主管领导审核签字

哈尔滨工业大学 2022 年春季学期

计算机系统（A）　试　题

题号	一	二	三	四	五	六	七	总分
得分								
阅卷人								

片纸鉴心　诚信不败

授课教师　姓名　学号　院系　密　封　线

一、 单项选择题（每小题 1 分，共 10 分）

1. 在 C 语言程序的链接阶段，链接器生成的文件不可能是（ A ）
   A. 后缀是.o 的目标文件
   B. 后缀是.so 的共享库文件
   C. 可执行程序
   D. ELF 格式的文件
2. 关于汇编指令 movl $0x1234, 8(%rax,%rbx,8)，错误的说法是(D)
   A. 目的操作数是内存操作数
   B. 内存操作数的地址是 8+(%rax+%rbx×8)
   C. 操作数宽度是 4 字节
   D. 可以改写成 mov $0x1234, 8(%rax, %rbx,8)，效果一样
3. 条件跳转指令 JC 是依据（ D ）做是否跳转的判断。
   A. ZF　B. OF　C. SF　D. CF
4. 在C语言程序中，按行优先顺序访问数组元素往往有较高效率的原因是（ D ）。
   A. 在 C 语言程序中，数组是按行优先顺序在内存中存放的
   B. 这样的存储访问有比较好的空间局部性
   C. 能提高缓存的命中率
   D. 以上都对
5. 内核保存的、用于实现进程调度的进程上下文信息是（ C ）
   A. 全局变量值　B. 局部变量值　C. 寄存器　D. 磁盘文件
6. 一个进程用 fork 函数创建子进程成功时，fork 函数在该进程(父进程)中的返回值为(B)。
   A. 0　B. 子进程 ID　C. 子进程组 ID　D. 父进程的子进程数量
7. 使用函数 execve 启动一个程序时，错误的叙述是（ C ）。
   A. 要删除原进程的页表和 vm_area_struct 链表
   B. 创建新的页表和 vm_area_struct 链表
   C. 将.bss 设定为请求二进制零、映射到匿名文件且初始长度为 0 的区域；
   D. 将栈设定为请求二进制零的、映射到匿名文件且初始长度为 0 区域也；
8. 在 X86-64 Linux 平台下，关于 C 语言函数的参数传递，说法错误的是(D)
   A. 采用寄存器，可能提高程序速度
   B. 使用 XMM 有固定的约定

C. 参数数量较多时，也采用栈来传递
D. 全部采用栈传递参数

9. 在 X86-64 Linux 平台下的 64 位程序中，关于 C 语言函数的说法错误的是（
A. 非 static 局部变量可以在栈中
B. 非 static 局部变量可能在寄存器中
C. static 类型的局部变量也可能在栈中
D. 参数可能在栈中

10. 可以用函数(　D　)，通过复制文件描述符指针和打开文件表节点引用计数的方式，实现输出重定向。
A. copy　　B.　write　　C. mov　D. dup2

二、填空题（每空 1 分，共 10 分）

11. Linux 下，将源程序 hello.c 编译生成可执行文件 hello 的指令是__gcc hello.c -o hello_，将可执行文件 hello 反汇编结果输出到文本文件 1.text 的命令是：___objdump hello -D > 1.txt。

12. Linux 下查看作业的指令是_jobs_____

13. C 语言程序定义了结构体 struct noname{ int n; char c; short k; char *p;};若该程序编译成 64 位可执行程序，则 sizeof(noname)的值是___16___。

14. C 语言程序中的局部变量可能在__栈___中或__寄存器__中，也可能被编译器优化掉而在可执行文件中消失。

15. 在计算机的存储体系中，速度最快的是__寄存器____。

16. 在 X86-64 Linux 平台下，C 语言函数调用时，整数值类型的参数用__%rdi、 %rsi 、%rdx、%rcx_、%r8、%r9 等最多 6 个寄存器来传递，整数值类型的函数返回值则由寄存器__%rax___传递。

三、判断对错（每小题 1 分，共 10 分，在题前打√X 符号）

17. （ √ ）小尾顺序是指整数值在内存中存放时，将低位数值存在低地址的内存字节中。

18. （ √ ）在 Y86-64 的流水线实现中，采用了阶段寄存器来保存上一条指令的运行结果，从而允许阶段部件能尽快运行下一条指令。

19. （ √ ）链接时，static 修饰的函数会被当成局部符号，不能被其他模块引用。

20. （ √ ）在 Intel CPU 中，有专门保存当前正在运行进程的页表首地址的寄存器。

21. （ × ）C 语言中数值从 float 转换成 int 后，数值不会溢出。

22. （ √ ）在 shell 中，当用户按下快捷键 ctrl+c 后，shell 应该将 SIGINT 信号发送给前台进程组的所有进程。

23. （ × ）向偶数舍入是为了产生统计上没有偏差的结果。

24. （ × ）在 C 语言程序中，用 malloc 函数申请内存空间后，程序占用的物理内存立即会有相应数量的增加。

25. （ √ ）多个运行中的进程可以通过私有的写时复制方式来共享物理内存中的共享库，节省计算机的内存开销。

26. （ √ ）C 语言的函数 printf 并不是异步信号安全的，在信号处理程序中使用该函数有潜在的安全风险。

四、简答题（每小题 6 分，共 30 分）

27. 简述链接器在将目标文件链接生成可执行文件时，如何处理符号引用（符号解析）的。

答：每个正确的知识点 1 分，包括但不限于：

1) 根据强弱符号类别，确定被引用的符号在哪里
2) 根据各目标文件的段落大小、系统代码等信息，合并同类型的段，并计算可执行文件各段的大小，
3) 根据程序的内存映像，确定被引用的符号在程序运行时的内存地址
4) 根据在磁盘上的可执行文件中，各段落的大小，确定符号引用的位置
5) 根据上述信息，计算符号引用处应该采用的数值，并写入文件。

28. 以 gets 读入字符串为例，简述缓冲区溢出攻击的原理以及防范方法。

答：

攻击原理（2 分）：向程序输入缓冲区写入特定的数据，例如在 gets 读入字符串时，使位于栈中的缓冲区数据溢出，用特定的内容覆盖栈中的内容，例如函数返回地址等，使得程序在读入字符串，结束函数 gets 从栈中读取返回地址时，错误地返回到特定的位置，执行特定的代码，达到攻击的目的。

防范方法(3 种即可，3 分)：

1. 代码中避免溢出漏洞：例如使用限制字符串长度的库函数。
2. 随机栈偏移：程序启动后，在栈中分配随机数量的空间，将移动整个程序使用的栈空间地址。
3. 限制可执行代码的区域
4. 进行栈破坏检查——金丝雀

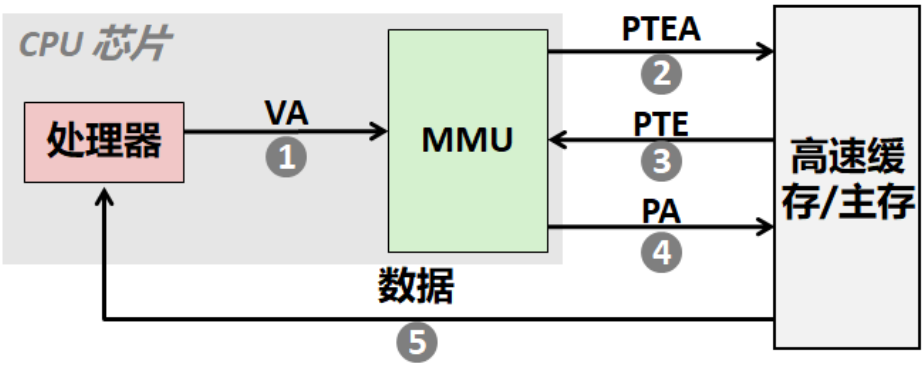
29. 程序速度性能优化的常用方法，及其能有效提高程序速度的原理。

答：答对一点就给 1 分，包括但不限于：采用循环展开、增加累积量、运算重组等手段

1） 减少数据依赖
2） 提高并发程度
3） 充分利用流水线（灌满）
4） 充分利用 CPU 中的多个计算单元
5） 减少循环开销

30. CPU 执行从内存读数据的指令，相应的虚拟地址翻译过程如下图所示。假设页面命中（在物理内存中），解释翻译步骤 1-5，包括每一步传送的数据、类型、如何得来等。

授课教师 密 姓名 封 学号 线 学院系



31. 简述在 **shell** 命令行中输入“**./hello**”后回车，系统如何实现 **hello** 程序从启动运行直至结束的整个过程。

答：略

五、系统分析题（30 分）

32. 阅读如下函数 **myproc** 的汇编代码，在横线上说明对应语句的功能（5 分）

```
myproc:
    movl      $0, %edx   _____
    movl      $0, %eax
    jmp  .L2
.L3:
    addq(%rdi,%rdx,8), %rax   _____
    addq$1, %rdx   _____
.L2:
    cmpq      %rsi, %rdx   _____
    jl    .L3   _____
    ret
```

33. 写出上述函数 **myproc** 的 **C** 语言源程序（5 分）

```
答案：long myproc0(long x[], long n)
{
long val = 0;
long i;

for(i=0;i<n; i++)
{
    val += x[i];
}
return val;
}
```

34. 顺序结构 **Y86 CPU** 的抽象结构如图 **1** 所示，该 **CPU** 每个周期执行一条指令，**CC** 是三位二进制状态码，从高到低每位分别对应 **ZF**、**SF**、**OF**。**%rbx**、**%rdx**、**CC** 的初值为都是 **100**，按图 **2** 顺序执行指令序列。

（1） 在周期 **3** 结束前一刻即图中时刻①时，**CC**、**PC**、**%rbx**、**%rdx** 的数值分别是多少**(5** 分**).**

（2） 图中时刻①时，组合逻辑是哪条指令的结果，并做解释？(5 分)

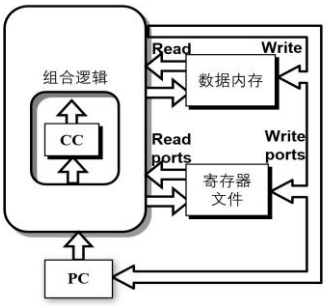
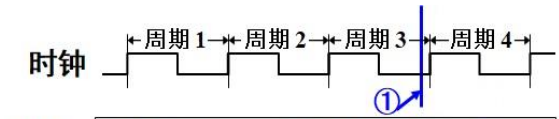


图 1 顺序结构 Y86 CPU 示意图



周期 1:	0x000: irmovq $0x123,%rbx
周期 2:	0x00a: irmovq $0x210,%rdx
周期 3:	0x014: addq %rdx,%rbx
周期 4:	0x016: je dest
周期 5:	0x01f: rmmovq %rbx,0(%rdx)

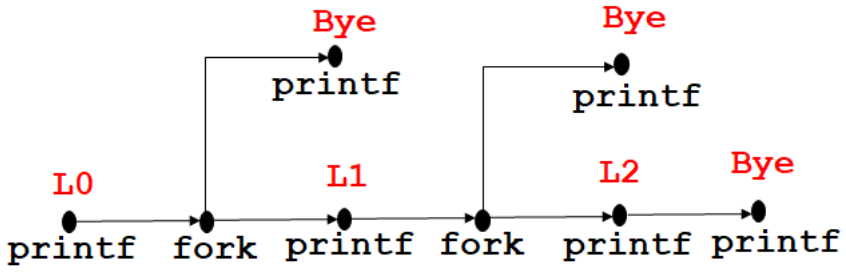
图 2 CPU 执行的指令序列

35. 有如下 C 语言程序，请画出程序的进程图，并给出可能的输出(10 分)。

```
int main()
{
    printf("L0\n");
    if (fork() != 0) {
        printf("L1\n");
        if (fork() != 0) {
            printf("L2\n");
 }
    }
    printf("Bye\n");
    return 0;
}
```

（1）进程图(5 分）：

答案：



（2）可能的输出（5 分）

答案：

L0	L0	L0	L0	L0
L1	L1	L1	Bye	Bye
L2	Bye	Bye	L1	L1
Bye	L2	Bye	L2	Bye
Bye	Bye	L2	Bye	L2
Bye	Bye	Bye	Bye	Bye

授课教师 姓名 学号 院系 密 封 线

六、综合设计题（共 10 分）

36. 若一计算机系统的情况如下：

- 内存是按字节寻址，字长为 1 字节（而非 4 字节）；
- 虚拟地址长度：25 位/bits，虚拟页面大小是 2*1024 字节；
- ；
- 物理页号 PPN 的长度是 8 位/bits；
- L1 d-cache 是物理寻址、直接映射（每组一行），行大小 8 字节，共 8 个组，当前数值如表 1 所示：

表 1 L1 d-cache 的数值

索引	标记位	有效位	块 0	块 1	块 2	块 3	块 4	块 5	块 6	块 7
0	35E	1	42	A0	75	50	42	0	05	50
1	27B	1	08	E3	00	A7	13	00	8B	52
2	C54	1	3F	75	AB	11	25	78	9A	00
4	A32	1	97	3A	91	D3	3F	12	86	22
5	C30	1	30	62	15	4C	48	A1	12	5C
6	B26	1	01	25	3E	62	1F	C4	85	12
7	01A	1	98	3A	12	D39	3F	3C	4D	5E

-

回答以下问题：

（1） 使用一级页表，页表条目 PTE 的总数量是多少？

2^{14}

（2） 物理地址的位数、虚拟页号 VPN 的位数？

19 、 14

（3） 在 L1 d-cache 中，块偏移、组索引、标记位的位长分别是多少？

块偏移是（ 3 ）位，组索引的位数是（ 3 ）位、标记的位数是（ 13 ）位。

（4） 请分析，在 CPU 从物理地址 0x31511 读取 1 字节数据，在访问缓存时，对应的组索引、标记位的数值、能否命中缓存、若能命中数值是多少？

组索引是（2 ）， 标记位数值是（ C54 ）、能命中、0x75