<https://www.cnblogs.com/chenyuming507950417/archive/2012/01/02/2310114.html>

## [C/C++之回调函数](http://www.cnblogs.com/danshui/archive/2012/01/02/2310114.html)

     今天讨论下C/C++中的回调函数。

     在理解“回调函数”之前，首先讨论下函数指针的概念。

函数指针

（1）概念：指针是一个变量，是用来指向内存地址的。一个程序运行时，所有和运行相关的物件都是需要加载到内存中，这就决定了程序运行时的任何物件都可以用指针来指向它。函数是存放在内存代码区域内的，它们同样有地址，因此同样可以用指针来存取函数，把这种指向函数入口地址的指针称为函数指针。

（2）先来看一个Hello World程序：

int main(int argc,char\* argv[])  
{  
 printf("Hello World!\n");  
 return 0;  
}

       然后，采用函数调用的形式来实现：

[复制代码](javascript:void(0);)

void Invoke(char\* s);  
  
int main(int argc,char\* argv[])  
{  
 Invoke("Hello World!\n");  
 return 0;  
}  
  
void Invoke(char\* s)  
{  
 printf(s);  
}

[复制代码](javascript:void(0);)

      用函数指针的方式来实现：

[复制代码](javascript:void(0);)

void Invoke(char\* s);  
  
int main()  
{  
 void (\*fp)(char\* s); //声明一个函数指针(fp)   
 fp=Invoke; //将Invoke函数的入口地址赋值给fp  
 fp("Hello World!\n"); //函数指针fp实现函数调用  
 return 0;  
}  
  
void Invoke(char\* s)  
{  
 printf(s);  
}

[复制代码](javascript:void(0);)

      由上知道：函数指针函数的声明之间唯一区别就是，用指针名（\*fp）代替了函数名Invoke，这样这声明了一个函数指针，然后进行赋值fp=Invoke就可以进行函数指针的调用了。声明函数指针时，只要函数返回值类型、参数个数、参数类型等保持一致，就可以声明一个函数指针了。注意，函数指针必须用括号括起来 void (\*fp)(char\* s)。

     实际中，为了方便，通常用宏定义的方式来声明函数指针，实现程序如下：

[复制代码](javascript:void(0);)

typedef void (\*FP)(char\* s);  
void Invoke(char\* s);  
  
int main(int argc,char\* argv[])  
{  
 FP fp; //通常是用宏FP来声明一个函数指针fp  
 fp=Invoke;  
 fp("Hello World!\n");  
 return 0;  
}  
  
void Invoke(char\* s)  
{  
 printf(s);  
}

[复制代码](javascript:void(0);)

函数指针数组

      下面用程序对函数指针数组来个大致了解：

[复制代码](javascript:void(0);)

#include <iostream>  
#include <string>  
using namespace std;  
  
typedef void (\*FP)(char\* s);  
void f1(char\* s){cout<<s;}  
void f2(char\* s){cout<<s;}  
void f3(char\* s){cout<<s;}  
  
int main(int argc,char\* argv[])  
{  
 void\* a[]={f1,f2,f3}; //定义了指针数组，这里a是一个普通指针  
 a[0]("Hello World!\n"); //编译错误，指针数组不能用下标的方式来调用函数  
  
 FP f[]={f1,f2,f3}; //定义一个函数指针的数组，这里的f是一个函数指针  
 f[0]("Hello World!\n"); //正确，函数指针的数组进行下标操作可以进行函数的间接调用  
   
 return 0;  
}

[复制代码](javascript:void(0);)

回调函数

（1）概念：回调函数，顾名思义，就是使用者自己定义一个函数，使用者自己实现这个函数的程序内容，然后把这个函数作为参数传入别人（或系统）的函数中，由别人（或系统）的函数在运行时来调用的函数。函数是你实现的，但由别人（或系统）的函数在运行时通过参数传递的方式调用，这就是所谓的回调函数。简单来说，就是由别人的函数运行期间来回调你实现的函数。

（2）标准Hello World程序：

int main(int argc,char\* argv[])  
{  
 printf("Hello World!\n");  
 return 0;  
}

      将它修改成函数回调样式：

[复制代码](javascript:void(0);)

//定义回调函数  
void PrintfText()   
{  
 printf("Hello World!\n");  
}  
  
//定义实现回调函数的"调用函数"  
void CallPrintfText(void (\*callfuct)())  
{  
 callfuct();  
}  
  
//在main函数中实现函数回调  
int main(int argc,char\* argv[])  
{  
 CallPrintfText(PrintfText);  
 return 0;  
}

[复制代码](javascript:void(0);)

      修改成带参的回调样式：

[复制代码](javascript:void(0);)

//定义带参回调函数  
void PrintfText(char\* s)   
{  
 printf(s);  
}  
  
//定义实现带参回调函数的"调用函数"  
void CallPrintfText(void (\*callfuct)(char\*),char\* s)  
{  
 callfuct(s);  
}  
  
//在main函数中实现带参的函数回调  
int main(int argc,char\* argv[])  
{  
 CallPrintfText(PrintfText,"Hello World!\n");  
 return 0;  
}

[复制代码](javascript:void(0);)

      至此，对回调函数应该有了一个大致的了解。