

2021秋ICS小班练习题(7)

姓名:

学号:

班号:

答卷说明:

- 答卷前,考生务必将自己的姓名填写在试卷指定位置.
- 答选择题时,请将答案填写在试卷相应位置.如需改动,请用签字笔将原答案划去,再在规定位置填写修正后的答案.未在规定区域作答的答案无效.
- 答非选择题时,用签字笔直接答在试卷相应位置,写在草稿纸等非答题区域的答案无效.
- 本卷共6页,卷面分100分.考试结束后,试卷由助教统一收回.
- 如出现中英文标点混用的情况属不严谨,请大家视作英文标点作答.

一、选择题(40分) 每题只有一个正确答案

1. 某机器内存为8位地址, cache设计参数为 $E=2$, $t=2$, $s=2$, $b=4$, cache块大小为16字节. cache为空的初始状态下, 数据访问的地址序列为 $0 \rightarrow 4 \rightarrow 34 \rightarrow 162 \rightarrow 128 \rightarrow 192 \rightarrow 2$ (以字节为单位), 请问一共发生多少次cache命中? 答: ()

- A. 0 B. 1 C. 2 D. 3

2. 对于下面这段程序说法正确的是().

- A. 假设s的长度为N, 该程序的时间复杂度为 $O(N \log N)$
B. 将“for(int i=0; i<strlen(s); i++) { }”修改为: “int len = strlen(s); for(int i=0; i<len; i++) { }”可以使程序运行时间与S的长度线性相关
C. B选项中的修改策略不影响程序的空间局部性与时间局部性
D. B选项中的修改策略仅影响程序的空间局部性, 不影响时间局部性

```
void upper(char *s)
{
    for(int i=0; i<strlen(s); i++)
        if(s[i]>='a' && s[i]<='z')
            s[i] += ('A' - 'a');
}
```

3. 通常情况下, 下面的哪些表述是正确的?

- A. 在一次读操作中, 返回的内容由高速缓存中的信息块决定
B. 高速缓存利用了时间局部性
C. 大部分情况下, 缓存需要用户程序采取显式的管理行为
D. 一级高速缓存更看重命中率, 二级高速缓存更看重命中时间

4. 下面关于缓存替换策略的说法哪个是正确的(N 为 cache 的大小)

- A. FIFO 的性能总是优于随机替换
B. LRU 适用于数组的顺序访问
C. 假定 cache 初始状态为空, 若某一输入片段使用 LRU 造成了 $N + 1$ 次miss, 则对任意策略至少产生两次 miss
D. 对于同一确定性替换策略, 增大 N 的大小一定减少 miss 次数

5. 已知一个磁盘每条磁道平均有 400 个扇区，旋转速度为 6000RPM，那么它的平均传送时间大约为()ms

- A. 0.020 B.0.025 C.0.040 D.0.050

6.阅读如下C代码和对应的汇编代码

C代码:

```
struct node {
    struct node *nxt;
    int num;
};
int sum(struct node *head)
{
    int rst = 1;
    while (head) {
        rst *= head->num;
        head = head->nxt;
    }
    return rst;
}
```

汇编代码:

```
sum:
    mov     $0x1,%eax
.L2:
    test    %rdi,%rdi
    je      .L1
    mov     0x8(%rdi),%ecx    // 2 cycle
    imul    %ecx,%eax         // 6 cycle
    mov     (%rdi),%rdi      // 2 cycle
    jmp     .L2
.L1:
    retq
```

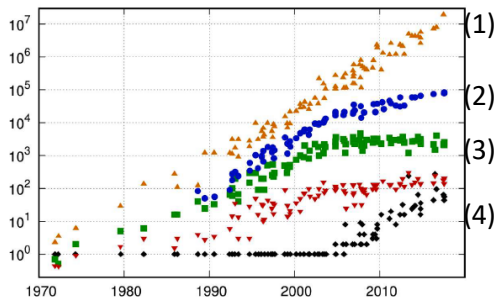
该段程序每轮循环处理一个元素。在理想机器上（处理器的执行单元足够多）， 每条指令消耗的时间周期如注释所示。此时程序的CPE为_____。

如果编译器采用类似循环展开的方法， 每轮循环可以处理多个元素， 程序的最优CPE为_____。

- A. 6, 2 B. 8, 2 C. 6, 1 D. 8, 1

7. 下面是计算机硬件指标随时间变化趋势图。根据摩尔定律，下面的(1)，(2)，(3)，(4)分别代表()。

- A. 晶体管数量，单核运算速度，处理器频率，处理器核数
B. 晶体管数量，单核运算速度，处理器核数，处理器频率
C. 单核运算速度，晶体管数量，处理器频率，处理器核数
D. 单核运算速度，晶体管数量，处理器核数，处理器频率



8. 使用经Aker' s Coding Scheme优化过的Lee' s Algorithm完成右图中由S到T的布线工作。则A和B处用于标记的用于回溯的符号分别为()，其中黑色方框代表障碍（block）。

- A. 2;3 B. 0;0 C. 0;1 D. 1;0

S	1	?	
1	1	?	?
	A	B	?
?	?	?	T

9. 现有如下汇编代码段，相关寄存器的初始值也在右侧表格中给出：(15分)

.L2:

```
movq    (%rdi), %rbx
addq    %rbx, %rax
addq    $8, %rdi
addq    $1, %rcx
cmpq    $8, %rcx
jl      .L2
```

%rdi	0x8008
%rax	0
%rbx	0
%rcx	0

(1) 请在下面 C 代码中，补充缺失的部分。

```
for(i=0; i<_____ ; i++)
{
    sum=sum+a[i];
}
```

(2) 现有一个类似课上 SEQ 的处理器来运行上述汇编代码，在理想情况下，其每个周期可以处理完一条汇编代码，请计算：理想情况下，执行完上述代码段（即结束循环）一共需要多少周期？_____

(3) 现在考虑数据缓存（data cache）对该处理器的影响。

该处理器配置了一个简单的直接映射（direct-mapped, E=1）数据缓存，总容量 32KB，每个缓存块的大小是 64Byte。对于 movq 指令，如果被访问的数据在该缓存中，则 1 个周期可以处理完成。否则，则需要 2 个周期才能完成。在运行上述汇编代码之前，数据缓存初始状态为空，其他设置同上一问。请再次计算：完成上述汇编代码段一共需要多少周期？_____

(4) 为了提升处理器的性能，增加了一个加法器。因此，在没有数据依赖、并且不改变指令顺序的情况下，新的处理器可以同时处理两个 addq 指令、或者一个 addq 加一个 cmpq 指令。注意：movq 和跳转指令仍旧需要单独执行，其他设置同上一问。请再次计算：完成上述汇编代码段一共需要多少周期？_____

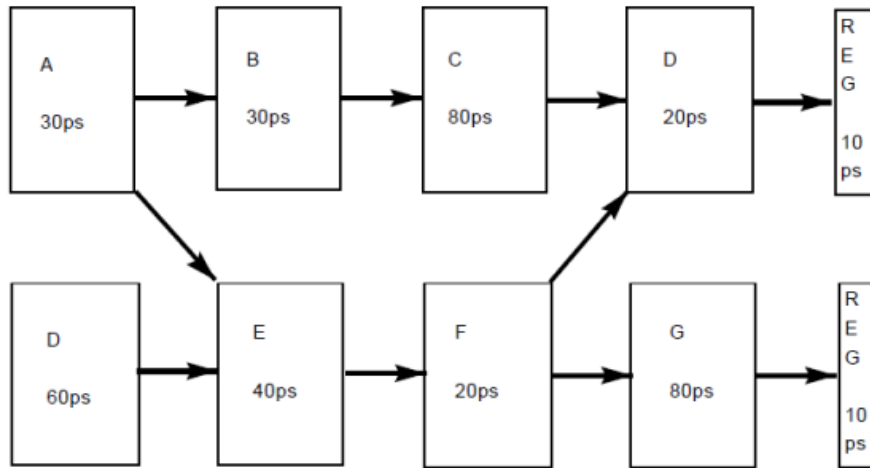
(5) 为了进一步减少处理时间，程序员将代码进行循环展开，新的代码如下：

.L2:

```
movq    (%rdi), %rbx
movq    8 (%rdi), %rdx
addq    %rbx, %rax
addq    %rdx, %rax
addq    $16, %rdi
addq    $2, %rcx
cmpq    $8, %rcx
jl      .L2
```

假设在不影响程序正确性的情况下，可以调整上述指令的顺序，请问最短需要多少个周期可以完成该代码段（注：其他设置和上一问相同）。

10. 如图所示，每个模块表示一个单独的组合逻辑单元，每个单元的延迟以及数据依赖关系已在图中标出。通过在两个单元间添加寄存器的方式，可以对该数据通路进行流水化改造。假设每个寄存器的延迟为 10 ps (15分)



1) 如果改造为一个二级流水线，为获得最大的吞吐率，该寄存器应在哪里插入？请计算该流水线的吞吐率，并说明计算过程。结果可以是分数形式也可以是小数形式。

2) 如果改造为一个三级流水线，为获得最大的吞吐率，寄存器应在哪里插入？请计算该流水线的吞吐率，并说明计算过程。结果可以是分数形式也可以是小数形式。

11. 阅读下列 C 代码以及它编译生成的汇编语言 (15分)

```
typedef struct _list {
    struct _list* next;
    int value;
} list;

int m(list* p) {
    int r = 100;
    while (___①___) {
        r = ___②___;
        p = ___③___;
    }
    if (p != 0) r = ___④___;
    return r;
}
```

```
m:
    testq %rdi, %rdi
    je .L6
    movq (%rdi), %rdx
    movl $100, %eax
    testq %rdx, %rdx
    jne .L4
    jmp .L3
.L14:
    movq (%rdi), %rdx
    testq %rdx, %rdx
    je .L3
.L4:
    movl 8(%rdx), %r8d
    ___⑤___ 8(%rdi), %r8d
    ___⑤___ %r8d, %eax
    movq (%rdx), %rdi
    testq %rdi, %rdi
    jne .L14
    ret
.L3:
    ___⑤___ 8(%rdi), %eax
    ret
.L6:
    movl $100, %eax
    ret
```

已知访存延迟 1 个指令周期。不访存的时候 addl 延迟 4 个指令周期，imull 延迟 8 个指令周期，其他指令延迟 1 个指令周期。指令访存时延迟的执行周期为不访存的时候延迟的指令周期和访存延迟的周期之和。函数 m 中的循环在处理链表 p 的时候 CPE 为 4 个指令周期。⑤处的指令为 movl, addl, imull 中的一个。请填写①-⑤处的代码。

① _____

② _____

③ _____

④ _____

⑤ _____

12. 有下面两个程序。将他们先分别编译为.o文件，再链接为可执行文件。(15分)

main.c	count.c
<pre>#include <stdio.h> A int foo(int n) { static int ans = 0; ans = ans + x; return n + ans; } int bar(int n); void op(void) { x = x + 1; } int main() { for (int i = 0; i < 3; i++) { int a1 = foo(0); int a2 = bar(0); op(); printf("%d %d ", a1, a2); } return 0; }</pre>	<pre>B int bar(int n) { static int ans = 0; ans = ans + x; return n + ans; }</pre>

(1) 当A处为 `int x = 1;`, B处为 `int x;`时, 完成下表。如果某个变量不在符号表中, 那么在名字那一栏打×; 如果它在符号表中的名字含有随机数字, 那么请用不同的四位数字区分多个不同的符号。对于局部符号, 不需要填最后一栏。

文件名	变量名	在符号表中的名字	是局部符号吗?	是强符号吗?
main.o	x			
	bar			
	ans			
count.o	x			
	bar			
	ans			

程序能够链接成功吗? 如果可以, 程序的运行结果是什么? 如果不可以, 链接器报什么错?

(2) 当A处为 `static int x = 1;`, B处为 `static int x = 1;`时, 重新做 (1) 问。

参考答案：

选择题：BBBCBAAB

非选择题：

(19年期末二)

8、48、50、42、26

(16年期末三)

(1) 插入在 BC 间以及 EF;

$$1000 / (80 + 20 + 10) = 1000 / 110 = 9.09 \text{ GIPS}$$

(2) 插入在 AE、FD、BC、CD、DE 和 FG 间;

$$1000 / (80 + 10) = 1000 / 90 = 11.11 \text{ GIPS}$$

(16年期中三.2)

(1) $p \neq 0 \ \&\& \ p \rightarrow \text{next} \neq 0$

(2) $r * (p \rightarrow \text{value} * p \rightarrow \text{next} \rightarrow \text{value})$ 注意：答案中运算可以交换，但不可结合

(3) $p \rightarrow \text{next} \rightarrow \text{next}$

(4) $r * p \rightarrow \text{value}$

(5) imull (因为每次处理两个元素，所以关键路径的时间周期为8。关键路径要么为两次访存和两次mov，要么为5处的运算。前者不可能达到8，所以只能是imull。)

(1) 当 A 处为 `int x = 1;`, B 处为 `int x;` 时, 完成下表。如果某个变量不在符号表中, 那么在名字那一栏打×; 如果它在符号表中的名字含有随机数字, 那么请用不同的四位数字区分多个不同的符号。对于局部符号, 不需要填最后一栏。

文件名	变量名	在符号表中的名字	是局部符号吗?	是强符号吗?
main.o	x	x	×	✓
	bar	bar	×	×
	ans	ans.1597	✓	
count.o	x	x	×	×
	bar	bar	×	✓
	ans	ans.0344	✓	

程序能够链接成功吗? 如果可以, 程序的运行结果是什么? 如果不可以, 链接器报什么错?

1 1 3 3 6 6

(2) 当 A 处为 `static int x = 1;`, B 处为 `static int x = 1;` 时, 完成下表。

文件名	变量名	在符号表中的名字	是局部符号吗?	是强符号吗?
main.o	x	x	✓	
	bar	bar	×	×
	ans	ans.1597	✓	
count.o	x	x	✓	
	bar	bar	×	✓
	ans	ans.0344	✓	

程序能够链接成功吗? 如果可以, 程序的运行结果是什么? 如果不可以, 链接器报什么错?

1 1 3 2 6 3。两个 x 在各自的 .o 文件中的名字都为 x, 因为它们不是过程中的静态变量。思考: 对于非过程间的静态变量, 为什么编译器不需要作这样的区分?