**概述**

**C语言的起源：**

贝尔实验室的Dennis Ritchie在1972年发明了C，当时他正在与Ken Thompson一起设计UNIX系统。C语言来自于B语言。

**设计性：**

强大的控制结构、快速、程序更小、可移植到其他计算机

**高效性：**

C是一种高效的语言。在设计上它充分利用了当前计算机在能力上的优点。C程序往往很紧凑且运行速度很快。事实上，C可以表现出通常只有汇编才具有的精细控制能力（汇编是特定的CPU设计所采用的一组内部指令的助记符，不同的CPU使用不同的汇编）。如果愿意，您可以细调程序以获得最大的速度或最大内存使用率

**可移植性：**

C是一种可移植性语言，这意味着在一个系统上编写的C经过很少的改动或不经改动就能在别的系统上再次运行起来

**计算机工作的基本原理：**

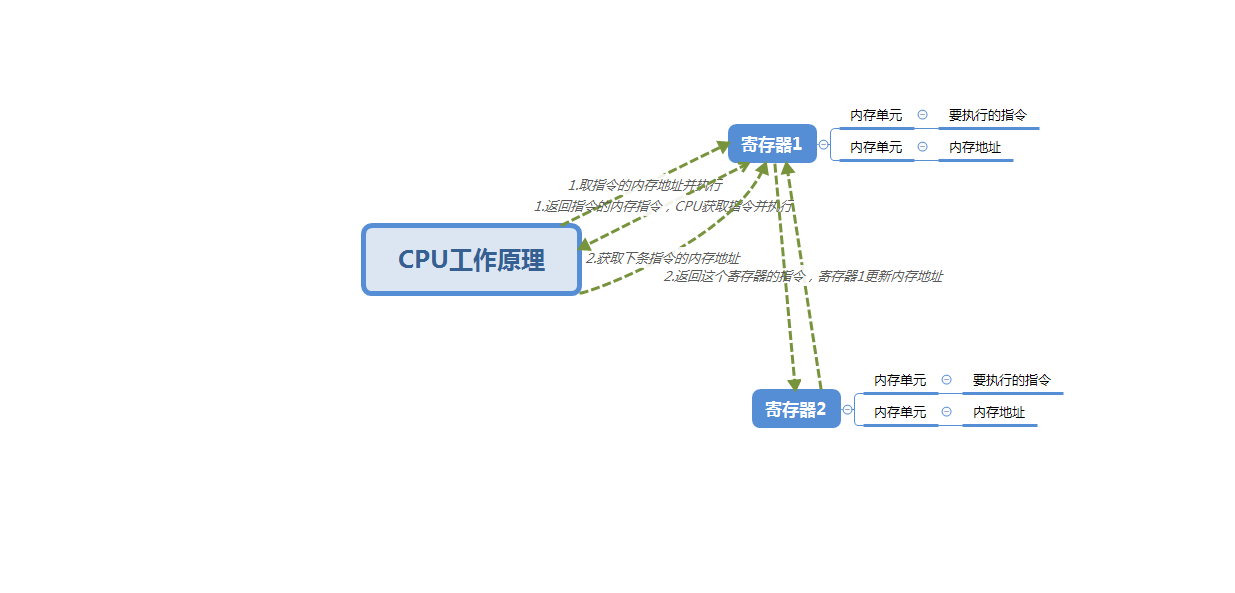
CPU：现代计算机可分为几个部件。中央处理单元（或称CPU）担负着绝大部分的计算机工作；（CPU负责处理程序）

内存条：随机访问存储器（或称RAM）作为一个工作区来保存程序和文件；

硬盘：永久存储器，即使计算机关机时也能记下程序和文件；

外围设备：键盘、鼠标、监视器等，用于提供人与计算机之间的通信

CPU的工作：它从内存中获取一个指令并执行该指令，然后从内存中获取下一个指令并执行。一个千兆CPU可以在一秒内进行大约一亿次这样的操作，所以CPU能以惊人的速度来从事其枯燥的工作。CPU有自己的小工作区，该工作区由若干个寄存器组成，每个寄存器可以保存一个数。一个寄存器保存下一条指令的地址，CPU使用该信息获取下一条指令。获取一条指令后，CPU在另一个寄存器中保存该指令并将第一个寄存器的值更新为下一条指令的地址。CPU只能理解有限的指令（指令集）。还有，这些指令是相当具体的，其中许多指令要求计算机将一个数从一个位置移动到另外一个位置，例如：从内存单元移动到寄存器



存储在计算机中的一切内容都是数字，数字是以数字形式存储的，字符也是用数字形式存储的，每个字符都有一个数字代码。计算机装载到寄存器的指令是以数字形式存储的，指令集中的每条指令具有一个数字代码。其次，计算机程序最终必须以这种数字指令代码（或称机器语言）来表示

计算机运行方式的一个结果是：如果希望计算机去做某件事，就必须提供一个特定的指令列表（一套程序）确切的告诉计算机要做的事及如何去做。必须以一种计算机可以直接理解的语言（机器语言）来创建该程序，这是一项繁琐、乏味、费力的任务。即使两个数字相加这种简单的事情也必须被分为若干个步骤：

1. 将内存单元为200中的数字复制到寄存器1
2. 将内存单元为2004中的数字复制到寄存器2
3. 将寄存器2的内存加到寄存器1的内容上，答案保留在寄存器1中
4. 将寄存器1的内容复制到内存单元2008

**使用C语言的七个步骤：**

定义程序目标->设计程序->编写代码->编译->运行程序->测试和调试程序->维护和修改程序

**定义程序目标：**

在开始时，希望程序做什么要有一个清晰的想法，考虑程序需要的信息、查询需要进行的计算和操作、以及程序应该向操作者报告的信息

**设计程序：**

设置好概念后，程序如何去完成它？用户界面是怎么样的，程序该如何组织，目标用户是谁，有多长时间来完成这个程序

PS：应该用一般的概念来考虑问题，而不是考虑具体代码，但是某些绝策要取绝于语言的一般特征

**编写代码：**

使用编辑器来编写规范的代码

C简单的例子：

#include <stdio.h>

Int main(void){

Int dogs;

Printf(“How many dogs do you have?\n”);

Scanf(“%d”,&dogs);

Printf(“So you have %d dog(s)!\n”,dogs);

Return 0;

}

**编译：**

将源代码通过编译器编程成系统可执行文件，在多数系统上编译器运行链接器。最后形成一个包含计算机可以理解的代码并且您能够运行的可执行文件。编译器还能检测程序是否有错误和警告，如果有错误的话编译器将错误信息返回给操作者，并不生成可执行文件

**运行程序：**

执行编译后的可执行文件

**测试和调试程序：**

程序运行是一个好的迹象，但有可能运行的不正确，此时应该进行检查。并修改程序

**维护和修改程序：**

一个好的程序需要不断的改进和测试，或许自己使用的时候没有遇见到bug。当你的程序广泛的被人使用的时候就可能会被发现存在操作者没发现的bug，这时就要修改

**编程机制：**

编写程序时必须遵守的确切步骤并取决于您的计算机环境。因为C是可移植的，所以它在许多环境可以用，包括：UNIX，LINUX，MAC，WINDOWS，MS-DOS

用C编写一个程序的时候，将编写的内容保存在一个叫源代码的文本文件中，后缀名为c

类似于：demo.c。名字最后不要取的太长

**目标文件、可执行文件和库：**

C编程的基本策略是使用程序将源代码文件转换为可执行文件，此文件包含可运行的机器语言代码。C分两步来完成这一工作：编译和链接

编译器：编译器将源代码转换为中间代码，链接器将此中间代码与其他代码相结合来生成可执行文件

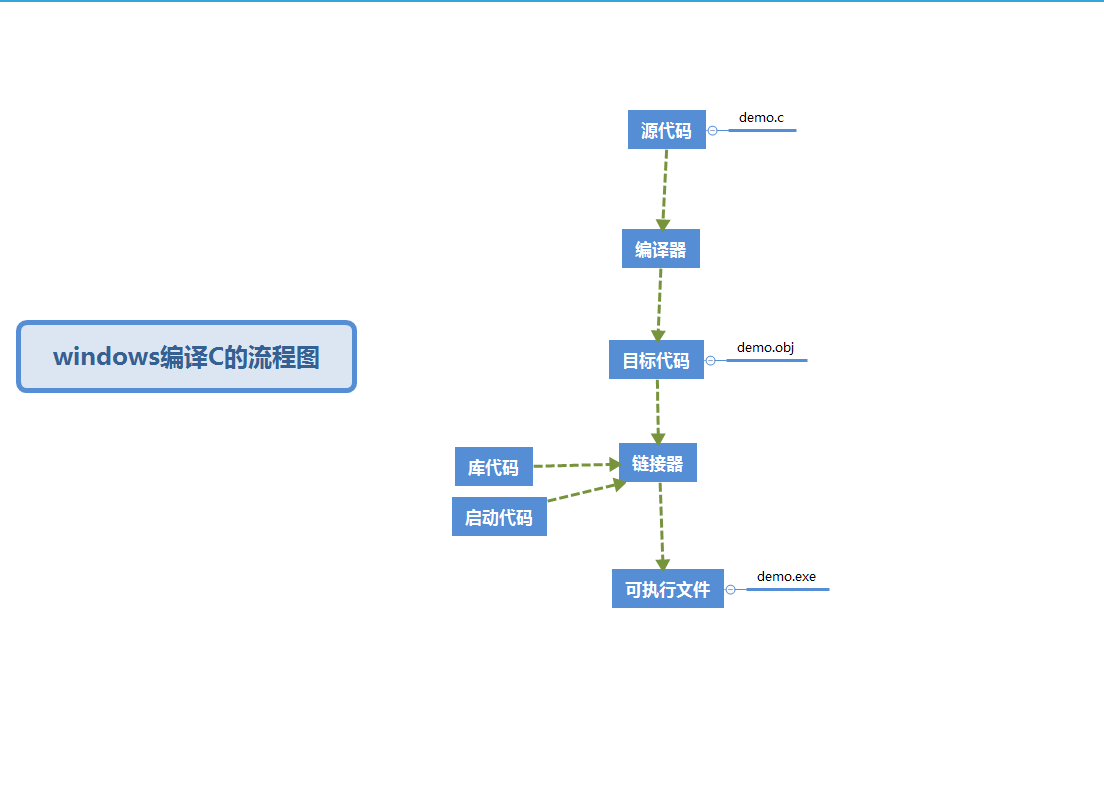
链接器：链接源代码和其他代码生成可执行文件

中间文件：将源代码转换成机器语言代码，将结果放置一个目标文件中。虽然目标文件包含机器语言代码，但该文件还不能执行，目标文件包含源代码的转换结果，但它不是一个完整的程序

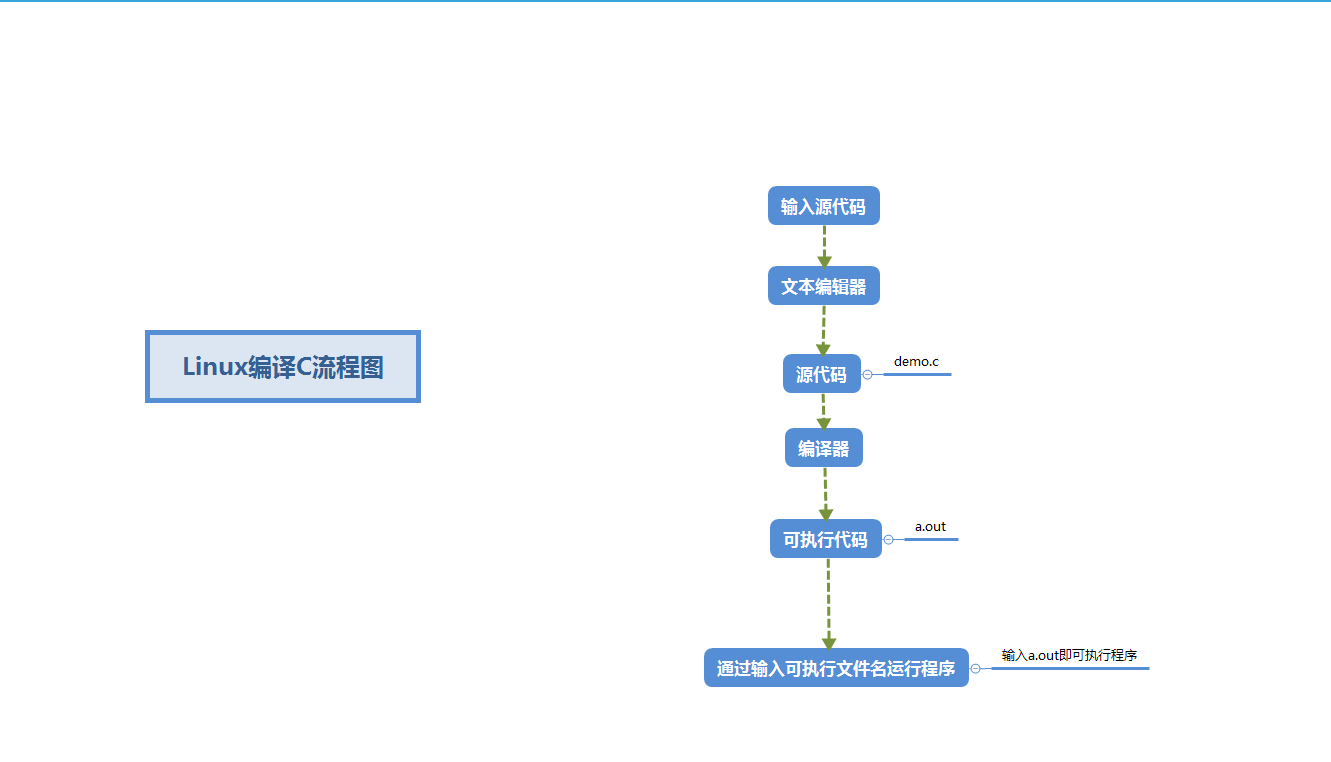
启动代码：此代码相当于程序和系统之间的接口，DOS与Linux要使用不同的启动代码，因为这两种系统处理程序的方法是不一样的

库代码：所以缺少的第二个元素是库列代码，几乎所有C都用C标准库，我们上面使用的printf()只是一个函数，存储在某个库中

Windows编译C的流程图：



Linux编译C的流程图：



**Linux上的编辑器：**

Vi,vim,Emacs,Jetbrains IDE,VSC

**Windows上的编辑器：**

Vs,Vsc,Jetbrains IDE,notepad++

**需要安装的编译器：**

Gcc

Gcc编译源码的方式：

Gcc demo.c