
*pa = 10 , *pb = 20
pa = 0x7fff35f009cc , pb = 0x7fff35f009c8

a = 10 , b = 20
*pa = 20 , *pb = 10
pa = 0x7fff35f009c8 , pb = 0x7fff35f009cc
```

从结果看到，a 和 b 的值没有变，变的是 pa 和 pb 这两个指针的值。

C++引用方式

可以通过传递指向指针的引用来实现指针的交换。

需用*定义指针，用&定义引用，如果要交换两个指针，形参类型如下（以 int 为例）：

int *&ptr

这个从右至左理解，首先 ptr 是个引用，需要用&符号，其次 ptr 与指向 int 型对象的指针相关联，需要用*符号。

则交换指针的核心代码如下：

```
void ptrSwp(int *&pi,int *&pj)
{
        int *pk = pi;
        pi = pj;
        pj = pk;
}
```

demo 如下：

```
#include <iostream>

using namespace std;


void ptrSwp(int *&pi,int *&pj)
{
        int *pk = pi;
        pi = pj;
        pj = pk;
}

int main()
{
        int i=42,j=24;
        int *pi=&i,*pj=&j;
        cout<<"i : "<<i<<"\tj : "<<j<<endl;
        cout<<"*pi : "<<*pi<<"\t*pj : "<<*pj<<endl;
```

```
        cout<<"pi : "<<pi<<"\tpj : "<<pj<<endl<<endl;
        ptrSwp(pi,pj);
        cout<<"i : "<<i<<"\tj : "<<j<<endl;
        cout<<"*pi : "<<*pi<<"\t*pj : "<<*pj<<endl;
        cout<<"pi : "<<pi<<"\tpj : "<<pj<<endl;
}
```