

| 清华大学本科生考试试题专用纸 | |
|----------------|---------------------------------|
| 考试课程：操作系统（A卷） | 时间：2021年04月12日上午 09:50~11:50 |

| | | | |
|--------------|--------------|--------------|--------------|
| 系别： _____ | 班级： _____ | 学号： _____ | 姓名： _____ |
|--------------|--------------|--------------|--------------|

| | |
|---------|--|
| 答卷注意事项： | 1. 答题前，请先在试题纸和答卷本上写明A卷或B卷、系别、班级、学号和姓名。 |
| | 2. 在答卷本上答题时，要写明题号，不必抄题。 |
| | 3. 答题时，要书写清楚和整洁。 |
| | 4. 请注意回答所有试题。本试卷有18个小题，共7页。 |
| | 5. 考试完毕，必须将试题纸和答卷本一起交回。 |

一、(16分)判断题

1. ☐ 如果父进程先于子进程退出，那么子进程将无法被回收，成为僵尸进程，一直到系统重启。
2. ☐ 内存虚拟地址的页内偏移位数与物理地址的页内偏移位数是相等的。
3. ☐ CLOCK算法属于LRU类算法，不会有Belady异常现象。
4. ☐ 在 RISC-V SV39模式下，可以指定逻辑页面和物理页面(帧)的大小为1024字节。
5. ☐ 本课程实验的操作系统ucore/rcore 的架构是微内核架构。
6. ☐ 本课程实验用的RISC-V CPU具有M-mode、S-mode和U-mode，但实验编码仅涉及S-mode和U-mode。
7. ☐ 在实验二中的用户态代码无法执行特权级指令，从而避免了操作系统的代码和数据被用户态代码直接访问到。
8. ☐ 在RISC-V CPU的用户态(U-Mode)执行的代码可以关闭（disable）异常，但无法关闭（disable）中断。

二、(11分)填空题

9. 每个单线程进程有 (---1---) 个用户栈和 (---2---) 个内核栈；每个OS支持的运行在用户态的线程有 (---3---) 个用户栈和 (---4---) 个内核栈；每个用户线程(green thread, stackful coroutine)有 (---5---) 个用户栈和 (---6---) 个内核栈；每个协程(stackless coroutine)各有 (---7---) 个用户栈和 (---8---) 个内核栈。

10. O(1)是 (---9---) 多处理机调度算法；CFS是 (---10---) 多处理机调度算法；BFS是 (---11---) 多处理机调度算法。

填空选项提示：A. 单队列；B. 多队列。

三、(73分)简答题

11. (9分)

在课堂上小张深深的记住了老师的一句话，更改页表和修改页表之后一定要及时刷新 TLB。所以小张在完成 lab4 实验时在 mmap 和 mupap 函数中加入了大量的刷新 TLB 的语句。请回答如下实验相关问题：(注：每小问回答的字数在40字以内)

1.) lab4 中如何更换页表？何时更换页表？
2.) 小张插入的语句有必要吗？为什么？
3.) 更换页表之后需要刷新 cache 吗？无需解释原因。
4.) 在内核页表映射和用户页表映射中都有 trampoline 这一段，解释其映射特点和作用。

14. (10分)

1.) 请描述stride调度算法的工作原理。

ch3 中实现的 stride 调度算法中，要求进程优先级 ≥ 2 。

2.) 为什么要求进程优先级 ≥ 2 ？
3.) 其实上述要求也是可以调整。现在假定仅要求进程优先级 ≥ 1 ，并使用 `n` bit 无符号整型储存 stride。那么，在不修改 lab3 问答题所设计的比较算法的前提下，BIG_STRIDE 最大可以是多少？
4.) 在第二小问的前提下，你会选择设置 BIG_STRIDE 的策略是什么？此时进程优先级的合理范围是多少？为什么？

12. (8分)

下面是ch3 进行进程切换的代码：

C:

```
.globl swtch
swtch:
    sd ra, 0(a0)
    sd sp, 8(a0)
    sd s0, 16(a0)
    sd s1, 24(a0)
    # ... save s2-s9
    sd s10, 96(a0)
    sd s11, 104(a0)

    ld ra, 0(a1)
    ld sp, 8(a1)
    ld s0, 16(a1)
    ld s1, 24(a1)
    # ... restore s2-s9
    ld s10, 96(a1)
    ld s11, 104(a1)

    ret
```

Rust:

```
# SAVE_SN and LOAD_SN
__swtch:
    addi sp, sp, -13*8
    sd sp, 0(a0)
    sd ra, 0(sp)
    .set n, 0
    .rept 12
        SAVE_SN %n
        .set n, n + 1
    .endr
    ld sp, 0(a1)
    ld ra, 0(sp)
    .set n, 0
    .rept 12
```

```

        LOAD_SN %n
        .set n, n + 1
    .endr
    addi sp, sp, 13*8
    ret

```

1.) 请描述 switch 函数的输入参数
2.) 通用寄存器共 32 个，为何 switch 函数仅处理了其中的一部分？
3.) switch 函数要完成执行流和内核栈的切换，请问这两者分别发生在哪一行？

注：执行流切换指：PC寄存器指向新运行进程（也称任务）的逻辑；内核栈切换指：从此开始的所以栈操作将在新栈上发生；每小问回答的字数在40字以内。

13. (3分)

请问如下程序执行完毕后，会输出多少个 'a' ？ 请给出计算过程。

注：

- 逻辑判断支持短路
- fork() 返回值： 父进程返回非零值，被 fork() 出的程序返回 0

```

#include <unistd.h>
#include <stdio.h>
int main() {
    (fork() && fork() || fork() && fork() || fork() &&
    fork() && fork());
    printf('a');
    return 0;
}

```

15. (10分)

在实验lab2~lab4中，操作系统有了特权级的支持和页机制的支持。如果程序运行在用户态，这时产生了异常，系统调用或中断，导致RISC-V 64 CPU从用户态陷入到了内核态。请问：

(注：每小问回答的字数在40字以内)

1.) 在RVSC-V中，用户态进程如何实现系统调用陷入到内核态的？
2.) 如果是系统调用陷入到内核态，跳转到stvec寄存器的第一条机器指令开始执行时，产生了一个时钟中断，会出现什么情况？

3.) 如果是类似非法指令异常陷入到内核态, 跳转到stvec寄存器的第一条机器指令开始执行时, 产生了一个时钟中断, 会出现什么情况?
4.) 在lab4中引入了页机制后, 在执行中断上下文切换处理的实现中, lab4与lab3相比, 有何不同?
5.) 在lab4中引入了页机制后, 在执行用户进程上下文切换处理的实现中, lab4与lab3相比, 有何不同?

16. (5分)

在具有基址和边界寄存器的系统中, 可以实现地址转换。系统具体设置如下:

- 地址空间大小 1k
- 物理内存大小 16k

Base-and-Bounds register (基址-界限 寄存器) 信息:

```
Base   : 0x00000d9d (十进制: 3485)
Limit  : 467

Virtual Address
VA  0: 0x000003c5 (十进制: 965)
VA  1: 0x000001f9 (十进制: 505)
VA  2: 0x00000197 (十进制: 407)
VA  3: 0x00000277 (十进制: 631)
VA  4: 0x000000ad (十进制: 173)
```

请判断访问上述5个虚拟地址是否会产生访存异常, 如果不会产生, 那它对应的物理地址是多少?

17. (12分)

假定一个物理内存管理系统采用Best Fit连续内存分配算法, 系统初始时有一个大小为299字节的空闲块, 从地址1000开始。为方便查询, 系统为每个被分配的块设定了额外的4个字节head字段, 并确保每个被分配块的可用空间是4字节的倍数(即按4字节对齐), 空闲列表按地址排序(递增)。

- `Alloc()`: 返回值: 申请到内存块的起始地址 or -1 (表示申请失败)
- `Free()`: 返回值: 0 (表示释放成功) or -1 (表示释放失败)

第一次分配操作 `ptr[0] = Alloc(3)` 的结果示例如下:

returned: 1004 表示 Alloc申请到了一个内存空间, 起始地址是1004
Free List: [Size 1]: [addr:1008 sz:291] 表示: 空闲块链
表中有一个空闲块, 起始地址是1008, 大小为291字节

请针对下面的申请和释放内存操作序列, 回答每个申请/释放操作返回值, 以及每个操作之后空闲列表的状态。

1.) ptr[0] = Alloc(3) returned ?
Free List?
2.) ptr[1] = Alloc(5) returned ?
Free List?
3.) Free(ptr[0]) returned ?
Free List?
4.) ptr[2] = Alloc(6) returned ?
Free List?
5.) Free(ptr[2]) returned ?
Free List?
6.) ptr[3] = Alloc(8) returned ?
Free List?

18. (16分)

1.) 描述Sv39的虚拟地址结构、物理地址结构、地址转换过程。（注：回答的字数在150字以内）
2.) 小张在完成 lab4 的过程中碰到了奇怪的现象，他明明正确的设置了页表，但是用户执行还是会碰到 page fault，以下是小张核心实现的伪代码，请问小张犯了什么错误？如何改正？（注：回答的字数在40字以内）

Rust:

```
fn sys_mmap(va: VirtAddr, len: usize, user_prot: usize) {  
    let mut mm = TASK_MANAGER.get_current_memset_mut();  
    let permission = MapPermission::from_bits((user_prot  
<< 1) as u8).unwrap();  
    mm.push(MapArea::new(va, (va + len).into(),  
MapType::Framed, permission), None);  
}
```

C:

```
int sys_mmap(uint64 va, uint64 len, uint64 user_prot) {  
    struct proc* p = curr_proc();  
    // 该函数可以分配给定长度的连续物理内存  
    uint64 pa = kalloc_continuous(len);  
    mappages(p->pagetable, va, len, pa, user_prot << 1)  
}
```