高级语言与机器语言的对应

四、指令系统

(一)指令系统的基本概念

(二)指令格式

(三)寻址方式

(四)数据的对齐和大/小端存放方式

(五)CISC和RISC的基本概念

(六)高级语言程序与机器级代码之间的对应

1.编译器，汇编器和链路器的基本概念

2.选择结构语句的机器级表示

3.循环结构语句的机器级表示

4.过程（函数）调用对应的机器级表示

王道考研/CSKAOYAN.COM

编译器、汇编器、链接器

编译器：高级语言程序→汇编语言程序，或 高级语言程序→机器语言程序

汇编器：汇编语言程序→机器语言程序，形成多个地址独立的目标模块

链接器：将多个目标模块链接成一个具有完整逻辑地址的装入模块

王道考研/CSKAOYAN.COM

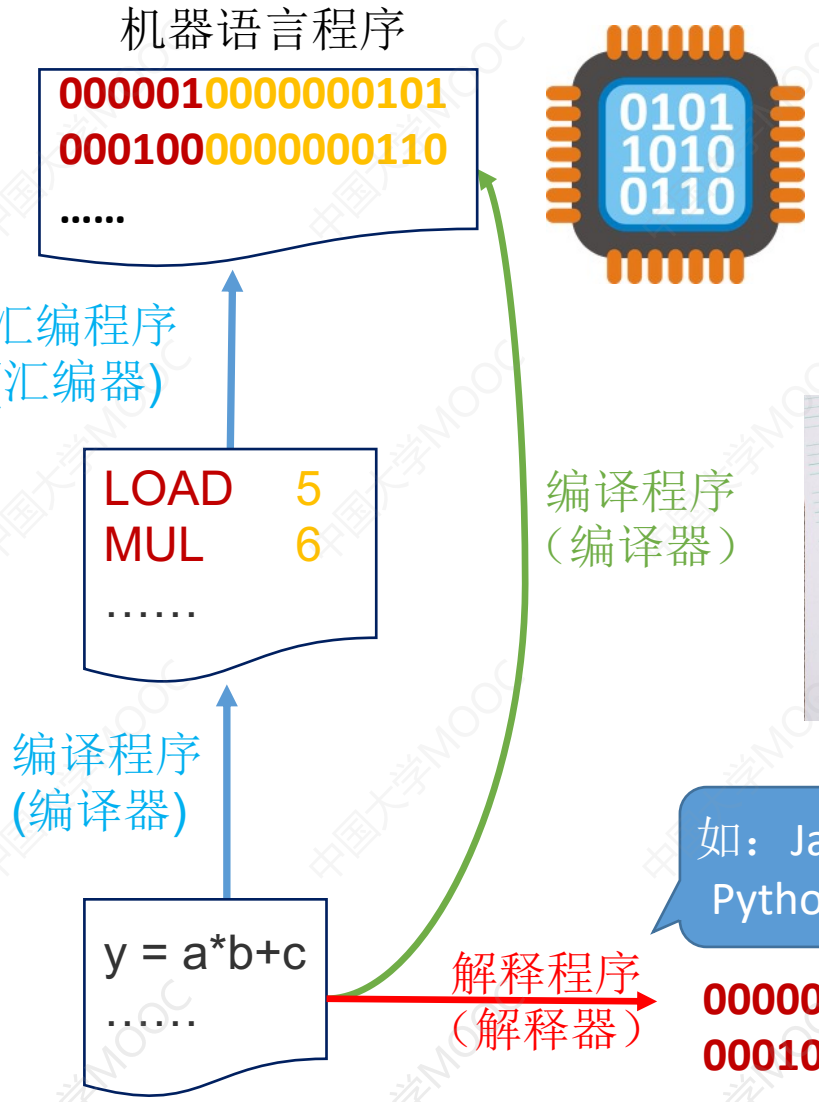
三种级别的语言（计组第一课件）

注：编译、汇编、解释程序，可统称“翻译程序”

机器语言：二进制代码

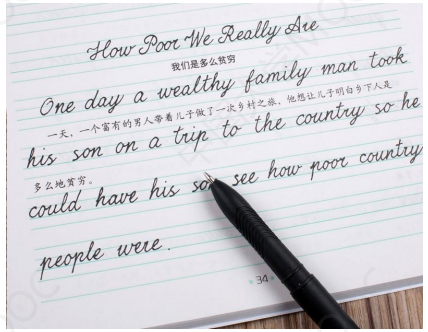
汇编语言：助记符

高级语言：C/C++、Java



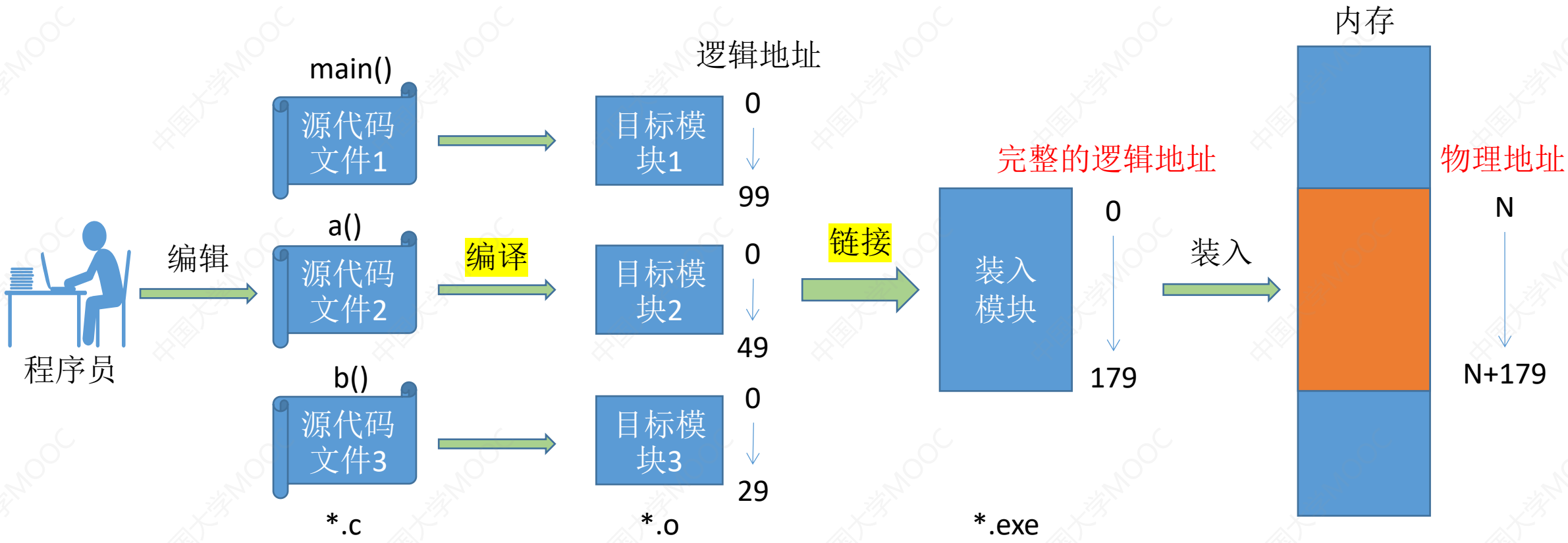
编译程序：将高级语言编写的源程序全部语句一次全部翻译成机器语言程序，而后再执行机器语言程序（只需翻译一次）

解释程序：将源程序的一条语句翻译成对应于机器语言的语句，并立即执行。紧接着再翻译下一句（每次执行都要翻译）



王道考研/CSKAOYAN.COM

从写程序到程序运行（操作系统第三章课件）



编译：由编译程序将用户源代码编译成若干个目标模块（编译就是把高级语言翻译为机器语言）
链接：由链接程序将编译后形成的一组目标模块，以及所需库函数链接在一起，形成一个完整的装入模块
装入（装载）：由装入程序将装入模块装入内存运行

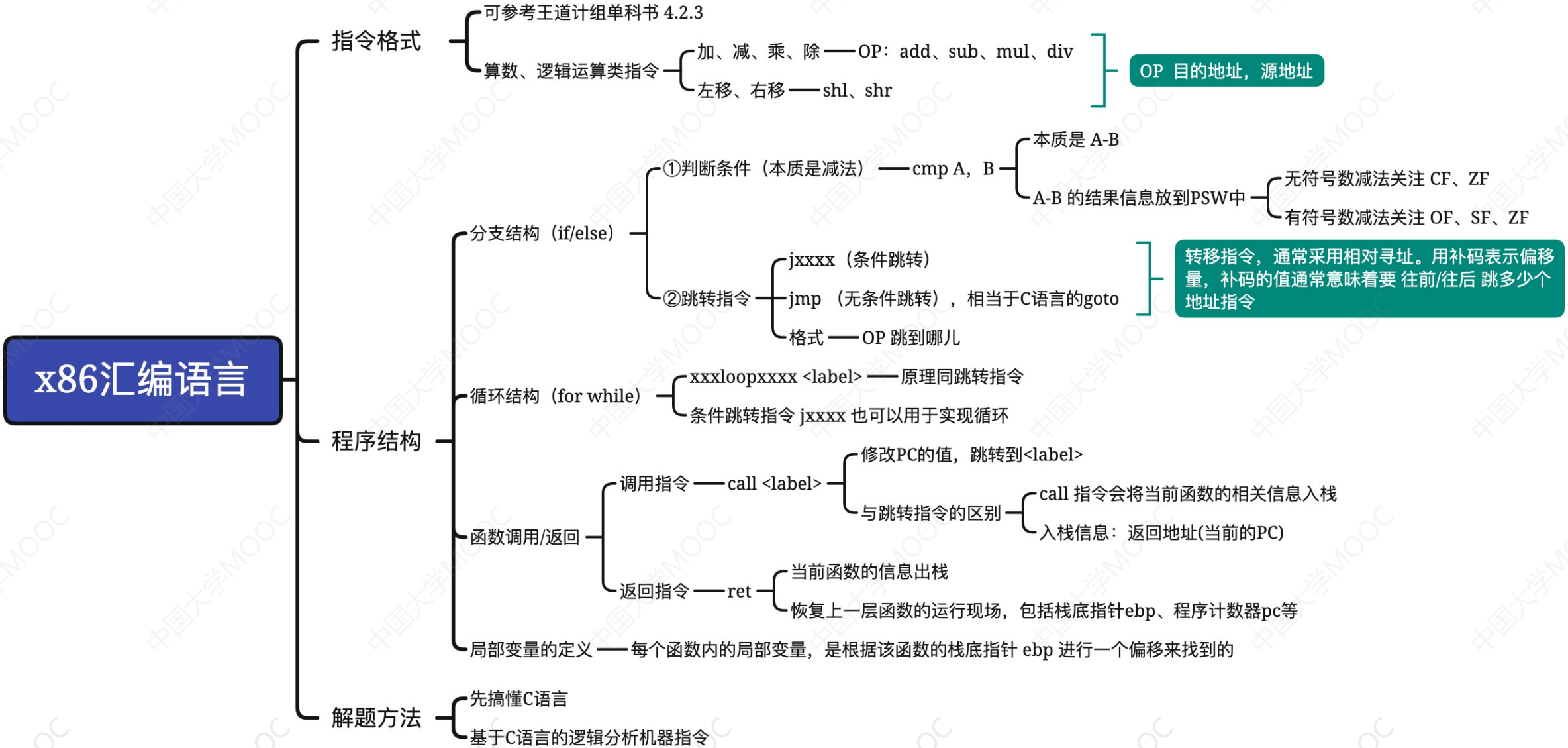
王道考研/CSKAOYAN.COM

高级语言与机器代码的对应

Tips：

- 不是让你看懂二进制代码（神仙都看不懂），是要了解基本程序结构（分支、循环、函数调用）的汇编语言代码表示
- 不算全新的考点，往年真题中考过
- 重点关注 x86汇编语言，MIPS汇编语言通常会给功能注释
- 需要掌握到哪种程度——能大致看懂题目里给的汇编语言代码就OK了，不用纠结细节，更不用自己写
- 重要例题——19年45题、17年44题

王道考研/CSKAOYAN.COM



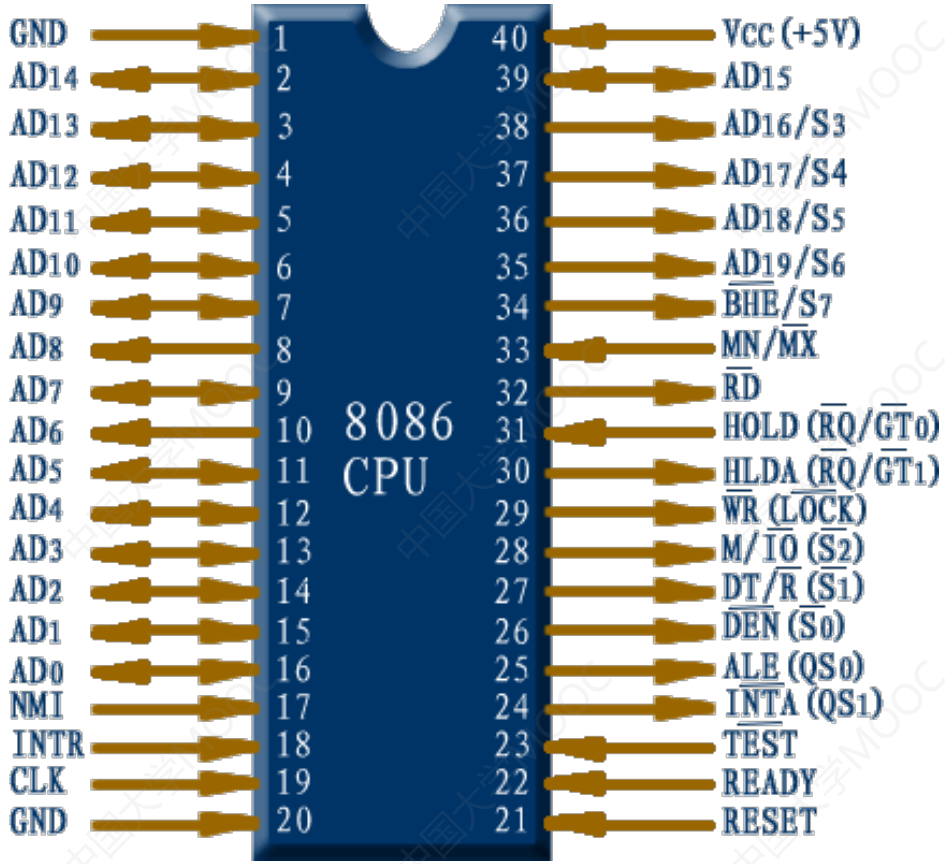
王道考研/CSKAOYAN.COM

扩展（了解一哈）

- 条件码：
  OF（Overflow Flag)溢出标志。溢出时为1,否则置0。
  SF（Sign Flag）符号标志。结果为负时置1,否则置0.
  ZF（Zero Flag)零标志，运算结果为0时ZF位置1,否则置0.
  CF（Carry Flag)进位标志，进位时置1,否则置0.
  AF（Auxiliary carry Flag）辅助进位标志，记录运算时第3位（半个字节）产生的进位置。有进位时1,否则置0.
  PF（Parity Flag）奇偶标志。结果操作数中1的个数为偶数时置1,否则置0.

- 控制标志位：
  DF（Direction Flag）方向标志，在串处理指令中控制信息的方向。
  IF（Interrupt Flag）中断标志。
  TF（Trap Flag）陷阱标志。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				OF	DF	IF	TF	SF	ZF		AF		PF		CF



- NMI:不可屏蔽中断请求信号。常用于处理电源掉电紧急情况。
- INTR:可屏蔽中断请求信号。

王道考研/CSKAOYAN.COM

2019年真题

45.（16 分）已知 $f(n)=n!=n\times(n-1)\times(n-2)\times\cdots\times2\times1$，计算 $f(n)$的 C 语言函数 f1 的源程序（阴影部分）及其在 32 位计算机 M 上的部分机器级代码如下：

```
int  f1(int n){
1    00401000   55              push  ebp
     ...        ...             ...
     if(n>1)
11   00401018   83 7D 08 01     cmp  dword ptr [ebp+8],1
12   0040101C   7E 17           jle  f1+35h (00401035)
     return n*f1(n-1);
13   0040101E   8B 45 08        mov  eax, dword ptr [ebp+8]
14   00401021   83 E8 01        sub  eax, 1
15   00401024   50              push  eax
16   00401025   E8 D6 FF FF FF  call  f1 ( 00401000)
     ...        ...             ...
19   00401030   0F AF C1        imul  eax, ecx
20   00401033   EB 05           jmp  f1+3Ah (0040103a)
     else return 1;
21   00401035   B8 01 00 00 00  mov  eax,1
     }
     ...        ...             ...
26   00401040   3B EC           cmp  ebp, esp
     ...        ...             ...
30   0040104A   C3              ret
```

其中，机器级代码行包括行号、虚拟地址、机器指令和汇编指令，计算机 M 按字节编址，int 型数据占 32 位。请回答下列问题：

（1）计算 $f(10)$需要调用函数 f1 多少次？执行哪条指令会递归调用 f1？

（2）上述代码中，哪条指令是条件转移指令？哪几条指令一定会使程序跳转执行？

（3）根据第 16 行的 call 指令，第 17 行指令的虚拟地址应是多少？已知第 16 行的 call 指令采用相对寻址方式，该指令中的偏移量应是多少（给出计算过程）？已知第 16 行的 call 指令的后 4 字节为偏移量，M 是采用大端方式还是采用小端方式？

（4）$f(13)=6227020800$，但 f1(13)的返回值为 1932053504，为什么两者不相等？要使 f1(13)能返回正确的结果，应如何修改 f1 的源程序？

（5）第 19 行的 imul 指令（带符号整数乘）的功能是 R[eax]←R[eax]×R[ecx]，当乘法器输出的高、低 32 位乘积之间满足什么条件时，溢出标志 OF = 1？要使 CPU 在发生溢出时转异常处理，编译器应在 imul 指令后应加一条什么指令？

王道考研/CSKAOYAN.COM

2017年真题

44.（10 分）在按字节编址的计算机 M 上，题 43 中 f1 的部分源程序（阴影部分）与对应的机器级代码（包括指令的虚拟地址）如下图所示。

```
            int f1( unsigned n)
1           00401020    55          push ebp
            …           …           …
                for(unsigned i=0; i<= n -1; i++)
            …           …           …
20          0040105E    39 4D F4    cmp dword ptr [ebp-0Ch],ecx
            …           …           …
                {   power *= 2;
            …           …           …
23          00401066    D1 E2       shl  edx,1
            …           …           …
                return sum;
            …           …           …
35          0040107F    C3          ret
```

其中，机器级代码行包括行号、虚拟地址、机器指令和汇编指令。请回答下列问题。

（1）计算机 M 是 RISC 还是 CISC？为什么？

（2）f1 的机器指令代码共占多少字节？要求给出计算过程。

（3）第 20 条指令 cmp 通过 i 减 n-1 实现对 i 和 n-1 的比较。执行 f1(0)过程中，当 i=0 时，cmp 指令执行后，进/借位标志 CF 的内容是什么？要求给出计算过程。

（4）第 23 条指令 shl 通过左移操作实现了 power*2 运算，在 f2 中能否也用 shl 指令实现 power*2？为什么？

王道考研/CSKAOYAN.COM

2014年真题

44.（12 分）某程序中有如下循环代码段 P：“for(int i = 0; i < N; i++) sum+=A[i];”。假设编译时变量 sum 和 i 分别分配在寄存器 R1 和 R2 中。常量 N 在寄存器 R6 中，数组 A 的首地址在寄存器 R3 中。程序段 P 起始地址为 0804 8100H，对应的汇编代码和机器代码如下表所示。

编号	地址	机器代码	汇编代码	注释
1	08048100H	00022080H	loop: sll R4, R2, 2	(R2)<<2 → R4
2	08048104H	00083020H	add R4, R4, R3	(R4) + (R3) → R4
3	08048108H	8C850000H	load R5, 0(R4)	((R4) + 0) → R5
4	0804810CH	00250820H	add R1, R1, R5	(R1) + (R5) → R1
5	08048110H	20420001H	add R2, R2, 1	(R2) + 1 → R2
6	08048114H	1446FFFAH	bne R2, R6, loop	if(R2)!=(R6) goto loop

执行上述代码的计算机 M 采用 32 位定长指令字，其中分支指令 bne 采用如下格式：

31 26	25 21	20 16	15 0
OP	Rs	Rd	OFFSET

OP 为操作码；Rs 和 Rd 为寄存器编号；OFFSET 为偏移量，用补码表示。请回答下列问题，并说明理由。

1）M 的存储器编址单位是什么？

2）已知 sll 指令实现左移功能，数组 A 中每个元素占多少位？

3）表中 bne 指令的 OFFSET 字段的值是多少？已知 bne 指令采用相对寻址方式，当前 PC 内容为 bne 指令地址，通过分析表中指令地址和 bne 指令内容，推断出 bne 指令的转移目标地址计算公式。

4）若 M 采用如下“按序发射、按序完成”的 5 级指令流水线：IF（取值）、ID（译码及取数）、EXE（执行）、MEM（访存）、WB（写回寄存器），且硬件不采取任何转发措施，分支指令的执行均引起 3 个时钟周期的阻塞，则 P 中哪些指令的执行会由于数据相关而发生流水线阻塞？哪条指令的执行会发生控制冒险？为什么指令 1 的执行不会因为与指令 5 的数据相关而发生阻塞？

2012年真题

44．某 16 位计算机中，带符号整数用补码表示，数据 Cache 和指令 Cache 分离。题 44 表给出了指令系统中部分指令格式，其中 Rs 和 Rd 表示寄存器，mem 表示存储单元地址，(x)表示寄存器 x 或存储单元 x 的内容。

指令系统中部分指令格式

名称	指令的汇编格式	指令功能
加法指令	ADD　Rs，Rd	(Rs) + (Rd)->Rd
算术/逻辑左移	SHL　Rd	2*(Rd)->Rd
算术右移	SHR　Rd	(Rd)/2->Rd
取数指令	LOAD　Rd, mem	(mem)->Rd
存数指令	STORE　Rs, mem	(Rs)->mem

该计算机采用 5 段流水方式执行指令，各流水段分别是取指(IF)、译码/读寄存器(ID)、执行/计算有效地址（EX)、访问存储器（M）和结果写回寄存器（WB），流水线采用“按序发射，按序完成”方式，没有采用转发技术处理数据相关，并且同一个寄存器的读和写操作不能在同一个时钟周期内进行。请回答下列问题：