**指针的指针、指针数组做main函数参数**

1. 指针的指针：指向指针变量的指针
   1. 定义
      1. char \*\*p；
      2. 代码

int a = 3;

int \*p;

p = &a; //指向整型的指针

int \*(\*pp);

pp = &p; //指向指针的指针

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

printf("%d", \*\*pp); //由指针定义，我们可以推出，\*pp 是指向的是地址i，那么我们的\*\*pp就是取值操作了

1. 指针函数做为main（） 的参数
   1. 代码
      1. 我们先来实验一下，默认的 main() 的参数的值

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

{

for(int i = 0; i < argc; i++){

printf("%s", argv[i]);

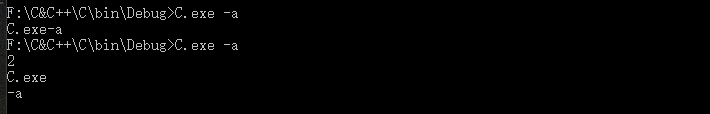
}

return 0;

}

// 我们的得出的结果是argc == 1， argv 的值是我们当前运行的目录

* + 1. 那我们怎么添加参数呢
       1. 我们先进去我们运行的目录下，有一个 exe 的文件 在后面可以添加参数



* + 1. 那有什么用呢，我们在Linux 下是不是很经常看到运行某个程序然后后面加上 -XXXXX 的，那我们也可以进行模拟一下，例如遇到 -a 我们输出 debug now
       1. 代码

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

{

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

for(int i = 0; i < argc; i++){

if(strcmp(argv[i], "-a") == 0){

printf("debug now");

}else{

printf("%s\n", argv[i]);

}

}

return 0;

}

* + - 1. 运行截图

