主管领导核签字

## 哈尔滨工业大学

学年 秋 季学期

2024

## 计算机系统(B) 试 题

题号	_	П	Ш	四	五	六	七	八	九	+	总分
得分											
阅卷人											

L		ال	
	ı		片纸鉴心 诚信不败
授课教师	**************************************	2.	<ul> <li>选择题(每小题 2 分, 共 20 分)</li> <li>C语言程序 hello.c 生成执行程序 ld 需链接的模块不包括( D )</li> <li>A. hello.o B. 标准库 libc-2.30.so C. ld-linux-x86-64.so.2 D.start.o</li> <li>Linux 下执行 int x=y/0 时, 非调试状态下的正确输出是( D )</li> <li>A. Segmentation Fault B. 无任何输出</li> <li>C. Divide OverFlow D. Floating Point Exception</li> <li>现代 Intel I7 计算机页表的 TLB 采用( B ) 级 Cache</li> </ul>
<b>春</b> 夕		4.	A. 1       B. 2       C.3       D. 4         静态局部变量是( D )。       A.全局符号 B.强符号 C.弱符号 D.本地符号         Intel I7 CPU 的第 3 级页表有( A )项/元素
			A. 512       B.1024       C. 2048       D.4096         关于固态硬盘错误的是(C)       A. 按页读写 B.按块擦除 C.顺序比随机访问慢 D.不宜存储太满下列函数调用后只返回一次的是(D)符号
合物		A. C.	A.fork       B. execve       C.setjmp       D.main         程序中的缓冲器溢出漏洞是由于(A)原因产生的         局部变量采用了数组       B.用了 malloc 等动态内存分配指令         程序采用了递归算法       D.编译时用了-fstack-protector         关于陷阱,描述错误的是(C)       A. 运行于内核态         B.运行结束后返回下一行指令处继续执行
弧然		10.	C. 按了 Ctrl-C 或 Ctrl-Z 等特殊键引起的         D. 是一种同步异常,内部由很多操作系统的共享功能模块组成         Intel I7 CPU 的机器语言中内存地址是(

```
二、 填空题(10 分,每空 2 分)
            11. 程序中 int i 为最大整数, int j=-i-1,则&j 开始的内存四个字节值为 00 00 00 80
            12. 函数调用 c=f (int a, int b)在 32 位环境下, b 采用 [ EBP + __12_ ] 进行访问
                                                                                EB(操作码) FE jmp
            13. 反汇编后 Intel x64 指令如下, L:
            14. Linux 下. a 结尾静态库用 AR
                                                                                     命令或软件生成
            15. void longjmp(jmp_buf j, int i)函数的返回值在_____EAX_
                                                                                                                                寄存器中
          判断对错(共10分,每题2分,正确打√、错误打×)
            16. ( × ) 子程序终止后向父进程发送 SIGINT 信号
            17. ( ✓ )缺页中断建立了虚拟内存到物理内存的映射
            18. ( × ) 采用连接时打桩需要提供被打桩的源程序
            19. ( × )局部变量是本地符号
            20. ( ✓ ) Unix IO 函数都不带任何缓冲
          简答分析题(共40分,每题10分)
四、
            21. 简述浮点数在数轴上的分布特点, 按步骤分析计算 float 数 -1.2 在内存各字节内容。
            答: 浮点数在数轴上成对称分布, 越靠近原点 0 则分布的密度越高, 越远离原点 0 则分
            步越稀疏。
                                        (2分)
            float: 32 bits, 1 个符号位, 8 位指数(127 移码), 23 位尾数(先导为 1 的规格化)。
            (1)转换成二进制: -1.00110011 [0011]...2
                                                                                                                     (1分)
                                                                                                                    (1分)
            (2)科学记数法, 先导为 1: -1.00110011 [0011] ... E 0
            (3)指数的 127 移码: 0+127=127 其二进制形式为 01111111
                                                                                                                       (2分)
            (4)尾数 23 位,向偶数舍入: 0011 0011 0011 0011 0011 0011 00...
                                                                                                                       (2分)
                                                        0011 0011 0011 0011 0011 010
            (5)IEEE754 编码: 1 0111 1111 0011 0011 0011 0011 0010 010 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 011 0
                                          其 16 进制 BF 99 99 9A
                                                                                                                    (1分)
            (6)内存小端存储: 9A 99 99 BF (1分)
            22. forkn 当前进程编号 100, 创建的每个子进程编号递增, sleep 时间较长进程会被切
                  换出去,分析当 c=1、p=2 及 c=2、p=1 时,各自最可能的执行结果是什么,为什么?
            答: P1:100 P2:101
                                                                                         void forkn(int c,int p){
                   P1:101 P2:0
                                                                                                 static int x=0;
                   child : x=1
                                                                                                 pid t pid=fork();
                   parent: x=-1
                                                                                                 printf("P1:%d P2:%d\n",getpid(),pid);
            fork 后先父, printf 切到子, 子 sleep1 到父, 父
            sleep2 到子,子 printf 后切到父,父 printf
                                                                                                 if (pid == 0) {
                                                                                                        sleep(c);
            答: P1:100 P2:101
                                                                                                        printf("child : x=%d\n", ++x);
                   P1:101 P2:0
                   parent: x=-1
                                                                                                        exit(0);
                   child: x=1
            fork 后先父, printf 切到子, 子 sleep2 到父, 父
                                                                                                 sleep(p);
            sleep1, printf 后,切到 csh,再切到子,子 printf
                                                                                                 printf("parent: x=%d\n", --x);
              (main 函数执行结果不用分析,题目中没有问) }
```

```
:23. 请面向 CPU 优化 A,面向存储器优化 B,写出优化的原理与结果程序。
        for(int i = 0; i < 2000; i++)
                                                             for(int j = 0; j < 4; j++){
               sum+=a[i];
                                                                 for(int i = 0; i < n; i++)
        printf("sum=%d", sum);
                                                                        m[i].v[j] = 0;
         程序A
                                                                 for(i = 0; i < n; i++)
        typedef struct{
                                                                        m[i].c[j] = 0;
           double v[4];
                                                            }
           double c[4];
                                                             程序 B
        } val m[N]
         答: A: 采用带累加器的分离的
                                                             sum=sum0+sum1+sum2+sum3
       循环展开原理, 充分利用超标量
                                                             printf("sum=%d", sum);
       CPU 的特征。
                                                             通过调整循环展开的级数,测试程
        int sum0=0;
                                                           序性能,确定下性能最优时的级数。
        int sum1=0:
                                                             B: 采用空间局部性原理, 使得
        int sum2=0;
                                                           连续访问存储器
        int sum3=0;
                                                             for(int i = 0; i < n; i++){
        for(int i = 0; i < 2000; i+=4){
                                                                 for(int j = 0; j < 4; j++)
               sum0+=a[i+0];
                                                                        m[i].v[j] = 0;
               sum1+=a[i+1];
                                                                 for(j = 0; j < 4; j++)
               sum2+=a[i+2];
                                                                        m[i].c[j] = 0;
               sum3+= a[i+3];
                                                            }
           //4 可以变成>=2 的数
    void switcher(long a, long b, long c, long *dest) switcher:
                                                          $7, %rdi
                                                   cmpq
           long val;
                                                          .L2
                                                   ja
                                                                            .L4:
           switch(a) {
                                                          *.L4(,%rdi,8)
                                                   jmp
                                                                                        .L3
                                                                              .quad
                             /* Case A */
           case
                                                   .section
                                                                 .rodata
                                                                                        .L2
              c =
                                                                              .quad
                                                  .L7:
中
              /* Fall through */
                                                                                        .L5
                                                                              .quad
业
                                                          $15, %rsi
                                                   xora
           case .
                             /* Case B */
                                                          %rsi, %rdx
                                                                                        .L2
                                                   movq
              val =
                                                                              .quad
                                                  .L3:
              break;
                                                                                        .L6
                                                                              .quad
                                                   leaq
                                                          112(%rdx), %rdi
                             /* Case C */
           case
                                                                                        .L7
                                                                              .quad
                             /* Case D */
                                                          .L6
           case
                                                   jmp
    线
                                                                              .quad
                                                                                        .L2
              val =
                                                  .L5:
                                                                                        .L5
              break;
                                                                              .quad
                                                   leaq
                                                          (%rdx, %rsi), %rdi
                             /* Case E */
                                                          $2, %rdi
                                                   salq
              val =
                                                          .L6
                                                   jmp
              break;
                                                  .L2:
           default:
                                                          %rsi, %rdi
                                                   movq
              val =
                                                  .L6:
           }
熈
           *dest = val;
                                                   movq
                                                          %rdi, (%rcx)
災
       }
                                                   ret
                  long val;
                  switch(a){
                                c=b^15;
                     case 5:
                                val=c+112; break;
                     case 0:
                     case 2:
                     case 7:
                                val=(b+c)*4;break;
                                val=a;
                     case 4:
                                            break:
                                val=b;
                     default:
                                           break;
                  *dest=val;
```

## 五、 综合设计题(20分)

25. 一计算机 CPU 字长为 2 字节,小端存储;内存按字节寻址,虚拟地址 26 位;使用一级 页表,页大小是 4KB;物理地址 20 位;TLB、L1 d-Cache 的当前全部数值如下:

表 1 TLB 数值表 (4 路组相联)

组	标记位 PPN		有效位 标记位 PPN		PPN	有效位	标记位 PPN		有效位	标记位 PPN		有效位
0	OCD	09	1	1F0	00	1	3E0	62	1	C4C	48	1
1	312	45	0	010	75	1	987	3A	1	D39	3F	0
2	038	E3	0	OA7	13	0	18B	52	1	49B	11	0
3	6C0	42	0	075	50	0	013	39	1	0F2	OD	0

表 2 L1 d-cache 的数值 (直接映射)

组	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	AO	75	50	42	0	05	50
1	27B	1	08	E3	00	A7	13	00	8B	52
2	014	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

此计算机系统的页表条目 PTE 的总数量是( 16K )个 ,TLB 中组索引是( 2 )位,标记是( 12 )位。在 L1 d-cache 中,组索引是( 3 )位、标记是( 14 )位。

CPU 从虚拟地址 0x7C0514 读取一个字的数值,将虚拟地址翻译成物理地址并获取数值的过程中,虚拟页号 VPN 是 ( 0X7C0 ),TLB 的索引是 0,TLB 的标记是 ( 0X1F0 ),TLB 命中,CPU 访存的物理页号是 ( 00 ),物理地址是 (0x00514),L1 D-Cache 命中,读取的数值是 ( 0x7825 ) 请按步骤给出这 10 个数据的分析与计算过程。

- 答: 1. 虚拟地址 26 位, 页大小 4K, 页内偏移地址占 12 位, 则页数 2^26 / 2^12 =2^14, 即 16K 个 PTE
  - 2. PTE 有 2<sup>14</sup> 个,则 VPN=14 位,或者 VPN==26-12=14 位,TLB 有 4 组,故 TLB 组索引为 2 位
  - 3. TLB 的 TAG 位=TLB 的 VPN 位-TLB 组索引位=14-2=12 位
  - 4. 物理地址 20 位, L1-dCache 有 8 组, 则组索引占 3 位
  - 5. 块大小 8 个字节, 块偏移占 3 位,则其标记位占 20-3-3=14 位
  - 6. 0X7C0514= 00 0111 1100 0000 <mark>0101 0001 0100</mark> B (26 位) 低 12 位为页内偏移,高 14 位为 VPN=00 0111 1100 0000 B =0x07C0
  - 7. 访问 TLB 的 VPN=00 0111 1100 00 00 B 低两位为索引 00,则高 12 位是标记 0x1F0
  - 8. 从 TLB 得到 PTE=00, 即物理页号 PPN=00
  - 9. PPN 与 页内偏移 一起构成物理地址 0000 0000 0101 0001 0100 B 即 0x00514
- 10. 物理地址 0000 0000 0101 0001 0100 B , 对应 010 即第 2 组,标记 014,命中。从 100 即 4 偏移处读出一个字,2 字节,为 25 78,按照小端模式,应为 0x7825

每个空答案正确1分,分析计算过程1分。