

13015 计算机系统原理

第1章 计算机系统概述

13015 计算机系统原理【第一章考点解析】

考点1 存储程序

存储程序方式的**基本思想**是：

必须将事先编好的**程序**和**原始数据**送入**主存**后才能执行程序，一旦程序被启动执行，计算机**不需要操作人员干预**就能**自动完成逐条指令**取出和执行的**任务**。

比如：北京地铁19号线牛街站 B 出口**机器人摊煎饼**

13015 计算机系统原理【第一章考点解析】

考点2 冯·诺依曼结构基本思想

冯·诺依曼结构计算机的**基本思想**主要包括以下**四个方面**

- 1) 采用“**存储程序**”工作方式。
- 2) 计算机由**运算器**、**控制器**、**存储器**、**输入设备**和**输出设备**五个基本部件组成。
- 3) 存储器能存放**数据**，也能存放**指令**。
- 4) 计算机内部以**二进制**形式表示**指令**和**数据**。

13015 计算机系统原理【第一章考点解析】

考点2 冯·诺依曼结构基本思想

2. 冯·诺依曼计算机工作方式的基本特点是

- A. 多指令流单数据流
- C. 堆栈操作

- B. 存储程序工作方式
- D. 存储器按内部选择地址

【答案】：B【2024年10月】

14. 冯·诺依曼结构计算机由运算器、_____、_____、输入设备和输出设备五大基本部件组成。

【答案】：控制器、存储器【2025年04月】

13015 计算机系统原理【第一章考点解析】

考点3 模型计算机的基本硬件结构

模型计算机的基本硬件结构如下：

- 1) **主存储器** (主存或内存)
- 2) **算术逻辑部件** (ALU)
- 3) **控制元件** (控制器)
- 4) **输入/输出设备**

总线：

- 1) **地址线**：用于传输**地址**信息
- 2) **数据线**：用于传输**数据**信息
- 3) **控制线**：用于传输**控制**信息

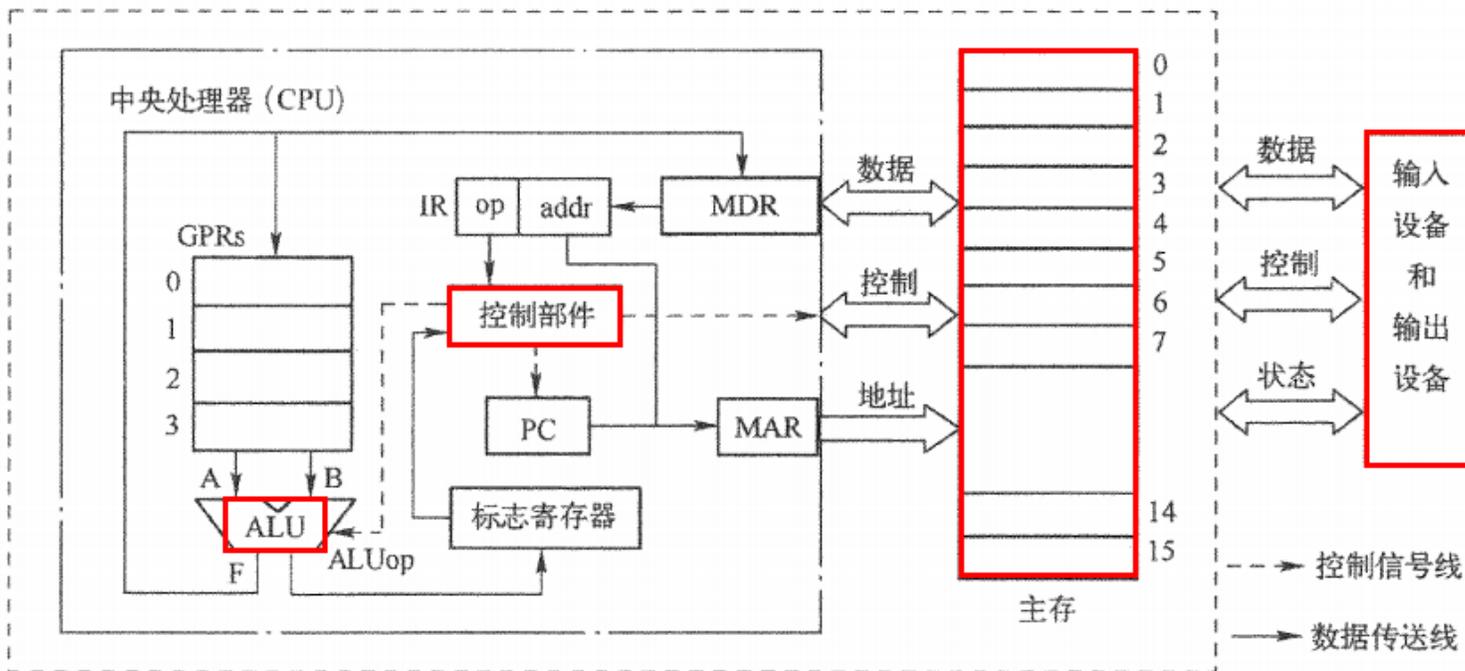


图 1.1 模型计算机基本硬件结构

13015 计算机系统原理【第一章考点解析】

考点3 模型计算机的基本硬件结构

4. 通用寄存器用于临时存放从主存取来的数据或运算的结果,下列不属于通用寄存器的部件是
- A. 指令寄存器 B. 标志寄存器 C. 程序计数器 D. 状态计数器

【答案】：D【2025年04月】

12. CPU 访问主存时,需先将主存地址、读/写命令分别送到总线的_____、控制线,然后通过总线的_____发送或接收数据。

【答案】：地址线、数据线【2025年04月】

13015 计算机系统原理【第一章考点解析】

考点4 指令

指令是用 0 和 1 表示的一串 0/1 序列，用来指示 CPU 完成一个特定的**原子**操作。【不可分割】

- 1) **取数**指令 (load) **主存** → 通用寄存器 【**M型**指令】
- 2) **存数**指令 (store) 通用寄存器 → **主存** 【**M型**指令】 **存取**与**内存**相关
- 3) **加法**指令 (add) 通用寄存器 → 通用寄存器 【**R型**指令】
- 4) **传送**指令 (mov) 通用寄存器 → 通用寄存器 【**R型**指令】

每条**指令**由**操作码**和**地址码**两部分组成。

下面模型机采用**8**位**定长**指令字

格式	4位	2位	2位	功能说明
R型	op	rt	rs	$R[rt] \leftarrow R[rt] \text{ op } R[rs]$ 或 $R[rt] \leftarrow R[rs]$
M型	op	addr		$R[0] \leftarrow M[\text{addr}]$ 或 $M[\text{addr}] \leftarrow R[0]$

图 1.2 定长指令字格式

13015 计算机系统原理【第一章考点解析】

考点4 指令

格式	4位	2位	2位	功能说明
R型	op	rt	rs	$R[rt] \leftarrow R[rt] \text{ op } R[rs]$ 或 $R[rt] \leftarrow R[rs]$
M型	op	addr		$R[0] \leftarrow M[\text{addr}]$ 或 $M[\text{addr}] \leftarrow R[0]$

图 1.2 定长指令字格式

op为**操作码**字段：

1) **R型指令**的 op 为**0000**定义为**传送(mov)**操作，op 为**0001**定义为**加(add)**操作

比如：指令 **0001 0001**的功能为 $R[0] \leftarrow R[0] + R[1]$ ，表示将 0 号和 1 号寄存器内容相加的结果送 0 号寄存器。

2) **M型指令**的 op 为**1110**定义为**取数(load)**操作，op 为**1111**定义为**存数(store)**操作。

比如：指令 **1110 0110**的功能为 $R[0] \leftarrow M[0110]$ ，表示将 6 号主存单元（地址为 0110）中的内容取到 0 号寄存器

13015 计算机系统原理【第一章考点解析】

考点5 程序和指令的执行过程

在上述模型机上实现“ $z=x+y$ ”， x 和 y 分别存放在主存 5 号和 6 号单元中，结果 z 存放在 7 号单元中，则相应程序在主存单元中的初始内容如图：

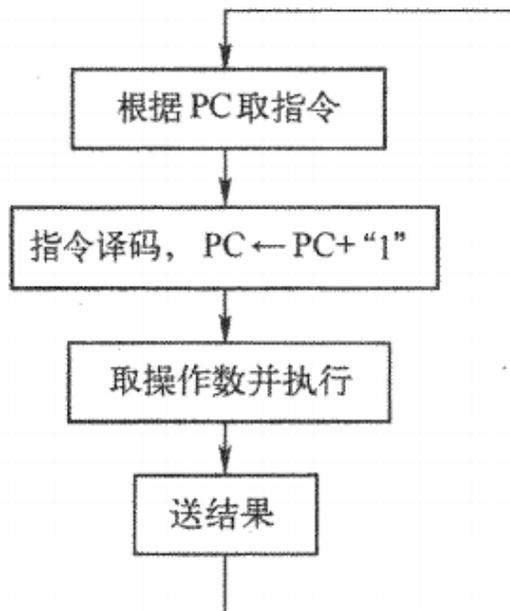
主存地址	主存单元内容	内容说明 (I_i 表示第 i 条指令)	指令的符号表示
0	1110 0110	$I_1: R[0] \leftarrow M[6]; op=1110$: 取数操作	load r0, <u>6</u> #
1	0000 0100	$I_2: R[1] \leftarrow R[0]; op=0000$: 传送操作	mov r1, r0
2	1110 0101	$I_3: R[0] \leftarrow M[5]; op=1110$: 取数操作	load r0, <u>5</u> #
3	0001 0001	$I_4: R[0] \leftarrow R[0] + R[1]; op=0001$: 加操作	add r0, r1
4	1111 0111	$I_5: M[7] \leftarrow R[0]; op=1111$: 存数操作	store 7#, r0
5	0001 0000	操作数 <u>x</u> , 值为 16	
6	0010 0001	操作数 <u>y</u> , 值为 33	
7	0000 0000	结果 z , 初始值为 0	

图 1.3 实现 $z=x+y$ 的程序在主存部分单元中的初始内容

13015 计算机系统原理【第一章考点解析】

考点5 程序和指令的执行过程

程序执行过程如右图：



每条指令的执行过程及结果如下图：

指令阶段	I1: 1110 0110	I2: 0000 0100	I3: 1110 0101	I4: 0001 0001	I5: 1111 0111
取指令	$IR \leftarrow M[0000]$	$IR \leftarrow M[0001]$	$IR \leftarrow M[0010]$	$IR \leftarrow M[0011]$	$IR \leftarrow M[0100]$
指令译码	op=1110, 取数	op=0000, 传送	op=1110, 取数	op=0001, 加	op=1111, 存数
PC 增量	$PC \leftarrow 0000+1$	$PC \leftarrow 0001+1$	$PC \leftarrow 0010+1$	$PC \leftarrow 0011+1$	$PC \leftarrow 0100+1$
取数并执行	$MDR \leftarrow M[0110]$	$A \leftarrow R[0]$, mov	$MDR \leftarrow M[0101]$	$A \leftarrow R[0]$, $B \leftarrow R[1]$, add	$MDR \leftarrow R[0]$
送结果	$R[0] \leftarrow MDR$	$R[1] \leftarrow F$	$R[0] \leftarrow MDR$	$R[0] \leftarrow F$	$M[0111] \leftarrow MDR$
执行结果	$R[0]=33$	$R[1]=33$	$R[0]=16$	$R[0]=16+33=49$	$M[7]=49$

13015 计算机系统原理【第一章考点解析】

考点6 程序设计和翻译程序

程序设计语言从**抽象层次**上来分，可以分成**两类**：

1) 低级语言

① 机器语言 比如：1110 0110

② 汇编语言 比如：load r0, 6#

2) 高级语言 比如：C、C++、Java等

13015 计算机系统原理【第一章考点解析】

考点6 程序设计语言和翻译程序

计算机无法直接理解和执行高级编程语言程序，因而需要将**高级语言**程序转换成**机器语言**程序。这个转换过程通常由计算机自动完成，进行这种转换的软件统称为**翻译程序**。

翻译程序有以下 3 类：

- 1) **汇编程序**：用于将**汇编语言源程序**翻译成**机器语言目标程序**
比如：工具有微软 x86 汇编器
- 2) **解释程序**：用于将**源程序**中的语句按其**执行顺序逐条翻译**成机器指令并立即执行
比如：JavaScript、Python
- 3) **编译程序**：用于将**高级语言源程序**翻译成**汇编语言**或**机器语言目标程序**
比如：C、C++

13015 计算机系统原理【第一章考点解析】

考点6 程序设计语言和翻译程序

3. 下列不属于翻译程序种类的是

A. 汇编程序

B. 解释程序

C. 编译程序

D. 链接程序

【答案】：D【2024年04月】

11. 用若干个_____表示的与机器指令一一对应的指令称为_____指令。

【答案】：助记符、汇编【2024年04月】

12. 与机器语言相对应的符号化表示语言称为_____语言。通常用容易记忆的英文单词或缩写表示指令操作码的含义,用标号、变量名、寄存器名等表示操作数或其地址码,这些英文单词或其缩写、标号、变量名等称为_____。

【答案】：汇编、助记符【2024年10月】

13015 计算机系统原理【第一章考点解析】

考点6 程序设计语言和翻译程序

11. 从抽象层次上来分,程序设计语言可分为_____语言和_____语言。

【答案】：低级语言、高级语言【2025年04月】

12. 属于低级语言的是_____和_____。

【答案】：机器语言、汇编语言【2025年10月】

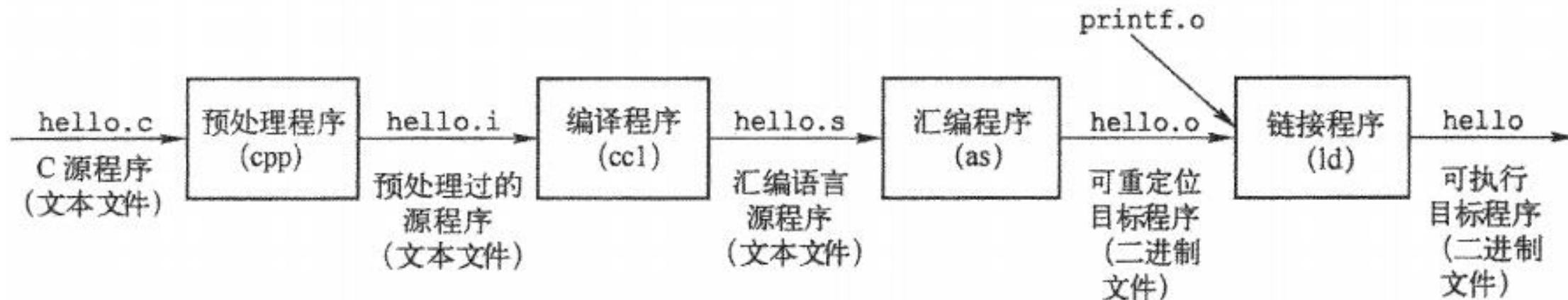
13015 计算机系统原理【第一章考点解析】

考点7 从源程序到可执行文件

右边是“hello.c”的C语言源程序代码

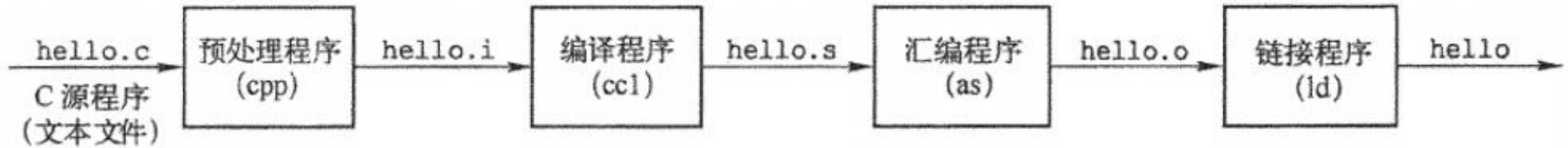
```
1  #include <stdio. h>
2
3  int main()
4  |
5      printf( "hello, world\n" );
6  |
```

从 hello.c 到可执行目标文件 hello 的转换过程如下图所示



13015 计算机系统原理【第一章考点解析】

考点7 从源程序到可执行文件



- 1) **预处理**阶段：预处理程序(**cpp**)对源程序中以字符“#”开头的命令进行处理
以 **.i** 为扩展名。→ **文本文件**
- 2) **编译**阶段：编译程序(**ccl**)对预处理后的源程序进行编译，生成一个**汇编语言源程序文件**
以 **.s** 为扩展名。→ **文本文件**
- 3) **汇编**阶段：汇编程序(**as**)对**汇编语言源程序**进行汇编，生成一个**可重定位目标文件**
以 **.o** 为扩展名。→ **二进制文件**，打开是乱码
- 4) **链接**阶段：链接程序(**ld**)将多个**可重定位目标文件**和**标准库函数库**中的**可重定位目标文件**合并成为一个**可执行目标文件**。→ **二进制文件**，打开是乱码

13015 计算机系统原理【第一章考点解析】

考点7 从源程序到可执行文件

6. 用来将若干可重定位目标文件组合起来,生成一个可执行目标文件的过程称为
- A. 预处理 B. 编译 C. 汇编 D. 链接

【答案】：D【2024年04月】

6. 从源程序变为可执行文件的第一个步骤是
- A. 链接 B. 编译
C. 汇编 D. 预处理

【答案】：D【2024年10月】

16. 将高级语言源程序转换为可执行文件通常分为_____、编译、汇编和_____4步。

【答案】：预处理、链接【2025年10月】

13015 计算机系统原理【第一章考点解析】

考点7 从源程序到可执行文件

17. 预处理是从_____变成_____文件的第一步。

【答案】：源程序、可执行【2025年10月】

13015 计算机系统原理【第一章考点解析】

考点8 计算机系统抽象层的转换

机器语言程序所运行的计算机**硬件**和**软件**之间需要有一个“**桥梁**”，这个在软件和硬件之间的界面就是**指令集体系结构**

(**I**nstruction **S**et **A**rchitecture, **ISA**)

简称**指令集架构**或**指令系统**

它是软件和硬件之间接口的一个完整定义。

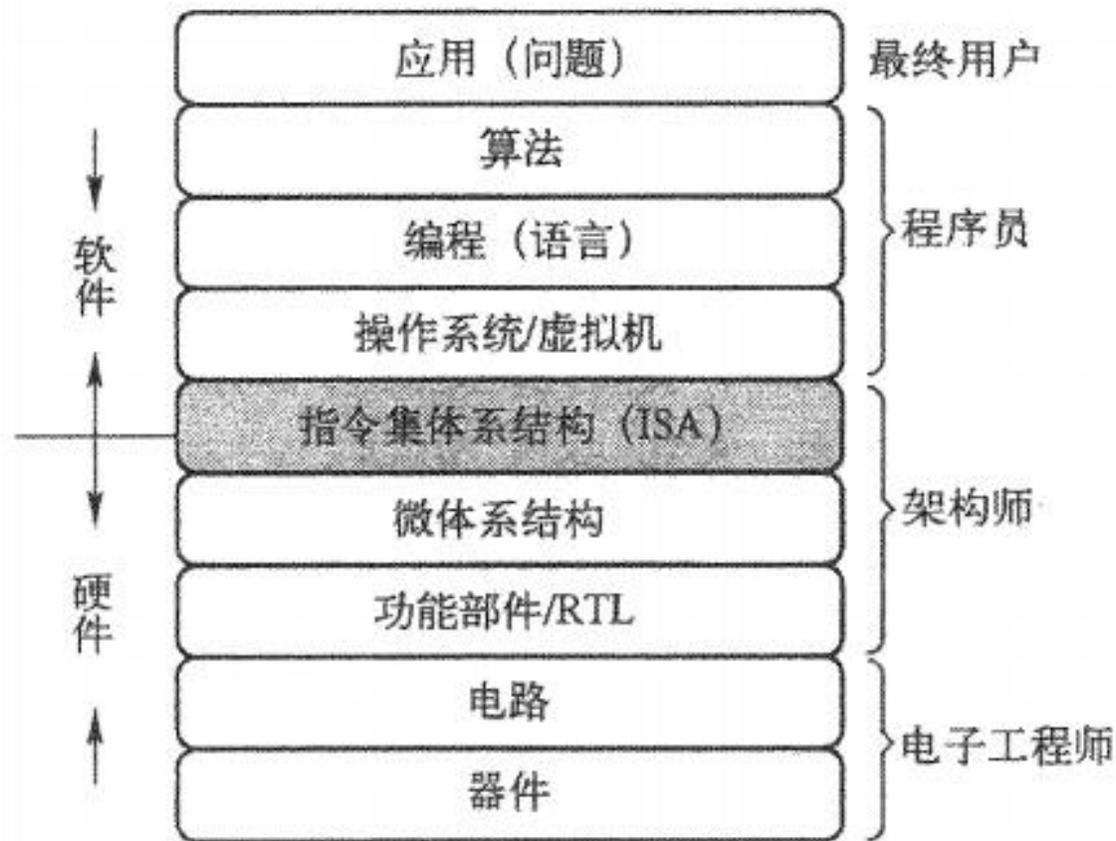


图 1.10 计算机系统抽象层及其转换

13015 计算机系统原理【第一章考点解析】

考点8 计算机系统抽象层的转换

1. 完整的计算机系统应包括

- A. 运算器、存储器、控制器
- C. 主机和实用程序

- B. 硬件和软件
- D. 外部设备和主机

【答案】： B【2024年04月】

12. 指令集体系结构(ISA)是整个计算机系统的核心部分,ISA 层上面是_____部分,下面是_____部分。

【答案】： 软件、硬件【2024年04月】

5. 下列不属于指令集体系结构设计所追求的目标的是

- A. 提高机器级程序的执行速度
- C. 缩短机器级指令的长度

- B. 增大控制存储器的容量
- D. 提高机器级程序设计的灵活性

【答案】： B【2025年04月】

13015 计算机系统原理【第一章考点解析】

考点8 计算机系统抽象层的转换

2. 机器语言程序所运行的计算机硬件和软件之间需要一个“桥梁”,这个在软件和硬件之间的界面是

A. 指令集体系结构

B. 微体系结构

C. 操作系统/虚拟机

D. 程序

【答案】：A【2025年10月】

13015 计算机系统原理【第一章考点解析】

考点9 计算机系统的不同用户

根据软件的用途，一般将软件分为两大类：

1) 系统软件

包括：**操作系统** (如 Windows、UNIX、Linux)、**语言处理系统** (如 Visual Studio、GCC)、**数据库管理系统** (如 Oracle)和**各类实用程序**(如磁盘碎片整理程序、备份程序)等软件。

2) 应用软件

例如，人们平时经常使用的电子邮件收发软件、多媒体播放软件、游戏软件、炒股软件、文字处理软件、电子表格软件、演示文稿制作软件等软件。

13015 计算机系统原理【第一章考点解析】

考点9 计算机系统的不同用户

按照在计算机上完成任务的不同，可以把使用计算机的用户分成以下 4 类：

1) 最终用户

例如：使用炒股软件的股民、玩计算机游戏的人、进行会计电算化处理的财会人员等。

2) 系统管理员

其职责主要包括：安装、配置和维护系统的硬件和软件

3) 应用程序员

例如：C++开发工程师，Java高级程序员

4) 系统程序员

例如：开发操作系统、编译器、数据库管理系统等系统软件的程序员。

13015 计算机系统原理【第一章考点解析】

考点9 计算机系统的不同用户

5. 介于计算机硬件与应用程序之间,包括为有效、安全地使用和管理计算机以及为开发和运行应用软件而提供的各种软件,称为
- A. 应用软件 B. 系统软件 C. 操作软件 D. 用户软件

【答案】 : B **【2024年04月】**

13. 按照在计算机上完成任务的不同,可以把使用计算机的用户分成以下 4 类:
_____、_____、应用程序员和系统程序员。

【答案】 : **最终用户、系统管理员** **【2024年04月】**

13. 用来管理整个计算机系统的资源,包括对它们进行_____、管理、监视和服务等的软件称为_____。

【答案】 : **调度、操作系统** **【2024年10月】** P36

13015 计算机系统原理【第一章考点解析】

考点9 计算机系统的不同用户

1. 属于系统软件的是

A. WPS Office

C. RealPlayer

B. Windows 10

D. 腾讯 QQ

【答案】： **B** **【2025年10月】**

13015 计算机系统原理【第一章考点解析】

考点10 计算机系统核心层之间的关联

程序的编译转换如右图所示

应用程序接口，有两种

1) 应用程序**二进制**接口 → **ABI**

(**A**pplication **B**inary **I**nterface, **ABI**)

面向：开发人员和上层应用

2) 应用程序**编程**接口 → **API**

(**A**pplication **P**rogramming **I**nterface, **API**)

面向：编译器、操作系统和硬件

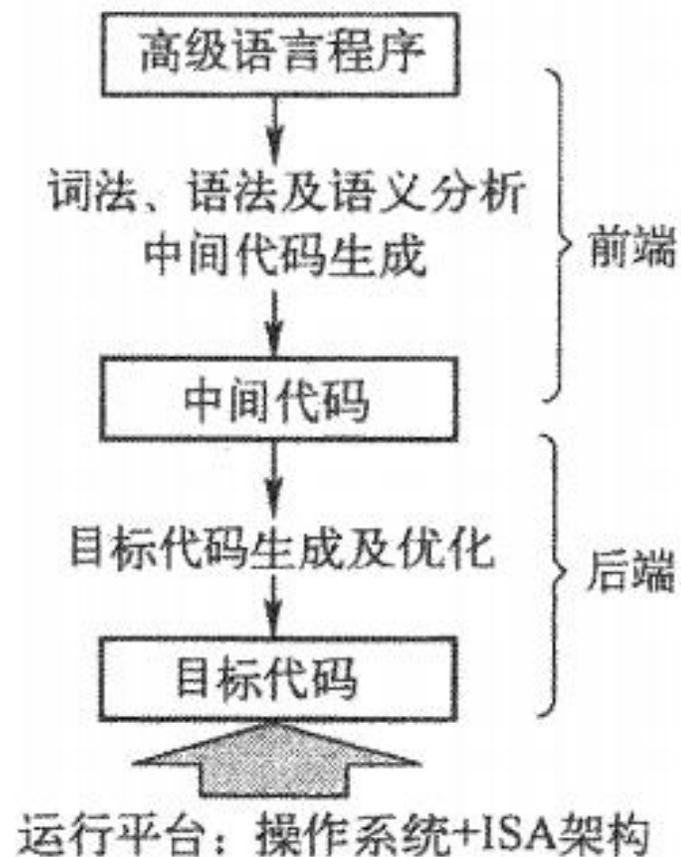


图 1.12 程序的编译转换

13015 计算机系统原理【第一章考点解析】

考点11 计算机性能的测试

考量一个计算机系统性能的两个基本指标

1) **吞吐率**

2) **响应时间**

通常把用户感觉到的**执行时间**分成以下两部分：

1) **CPU时间**

① **用户CPU时间**

② **系统CPU时间**

2) **其他时间**

13015 计算机系统原理【第一章考点解析】

考点11 计算机性能测试

CPU性能是指**用户CPU时间**，它只包含 CPU 运行用户程序代码的时间。在对 CPU 时间进行计算时需要用到以下几个重要的概念和指标：

1) **时钟周期**：CPU里最小的**时间单位**

比如：常见的 3.5GHz，就是 CPU 每秒有 35 亿次"心跳"(时钟脉冲)

2) **时钟频率**：CPU的**主频**，**时钟周期的倒数**

比如：3.5GHz 对应的时钟周期 ≈ 0.286 纳秒 (1秒 \div 35 亿)，也就是每 0.286 纳秒完成一次"心跳"(时钟脉冲)

3) **CPI(Cycles Per Instruction)**：表示执行一条指令所需的**时钟周期数**。

① 对于**一条特定指令**而言，此时 CPI 是一个**确定的值**；

② 对于**一个程序**或**一台机器**来说，其 CPI 指该程序或该机器指令集中的所有指令执行所需的**平均时钟周期数**，此CPI是一个**平均值**。

13015 计算机系统原理【第一章考点解析】

考点11 计算机性能测试

【基础计算公式】

如果**已知**程序**总指令条数**和**综合CPI**，则可用如下公式

$$\text{程序总时钟周期数} = \text{程序总指令条数} \times \text{CPI}$$

如果**已知**程序中**共有 n 种不同类型的指令**，第 **i 种指令的条数和 CPI 分别为 C_i 和 CPI_i** ，则

$$\text{程序总时钟周期数} = \sum_{i=1}^n (\text{CPI}_i \times C_i)$$

$$\text{用户CPU 时间} = \text{程序总时钟周期数} \div \text{时钟频率} = \text{程序总时钟周期数} \times \text{时钟周期}$$

程序的**综合CPI(平均CPI)**也可由以下公式求得，其中， **F_i** 表示第 **i 种指令在程序中所占的比例**。

$$\text{综合CPI} = \text{程序总时钟周期数} \div \text{程序总指令条数}$$

$$\text{综合CPI} = \sum_{i=1}^n (\text{CPI}_i \times F_i)$$

13015 计算机系统原理【第一章考点解析】

考点11 计算机性能测试

例 1.1 假设某个频繁使用的程序 P 在机器 M1 上运行需要 10 s，M1 的时钟频率为 2 GHz。设计人员想开发一台与 M1 具有相同 ISA 的新机器 M2。采用新技术可使 M2 的时钟频率增加，但同时也会使 CPI 增加。假定程序 P 在 M2 上的时钟周期数是在 M1 上的 1.5 倍，则 M2 的时钟频率至少达到多少才能使程序 P 在 M2 上的运行时间缩短为 6 s？

分析：

- 1) 程序 P 在机器 M1 上运行需要 10s，说明：M1 的**用户CPU时间**是 10s
- 2) M1 的时钟频率为 2GHz，结合【公式】**用户CPU时间** = 程序**总时钟周期数** ÷ **时钟频率**，可以求出程序 P 在 M1 上 **总时钟周期数**，程序**总时钟周期数** = **用户CPU时间** × **时钟频率**
- 3) 程序 P 在 M2 上的时钟周期数是 M1 的 1.5 倍，可以求出：程序 P 在 M2 上 **总时钟周期数**
- 4) 程序 P 在 M2 上的运行时间缩短为 6s，说明：M2 的**用户CPU时间**是 6s
- 5) **时钟频率** = 程序**总时钟周期数** ÷ **用户CPU时间**，可以求出：程序 P 在 M2 上的**时钟频率**

13015 计算机系统原理【第一章考点解析】

考点11 计算机性能的测试

例 1.1 假设某个频繁使用的程序 P 在机器 M1 上运行需要 10 s，M1 的时钟频率为 2 GHz。设计人员想开发一台与 M1 具有相同 ISA 的新机器 M2。采用新技术可使 M2 的时钟频率增加，但同时也会使 CPI 增加。假定程序 P 在 M2 上的时钟周期数是在 M1 上的 1.5 倍，则 M2 的时钟频率至少达到多少才能使程序 P 在 M2 上的运行时间缩短为 6 s？

解：

1) 程序 P 在机器 M1 上 **总时钟周期数** = $10\text{s} \times 2\text{GHz} = 20\text{G}$

2) 程序 P 在机器 M2 上 **总时钟周期数** = $20\text{G} \times 1.5 = 30\text{G}$

3) 要使程序 P 在 M2 上的运行时间缩短为 6s，

则 M2 的**时钟频率** = 程序**总时钟周期数** ÷ 用户 CPU 时间 = $30\text{G}/6\text{s} = 5\text{GHz}$

13015 计算机系统原理【第一章考点解析】

考点11 计算机性能测试

【基础计算公式】

如果**已知**程序**总指令条数**和**综合CPI**，则可用如下公式

$$\text{程序总时钟周期数} = \text{程序总指令条数} \times \text{CPI}$$

如果**已知**程序中**共有 n 种不同类型的指令**，第 **i 种指令的条数**和 **CPI** 分别为 **C_i** 和 **CPI_i** ，则

$$\text{程序总时钟周期数} = \sum_{i=1}^n (\text{CPI}_i \times C_i)$$

$$\text{用户CPU 时间} = \text{程序总时钟周期数} \div \text{时钟频率} = \text{程序总时钟周期数} \times \text{时钟周期}$$

程序的**综合CPI(平均CPI)**也可由以下公式求得，其中， **F_i** 表示第 **i 种指令**在程序中所占的**比例**。

$$\text{综合CPI} = \text{程序总时钟周期数} \div \text{程序总指令条数}$$

$$\text{综合CPI} = \sum_{i=1}^n (\text{CPI}_i \times F_i)$$

13015 计算机系统原理【第一章考点解析】

考点11 计算机性能测试

例 1.2 假设计算机 M 的指令集中包含 A、B、C 三类指令，其 CPI 分别为 1、2、4。某个程序 P 在 M 上被编译成两个不同的目标代码序列 P1 和 P2，P1 所含 A、B、C 三类指令的条数分别为 8、2、2，P2 所含 A、B、C 三类指令的条数分别为 2、5、3。哪个代码序列总指令条数少？哪个执行速度快？它们的 CPI 分别是多少？

分析：

- 1) 有 A、B、C 三类指令，其 CPI 分别为 1、2、4，结合【公式】程序总时钟周期数 = $\sum_{i=1}^n (CPI_i \times C_i)$
- 2) 比较哪个代码序列总指令条数？加和比大小即可
- 3) 比较哪个执行速度快？用户 CPU 时间
- 4) 它们的 CPI 分别是多少，就是求综合 CPI

综合 CPI = 程序总时钟周期数 ÷ 程序总指令条数

13015 计算机系统原理【第一章考点解析】

考点11 计算机性能的测试

例 1.2 假设计算机 M 的指令集中包含 A、B、C 三类指令，其 CPI 分别为 1、2、4。某个程序 P 在 M 上被编译成两个不同的目标代码序列 P1 和 P2，P1 所含 A、B、C 三类指令的条数分别为 8、2、2，P2 所含 A、B、C 三类指令的条数分别为 2、5、3。哪个代码序列总指令条数少？哪个执行速度快？它们的 CPI 分别是多少？

解：

1) P1 的总指令条数为： $8+2+2=12$ ，P2 的总指令条数为： $2+5+3=10$ ，所以，P2 的总指令条数少。

2) P1 的总时钟周期数为： $8\times 1+2\times 2+2\times 4=20$ ，P2 的总时钟周期数为： $2\times 1+5\times 2+3\times 4=24$ ，因为两个指令代码序列在同一台机器上运行，所以**时钟周期**一样，故总时钟周期数少的代码序列所用时间短、执行速度快。显然，P1 比 P2 快。**【公式】**用户 CPU 时间 = 程序**总时钟周期数** × 时钟周期

3) P1 的**综合CPI** = 程序**总时钟周期数** ÷ 程序**总指令条数**，因此，

P1 的 CPI 为 $20/12=1.67$ ；P2 的 CPI 为 $24/10=2.4$ 。

13015 计算机系统原理【第一章考点解析】

考点12 用指令执行速度进行性能评估

MIPS (**M**illion **I**nstructions **P**er **S**econd) , 其含义是平均每秒钟执行多少**百万**条指令。

反映了机器执行**定点指令**的速度。

$$\text{公式1: MIPS} = \frac{\text{指令总数}/10^6}{\text{CPU 执行时间 (秒)}}$$

举例: 指令总数=50000000, CPU执行时间=2秒, 则MIPS = $5 \times 10^7 / 10^6 \div 2 = 25$ **MIPS**

$$\text{公式2: MIPS} = \frac{\text{时钟频率 (Hz)}}{\text{CPI} \times 10^6}$$

举例:

时钟频率是50**K**Hz, CPI=2, 则MIPS = $50 \times 10^3 / (2 \times 10^6) = 0.025$ **MIPS**

时钟频率是50**M**Hz, CPI=2, 则MIPS = $50 \times 10^6 / (2 \times 10^6) = 25$ **MIPS**

时钟频率是50**G**Hz, CPI=2, 则MIPS = $50 \times 10^9 / (2 \times 10^6) = 25000$ **MIPS**

13015 计算机系统原理【第一章考点解析】

考点12 用指令执行速度进行性能评估

例 1.3 假定某程序 P 编译后生成的目标代码由 A、B、C、D 四类指令组成，它们在程序中所占的比例分别为 43%、21%、12%、24%，已知它们的 CPI 分别为 1、2、2、2。现重新对程序 P 进行编译优化，生成的新目标代码中 A 类指令条数减少了 50%，其他类指令的条数没有变。请回答下列问题。

① 编译优化前后程序的 CPI 各是多少？

② 假定程序在一台主频为 50 MHz 的计算机上运行，则优化前后的 MIPS 各是多少？

分析：

1) A类指令条数减少50%，可知：想到：A类指令条数减少，总指令条数减少（分母变了）；

比如：我有4个鸡蛋，张三有6个鸡蛋，我的鸡蛋占比 $4 \div (4 + 6) = 40\%$ ，若我的鸡蛋个数减少了50%，那么我就剩下 $4 \div 2 = 2$ 个鸡蛋，这时我的鸡蛋占比 $2 \div (2 + 6) = 25\%$

2) 求编译优化前后程序的 CPI 各是多少？实际求的是综合CPI = $\sum_{i=1}^n (CPI_i \times F_i)$

3) 求MIPS，注意时钟频率单位：KHz、MHz、GHz

13015 计算机系统原理【第一章考点解析】

考点12 用指令执行速度进行性能评估

例 1.3 假定某程序 P 编译后生成的目标代码由 A、B、C、D 四类指令组成，它们在程序中所占的比例分别为 43%、21%、12%、24%，已知它们的 CPI 分别为 1、2、2、2。现重新对程序 P 进行编译优化，生成的新目标代码中 A 类指令条数减少了 50%，其他类指令的条数没有变。请回答下列问题。

① 编译优化前后程序的 CPI 各是多少？

② 假定程序在一台主频为 50 MHz 的计算机上运行，则优化前后的 MIPS 各是多少？

解：

优化后A类指令的条数减少了**50%**，因而各类指令所占比例分别计算如下。

A类指令： $21.5/(21.5+21+12+24)=27\%$

B类指令： $21/(21.5+21+12+24)=27\%$

C类指令： $12/(21.5+21+12+24)=15\%$

D类指令： $24/(21.5+21+12+24)=31\%$

13015 计算机系统原理【第一章考点解析】

考点12 用指令执行速度进行性能评估

例 1.3 假定某程序 P 编译后生成的目标代码由 A、B、C、D 四类指令组成，它们在程序中所占的比例分别为 43%、21%、12%、24%，已知它们的 CPI 分别为 1、2、2、2。现重新对程序 P 进行编译优化，生成的新目标代码中 A 类指令条数减少了 50%，其他类指令的条数没有变。请回答下列问题。

① 编译优化前后程序的 CPI 各是多少？

② 假定程序在一台主频为 50 MHz 的计算机上运行，则优化前后的 MIPS 各是多少？

解：

① 优化前后程序的 **CPI** 分别计算如下。

$$\text{优化前: } 43\% \times 1 + 21\% \times 2 + 12\% \times 2 + 24\% \times 2 = 1.57$$

$$\text{优化后: } 27\% \times 1 + 27\% \times 2 + 15\% \times 2 + 31\% \times 2 = 1.73$$

② 优化前后程序的 **MIPS** 分别计算如下。

$$\text{优化前: } 50\text{M} / (1.57 \times 10^6) = 31.8 \text{ MIPS}$$

$$\text{优化后: } 50\text{M} / (1.73 \times 10^6) = 28.9 \text{ MIPS}$$

13015 计算机系统原理【第一章考点解析】

考点12 用指令执行速度进行性能评估

MIPS (**M**illion **I**nstructions **P**er **S**econd) , 其含义是平均**每秒钟**执行多少**百万**条指令。

反映了机器执行**定点指令**的速度。

MFLOPS (**M**illion **FLO**ating-point operations **P**er **S**econd) , 其含义是**每秒**所执行的**浮点运算**

有多少**百万次**, 它是基于所完成的**操作次数**而不是**指令数**来衡量的。

反映了机器执行**浮点操作**的速度。

类似的浮点操作速度还有:

GFLOPS (10^9 次/s)

TFLOPS (10^{12} 次/s)

PFLOPS (10^{15} 次/s)

EFLOPS (10^{18} 次/s)

13015 计算机系统原理【第一章考点解析】

考点13 用基准程序进行性能评估

基准程序是进行**计算机性能评测**的一种重要工具。基准程序是专门用来进行性能评价的一组程序，能够很好地反映机器在运行实际负载时的性能，可以通过在不同机器上运行相同的基准程序来比较在不同机器上的运行时间，从而评测其性能。

谢谢大家