回调函数就是函数指针的一种是用，A要调用

看了这么多的资料，我只将每位的定义总结一下就一句话：回调函数就是函数指针的一种用法。

在部分资料上，大量讨论了回调函数怎么被调用，到底被谁调用，还有好多的图形，我认为都没有看到问题的本质。

第二个问题：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

我实现了一个很简单的回调函数。

#include <stdio.h>

void printWelcome(int len)

{

       printf("欢迎欢迎 -- %d/n", len);

}

void printGoodbye(int len)

{

       printf("送客送客 -- %d/n", len);

}

void callback(int times, void (\* print)(int))

{

       int i;

       for (i = 0; i < times; ++i)

       {

              print(i);

       }

       printf("/n我不知道你是迎客还是送客!/n/n");

}

void main(void)

{

       callback(10, printWelcome);

       callback(10, printGoodbye);

       printWelcome(5);

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

上面的代码没有被任何系统函数调用，说明那些东西只是撒撒土迷迷路人眼而已。还有面相对象编程时，用class给封装起来也是掩人耳目，不要被外表所迷惑。

第三个问题：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

用过STL的人都知道，在STL中众多算法和程序都用到回调函数，这实现了一种策略。只要任何符合我的标准的函数和计算都可以用我这个公式。你可以实现各种各样的回调函数，只要符合我的格式就能用。

就上面的程序来说，你只要函数格式符合cllback第二个参数的格式不论你给别人做饭、铺床叠被都可以正常工作。这就是回调的作用，把回调实现留给别人。

这是一个用法。

有一位朋友用分层的概念来解释了回调机制：callback函数为B层，main函数和print\*函数为A层，A层调用了B层的回调函数callmeback，而B层的回调函数调用了A层的实现函数print\*。说白了B层就是一个接口。