

操作系统期中测试（2023）

一. 简答题（共9题）

1. 什么是用户态，什么是内核态，这两种状态如何切换？

参考答案：用户态是指程序在操作系统中以普通的用户权限运行，只能访问自己的内存空间和系统分配的资源，无法直接访问操作系统的核心部分和底层硬件。内核态是指操作系统运行在高特权级别的状态，可以访问系统的所有资源，包括 CPU、内存、I/O 设备等，并直接控制系统硬件。

通过系统调用、中断、异常事件出现时，会切换到内核态。执行完相应的系统调用、中断处理完成、异常处理完成、新建一个进程等都会切换到用户态。

2. 什么是系统调用？简述系统调用的基本流程。

参考答案：系统调用（system call）是操作系统提供给用户程序访问内核功能和系统资源的一种接口。系统调用是一种特殊的函数调用，它通常通过自陷（trap）的方式实现。当用户程序需要执行一个系统调用时，它会将系统调用的参数传递给操作系统，并触发自陷。操作系统收到中断信号后，会将控制权转移到内核态，执行相应的系统调用处理程序。系统调用处理程序完成操作后，将结果返回给用户程序，并将控制权交还给用户程序，使其继续执行。

3. Linux 中的僵尸进程是怎么产生的，怎么被回收？

参考答案：僵尸进程是指已经完成执行任务的进程，但它的进程描述符仍然存留在进程表中，其父进程却没有对其进行回收处理的一种进程状态。父进程通过 `wait()` 会回收掉这些僵尸进程。

4. 在用户态实现线程分别有何优点和缺点？

参考答案：在用户态实现线程的主要优点是：1) 轻量：用户态线程的创建、切换和销毁等操作不需要内核参与，因此开销较小，比内核态线程更轻量级。2) 灵活：用户态线程的调度策略、优先级、同步机制等都可以自己定义，从而能够更好地适应各种不同的应用场景。3) 简单：可以实现在只支持单线程的内核里。

在用户态实现线程的主要缺点：1) 阻塞问题：由于用户态线程是基于进程的，当一个用户态线程阻塞时，整个进程都会被阻塞。2) 操作系统由于没有这些线程的信息，因此可能会导致没有优化的调度策略。3) 某个线程长期占用 CPU：线程除了自己 `yield` 之外，不会放弃 CPU，会导致其它线程饿死现象。

5. 考虑如下多级反馈队列策略：a) 同优先级队列采用时间片轮转 RR；b) 高优先级队列拥有小一点的时间片，低优先级队列拥有大一点的时间片；c) 调度器首先选择高优先级队列中的第一个任务，用完它的时间片之后，其优先级会下降；d) 如果其时间片没有用完(例如被一个 I/O 任务阻塞)，其优先级不变，等待下次调度。请问如何愚弄(game)该多级反馈队列算法，以及如何应对这种愚弄？

参考答案：可以周期性的强制让任务在时间片用完之前被 I/O 短暂的阻塞，这样就能够始终获得高优先级的调度。可用通过额外的追踪其真正的使用 CPU 时长，来相应地

调整其优先级。

6. 内存管理中为什么要引入虚拟地址空间的概念？

参考答案：在多个进程共享使用物理内存的场景下，给所有进程都提供一致的地址空间，每个进程都认为自己是在独占使用计算机的内存资源；高效使用内存，支持多个进程共享物理内存、以及支持对物理内存的扩展（进程所需的地址空间大小超过物理内存空间大小）；实现内存保护，防止单个进程的地址空间被其他进程破坏。

7. 从地址转换、保护和共享、内部和外部碎片这几方面对分段和分页进行简要比较。

参考答案：1) 在分段机制下，虚拟地址划分为段号和偏移（offset），根据段号在段表中获得对应段的 base 和 limit，比较 offset 和 limit 实现对地址空间的保护，base + offset 获得物理地址；在分页机制下，虚拟地址划分为页号和偏移，根据页号在页表中获得对应页的物理页号（页框号），物理页号替换虚拟地址中的页号得到物理地址。2) 分段和分页都能保护进程地址空间不被其他进程破坏（段表项的 limit，按页进行分配），分段和分页分别以段和页的粒度实现共享。3) 内部和外部碎片：分段试图最小化内部碎片，但存在外部碎片；分页没有外部碎片，但存在内部碎片（取决于页大小）

8. 什么是缺页错误？操作系统处理缺页错误的基本流程是什么？

参考答案：所访问的虚拟页不在物理内存中时会产生 Page Fault。当产生 Page Fault 时，操作系统将挂起正在执行的进程；陷入内核；在后备存储上找到需要的页；将新的页面载入内存中的一个空闲页框，如果没有空闲页框，调用页面替换算法；更新进程的页表；等待进程调度，重新执行引发缺页中断的指令。

9. 操作系统经常利用局部性来提高性能，请给出操作系统中与此相关的两个例子，并简要说明其如何利用了局部性。

参考答案：虚拟内存利用了程序在数据访问上的局部性，即程序在某个运行时刻通常仅要求某些页面在内存中，并且通常会访问最近一段时间内访问过的页面。其它例子包括 TLB、工作集算法等（正确解释基本概念即可）。

二. 应用题（共 5 题）

10. Windows 的 CreateProcess 和 Linux 的 fork 都具有创建进程的功能，但语义稍有不同，请分析以下三种情况，说明 CreateProcess 和 fork 哪个更加合适，并解释原因。

- a) Shell 接收用户输入的命令 ls，并创建一个新的进程执行该命令。
- b) Web 服务器收到请求，并创建一个新的进程处理该请求。
- c) 父进程创建一个子进程，并利用管道通信。

参考答案：

a) Linux 的 fork 更加合适。因为在 Linux 中，Shell 通过 fork 创建一个子进程，然后通过 exec 函数族在子进程中加载执行新的命令 ls。这种方式可以保持父进程的环境变量、文件描述符等信息不变，可以顺利的在父进程的终端打印目录文件信息。而在 Windows 中，CreateProcess 函数需要指定要执行的可执行文件和命令行参数，因此无法像 Linux 中那样简单地执行一个命令。

b) Windows 的 CreateProcess 更加合适。因为处理一个新的请求往往是独立的，不需要继承父进程的信息。利用 CreateProcess 可以更快的新建响应的进程。

c) Linux 的 fork 更加合适。因为管道需要共享父进程的文件信息。

11. 下面代码总共创建多少进程（包含该父进程）？

```
#include <stdio.h>
#include <unistd.h>

int main() {
    /* fork a child process */
    fork();
    /* fork another child process */
    fork();
    return 0;
}
```

参考答案：4 个进程（1 个原始的父进程，3 个子进程）

12. 假设有如下一组进程，他们的 CPU 执行时间以毫秒来计算：

进程	执行时间	优先级（越大越高）
P1	2	2
P2	1	1
P3	8	4
P4	4	2
P5	5	3

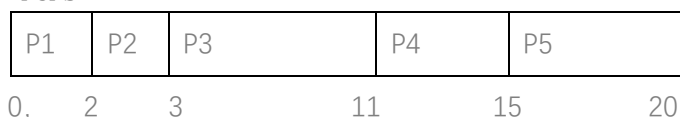
假设进程按照 P1, P2, P3, P4, P5 顺序在 0 时刻到达。

a) 画出甘特图，分别演示采用 FCFS、SJF、非抢占式优先级和 RR（时间片为 2）调度算法的进程执行。

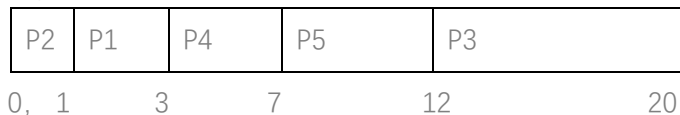
b) 对于上述问题，给出每种算法的平均等待时间。

参考答案：

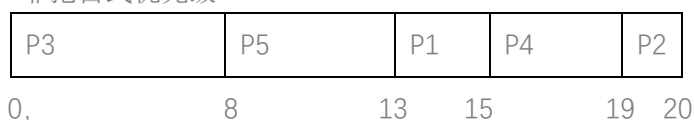
a) FCFS



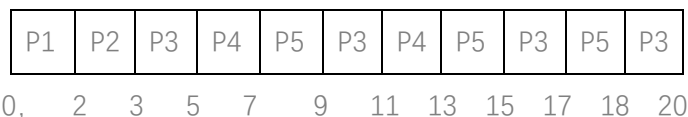
SJF



非抢占式优先级



RR



- b) FCFS 平均等待时间： $(0 + 2 + 3 + 11 + 15) / 5 = 6.2$
 SJF 平均等待时间： $(0 + 1 + 3 + 7 + 12) / 5 = 4.6$
 非抢占式优先级 平均等待时间： $(0 + 8 + 13 + 15 + 19) / 5 = 11$
 RR 平均等待时间： $(0 + 2 + 3 + 9 - 5 + 15 - 11 + 18 - 17 + 5 + 11 - 7 + 7 + 13 - 9 + 17 - 15) / 5 = 7.2$

13. 假设一个支持分页的计算机系统使用 32 位的虚拟地址和 40 位的物理地址，页面大小为 4KB，每个页表项除了页框号外还需要 4 bits 存储额外信息。

- a) 如果该系统采用二级页表，则应如何对虚拟地址进行划分？请简要解释依据
 b) 假设一个进程使用了其虚拟地址空间的 0~64K，此时一级页表和二级页表分别需要多少物理内存来存储？相应产生的内部碎片有多少？

参考答案：

a) 页面大小对应 12 位偏移，页表项（PTE）中页框号需要 $40 - 12 = 28$ bits，加上 4 bits 保护位，因此每个 PTE 为 4 bytes，一个页可以放 $2^{12} / 2^2 = 2^{10}$ 个 PTE，因此虚拟地址按 10 | 10 | 12 bits 划分。

b) 64KB 总共需要 $64/4 = 16$ 个页来存储，而每个页表可以存放 1024 个 PTE，因此只需要 1 个一级页表和 1 个二级页表，即 8KB 的物理内存；一级页表内部碎片为 4KB - 4 bytes，二级页表内部碎片为 4KB - 16 * 4 bytes，总共 8KB - 68 bytes。

14. 假设系统为某进程分配了 4 个物理页框，其中每一页对应的载入时间、最近一次访问时间、以及 Reference 和 Modified 位如下表所示。此时，FIFO、LRU、Second Chance 和 NRU (Not Recently Used) 页面替换算法分别会选择哪个页面换出？为什么？

页面	载入时间	最近一次访问时间	R	M
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

参考答案： FIFO 选择 Page 3（载入时间最早）、LRU 选择 Page 1（最近访问时间最早）、Second Chance 选择 Page 2（按 FIFO 序列）、NRU 选择 Page 2（R=0 & M=0）。