第四章 C语言的补充概念

模块4.3：带参数的main函数

4.3 带参数的main函数

4.3.1 引入

- 可执行文件运行时，目前只能简单的运行，如能加上参数，使用中更灵活

4.3.2 方法

- 带参数的main函数的定义形式

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

两者均可

- 参数解释

  argc：参数的个数，若不带参数，则为1(自身)

  argv：参数的内容，用指针数组表示，每个元素是一个字符串(char *)，最后一个是NULL

- 参数名argc/argv可变，类型不能变（例如：int ac, char **av）

4.3 带参数的main函数

- argv[0]为该可执行文件的文件名（含目录）

```
//集成环境运行
#include<iostream>
using namespace std;
int main(int argc, char* argv[])
{
    int i;
    cout <<"当前文件的路径："<< argv[0] << endl;
    for (i = 1; i < argc; i++){
        cout << i <<","<< argv[i] << endl;
    }
    return 0;
}
```

只输出该可执行文件的路径，即argv[0]中存储的字符串。

```
当前文件的路径：D:\test\Debug\demo.exe
```

4.3 带参数的main函数

- argv[0]为该可执行文件的文件名（含目录）

```
//命令行运行
#include<iostream>
using namespace std;
int main(int argc, char* argv[])
{
    int i;
    cout << "当前文件的路径：" << argv[0] << en
    for (i = 1; i < argc; i++){
        cout << i << "," << argv[i] << endl;
    }
    return 0;
}
```

假设编译后形成demo.exe，运行：

```
demo
demo hello world
demo who are you
```

```
D:\test\Debug>demo
当前文件的路径：demo

D:\test\Debug>demo hello world
当前文件的路径：demo
1,hello
2,world

D:\test\Debug>demo who are you
当前文件的路径：demo
1,who
2,are
3,you
```

```
//两数交换
#include <iostream>
#include <cstdlib> //atoi函数用到
using namespace std;
int main(int argc, char *argv[])
{   int a, b, t;
    if (argc<3) { /* 参数不足3个则出现提示 */
       cerr << "请带两个整数作为参数"<< endl;
       return -1;
    }
    for (t=0; t<argc; t++) /* 打印所有的参数值 */
       cout << "argv[" << t << "]=" << argv[t] << endl;
    a = atoi(argv[1]); //atoi是将字符串转为整数的函数
    b = atoi(argv[2]);
    cout << "交换前：a=" << a << "  b=" << b << endl;
    t = a; a = b; b = t;
    cout << "交换后：a=" << a << "  b=" << b << endl;
    return 0;
}
```

1.

```
请带两个整数作为参数
```

假设编译后形成demo.exe

1、集成环境运行

2、命令行运行

```
demo
demo 10
demo 10 15
demo 10 15 20
```

2.

```
D:\test\Debug>demo
请带两个整数作为参数

D:\test\Debug>demo 10
请带两个整数作为参数

D:\test\Debug>demo 10 15
argv[0]=demo
argv[1]=10
argv[2]=15
交换前：a=10  b=15
交换后：a=15  b=10

D:\test\Debug>demo 10 15 20
argv[0]=demo
argv[1]=10
argv[2]=15
argv[3]=20
交换前：a=10  b=15
交换后：a=15  b=10
```

4.3 带参数的main函数

- 例：高程大作业-文件压缩小程序

» 要求使用cmdline方式读取参数，参数格式为 压缩文件名 输出文件名 压缩指令(zip/(unzip，选做))

```
int main(int argc, char* argv[]) {
    cout << "Zipper 0.001! Author: root" << endl;
    if (argc != 4) {
        cerr << "Please make sure the number of parameters is correct." << endl;
        return -1;
    }

    if (strcmp(argv[3], "zip"
        cerr << "Unknown para
        return -1;
    }
```

```
命令提示符
Microsoft Windows [版本 10.0.19042.867]
(c) 2020 Microsoft Corporation. 保留所有权利。

C:\Users\april>cd C:\Users\april\source\repos\Gaocheng\Debug

C:\Users\april\source\repos\Gaocheng\Debug>demo C:\Users\april\source
\repos\Gaocheng\ser.log C:\Users\april\source\repos\Gaocheng\ser_comp
ressed.log zip
Zipper 0.001! Author: root
Complete!
```

4.3 带参数的main函数

- 带参数的main函数的扩展形式(仅了解)

形式：　　int main(int argc, char **argv, char **env)

或：　char *env[]

参数解释：

- argc：同前
- argv：同前
- env ：操作系统的环境变量，用指针数组来表示，每个元素是一个字符串(char *)，最后一个元素是NULL

4.3 带参数的main函数

- 带参数的main函数的扩展形式(仅了解)

使用：需要判断/取操作系统的某些设置时才用到

```
//取操作系统的环境变量
#include <iostream>
using namespace std;
int main(int argc, char **argv, char **env)
{
   int i;
   for (i=0; env[i]; i++)
      cout<< "env[" << i << "]=" << env[i] <<endl;
   return 0;
}
```