

## main 函数概述

日常生活中，要完成一件复杂的功能，我们总是习惯把“大功能”分解为多个“小功能”以实现。在 C++ 程序的世界里，“功能”可称为“函数”，因此“函数”其实就是一段实现了某种功能的代码，并且可以供其它代码调用。

一个程序，无论复杂或简单，总体上都是一个“函数”；这个函数就称为“main 函数”，也就是“主函数”。比如有个“做香辣虾”程序，那么“做香辣虾”这个过程就是“主函数”。在主函数中，根据情况，你可能还需要调用“买虾，洗菜，炒菜”…等子函数。

### main 函数的参数

C 程序最大的特点就是所有的程序都是用函数来装配的。main() 称之为主函数，是所有程序运行的入口。其余函数分为有参或无参两种，均由 main() 函数或其它一般函数调用，若调用的是有参函数，则参数在调用时传递。

```
int main(void)
{ ...
y1=f1(x1, x2);
...
}
f1(int a, int b)
{....
Y2=f2(x3, x4);
....
}
f2(int m, int n)
{....
.....
}
```

在前面课程的学习中，对 main() 函数始终作为主调函数处理，也就是说，允许 main() 调用其它函数并传递参数。事实上，main() 函数既可以无参函数，也可以是有参的函数。对于有参的形式来说，就需要向其传递参数。但是其它任何函数均不能调用 main() 函数。当然也同样无法向 main() 函数传递，只能由程序之外传递而来。这个具体的问题怎样解决呢？

我们先看一下 main() 函数的带参形式：

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

从函数参数的形式上看，包含一个整型和一个指针数组。当一个 C 的源程序经过编译、链接后，会生成扩展名为 .EXE 的可执行文件。这是可以在操作系统下直接运行的文件，换句话说，就是由系统来启动运行的。对 main() 函数既然不能由其它函数调用和传递参数，就只能由系统在启动运行时传递参数了。

在操作系统环境下，一条完整的运行命令应包括两部分：命令与相应的参数。其格式为：

命令 参数 1 参数 2... 参数 n&iquest;

此格式也称为命令行。命令行中的命令就是可执行文件的文件名，其后所跟参数需用空格分隔，并为对命令的进一步补充，也即是传递给 main() 函数的参数。命令行与 main() 函数的参数存在如下的关系：

设命令行为：program str1 str2 str3 str4 str5

其中 program 为文件名，也就是一个由 program.c 经编译、链接后生成的可执行文件 program.exe，其后各跟 5 个参数。对 main() 函数来说，它的参数 argc 记录了命令行中命令与参数的个数，共 6 个，指针 [数组](#) 的大小由参数 argc 的值决定，即为 char\*argv[6]，数组的各指针分别指向一个字符串。应当引起注意的是接收到的指针数组的各指针是从命令行的开始接收的，首先接收到的是命令，其后才是参数。

下面用实例来说明带参数的 main() 函数的正确使用。

[例 1] 利用图形库函数绘制一个变化的环。它是把一个半径为 R 1 的圆周分成 n 份，然后以每个等分点为圆心，以 R s 为半径画 n 个圆（建议大家上机试试，很漂亮动画）。利用 main() 函数的带参数形式，我们可以从键盘以命令行的方式输入 R 1 和 R s 及屏幕的背景色。

```
#include <graphics.h> /*包含图形库函数的头文件* /
#include <math.h>
#define pi 3.1415926
int main ( int argc;char *argv[]; )
{
    int x,y,r1,rs,color;
    double a;
    int gdriver=DETECT,gmode;
    initgraph(&gdriver,&gmode,"..\\bgi ") 启; /*图形工作方式*/
    r1=atoi(argv[1]); /*计算基础圆半径*/
    rs=atoi(argv[2]); /*计算同心圆半径*/
    color=atoi(argv[3]); /*背景色*/
    cleardevice(); /*清除图形屏幕*/
    setbkcolor(color); /*设置背景色*/
    setcolor(4); /*设置图形显示颜色*/
    for(a=0;a<=2*pi;a+=pi/18) /*绘制同心圆*/
    {
        x=r1*cos(a)+320;
        y=r1*sin(a)+240;
        circle(x,y,rs); /*以圆心坐标为 x、y，半径为 rs 画圆*/
    }
    getch(); /*等待按键继续*/
    close_graph(); /*关闭图形工作方式*/
    return 0;
}
```

若程序名为 L6-29.c，经编译、连结生成可执行文件 L6-29.exe。在操作系统的环境下运行程序，命令行方式为：

16-29 40 20 3

则命令行与 main() 函数的参数关系为：argv[0] 是程序名，argv[1] 是 r1 的值，argv[2] 是 rs 的值，argv[3] 是屏幕的背景色。

由于指针数组均存放字符串，所需的圆半径及背景. 色彩通过 atoi() 函数转换为整型。

通过带参数的 main() 函数，我们可以为自己的程序设置口令，在运行程序的命令行中给出所需的口令，正确则继续，否则退出。

[例 2] 将上述程序作修改，在程序的入口处添置密码，若给定密码正确，则显示图形。

```
#include<graphics.h>
#include <math.h>
#define pi 3.1415926
int m a i n (int argc;char *argv[];)
{
    int x,y,r1,rs,color;
    double a;
    int gdriver=DETECT,gmode;
    if (strcmp(argv[1],"pass")!=0) 设/置* 口令的比较*/
    { printf("password error!\n");
    e x i t ( 0 ) ;
    }
    initgraph(&gdriver,&gmode,"..\\bgi ");
    r1=atoi(argv[2]);
    rs=atoi(argv[3]);
    color=atoi(argv[4]);
    cleardevice();
    setbkcolor(color);
    setcolor(4);
    for(a=0;a<=2*pi;a+=pi/18)
    {
        x=r1*cos(a)+320;
        y=r1*sin(a)+240;
        circle(x,y,rs);
    }
    getch();
    closegraph();
    return 0;
}
```

在操作系统的环境下运行程序， 命令行中增加口令“pass”， 命令行方式为：16-30 pass 20 40 3

若给定字符串 argv[1] 的值是 pass，则程序正确运行，否则程序退出。