1.对于编程而言,可移植性意味着什么?

解答:计算机系统的硬件系统在指令系统和编码格式上有很大差异,开发的可执行应用程序无法在所有平台上运行.为了保证开发的程序能够兼容其他平台,针对相应的目标平台,需要将高级程序设计语言的源代码文件进行修改和再次编译.这样才能够生成目标系统的适配可执行应用程序.程序设计语言的可移植性就是指通过编译器,将源代码编译,生成对应目标系统的可执行程序.C语言历史悠久,在多种软硬件平台上都有广泛的支持,C99,C11语言规范也能够很好地向下兼容.因此,应用C语言进行程序开发,略加修改C语言源代码,就可以编译,生成多种目标系统的应用程序.

2.源代码文件,目标代码文件和可执行文件有什么区别?

解答:源代码文件是指由高级程序语言编写的指令文件.源代码文件是符合高级程序设计语言规范的高级指令系统.因此不能直接被计算机系统识别和运行,而需要通过编译器,将源代码编译,生成计算机能够直接识别和运行的机器语言的指令码.这种机器语言的指令码系统组成的文件能够直接被计算机系统识别和执行,因此通常也称为可执行文件.

在C语言程序设计中,程序员使用的外部库函数无法通过一次编译操作就生成完整的可执行文件.在代码编译过程中首先将源代码编译成一种叫做目标代码的中间代码文件.目标代码中部分标准库函数的代码部分临时空缺.在链接阶段为目标代码文件添加对应的系统标准启动代码和库代码,组合,生成最重的可执行文件

3.编程的7个主要步骤是什么?

解答:(1)定义程序的目标

(2)设计程序

(3)编写代码

(4)编译

(5)运行程序

(6)测试和调试程序

(7)维护和修改代码

4.编译器的任务是什么

对于编译型的高级程序设计语言,通常所说的编译是指将以高级程序设计语言编写的源代码,转换成目标平台的机器语言代码的过程.对于C语言或其他部分语言,编译器的编译工作一般分为两个步骤—编译和链接.其中编译是将源代码转为成目标代码的过程.目标代码文件不是一个完整的可执行文件,其中还缺少库代码和启动代码.目标代码文件必须通过链接器将中间代码和其他运行库代码合并才能形成目标平台的可执行文件.

5.链接器的任务是什么

链接器的主要工作是将编译器形成的中间代码,编译系统原有的系统库代码和其他一些第三方代码合并,形成目标平台的可执行文件.通过编译器和链接器的分步编译,首先能够尽量提高代码的可重用性和代码的可移植性.其次,也能够提高编译效率.原有系统库代码可以补编译,部分无修改的代码也可以直接使用原有的目标文件,直接通过高效率的链接形成可执行文件