北京师范大学 2023-2024 学年夏季学期期末考试模拟试题

计算机系统导论

任课教师：计卫星

本试卷共 45 道题。全卷满分 120 分。考试用时 100 分钟。

注意事项：

1. 本试题为学生自命题的模拟试题，仅作《计算机系统导论》课程复习使用，与考试的实际题目无关。

2. 本试卷的题目有借鉴本门课程的教材、PPT、北京理工大学 2020 年的试题、北京大学 2013 年到 2021 年的期中、期末考试题。

3. 本试题的命题范围是：计算机系统漫游、数据的表示与处理、程序的机器级表示、程序的优化、链接、异常控制流、虚拟内存。共七章内容。

4. 由于本试题仅仅为学生戏作，本试题的题目或答案可能存在错误或纰漏，还望读者批评指正。

一、判断题：本题共 10 小题，每小题 2 分，共 20 分。

1. x86-64 位系统中的 long 型数据占用 8 个字节，指针也占用 8 个字节。（　　）

2. 按字节小端存储的计算机中，地址 0x123456 的存储格式为 65 43 21。（　　）

3. 按 IEEE 754 的浮点数（float）表示解读：0x7fc00000 的结果是 NaN。（　　）

4. 对于任意结构体，将其成员按照其实际占用内存大小从小到大的顺序进行排列一定不会使之内存占用最大。（　　）

5. 对于 IEEE 浮点数，如果把 1 位指数位交给小数部分，则其能表示的实数数量变多。（　　）

6. C 语言中，如果定义了两个变量名相同的强符号，链接器会随机选择一个使用。（　　）

7. 空闲块中互相指向的指针所占据的空间是内碎片。（　　）

8. 异常是由于计算机系统发生了不可恢复的错误导致的。（　　）

9. 某些 switch 语句不会被编译器 gcc 用跳转表的方式实现。（　　）

10. 表达式：`(-5) + sizeof(int) < 0` 为真。（　　）

二、选择题：本题共 20 小题，每小题 2 分，共 40 分。

11. 在 C 语言中输入下列指令后，a, b, c 的大小关系是 （　　）

```
int x = -2024;
int a = (x / 8) * 8;
int b = (x >> 3) << 3;
int c = (-(~x + 1) << 3) >> 3;
```

A. $a = b, b = c$　　B. $a = b, b > c$　　C. $a < b, b < c$　　D. $a < b, b = c$

12. 某系统由 4 个组件组成，组件 1（占系统的 5%）的性能可以提高一倍，组件 2（占系统的 20%）的性能可以提高 80%，组件 3（占系统的 45%）的性能可以提高 50%，组件 4 的性能暂时无法提高。若只能改进其中一个组件，则改进后可以取得系统性能最大提高的组件是 （　　）

A. 组件 1　　B. 组件 2　　C. 组件 3　　D. 组件 4

13. 下列有关 C 语言的浮点数说法错误的是 （　　）

A. C 语言的 float 类型浮点数为 32 位，而 double 类型则是 64 位。

B. 浮点数的加减运算可能出现“大数吃小数”现象，因为运算要向高阶对齐。

C. 在对浮点数 a, b 做加法运算时，$a + b$ 和 $b + a$ 的值总不相等。

D. 如果训练某 AI 不需要太大的浮点精度，就可以使用 16 比特表示的浮点数。

14. 下列结构体在 x86-64 机器上存储时，如果严格按照数据类型字节宽度对齐，且不考虑结构体成员存储顺序，需要的字节数是 （　　）

```
struct T
{
    int i;
    long l;
    int* p;
};
```

A. 16　　B. 24　　C. 32　　D. 40

15. 单精度浮点数使用 IEEE 格式，则下图中的 x 表示的是 （　　）

```
float f;
unsigned u = *(unsigned *)&f;
int x = (u >> 23) & 0xFF;
```

A. 阶码　　B. 符号位　　C. 尾数　　D. 溢出位

16. 下列避免缓冲区溢出的方式中，无效的是 （　　）

A. 编程时定义容量较大缓冲区的数组。

B. 编程时避免使用 `gets`，使用 `fgets`。

C. 程序运行时，随机化进程栈的偏移地址。

D. 在硬件级别引入“不可执行代码段”机制。

17. 若处理器实现了三级流水线，每一级流水线实际需要的运行时间分别为 1ns、2ns 和 3ns，则此处理器不停顿地执行完毕 10 条指令需要的时间为 （ ）

A. 21ns　　B. 12ns　　C. 24ns　　D. 36ns

18. 在 x86-64 系统中，要将寄存器 `%ax` 清零，下列指令错误的是 （ ）

A. `test %ax, %ax`　　B. `sub %ax, %ax`　　C. `xor %ax, %ax`　　D. `and $0, %ax`

19. 在一个具有 TLB 和高速缓存的系统中，假设地址翻译使用四级页表来进行，且不发生缺页异常，那么在 CPU 访问某个虚拟内存地址的过程中，最少会访问物理内存的次数、最多会访问物理内存的次数分别是 （ ）

A. 0，4　　B. 0，5　　C. 1，4　　D. 1，5

20. 关于内存别名（Memory Aliasing），以下说法正确的是 （ ）

A. 内存别名是指两个不同类型的指针指向不同的内存地址。

B. 内存别名是指两个不同的指针变量指向相同的内存地址。

C. 内存别名只会在编译期产生影响，而不会影响运行时性能。

D. 编译器可以总是准确地预测所有内存别名情况，从而优化代码。

21. 在链接时，下列符号需要进行重定位的是 （ ）

(1) 不同 C 语言源文件中定义的函数
(2) 同一 C 语言源文件中定义的全局变量
(3) 同一函数中定义时不带 static 的变量
(4) 同一函数中定义时带有 static 的变量

A. (1)(3)　　B. (2)(4)　　C. (1)(2)(4)　　D. (1)(2)(3)(4)

22. 下列关于局部性（locality）的描述错误的是 （ ）

A. 循环通常具有良好的时间局部性　　B. 循环通常具有良好的空间局部性

C. 数组通常具有良好的空间局部性　　D. 数组通常具有良好的时间局部性

23. 下列关于 x86-64 系统里过程调用的叙述中，不正确的是 （ ）

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. 调用返回时，向 PC 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。

24. 在虚拟存储系统的一次访存过程中，下列命中组合不可能发生的是 （ ）

A. TLB 未命中，Cache 未命中，Page 未命中。

B. TLB 未命中，Cache 命中，Page 命中。

C. TLB 命中，Cache 未命中，Page 命中

D. TLB 命中，Cache 命中，Page 未命中。

25. 在 foo.c 文件中包含如下代码。则在经过编译和链接之后，字符串“You ran into a problem!\n”会出现在 （ ）

```
int foo(void)
{
    int error = printf("You ran into a problem!\n");
    return error;
}
```

A. .bss 段　　B. .data 段　　C. .rodata 段　　D. .text 段

26. 文件 f1.c 和 f2.c 的 C 源代码如下所示。已知这两个文件在同一个目录下，在该目录下用 `gcc -Og -o f f1.c f2.c` 编译，然后用 `./f` 运行，这个过程中会出现的情况是 （ ）

```
// f1.c
#include<stdio.h>
extern float x;
extern void f();
int main()
{
    f();
    printf("%d\n",(int)x);
    return 0;
}
```

```
//f2.c
int x = 0;
void f()
{
    x++;
}
```

A. 编译错误　　B. 输出 0　　C. 输出 1　　D. 链接错误

27. 假设一些变量的定义如下所示。则在下列选项的优化中，编译器总是可以自动进行的是（ ）

```
const int n = 1e5;
int i; int j;  float x; int y;
int a[n]; int b[n]; int *p; int *q; int *r;
int foo(int);
```

A	`for (i = 0; i < n; i++)` `  x = (x+ a[j]) + b[j]`	`for (i = 0; i < n; i++)` `  x= x + (a[j] + b[j])`

B	`*p += *q`
	`*p += *r`
		`int tmp;`
		`tmp = *q + *r`
		`*p += tmp`

C	`for(i = 0; i < foo(n);`
	`i++)`
	`  sum += i;`
		`int tmp = foo(n)`
		`for(i = 0; i < tmp; i++)`
		`  sum += i;`

D	`for (i = 0; i < n; i++)`
	`  y = (y * a[j]) *`
	`b[j]`
		`for (i = 0; i < n; i++)`
		`  y = y * (a[j] *`
		`b[j])`

28. 关于 x86-64 系统中的异常（Exception），下列说法正确的是 （ ）

A. 除法错误是异步异常，Unix 会终止程序。

B. 键盘输入中断是异步异常，异常服务后会返回当前指令执行。

C. 缺页是同步异常，异常服务后会返回当前指令执行。

D. 时间片到时中断是同步异常，异常服务后会返回下一条指令执行。

29. 下列关于静态库链接的叙述中正确的是 （ ）

A. 如果库不是相互独立的，那么它们必须排序。

B. 链接时，链接器会拷贝静态库中的所有目标模块。

C. 使用库的时候必须将静态库文件放在程序的同一目录下。

D. 每个库在命令行只须出现一次即可。

30. Intel 的 IA32 体系结构采用二级页表，称第一级页表为页目录（Page Directory），第二级页表为页表（Page Table）。页面的大小为 4KB，页表项 4 字节。以下给出了页目录与若干页表中的部分内容，例如，页目录中的第 1 个项索引到的是页表 3，页表 1 中的第 3 个项索引到的是物理地址中的第 7 个页。则十六进制逻辑地址 8052CB 经过地址转换后形成的物理地址应为十进制的 （ ）

页目录		页表 1		页表 2		页表 3	
VPN	页表号	VPN	页号	VPN	页号	VPN	页号
1	3	3	5	2	1	2	9
2	1	4	2	4	4	3	8
3	2	5	7	8	6	5	3

A. 21195　　B. 29387　　C. 21126　　D. 47195

三、多选题：本题共 10 小题，每小题 2 分，共 20 分。在每小题给出的选项中，有多项符合题目要求的。全部选对的得 2 分，部分选对的按选对的数量占正确答案比例给分，有选错的得 0 分。

31. C 语言中的 int 和 unsigned 类型的常数进行比较时，下列**表达式为真**且**描述正确**的是（注：位宽为 32 位，$INT_{min} = -2147483648$，$INT_{max} = 2147483647$） （ ）

A. 0 == 0U，按无符号数进行比较。

B. 2147483647U > -2147483647 - 1，按无符号数进行比较。

C. (unsigned)-1 < -2，按无符号数进行比较。

D. 2147483647 > (int)2147483648U，按有符号数进行比较。

32. 下列选项中属于指令集并行的方式的是 （ ）

A. 时间片　　B. SIMD　　C. 流水线　　D. 超标量

33. 在 x86-64 系统中，下列指令会引起寄存器 esp 变化的是 （ ）

A. `movl %esp, %ebp`　B. `pushl %ebp`　　C. `call printf`　　D. `subl $20, %esp`

34. 关于内存管理中的隐式空闲链表和显式空闲链表，下列说法正确的是 （ ）

A. 隐式空闲链表通常使用一个标记位来表示一个块是否空闲。

B. 显式空闲链表在空闲块中存储指向下一个和上一个空闲块的指针。

C. 隐式空闲链表比显式空闲链表在找到一个空闲块时的效率更高。

D. 在隐式空闲链表中，找到一个空闲块可能需要遍历所有块。

35. 下列对于进程 P 的叙述中，可能导致进程 P 终止的是 （ ）

A. P 收到信号 SIGKILL　　B. P 从 main 函数返回

C. P 调用了 exit 函数　　D. P 调用了 wait 函数

36. 一段程序中阻塞了 SIGCHLD 和 SIGUSR1 信号。接下来向它按顺序发送如下信号：SIGCHLD，SIGUSR1，SIGCHLD，当程序取消阻塞继续执行时，其将执行的操作是 （ ）

A. 处理两次 SIGUSR1 信号　　B. 处理一次 SIGSUR1 信号

C. 处理一次 SIGCHLD 信号　　D. 处理三次 SIGCHLD 信号

37. 在编译并链接 main.c 文件时，需要链接 a.o、b.o 和 c.o 目标文件。下面几种命令可以正确生成可执行文件 program 的是 （ ）

A. `gcc main.c -o program a.o b.o c.o`

B. `gcc -c main.c a.o b.o c.o -o program`

C. `gcc main.c a.o b.o c.o -o program`

D. `gcc -o program main.c a.o b.o c.o`

38. 关于 fork 函数，以下说法正确的是 （ ）

A. fork 函数用于创建一个子进程，该子进程是父进程的完整副本。

B. 在父进程中，fork 函数会返回 0，表示成功创建子进程。

C. 在子进程中，fork 函数会返回子进程的进程 ID。

D. 如果子进程先于父进程终止且父进程未调用 wait，则子进程会变僵尸进程。

39. 对于下面的 `main` 函数，在系统调用成功的情况下，下列输出可能的是 （ ）

```
void handler()
{
    printf("h");
}

int main()
{
    signal(SIGCHLD, handler);

    if ( fork() == 0 ) {
        printf("a") ;
    } else {
        printf("b") ;
    }
    printf("c") ;
    exit(0);
}
```

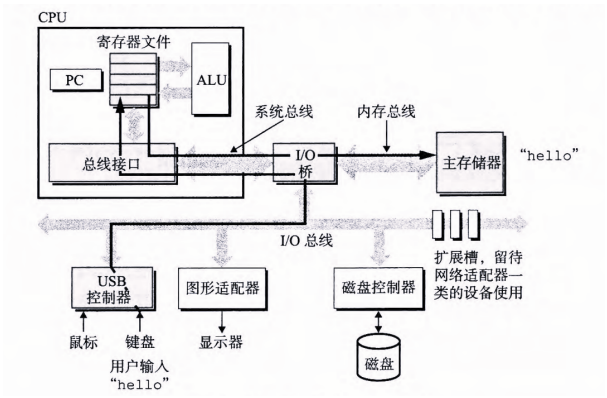
A. achbc　　B. abchc　　C. bcach　　D. bchac

40. 下列操作中，程序员可以做的是　　（　　）

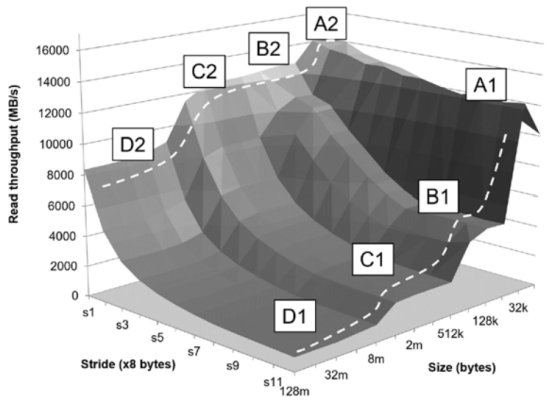
A. 更改 Cache　　B. 更改页表表项　　C. 寄存器分配　　D. 分配虚拟内存

四、解答题：本题共 5 小题，共 40 分。解答应写出文字说明、证明过程或者演算步骤。

41.（5 分）下图是程序 hello.c 在系统中运行过程的示意图，基于该图，简述在 shell 中输入“./hello”之后，系统运行的过程。



42.（5 分）在一台现代计算机上，我们可以通过运行某个函数扫描一个数组，从而测试计算机的性能。对于某台计算机，我们得到了如下图所示的性能表现。因变量为“读吞吐量”，即一个程序从存储系统中读数据的速率；自变量为扫描数组的步长和每个数组元素的大小。据此回答下列问题。



（1） 图中 A_1 到 D_1 这几个点有一些较为平缓的“平台”，这主要体现了访存行为的什么特性？如果 D_1 体现了内存的性能，那 A_1、B_1、C_1 分别体现存储器层次结构中的什么部件的性能？

（2） 图中 B_2 和 B_1、C_2 和 C_1 之间的“斜坡”，主要是由于访存行为的什么特性导致的？实际评测时，需要把题目中描述的函数先运行一遍，然后再运行一次获取评测数据，其原因是什么？

43.（10 分）x86-64 服务的 Linux 系统下有下列 m.c 及 foo.c 文件。它们经编译分别生成了 m.o 和 foo.o，编译过程未加优化选项。

```c
// m.c
void foo();
int buf[2] = {1, 2};
int main()
{
    foo();
    return 0;
}
```

```c
// foo.c
extern int buf[];
int *bufp0 = &buf[0];
int *bufp1;
void foo()
{
    static int count = 0;
    int temp;
    bufp1 = &buf[1];
    temp = *bufp0;
    *bufp0 = *bufp1;
    *bufp1 = temp;
    count++;
}
```

（1） 请写出链接 m.o 和 foo.o 并生成文件 mf 的 gcc 指令。并写出 mf 的文件格式。

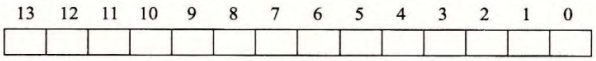
（2） 对于每个 foo.o 中定义和引用的符号，填写下表，指出其是否在符号表条目中。如果在，则指出定义该符号的模块（foo.o 或 m.o）、符号类型（局部、全局或外部）以及它在模块中所处的节；如果不在，这些列都填充“/”。

符号	是否在符号表中	符号类型	定义符号的模块	节
bufp0				
buf				
bufp1				
temp				
foo				
count				

44.（10 分）假设在某存储系统中，内存是按字节寻址的；内存访问是针对 1 字节的字的；虚拟地址是 14 位长的；物理地址是 12 位长的；页面大小是 64 字节的；TLB 是四路组相联的，总共有 16 个条目；Ll cache 是物理寻址、直接映射的，行大小为 4 字节，而总共有 16 个组。某时刻下内存系统的快照如下图所示。对于下列虚拟地址，请指出访问的 TLB 条目、物理地址，以及返回的缓存字节值。请指明是否 TLB 不命中、是否发生了缺页、是否发生了缓存不命中。如果有缓存不命中，对于“返回的缓存字节”用“-”来表示。如果有缺页，对于“PPN”用“-”来表示，而 C 部分和 D 部分就空着。

（1） 0x027c

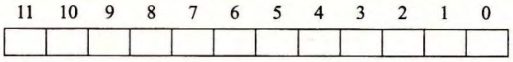
A. 虚拟地址格式



B. 地址翻译

参数	值
VPN	
TLB索引	
TLB标记	
TLB命中？（是/否）	
缺页？（是/否）	
PPN	

C. 物理地址格式

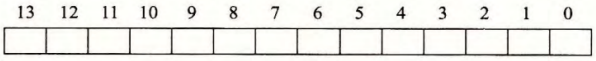


D. 物理地址引用

参数	值
字节偏移	
缓存索引	
缓存标记	
缓存命中？（是/否）	
返回的缓存字节	

（2） 0x03a9

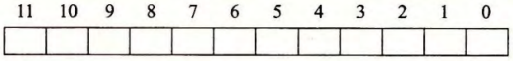
A. 虚拟地址格式



B. 地址翻译

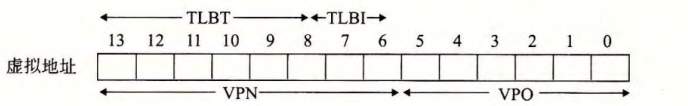
参数	值
VPN	
TLB索引	
TLB标记	
TLB命中？（是/否）	
缺页？（是/否）	
PPN	

C. 物理地址格式



D. 物理地址引用

参数	值
字节偏移	
缓存索引	
缓存标记	
缓存命中？（是/否）	
返回的缓存字节	



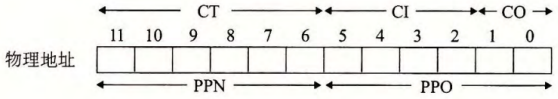
位	标记位	PPN	有效位	标记位	PPN	有效位	标记位	PPN	有效位	标记位	PPN	有效位
0	03	–	0	09	0D	1	00	–	0	07	02	1
1	03	2D	1	02	–	0	04	–	0	0A	–	0
2	02	–	0	08	–	0	06	–	0	03	–	0
3	07	–	0	03	0D	1	0A	34	1	02	–	0

a）TLB：四组，16 个条目，四路组相联

VPN	PPN	有效位
00	28	1
01	—	0
02	33	1
03	02	1
04	—	0
05	16	1
06	—	0
07	—	0

VPN	PPN	有效位
08	13	1
09	17	1
0A	09	1
0B	–	0
0C	–	0
0D	2D	1
0E	11	1
0F	0D	1

b）页表：只展示了前 16 个 PTE



索引	标记位	有效位	块 0	块 1	块 2	块 3
0	19	1	99	11	23	11
1	15	0	—	—	—	—
2	1B	1	00	02	04	08
3	36	0	—	—	—	—
4	32	1	43	6D	8F	09
5	0D	1	36	72	F0	1D
6	31	0	—	—	—	—
7	16	1	11	C2	DF	03
8	24	1	3A	00	51	89
9	2D	0	—	—	—	—
A	2D	1	93	15	DA	3B
B	0B	0	—	—	—	—
C	12	0	—	—	—	—
D	16	1	04	96	34	15
E	13	1	83	77	1B	D3
F	14	0	—	—	—	—

c）高速缓存：16 个组，4 字节的块，直接映射

45.（10 分）现有 x86-64 指令集服务的 Linux 系统。请基于此系统回答下列问题。

（1） 已知该系统内有如下汇编代码段，相关寄存器的初始值在表格中给出。请基于汇编代码段，在下面 C 代码中补充缺失的部分。并回答：汇编代码中存储循环变量 i 的寄存器是哪个？C 代码中 sum 的初值是多少？已知在理想情况下，处理器能在一个周期内处理一条汇编代码。则理想情况下，处理器执行完全部循环语句一共需要的周期数是多少？

```
.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

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

（2） 对于下列指令，已知 `divl [立即数]` 指令会将寄存器%eax 的值除以这个立即数。请基于汇编代码段，在下面 C 代码中补充缺失的部分，然后描述该程序的功能。

```
        pushl %ebp
        movl %esp,%ebp
        movl $0x0, %ecx
        cmpl $0x0, 8(%ebp)
        jle .L1
.L2
        movl $0x0, %edx
        movl 8(%ebp), %eax
        divl $0x0a
        addl %edx, %ecx
        movl %eax, 8(%ebp)
        cmpl $0x0, 8(%ebp)
        jg .L2
.L1
        movl 0x0, %edx
        movl %ecx, %eax
        divl 0x3
        cmpl 0x0, %edx
        jne .L3
        movl 0x1, %eax
        jmp .L4
.L3
        movl 0x0, %eax
.L4
```

```
int fun(unsigned x) {
    int bit_sum = 0;
    while ( (int) x > 0 ) {
        bit_sum += _________;
        x = x / 10 ;
    }
    if (_______________)
        return 1;
    else
        return 0;
 }
```

参 考 答 案

1.【解析】正确。32-bit 系统和 64-bit 系统的 C 语言基础变量类型的字节表示中，只有 long 和 pointer 两种类型的表示字节数不同。其中，32-bit 字节的 long 类型和 int 类型等长，为 4 个字节；而 64-bit 为 8 个字节。pointer 类型同上。

2.【解析】错误。0x123456 按字节小端存储的结果是 56 34 12。

3.【解析】正确。IEEE 754 的浮点数（float）表示：S（1 位）、E（8 位）、M（23 位）。原题表示的二进制串为：0/111 1111 1/100 0000 0000 0000 0000 0000，E 全为 1，而 M 不全为 0，故表示的是 NaN。

4.【解析】错误。结构体的成员之间可能存在内存对齐的要求。因此，如果结构体成员有一个 char 和一个 long，显然先放 char 就更浪费空间。

5.【解析】错误。NaN 变多了，所以能表示的实数更少。

6.【解析】错误。如果定义了两个变量名相同的强符号，链接器会报错；如果定义了两个变量名相同的弱符号，链接器会随机选择一个使用。

7.【解析】正确。解析略。

8.【解析】错误。异常不一定是不可恢复的，也可能是可恢复的，甚至是有意产生的。

9.【解析】正确。如果 switch 语句中的 case 比较稀疏，则不会被用跳转表的方式实现。

10.【解析】错误。由于 `sizeof()` 函数的返回值是无符号数，所以左边的表达式被强制转化为无符号数。最后结果为 UMax。

11.【解析】本题考查位运算黑魔法，最后 $a=b=c=-2024$，结果选 A。

12.【解析】本题考查 Amdahl 定理。Amdahl 定理：对系统的某个部分加速时，其对系统整体性能的影响取决于该部分的重要性和加速程度，公式表达为 $S=\dfrac{1}{(1-\alpha)+\alpha/k}$，$\alpha$ 为某部分占全部的比例；k 为该部分性能提高的倍数。经计算，$S_1=\dfrac{1}{0.95+0.05/1}=1$；$S_2=\dfrac{1}{0.8+0.2/0.8}=\dfrac{1}{1.05}$；$S_3=\dfrac{1}{0.55+0.45/0.5}=\dfrac{1}{1.45}$，故选 C.

13.【解析】本题考查浮点数。A、B 显然正确，C 选项，浮点数运算中由于精度误差和舍入问题，$a+b$ 等于 $b+a$，错误；D 为 jwx 老师上课讲过的原话，正确。选 C。

14.【解析】本题考查字节对齐。在 x86-64 架构中，int 通常是 4 字节对齐；long 和指针类型通常是 8 字节对齐。所以 int i 占用 4 字节，后面需要 4 字节填充，以满足 long 的对齐要求。long l 占用 8 字节。int* p 占用 8 字节。因此，结构体总大小应是 4（int）+ 4（填充）+ 8（long）+ 8（指针）= 24 字节，选 B。

15.【解析】本题考查代码阅读。这段代码的作用是：将浮点数 f 的位模式解释为无符号整数 u；右移 u 的 23 位，取接下来的 8 位，这就是阶码。选 A。

16.【解析】本题考查缓冲区溢出。B、C、D 均为讲义中所描述的避免缓冲区溢出的方式。A 选项则

没有用。对于一个大小为 N 的缓冲区，当输入规模为 $N+1$ 的数据时便可导致缓冲区溢出。选 A。

17.【解析】本题考查流水线。$3+3+3\times10=36ns$。选 D。

18.【解析】本题考查指令阅读。B，C，D 都能达到清零效果。A 中，`test` 指令相当于与操作，不是清零操作。选 A。

19.【解析】本题考查页表和 TLB 的位置。最好情况是 Cache 和 TLB 都命中，不需要访问内存；最坏情况是都不命中，需要访问四级页表项以及主存中的数据，一共 5 次。选 B。

20.【解析】本题考察内存别名。A 选项，内存别名是指两个不同类型的指针指向不同的内存地址，错误；B 正确；C 选项，内存别名会影响运行时性能，因为编译器在有别名的情况下可能无法进行某些优化，错误；D 选项，编译器无法总是准确预测所有内存别名情况，因此在有别名的情况下，编译器可能会保守地生成代码，错误。选 B。

21.【解析】本题考查重定位。重定位指的是将编译器生成的目标文件中的相对地址或符号引用转换成在最终可执行文件或库中的绝对地址。涉及到重定位的符号一般有：不同源文件中定义的函数和变量、全局变量、静态局部变量、动态库或共享库、代码的相对地址引用（分支、跳转）。故选 (1)(2)(4)，即 C。

22.【解析】本题考察局部性，但是是送分题。数组本身只能体现出空间局部性，显然选 D。

23.【解析】本题考查过程调用，但是是送分题。被调用函数在栈上分配空间以存储局部变量，在函数返回之前必须释放这些空间，以确保栈的正确使用，显然选 C。

24.【解析】本题考查 TLB、Cache、Page 的性质，是考研原题。D 中，TLB 和 Cache 都命中了，但所需数据不在内存中，显然不可能发生。选 D。

25.【解析】本题考查程序的堆栈结构。为防止更改，字符串常量通常存储在只读数据段.rodata 段，选 C。

26.【解析】本题考查符号引用。上述代码执行的逻辑是：由于 f2.c 中的 x 是强符号，故使用 (int)0 来解释 x；随后函数 f 作用 x，使得 $x=1$；在 printf 的强制类型转换之前，将 x 使用单精度浮点数解读，即一个很小的浮点数。类型转换后输出是 0，选 B。

27.【解析】本题考查编译优化障碍。A，浮点数不具有结合律，错误；B，可能存在内存别名，错误；C，函数 foo() 可能具有副作用，错误；D，整数可以使用结合律，正确。

28.【解析】本题考查异常。异常中同步、异步的判断要看指令。如果异常由指令引起，则是同步异常；反之则是异步异常。A，显然是同步异常，错误；B，显然是异步异常，但是会返回下一条指令执行，错误；C，缺页是由于指令执行时找不到某地址引起的，是同步异常，也会返回当前指令执行，正确；D，时间片到时是操作系统调控的，故是异步异常，错误。选 C。

29.【解析】本题考查库链接。A，正确；B，链接时只需要拷贝有用的模块，错误；C，在链接时并不需要将静态库文件放在程序的同一目录下，可以通过链接器的参数来指定静态库的路径，错误；D，相互调用的库在命令行必须重复出现，错误。选 A。

30.【解析】本题考查页表。4KB=2 ^ 12，所以页内地址有 12 位。4KB/4B=1K，所以页目录和每个页表中的页表项数为 1K 个。因此，在 32 位的虚拟地址中，最低的 12 位为页内地址（Offset），最高的 10 位为页目录的虚拟地址（Dir），中间 10 位为页表的虚拟地址（Table）。十六进制逻辑地址 8052CB 转换为二进制后为 10 0000000101 001011001011，Dir 为 10，即 10 进制的 2，在表中对应到页表 1。Table 为 101，即 10 进制的 5，在表中对应到物理页面 7。因此，物理地址应为 7

的二进制 111 和 Offset 的拼合，即 111001011001011，转换为十进制为 29387，答案为 B。

31.【解析】本题考查类型转换。A，表达式为真且解释正确；B，表达式应为“<”；C，表达式应为“>”，D，表达式为真且解释正确。选 AD。

32.【解析】本题考查指令集并行。A 显然是并发方式，B、C、D 都是指令集并行方式。选 BCD。

33.【解析】本题考查对指令和内存栈的理解。%esp 是栈指针的低 32 位。A 显然不会改变；B 将%ebp 压入栈中，所以%esp 会变；C 的调用函数指令会把返回地址压入栈中，所以%esp 会变；D 显然会变。选 BCD。

34.【解析】本题考查空闲链表。A、B 显然正确。C，式空闲链表在找到一个空闲块时的效率通常更高，因为它只需要遍历空闲块，错误；D，Best Fit 算法要求遍历所有块，正确。选择 ABD。

35.【解析】本题考查进程终止的条件。A、B、C 都是 PPT 中原话。D 是干扰项，wait 函数用于等待子进程的状态变化，特别是等待其终止，但不会导致调用该函数的进程本身终止。选择 ABC。

36.【解析】本题考查异常控制流。当程序阻塞了 SIGCHLD 和 SIGUSR1 信号时，这些信号将会被挂起，而不是被忽略。当取消阻塞后，挂起的每种信号类型将至少处理一次。这意味着即使同一个信号被发送多次，它也只会被处理一次，而不会重复处理。因此，信号无法支持子进程的计数。选 BC。

37.【解析】本题考查链接命令书写。A、C、D 都正确地指定了源文件 main.c 和目标文件 a.o、b.o、c.o，并生成名为 program 的可执行文件；B 是错误的，因为 -c 选项只会将 main.c 编译成目标文件，不会进行链接。选择 ACD。

38.【解析】本题考查 fork 函数。A 选项显然正确；B 和 C 正好相反，故均错误；D 选项，如果父进程调用 wait 系统调用，子进程的状态会被父进程回收，不会变成僵尸进程，反之则可能会变成僵尸进程。故选 AD。

39.【解析】本题考查 fork 函数。父进程输出“b，c”，子进程输出“a，c”，signal 命令的含义是：在子进程退出时调用 handler 函数输出“h”，所以“h”的输出一定要在一个“a”和一个“c”之后。选 ABC。

40.【解析】送分题。A、B，显然不能直接操作；C，古早的程序员需要做寄存器分配，正确；D，程序员可以使用 malloc 分配虚拟内存，正确。选择 CD。

41.【解析】首先，shell 将字符“./hello”逐一读入寄存器，再将其放入内存中。在我们输入回车之后，shell 执行一系列指令运行 hello 文件，这导致 hello 目标文件的代码从磁盘被读入主存，于是 CPU 就开始执行 hello 程序的 main 函数中的指令。这些指令中的输出指令会把输出字符串的每一个字节从主存复制到寄存器，再从寄存器复制到显示设备，最后显示在屏幕上。

42.【解析】（1）时间局部性；L_1 Cache、L_2 Cache、L_3 Cache。

（2）空间局部性；避免在缓存刚刚启动时，或对某个数据块的第一次访问时，由于该数据块还没有被加载到缓存中而发生的缓存未命中。

43.【解析】（1）`gcc -o mf m.o foo.o`；.out（答 ELF 文件给一半）。

（2）

44.【解析】

45.【解析】（1）8；%rcx；0；$8 \times 6 = 48$ 周期。这是一道送分题。

（2）x % 10；bit_sum % 3 == 0；该程序的功能是判断一个 unsigned 整数是否是 3 的倍数。

符号	是否在符号表中	符号类型	定义符号的模块	节
bufp0	是	全局	foo.o	.data
buf	是	外部	m.o	.bss
bufp1	是	全局	foo.o	COMMON
temp	否	/	/	/
foo	是	全局	foo.o	.text
count	是	局部	foo.o	.bss