

# 北京大学信息科学技术学院考试试卷

考试科目：计算机系统导论 姓名：\_\_\_\_\_ 学号：\_\_\_\_\_

考试时间：2020 年 11 月 16 日 小班号：\_\_\_\_\_

题号	一	二	三	四	五	总分
分数						
阅卷人						

## 北京大学考场纪律

1、考生进入考场后，按照监考老师安排隔位就座，将学生证放在桌面上。无学生证者不能参加考试；迟到超过 15 分钟不得入场。在考试开始 30 分钟后方可交卷出场。

2、除必要的文具和主考教师允许的工具书、参考书、计算器以外，其它所有物品（包括空白纸张、手机、或有存储、编程、查询功能的电子用品等）不得带入座位，已经带入考场的必须放在监考人员指定的位置。

3、考试使用的试题、答卷、草稿纸由监考人员统一发放，考试结束时收回，一律不准带出考场。若有试题印制问题请向监考教师提出，不得向其他考生询问。提前答完试卷，应举手示意请监考人员收卷后方可离开；交卷后不得在考场内逗留或在附近高声交谈。未交卷擅自离开考场，不得重新进入考场答卷。考试结束时间到，考生立即停止答卷，在座位上等待监考人员收卷清点后，方可离场。

4、考生要严格遵守考场规则，在规定时间内独立完成答卷。不准交头接耳，不准偷看、夹带、抄袭或者有意让他人抄袭答题内容，不准接传答案或者试卷等。凡有违纪作弊者，一经发现，当场取消其考试资格，并根据《北京大学本科考试工作与学术规范条例》及相关规定严肃处理。

5、考生须确认自己填写的个人信息真实、准确，并承担信息填写错误带来的一切责任与后果。

学校倡议所有考生以北京大学学生的荣誉与诚信答卷，共同维护北京大学的学术声誉。

以下为试题和答题纸，共 14 页。

<注>：本试卷未做特别说明的，均视为在 x86-64 环境下执行

得分

第一题 单项选择题（每小题 2 分，共 24 分）

注：选择题的回答必须填写在下表中，写在题目后面或其他地方，视为无效。

题号	1	2	3	4	5	6
回答						
题号	7	8	9	10	11	12
回答						

1、某 C 语言程序中对数组变量 a 的声明为“int a[10][10];”，有如下一段代码：

```
for (i=0; i<10; i++)  
    for (j=0; j<10; j++)  
        sum+= a[i][j];
```

假设执行到“sum+= a[i][j];”时，sum 的值在 %rax 中，a[i][0] 所在的地址在 %rdx 中，j 在 %rsi 中，则“sum+= a[i][j];”所对应的指令是（ ）。

- A. addl 0 (%rdx, %rsi, 4), %eax
- B. addl 0 (%rsi, %rdx, 4) , %eax
- C. addl 0 (%rdx, %rsi, 2) , %eax
- D. addl 0 (%rsi, %rdx, 2) , %eax

2、条件码描述了最近一次算术或逻辑操作的属性。下列关于条件码的叙述中，哪一个是不正确的？

- A. set 指令可以根据条件码的组合将一个字节设置为 0 或 1
- B. cmp 指令和 test 指令可以设置条件码但不更改目的寄存器
- C. leaq 指令可以设置条件码 CF 和 OF
- D. 除无条件跳转指令 jmp 外，其他跳转指令都是根据条件码的某种组合跳转到标号指示的位置

3、假定静态 int 型二维数组 a 和指针数组 pa 的声明如下：

```
static int a[4][4]={ {3, 8, -2, 6}, {2, 1, -5, 3 }, {1, 18, 4, 10},{4, -2, 0, 8}};
```

```
static int *pa[4]={a[0], a[1], a[2], a[3]};
```

若 a 的首地址为 0x601080, 则&pa[0]和 pa[1]分别是:

- A. 0x6010c0、0x601090
- B. 0x6010e0、0x601090
- C. 0x6010c0、0x6010a0
- D. 0x6010e0、0x6010a0

4、下列关于 x86-64 过程调用的叙述中, 哪一个是不正确的?

- A. 每次递归调用都会生成一个新的栈帧, 空间开销大
- B. 当传递给被调用函数的参数少于 6 个时, 可以通过通用寄存器传递
- C. 被调用函数要为局部变量分配空间, 返回时无需释放这些空间
- D. 过程调用返回时, 向程序计数器中推送的地址是调用函数中调用指令的下一条指令的地址

5、假设结构体类型 student\_info 的声明如下:

```
struct student_info {  
    char id[8];  
    char name[16];  
    unsigned zip;  
    char address[50];  
    char phone[20];  
}x;
```

若 x 的首地址在 %rdx 中, 则“unsigned xzip=x.zip;”所对应的汇编指令为:

- A. movl 0x24(%rdx), %eax
- B. movl 0x18(%rdx), %eax
- C. leaq 0x24(%rdx), %rax
- D. leaq 0x18(%rdx), %rax

6、在Y86-64的PIPE实现中, 仅考虑ICALL、IPOPQ、IPUSHQ、IRET指令, 对mem\_addr的HCL描述正确的是:

```
word mem_addr = [  
    M_icode in { ①, ② } : M_valE;  
    M_icode in { ③, ④ } : M_valA;  
];
```

- A. ①IPUSHQ ②ICALL ③IPOPQ ④IRET  
 B. ①IPUSHQ ②IRET ③ICALL ④IRET  
 C. ①IPUSHQ ②IPOPQ ③IRET ④ICALL  
 D. ①IPUSHQ ②IRET ③IPOPQ ④ICALL

7、下列说法正确的是：

- A. 在SEQ机器中，我们采用预测跳转总是选择（always taken）的策略比从不选择（never taken）的策略要略好。  
 B. 流水级划分应尽量均衡，不均衡的流水线会增加控制冒险。  
 C. 如果一台机器的CPI小于1，则它必然不是普通流水线结构。  
 D. 由于rrmovq %rax, %rax不影响标记位，所以可使用其代替nop指令。

8、仅考虑以下代码，哪个或哪些代码片段在当前主流编译器的标准优化选项下一定会被优化？（假设int i, int j, int A[N], int B[N], int \*p都是局部变量，int foo(int) 是一个函数）

优化前	优化后
A. for (j = 0 ; j < N ; j++) B[i] *= A[j];	int temp = B[i]; for (j= 0 ; j < N ; j++) temp *= A[j]; B[i] = temp;
B. for (j = 0 ; j < N ; j++) m += i*N*j;	temp = i*N; for (j= 0 ; j < N ; j++) m += temp * j;
C. i = foo(N); j = foo(N)+1; if (i != j) m = j ;	m = j ;
D. if(m==1){ i=3;j=4; }else{ i=4;j=3; } m=i+j;	m = 7;

9、下面代码中不能体现以下哪种局部性：

- A. 数据的时间局部性
- B. 数据的空间局部性
- C. 指令的时间局部性
- D. 指令的空间局部性

```
for(i=0; i<100; i++){  
    a[0]=a[0]+i;  
}
```

10、以下关于存储器的说法中，正确的是：

- A. SRAM 单元比 DRAM 单元的能耗更大
- B. DRAM 通常用做高速缓存
- C. DRAM 单元比 SRAM 单元的速度更快
- D. 固态硬盘的读写速度基本相当

11、以下关于缓存的说法中，正确的是：

- A. LRU 不会比 MRU (most-recently used) 替换策略差
- B. 先入先出 (FIFO) 比随机替换命中率高
- C. 保持缓存容量、缓存块大小 (B) 不变，增加路数 (E)，命中率不会下降
- D. 以上说法都不正确

12、以下关于存储器的说法中，正确的是：

- A. SRAM 和 DRAM 的融合形成 SDRAM
- B. SRAM 单元比 DRAM 单元的晶体管数目更多
- C. DRAM 是一种非易失性存储器
- D. 固态硬盘的读写速度基本相当

得分

## 第二题（20 分）

分析两段在 x86-64 CPU 上运行的程序代码，回答相关问题。

### 第一段代码：

```
#include<stdio.h>

typedef struct
{
    int x;
    int y;
}struct_e;

typedef union
{
    struct_e s;
    double d;
}union_e;

int main () {
    union_e test;
    test.s.x = 0;
    test.s.y = -1;
    printf("%lf\n", test.d);

    test.s.y = 0x3ff00000;
    printf("%lf\n", test.d);
}
```

(1) 以上程序的两次输出分别是 \_\_\_\_\_， \_\_\_\_\_。

(2) sizeof(union\_e) = \_\_\_\_\_。

(3) 如果某次程序输出的 test.d 是最大的负非规格化数，则

此时从 test 的起始地址开始的 8 个字节（用 16 进制表示）依次是：

--	--	--	--	--	--	--	--

此时  
test.s.x = \_\_\_\_\_  
test.s.y = \_\_\_\_\_

**第二段代码：**

```
int a = 33554442; // 2^25+10
int b = a + 5;
for (; a < b; a++) {
    float f = a;
    printf("%d ", (int)f - a);
}
```

(1) 这段代码的运行输出结果是（若有多个输出结果，用逗号分隔表示）：

\_\_\_\_\_

(2) 进一步分析，float 类型不可以精确表示的最小正整数是 \_\_\_\_\_。

得分

### 第三题 (20 分)

请分析下面的 C 语言程序和对应的 x86-64 汇编代码。

1. 其中，有一部分缺失的代码（用标号标出），请在标号对应的横线上填写缺失的内容。注：汇编与机器码中的数字用 16 进制数填写。

C 语言代码如下：

```
typedef struct _parameters {
    int n;
    int product;
} parameters;
int bar(parameters *params, int x) {
    params->product *= x;
}
void foo (parameters *params) {
    if (params->n <= 1)
        ____ (1) ____ (1) _____
    bar(params, ____ (2) ____); (2) _____
    params->n--;
    foo(params);
}
```

x86-64 汇编代码如下（为简单起见，函数内指令地址只给出后四位，需要时可补全）：

```
0x00005555555555189 <bar>:
5189: f3 0f 1e fa    endbr64
518d: 55            push    %rbp
518e: 48 89 e5      mov     %rsp,%rbp
5191: 48 89 7d f8    mov     _(3)_,-0x8(%rbp) (3)_____
5195: 89 75 f4      mov     %esi,-0xc(%rbp)
5198: 48 8b 45 f8    mov     -0x8(%rbp),%rax
519c: 8b 40 04      mov     0x4(%rax),%eax
519f: 0f af 45 f4    imul    _(4)_(%rbp),%eax (4)_____
51a3: 89 c2        mov     %eax,%edx
51a5: 48 8b 45 f8    mov     -0x8(%rbp),%rax
```



```

51a9: 89 50 04      mov    %edx,0x4(%rax)
51ac: 90             nop
51ad: 5d            pop    _ (5) _      (5) _____
51ae: c3           retq

```

00005555555551af <foo>:

```

51af: f3 0f 1e fa    endbr64
51b3: 55             push   %rbp
51b4: 48 89 e5       mov    %rsp,%rbp
51b7: 48 83 ec 10     _ (6) _ $0x10,%rsp      (6) _____
51bb: 48 89 7d f8     mov    %rdi,-0x8(%rbp)
51bf: 48 8b 45 f8     mov    -0x8(%rbp),%rax
51c3: 8b 00          mov    (%rax),%eax
51c5: 83 f8 01       cmp    $0x1,%eax
51c8: 7e 31          _ (7) _ 51fb <foo+0x4c>  (7) _____
51ca: 48 8b 45 f8     mov    -0x8(%rbp),%rax
51ce: 8b 10          mov    (%rax),%edx
51d0: 48 8b 45 f8     mov    -0x8(%rbp),%rax
51d4: 89 d6          mov    %edx,%esi
51d6: 48 89 c7       mov    %rax,%rdi
51d9: e8 ab ff ff ff callq  0x0000555555555189 <bar>
51de: 48 8b 45 f8     mov    -0x8(%rbp),%rax
51e2: 8b 00          mov    (%rax),%eax
51e4: 8d 50 ff       lea    -0x1(_ (8) _),%edx  (8) _____
51e7: 48 8b 45 f8     mov    -0x8(%rbp),%rax
51eb: 89 10          mov    _ (9) _ ,(%rax)    (9) _____
51ed: 48 8b 45 f8     mov    _ (10) _ ,%rax     (10) _____
51f1: 48 89 c7       mov    %rax,%rdi
51f4: e8 b6 ff ff ff callq  _ (11) _      (11) _____
51f9: eb 01          jmp     51fc <foo+0x4d>
51fb: 90             nop
51fc: c9             leaveq
51fd: c3           retq

```

2. 在程序执行到 0x00005555555518e 时(该指令还未执行), 此时的栈帧如下, 请填写空格中对应的值。

地址	值
0x7fffffffef308	0xffffef340
0x7fffffffef304	0x00000000
0x7fffffffef300	0x00000000
0x7fffffffef2fc	0x00005555
0x7fffffffef2f8	(12) _____
0x7fffffffef2f4	0x00007fff
0x7fffffffef2f0	0xffffef310
0x7fffffffef2ec	0x00007fff
0x7fffffffef2e8	0xffffef340
0x7fffffffef2e4	0x00000004
0x7fffffffef2e0	0xffffef350
0x7fffffffef2dc	0x00005555
0x7fffffffef2d8	(13) _____
0x7fffffffef2d4	0x00007fff
0x7fffffffef2d0	(14) _____

3. 当 params={n, 1} 时, foo(&params) 函数的功能是什么?

得分

#### 第四题（20 分）

基于教材所描述的Y86-64 ISA和SEQ、PIPE处理器结构，完成下列问题。

（1）拟新加入指令leave。其语义相当于

```
rrmovq    %rbp, %rsp
popq      %rbp
```

请按下表补全SEQ处理器每个阶段的操作。需说明的信号可能会包括：icode, ifun, rA, rB, valA, valB, valC, valE, valP, Cnd; 寄存器堆R[], 存储器M[], 程序计数器PC, 条件码CC。其中对存储器的引用必须标明字节数。

取指	$\text{icode:ifun} \leftarrow M_1[\text{PC}]$ $\text{valP} \leftarrow \text{PC} + 1$
译码	
执行	
访存	
写回	
更新PC	$\text{PC} \leftarrow \text{valP}$

(2) 类似的, 如果尝试新加入指令enter。其语义相当于

```
pushq    %rbp
rrmovq   %rsp, %rbp
```

为执行拟新增的指令, 你尝试改进PIPE处理器, 最终完成的工作为(可多选):\_\_\_\_\_。

- A. 增加了一个计算栈帧调整 $\pm 8$ 的ALU单元;
- B. 为寄存器堆增加了一个读口;
- C. 为寄存器堆增加了一个写口;
- D. 在执行(E: Execution)阶段引入寄存器以保持新增的流水线信号;
- E. 在访存(M: Memory Access)阶段引入寄存器以保持新增的流水线信号;
- F. 在回写(W: Write Back)阶段引入寄存器以保持新增的流水线信号;
- G. 认为现有框架无法有效支持这一改造。

得分

第五题（16 分）

现有一个直接映射（Direct-mapped）的高速缓存（cache），其总容量为 16 字节，每个高速缓存块（cache block）为 4 字节（ $B=4\text{Byte/block}$ ）。Cache 的初始状态为空。

（1）经过如下地址访问序列后，请填写 cache 第 2 组（即 Set 2，注：从 0 开始编号）的状态。数据块用 10 进制表示，举例： $M[4-7]$  表示地址 4-7 的数据。地址序列全部为读访存，每次读取 2 个字节，地址单位为字节。

（提示：一次访存可能访问多个高速缓存块。）

R	0	$[000000_2]$ ,			
R	1	$[000001_2]$ ,			
R	7	$[000111_2]$ ,			
R	4	$[000100_2]$ ,			
R	0	$[000000_2]$			

	v	Tag	Block
Set 2			

本次访存序列，共发生\_\_\_\_\_次高速缓存缺失 (miss)？

（提示：一次访存可能引发多次高速缓存缺失。）

（2）接上题，高速缓存重新清空后，执行如下访问序列，同时包含读访存（R）和写访存（W）操作，每次读/写 2 个字节，高速缓存采用 write back + no-write-allocate 的管理策略，其他配置不变。请填写访存后 cache 第 1 组（Set 1）的状态。

R	0	$[000000_2]$ ,			
R	1	$[000001_2]$ ,			
W	7	$[000111_2]$ ,			
R	12	$[001100_2]$ ,			
R	9	$[001001_2]$			

	v	Tag	Block
Set 1			

本次访存序列，共发生\_\_\_\_\_次高速缓存命中 (hit)？

(3) 为了提升高速缓存的效率，设计师提出两种修改方案

- a) 将高速缓存块增大到 8 字节 ( $B=8\text{Byte/block}$ )，其他配置不变，每次访存命中 (hit) 的时间开销为 4 个时钟周期；
- b) 将高速缓存变为全相连 (fully-associative, 即  $S=1$ )，其他配置不变，每次访存命中 (hit) 的时间开销为 2 个时钟周期；

对于如下访问序列，分别采用两种修改方案的访存时间总开销相等，请问每次高速缓存缺失 (miss) 的时间开销是\_\_\_\_\_时钟周期？

R	0	[000000] <sub>2</sub> ],
R	1	[000001] <sub>2</sub> ],
R	7	[000111] <sub>2</sub> ],
R	8	[001000] <sub>2</sub> ],
R	9	[001001] <sub>2</sub> ]

提示：修改高速缓存配置不会影响高速缓存缺失的时间开销