指针与函数：

指向函数的指针：函数指针

C/C++中，一个函数在编译时被分配给一个入口地址，这个入口地址就是函数的指针。

函数指针是指向函数而非指向对象的指针。函数指针指向某个特定的函数类型。

函数类型由其返回类型以及形参表确定，与函数名无关。

函数指针总是指向代码区。

int (\*pf) (float);

pf声明为指向函数的指针，它所指向的函数带有1个float类型的形参和int类型的返回值。

使用typedef可以简化函数指针的定义：

typedef int (\*cmpFcn) (float);

cmpFcn是一种指向函数的指针类型的名字，就如int，char一样

int lengthCompare(float);

cmpFcn pf1 = lengthCompare; // pf1就是一个指向函数的指针

lengthCompare(5.0); // ok:直接调用

pf1(5.0); // ok:等价调用

简单的应用：

例：程序func\_array\_pointer9

int (\*pf) (float);

int func(float b);

int main(int arg, char\* argv)

{

float a = 2.5;

int b;

int c;

pf = func;

b = pf(a);

c = func(a);

printf("%d\n", b);

printf("%d\n", c);

return 0;

}

int func(float b)

{

return (int)b;

}

输出b和c的值完全相同，因为pf(a)与func(a)的作用完全相同。

用函数指针作为函数的参数：

// 指针与函数

例：程序func\_array\_pointer10

#include <stdio.h>

int max(int, int);

int min(int, int);

void processor(int, int, int (\*p)(int, int));

int main(int argc, char\* argv[])

{

int a = 6;

int b = 7;

printf("max = ");

processor(a, b, max);

printf("min = ");

processor(a, b, min);

return 0;

}

int max(int a, int b)

{

if (a > b)

{

return a;

}

else

{

return b;

}

}

int min(int a, int b)

{

if (a < b)

{

return a;

}

else

{

return b;

}

}

void processor(int a, int b, int (\*p)(int, int))

{

int result = p(a, b);

printf("%d\n", result);

}

指针函数：

返回值为指针的函数为指针函数，但不能把函数内部声明的具有局部作用域的数据地址返回（编译不会报错，但运行时有可能会造成未知的风险），可以返回堆地址，全局或静态变量地址。

例：程序func\_array\_pointer11

int\* func();

int main(int argc, char\* argv[])

{

int\* p = NULL;

p = func();

printf("%d\n", \*p); // 编译时没错，输出的值是20，但不建议返回局部作用域的地址。

return 0;

}

int\* func()

{

int a = 20;

return &a;

}

// 返回堆内存

int\* func1();

int main(int argc, char\* argv[])

{

int\* p = NULL;

p = func1();

printf("%d, %d\n", \*p, \*(p + 1));

free(p); // 释放堆内存

return 0;

}

int\* func1()

{

int\* p = (int\*)malloc(2 \* sizeof(int));

p[0] = 2;

p[1] = 3;

return p;

}

要注意函数指针与返回指针的函数的区别

int fn1(char x,char y)//两个字符参数和返回整型值的函数

int\* fn2(char x,char y)//两个字符参数和返回整型指针的函数

int fn3(int a)//一个整型参数和返回整型值的函数

int (\*fp1) (char a,char b)//两个字符参数和返回整型值的函数指针

int (\*fp2) (int a)//一个整型参数和返回整型值的函数指针

fp1=fn1;//ok,fn1函数与指针fp1指向的函数一致

fp1=fn2;//error, fn2函数与指针fp1指向的函数不一致，返回值不一致

fp2=fn3;//ok

fp2=fp1;//error,两个指针指向的函数不一致

fp2=fn3(5);//error,函数赋给函数指针时，不能加括号（初始化）

通过函数指针调用函数时：

fp2=fn3;

fp2(5);

函数指针与其他数据类型的指针尽管都是地址，但在类型上有很大的差别，不允许互相赋值。函数指针指向程序的code区，而数据指针指向data数据区，stack栈区和heap堆区