

## 2020 年计算机系统 试题 (A) 答案

### 一、选择题 (每题两分)

DCBCD    ACBAB    ADDCC

### 二、填空题 (每题两分)

21:    16

22:    C480 0000

23:    0x00b

24:    2

25:    30.0125ms (这个好像是 20.0125ms, 具体答案如何我忘记了)

### 三、判断题 (每题两分)

× √ √ √ ×

### 四、系统分析题

26. 共 10 分, 每条 2 分。

- ①入栈指令, 将 rbp 入栈
- ②传送指令, 将栈顶指针 rsp 的值传送给 rbp
- ③传送指令, 向%rbp-4 的内存位置传送第一个参数(%rdi) 的值
- ④比较指令, %rbp-4 的内存数值 (第一个参数 x 的值) 与 3 进行比较
- ⑤条件跳转指令, 小于则跳转 (跳转到 4004f4 处)

27. 满分 12 分, 每点 2 分。

(1) 组索引位 8 位, 块内偏移 5 位。

(2) 所以组索引 5 位, 组索引对应的位为 0x13。

(3) CT 位为 0x8293 C0 位为 0x0B。

27-详细解答:

(1)  $64 \text{ KB} / (8 * 32\text{B}) = 256$  组 Tag 位 37 35? 位, 组索引位 8 位, 块内偏移 5 位。

(2) VPN 20 位 VPO 12 位 (←虚拟页大小是 4KB)。TLBI 128 条目, 4 路, 一共 32 组。所以组索引 5 位。0x829358B = 1000 0010 1001 0011 0101 1000 1011。所以组索引对应的位为 0x13。VPN (TLBT-TLBI) -VPO

(3)  $64 \text{ KB} / (16 * 32\text{B}) = 128$  组 组索引位 7 位。块内偏移 5 位。Tag 位 36 位。所以对应的 0x829358B = 1000 0010 1001 0011 0101 1000 1011  
CT 位为 0x8293 C0 位为 0x0B。

28. 满分 8 分。

1) 全局符号: x、y、z、main、addvec、addcnt (2 分)

强符号: x、y、main、addvec、addcnt (1 分)

若符号: z (1 分)

x、y 在.data 节, z 在.bss 节, main.text 节。(1 分)

addvec 在未定义节 (UND), addcnt 被优化掉 (addvec、addcnt 未说明不扣分)

2) 满分 3 分 (每点一分, 写出任意三点即可)

- ① 3c 10 60 00                      ② 28 10 60 00
- ③ 30 10 60 00                      ④ b3 fe ff ff
- ⑤ 2d 09 20 00                      ⑥ 23 09 20 00
- ⑦ a3 fe ff ff

## 五、综合设计题

29. 满分 12 分, 每写出一条微指令得 1 分。

指令	call XXX	ret
取指		icode:ifun $\leftarrow M_1[PC]$
		① valP $\leftarrow PC+1$
译码		valA $\leftarrow R[\%rsp]$
	valB $\leftarrow R[\%rsp]$	valB $\leftarrow R[\%rsp]$
执行	② valE $\leftarrow valB+(-8)$	valE $\leftarrow valB+8$
访存	M <sub>8</sub> [valE] $\leftarrow valP$	valM $\leftarrow M_8[valA]$

写回	R[%rsp] <- valE	R[%rsp] <- valE
更新 PC	PC <- valC	PC <- valM

注意：①处 PPT 上没有，但是课本上有，还是以课本为准。

②处很多同学写了 `valE <- valB - 8`，这个以后还是统一，写成 `+(-8)`。

30.满分 8 分。

```
void MatrixMult(double *a, double *b, double *c, int n){
    for(int i=0; i<n; i++){
        for(int j=0; j<n; j++){
            int k;
            double acc1=0;
            double acc2=0;
            for(k=0; k<n-1; k=k+2){//不一定要是 2 路循环展开，多路循环展开都可以
                acc1+=a[i][k]*b[k][j];
                acc2+=a[i][k+1]*b[k+1][j];
            }
            for (; k < n; k++) {
                acc2 +=a[i][k]*b[k][j];
            }
            c[i][j] = acc1+acc2;
        }
    }
}
```

编译优化：尽量用局部变量，减少不必要的内存访问；循环展开，减少循环边界检查和循环开销。（4 分，如果没有使用局部变量做累加，可以酌情扣分）

CPU 优化：尽量制造指令并行（流水）充分利用计算机内部的运算单元，方法也是循环展开，多路累积。（4 分，如果循环展开有错误，可以酌情扣分）