计算机原理 E 卷

一、 选择题 (3 * 10 = 30 分)

1. 关于 IEEE 754 单精度浮点数表示，以下说法错误的是：
   - A. 规格化数的阶码 (Exponent) 字段通过偏置常数 127 编码。
   - B. 所有的单精度浮点数都能在数值轴上均匀分布。
   - C. 阶码全 1 且尾数全 0 表示正负无穷大 ($\pm\infty$)。
   - D. 非规格化数 (阶码全 0) 的引入是为了解决“逐步下溢”问题，使得 0 附近的表示更平滑。
2. 在现代处理器架构中，关于指令流水线 (Pipelining) 的“数据冒险 (Data Hazard)”，以下说法正确的是：
   - A. 只要增加流水线的深度，就可以完全消除数据冒险。
   - B. 写后读 (RAW) 冒险可以通过“旁路/转发 (Forwarding)”技术在某些情况下避免停顿。
   - C. 读后写 (WAR) 冒险在经典的 5 级顺序流水线中是非常严重的性能瓶颈。
   - D. 所有的分支指令都会引发数据冒险。
3. 链接器 (Linker) 在处理符号解析时，关于“强符号”和“弱符号”的规则，描述错误的是：
   - A. 不允许有多个同名的强符号。
   - B. 如果有一个强符号和多个弱符号同名，则选择强符号。
   - C. 函数和初始化的全局变量是强符号，未初始化的全局变量是弱符号。
   - D. 链接器在发现多个同名弱符号时会报错并停止链接。
4. 关于 Cache 的写策略，以下哪种组合在处理“写缺失 (Write Miss)”时能提供较好的空间局部性利用？
   - A. 直写 (Write-through) + 写分配 (Write Allocate)
   - B. 写回 (Write-back) + 非写分配 (No-write Allocate)
   - C. 写回 (Write-back) + 写分配 (Write Allocate)
   - D. 直写 (Write-through) + 非写分配 (No-write Allocate)
5. 在异常控制流 (ECF) 中，关于信号 (Signal) 的描述，正确的是：
   - A. 信号可以在任意时刻被进程接收，因此信号处理函数中可以安全地调用 `printf`。
   - B. 待处理信号 (Pending Signal) 会在位向量中排队，同一类型的信号如果发送多次，接收方会依次处理。
   - C. `sigprocmask` 函数可以用来阻塞特定的信号，使其在解除阻塞前保持待处理状态。
   - D. 信号处理函数执行在内核态，不能访问用户栈。

6. **关于虚拟内存中的“逆向页表 (Inverted Page Table)”，其主要优势在于：**
   - A. 减少了每个进程在页表查找时的访问次数。
   - B. 它的空间复杂度与物理内存的大小成正比，而不是与虚拟地址空间成正比。
   - C. 它完全消除了 TLB 缺失的可能性。
   - D. 它使得进程间的内存共享变得更加直接和简单。
7. **在编译优化中，以下哪种技术主要针对减少循环内部的冗余计算？**
   - A. 代码移动 (Code Motion / Loop-Invariant Reflection)
   - B. 循环展开 (Loop Unrolling)
   - C. 寄存器分配 (Register Allocation)
   - D. 条件移动 (Conditional Move)
8. **在 64 位系统下，关于过程调用标准 (ABI)，寄存器使用规则正确的是：**
   - A. `%rax` 用于传递第一个参数。
   - B. 函数的前 6 个整数参数依次通过 `%rdi`，`%rsi`，`%rdx`，`%rcx`，`%r8`，`%r9` 传递。
   - C. `%rbx` 是调用者保存 (Caller-saved) 寄存器。
   - D. 返回值总是通过栈来传递。
9. **关于 I/O 控制方式，DMA (直接内存访问) 相比于中断驱动 I/O 的核心优势是：**
   - A. DMA 不需要 CPU 参与任何初始化工作。
   - B. DMA 可以处理更复杂的协议转换。
   - C. DMA 在数据传输过程中不需要 CPU 的干预，减轻了 CPU 负担。
   - D. DMA 显著提高了单个字节传输的速度。
10. **动态内存分配器 (如 `malloc`) 中的“外部碎片”是指：**
    - A. 已分配块内部由于对齐要求产生的浪费空间。
    - B. 分配器本身占用的元数据空间。
    - C. 合计空闲空间足够满足请求，但没有一个单独的空闲块足够大。
    - D. 堆空间超出了操作系统设定的限制。

二、 填空题 (4 * 4 = 16 分)

1. 考虑一个 8 位的浮点数格式，1 位符号位，4 位阶码 (偏置量为 7)，3 位尾数。该格式下能表示的最大规格化正数为：___ (请用十进制表示)。
2. 在存储器层次结构中，若命中率为 99%，命中时间 (Hit Time) 为 1ns，缺失惩罚 (Miss Penalty) 为 100ns，则平均访问时间 (AMAT) 为：___ ns。

3. 利用位运算实现一个表达式，判断整数 x 是否为 2 的幂次 (且 $x > 0$)。表达式为：____ (结果为真表示是 2 的幂)。
4. *(题目开头部分缺失)* ... 为 8 字节，则该进程页表共需占用 ___ GB 内存空间。

三、 解答题 (54 分)

15. (10分) 浮点数与整数转换分析

给定 C 语言代码片段如下：

```c
int x = some_int_value;
float f = (float)x;
int y = (int)f;
```

(1) 是否对于所有的 int 值，都有 `x == y`？请说明原因。

(2) 如果将 `float` 换成 `double`，情况会有所不同吗?

16. (12分) 汇编代码逆向分析：递归函数

观察以下 x86-64 汇编代码：

代码段

```asm
func:
    testq   %rdi, %rdi
    jle     .L3
    pushq   %rbx
    movq    %rdi, %rbx
    subq    $1, %rdi
    call    func
    imulq   %rbx, %rax
    popq    %rbx
    ret
.L3:
    movl    $1, %eax    ; (推测代码，图中被遮挡但根据逻辑补全)
    ret
```

(1) 补全对应的 C 语言代码。

(2) 该函数实现了什么数学功能?

(3) 简述汇编中 `pushq %rbx` 和 `popq %rbx` 的作用。

17. (12分) 高速缓存 (Cache) 性能计算

假设一个直接映射缓存：$C = 512$ 字节，$B = 32$ 字节，$E = 1$。

考虑以下程序：

```
struct {int a; /* 4字节 */ int b; /* 4字节 */ } data[64];

int sum = 0;
for (int i = 0; i < 64; i++) {
    sum += data[i].a;
    sum += data[i].b;
}
```

假设 data 数组从内存地址 0x0 开始，Cache 初始为空。

(1) 缓存共有多少组 (Sets)?

(2) 该程序执行过程中的不命中率 (Miss Rate) 是多少?

(3) 如果我们将 `data` 数组的大小增加到 128，且 Cache 保持不变，不命中率会发生变化吗?

18. (10分) 信号处理与竞争冒险

考虑以下代码：

```
/* ...省略部分代码... */
// 关键位置 A
if (fork() == 0) {
    execve(...);
}
job_count++; // 关键位置 B // 关键位置 C
/* ... */
```

(1) 关键位置 B 的 `job_count++` 存在什么潜在的竞争风险?

(2) 如何使用 `sigprocmask` 修改代码以消除此风险?

19. (10分) 链接器与库解析

(1) 解释静态链接库 (`.a`) 和动态共享库 (`.so`) 在程序加载时的区别。

(2) 什么是“位置无关代码 (PIC)”？为什么共享库需要它？