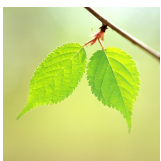
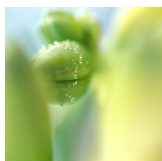
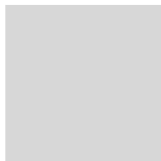
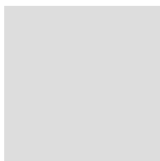


C 语言程序设计

第 1 章 初识 C 语言



Contents



任务 1 关机程序

1

任务描述

2

知识准备

3

知识应用

4

任务实现



任务描述



本任务设计完成一个计算机关机软件。程序运行后，计算机自动处于关机倒计时（120 秒），当用户输入正确的指令后计算机解除关机状态，否则到达时间后计算机自动关机。

即将注销你的登录

Windows 将在 2 分钟后关闭。

2015年11月8日 18:50:29 开始关闭。

关闭(C)

请输入正确口令：123
Press any key to continue

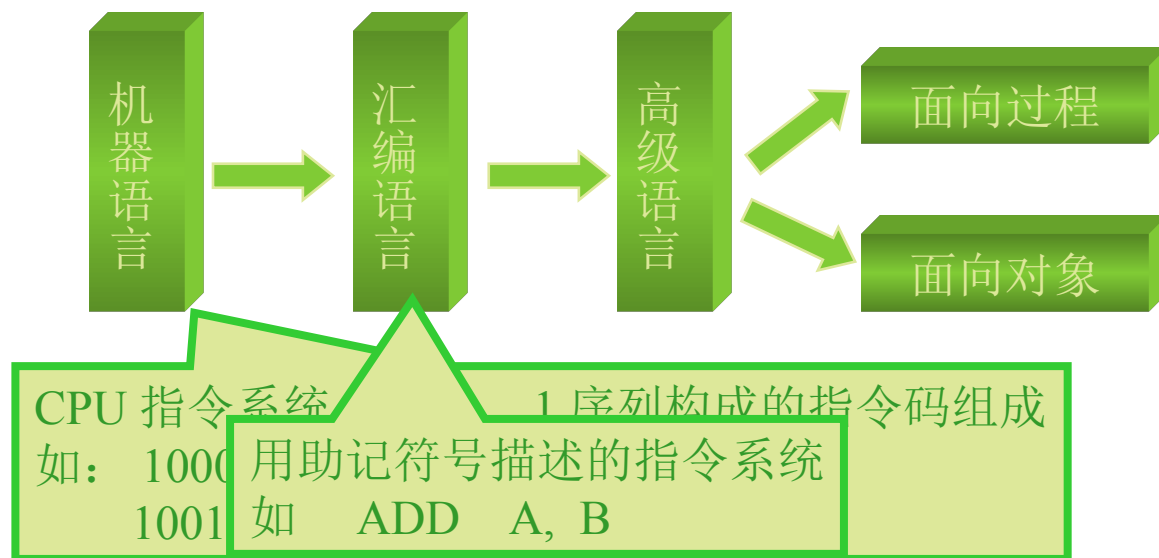


知识准备



一、C语言的历史背景

— 程序设计语言的发展



知识准备



一、C语言的历史背景

- 时间 : 1972~1973
- 地点 : 美国贝尔实验室

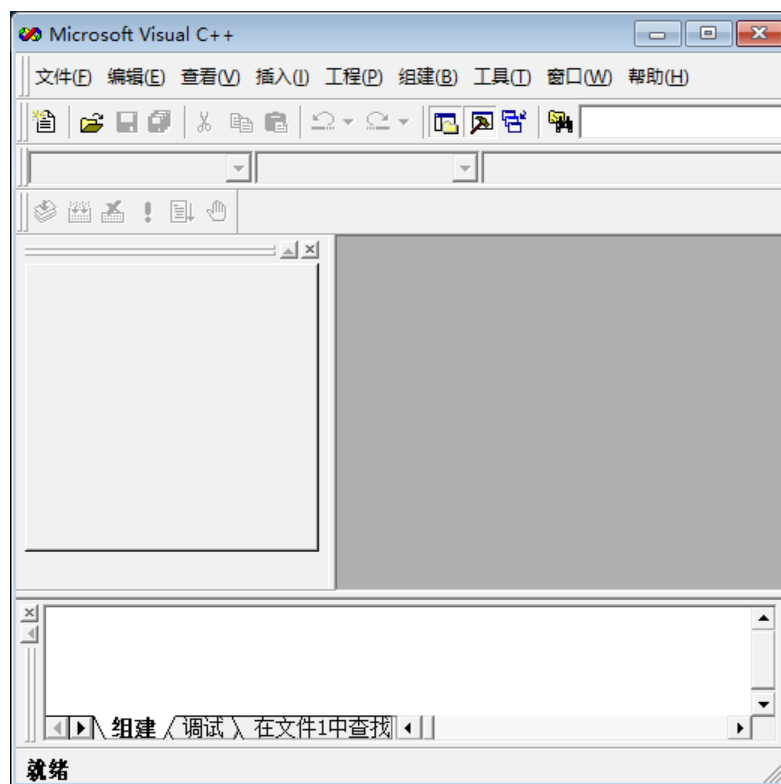


知识准备



二、利用 VC6.0 开发 C 程序的基本过程

1. 启动 VC6.0 应用程序：可通过 windows 的开始菜单或者桌面快捷方式运行 VC6.0 应用程序

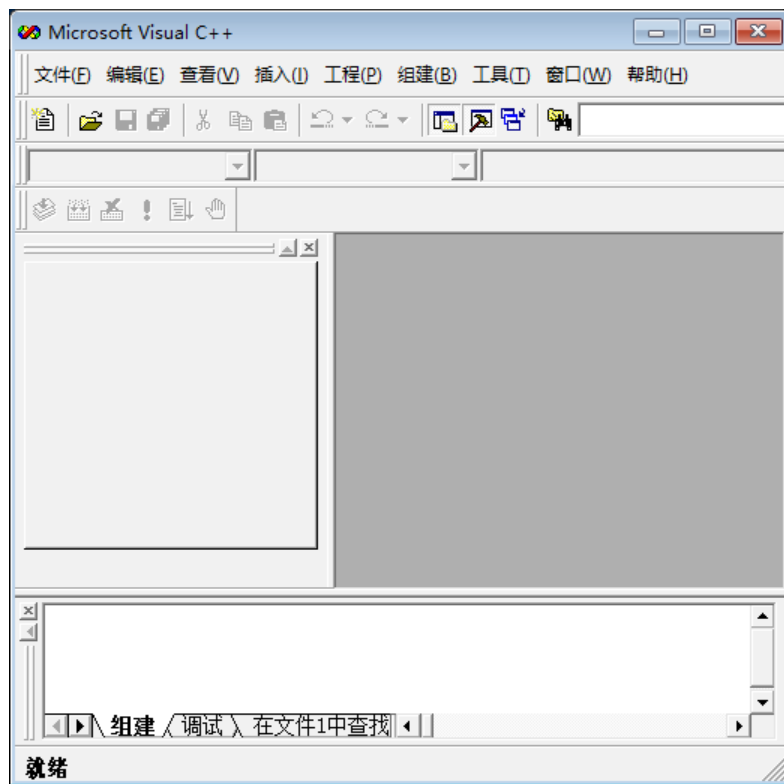


知识准备



二、利用 VC6.0 开发 C 程序的基本过程

2. 创建工程：点击文件 -> 新建



知识准备



二、利用 VC6.0 开发 C 程序的基本过程

2. 创建工程：选择创建 Win32 Console Application 工程，并输入要创建的工程名称，选择工程所在的位置



知识准备



二、利用 VC6.0 开发 C 程序的基本过程

2. 创建工程：选择一个空的工程，点击完成

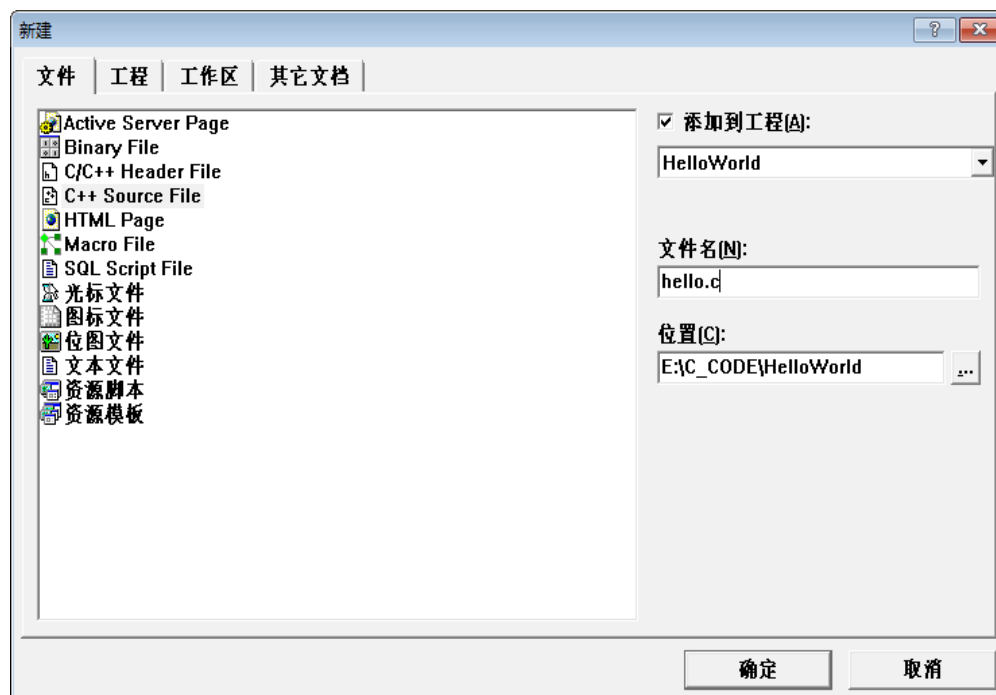


知识准备



二、利用 VC6.0 开发 C 程序的基本过程

3. 创建 C 语言源文件：点击文件 -> 新建



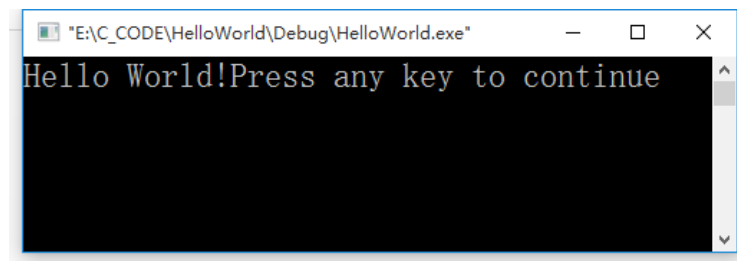
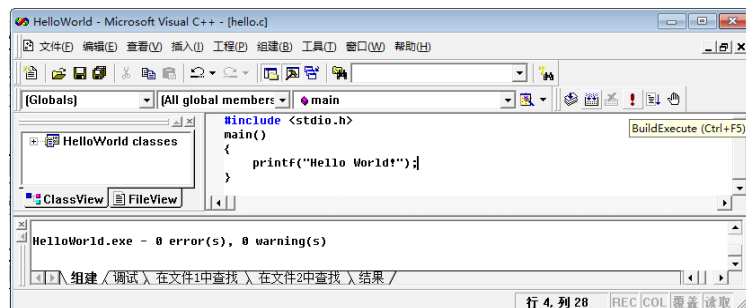
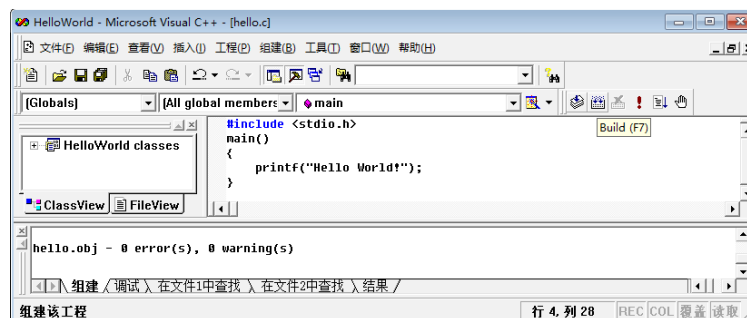
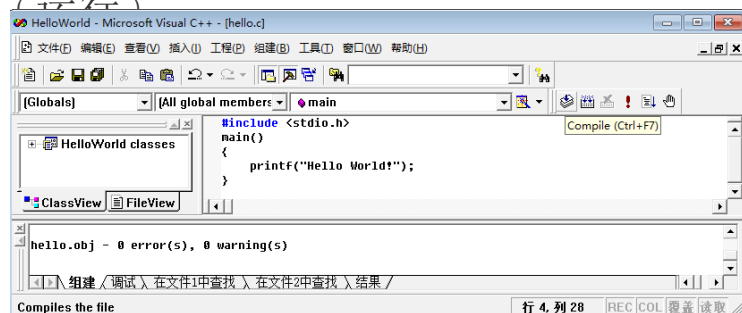
知识准备



二、利用 VC6.0 开发 C 程序的基本过程

4. 编写 C 语言程序：根据不同需求完成 c 源代码的编写，点击 **Compile**（编译）图标 - **Build**（组建）图标 - **BuildExecute**

(运行)



知识准备



三、VC6.0 开发环境

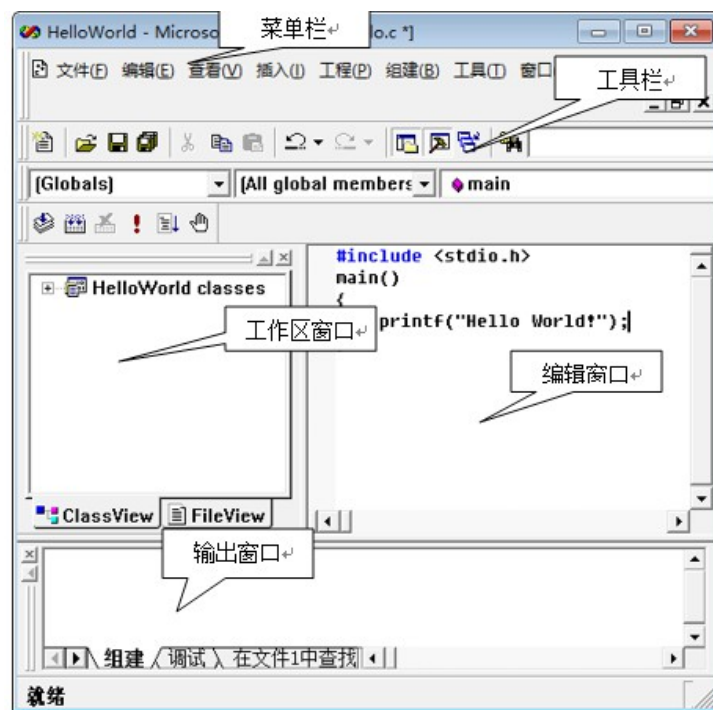
菜单栏：包含文件、编辑、查看、插入、工程、组建、工具、窗口、帮助等选项，包含 VC6.0 所能提供的所有功能。

工具栏：显示工程在开发过程中常用的一些工具选项。

工作区窗口：以树形结构列出当前工程所包含的所有成分。

编辑窗口：可以同时有多个编辑窗口，用来编译程序代码。

输出窗口：用来显示程序编译、组建过程信息。



知识准备



三、VC6.0 开发环境

VC6.0 常用的快捷键

快 捷 键	功 能
Ctrl+N	新建工程、文件
Ctrl+S	保存源程序文件
Ctrl+F7	编译当前工程
F7	组建工程
Ctrl+F5	执行程序
Ctrl+Break	停止编译
F5	开始或继续程序
Alt+F8	C 语言的风格格式化代码



知识准备



四、C 程序的构成

1. C 程序是函数构成的，其基本单位是函数；
2. 一个 C 程序由一个或多个函数组成，但其中必须有且只能有一个主函数即 `main()`；
3. 一个函数由函数的首部和函数体两部分组成。
4. 函数名后必须有对圆括号“`()`”，这是函数的标志。如 `main()`，`printf()`；
5. 函数体必须由一对大括号“`{ }`”括起。一个函数至少有一对大括号，若有多个大括号，则最外层的一对为函数体的范围。函数体可以没有，这样的函数称为空函数；
6. C 语言本身没有输入输出语句。输入和输出的操作是由库函数 `scanf()` 和 `printf()` 等函数来完成的。



知识准备



五、C 程序的特点

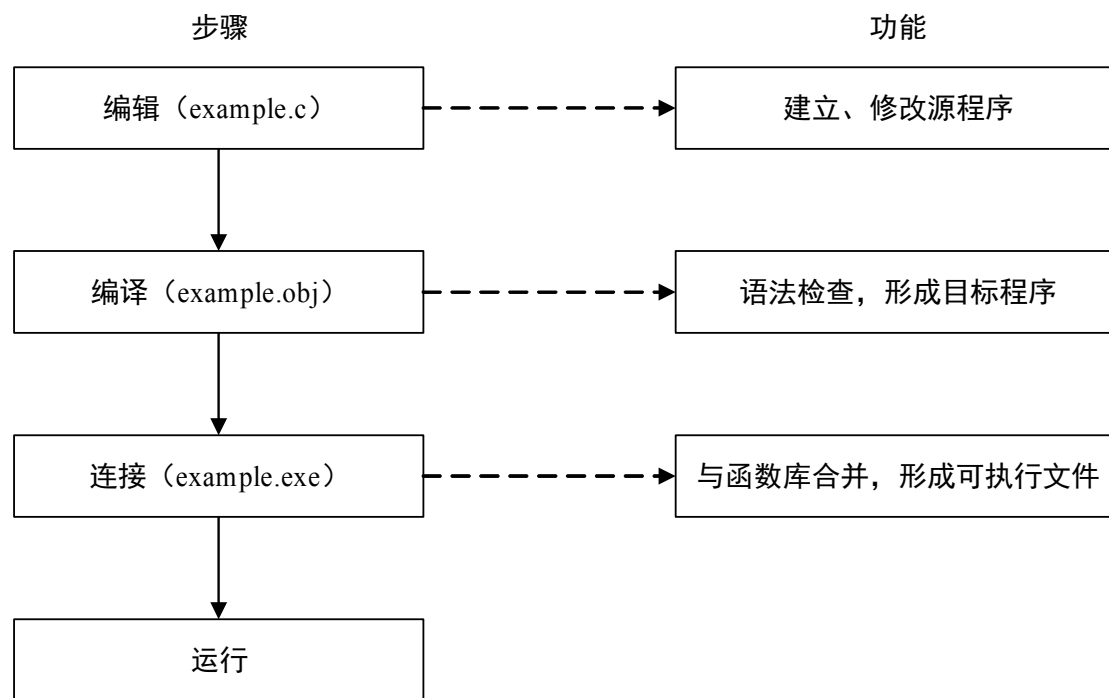
1. C 程序中所有函数的位置都是任意的，但总是从主函数开始执行到主函数结束；
2. 一行内可以写多条语句，一条语句也可写在多行上，用“\”作续行符；
3. 分号“；”是 C 语句的必要组成部分，即每条语句都必须以分号
4. 可以用 `/*.....*/` 对程序中的任何部分作注释，其作用是增强程序的可读性。



知识准备



六、C 程序的运行过程



知识应用



1. 在屏幕上显示“HELLO WORLD!”。

```
#include <stdio.h>    /* 当前程序包含标准输入输出信息库 */
main()
{
    printf("HELLO WORLD!");    /* 标准输出语句，向屏幕
    上打印双引号内信息 */
}
```



知识应用



2. 在屏幕上显示一个小猪的卡通形象。

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    printf("  ^ ^ ^ \n");
```

/*\n 是转义字符，表示换行 */

```
    printf("{/ o o /}\n");
```

```
    printf(" ( oo ) \n");
```

```
    printf("  ^ ^ ^ \n");
```

```
}
```



知识应用



3. 在屏幕上做一个简单的刷屏程序。

```
#include <stdio.h>
main()
{
    /*while 是循环语句，控制循环内语句反复执行 */
    while(1)
        printf(" 刷屏幕啦 ~ !  ");
}
```



任务实施



一、任务流程分解

1. 程序流程描述：程序开始执行后，程序会自动执行计算机关机命令（倒计时 120 秒），要求用户输入正确的关机解除口令（ 123 ），当输入正确时，解除计算机关机，否则继续执行关机命令。

（ 1 ） 程序初始化分析：产生计算机关机命令，定义关机解除命令存储空间

（ 2 ） 数据录入分析：用户录入关机解除命令

（ 3 ） 数据处理分析：用户输入命令和用户预设命令比较

（ 4 ） 输出结果分析：

结果 1：用户输入命令和用户预设命令相同，执行解除关机命令；

结果 2：用户输入命令和用户预设命令不相同，继续运行关机命令。



任务实施



二、知识扩展

1、调用系统命令 `system()` :

(1) 包含头文件: `stdlib.h`

所谓文件包含,是指在一个文件中将另一个文件的全部内容包含进来。文件包含命令的功能是把指定的文件插入该命令行的位置取代该命令行,从而把指定的文件包含进当前的源程序文件中,并连成一个源文件。

文件包含的一般形式:

`#include` “文件名” 或 `#include` < 文件名 >



任务实施



三、代码实现

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    int password; /* 定义输入密码的存储变量 */
    system("shutdown -t 120 -s"); /* 120 秒后关闭计算机 */
    printf(" 请输入正确口令: "); /* 提示信息 */
    scanf("%d",&password); /* 录入口令 */
    if(password==123) /* 判断录入口令与预设口令是否相同 */
        system("shutdown -a"); /* 相同时解除关机 */
}
```



任务实施



三、结果演示（ Windows10 环境下）

即将注销你的登录

Windows 将在 2 分钟后关闭。

2015年11月8日 18:50:29 开始关闭。

关闭(C)

请输入正确口令： 123
Press any key to continue



总结



本章重点内容为掌握 C 语言运行环境、C 语言的基本单位、C 程序的构成。C 语言的基本单位是函数，一个函数由两部分组成，函数的首部与函数体。一个 C 程序总是从 `main` 函数开始执行的。

本章难点内容为 C 程序的运行过程，主要包括编辑、编译、连接和运行；编译源程序时选择“`Compile`”按钮，则进行编译，得到一个后缀为 `.obj` 的目标程序。然后再选菜单“`Build`”按钮，进行连接操作，可得到一个后缀为 `.exe` 的可执行文件。也可通过“`BuildExecute`”按钮，实现程序的编译和运行测试。



C 语言程序设计



Thank You!

