

北京邮电大学 2017-2018 学年第一学期

计算机学院

### “Operating Systems” Test (1)

Class \_\_\_\_\_

一、 Fill in blanks (10 points 注: 要求填入英文答案, 中文答案扣 0.5 分)

- 1) When a computer is powered on, the procedure of starting the computer by loading the OS kernel is known as booting the system.
- 2) The operating systems that allow users to use computers in interactive manners are called the time-sharing operating systems.
- 3) A software-generated interrupt caused either by an error or by specific requests from user programs that an operating-system service be performed is called a trap.
- 4) For  $n$  concurrent processes that mutual exclusively use some resources, the code segmentations, in which the processes access the resources, are called Critical Section.
- 5) To protect the OS and all other programs and their data from any malfunctioning program, hardware protection is needed. Two separate modes of CPU operations, that is the kernel mode and the user mode, are provided.
- 6) The data structure in kernel space used by OS to describe and manage processes is called PCB.
- 7) The CPU scheduler selects one among the processes that are ready to execute and allocates the CPU to it.

process

- 8) Real-time System are computers and/or software systems that react to external events in limited time intervals before the events become obsolete.
- 9) There are two common models of process communications, i.e. message-passing and memory-sharing (~~shared~~) shared memory (~~memory sharing~~)
- 10) The scheduling criteria include CPU utilization, throughput, turnaround time, waiting time, and response time.  
 响应时间      周转时间

## 二、 Choices (21 points)

1. 以下陈述中，正确的是 A.

- I. Android 是一种广泛应用于智能手机和平板电脑的操作系统;
- II. Unix 操作系统的一些发行版本可以支持大型主机和服务器;
- ~~III. Windows 和 Linux 操作系统属于开源操作系统;~~
- IV. 在 Windows 操作系统中，通过任务管理器可以查看系统内并非执行的进程信息，如 CPU 占用率、进程内线程数目、内存占用情况等。

A. I, II and IV      B. II and III      C. III and IV      D. II, III and IV

2. Among the following comments, only B are correct.

- ~~I. In a system, the state of a process can migrate from waiting to running.~~ <sup>ready</sup>
- II. PCB contains the process state, the program counter, CPU registers and user data.
- III. In a system with the operating system supporting kernel-level threads, the thread is the basic unit for CPU scheduling, and the process is the basic unit for resources allocation.
- IV. For several threads created by one process, they can share the files opened in the process.

~~A. I and II~~      ~~B. III and IV~~      ~~C. I and IV~~      ~~D. II and III~~

3. Which of the following is true about process's execute time, turnaround time, waiting time and response time D.

- waiting time*
- A. turnaround time = ~~response time~~ + execute time ~~X~~  
 B. response time  $\leq$  waiting time  $<$  turnaround time  
 C. waiting time  $\leq$  response time  $<$  turnaround time  
 D. No of the above

4. 下述 CPU 调度算法中, 具有 最小平均等待时间 的调度算法是 C.

- A. 时间片轮转法                      B. 先来先服务  
 C. 最短作业优先                      D. 基于优先级的抢占式调度  
 E. 多级队列                              F. 多级反馈队列

5. When we power up or reset a PC computer, the booting procedure starts, and the CPU will sequentially executes several chunks of codes, including

- I. boot block on disk    II. bootstrap in BIOS    III. OS kernel  
 IV. application programs    V. system programs

The correct order of code execution is D.

- ~~A. I, II, III, IV, V~~                      ~~B. I, II, III, V, IV~~  
~~C. II, I, III, IV, V~~                      D. II, I, III, V, IV

(09: 2)

6. 23. 单处理机系统中, 可并行的是 D.

- ~~I. 进程与进程~~    II. 处理机与设备    III. 处理机与通道    IV. 设备与设备  
 A. I, II, III    B. I, II, IV    C. I, III, IV    D. II, III, IV

7. 24. 下列进程调度算法中, 综合考虑 进程等待时间 和 执行时间 的是 D.

- A. 时间片轮转法调度                      B. 短进程优先调度  
 C. 先来先服务调度                          D. 高响应比优先调度



(10: 5)

8. 23. 下列选项中, 操作系统提供给应用程序的接口是 A.

- A. 系统调用      B. 中断      C. 库函数      D. 原语

9. 24. 下列选项中, 导致创建新进程的操作是 C.

- I. 用户成功登陆    II. 设备分配    III. 启动程序执行

- A. 仅 I, II      B. 仅 II, III      C. 仅 I, III      D. 仅 I, II, III

10. 25. 设与某资源相关联的信号量初值为 3, 当前值为 1。若 M 表示该资源的可用个数, N 表示等待资源的进程数, 则 M, N 分别是 B.

- A. 0, 1      B. 1, 0      C. 1, 2      D. 2, 0

(11: 3)

11. 23. 下列选项中, 满足短任务优先且不会发生饥饿现象的调度算法是 B.

- A. 先来先服务      B. 高响应比优先  
C. 时间片轮转法      D. 非抢占式短任务优先

12. 24. 下列选项中, 在用户态执行的是 A.

- A. 命令解释程序      B. 缺页处理程序  
C. 进程调度程序      D. 时钟中断处理程序

13. 25. 在支持多线程的系统中, 进程 P 创建的若干个线程不能共享的是 D.

- A. 进程 P 的代码段      B. 进程 P 中打开的文件  
C. 进程 P 的全局变量      D. 进程 P 中某线程的栈指针

(12: 4)

14. 28. 若 1 一个用户进程通过 read 系统调用读取一个磁盘文件中的数据, 则关

于此过程的叙述中, 正确的是 A.

I. 若该文件不在内存, 则该进程进入睡眠等待状态

II. 请求 read 系统调用会导致 CPU 从用户态切换到核心态

~~III. Read 系统调用的参数应包含文件的名称~~

A. 仅 I、II    B. 仅 I、III    C. 仅 II、III    D. I、II、III

15. 30. 若某单处理器多进程系统中有多就绪态进程, 则下列关于处理机(CPU)

调度的叙述中 错误 的是 C.

A. 在进程结束时能进行处理机调度

B. 创建新进程后能进行处理机调度

C. 在进程处于临界区时不能进行处理机调度

D. 在系统调用完成并返回用户态时能进行处理机调度

16. 31. 下列关于进程和线程的叙述中, 正确 的是 A.

A. 不管系统是否支持线程, 进程都是系统资源分配的基本单位

B. 线程是资源分配的基本单位, 进程是调度的基本单位

C. 系统级线程和用户级线程的切换都需要内核的支持

D. 同一进程中的各个线程拥有各自不一的地址空间

(13:)

17. 28. 下列选项中, 能导致用户进程从 用户态 切换到 内核态 的操作是 B.

I. 整数除零    ~~II. sin()函数调用~~    III. read 系统调用

A. 仅 I、II    B. 仅 I、III    C. 仅 II、III    D. I、II、III

18. 29. 计算机开机后, 操作系统最终被加载到 D.

A. BIOS    B. ROM    C. EPROM    D. RAM

(14: 3)

19. 23. 下列调度中, 不可能导致饥饿现象的是 A

- A. 时间片轮转      B. 静态优先级调度  
C. 非抢占式作业优先      D. 抢占式短作业优先

20. 25. 下列指令中, 不能在用户态下执行的是 D

- A. trap      B. 跳转      C. 后栈指令      D. 关中断

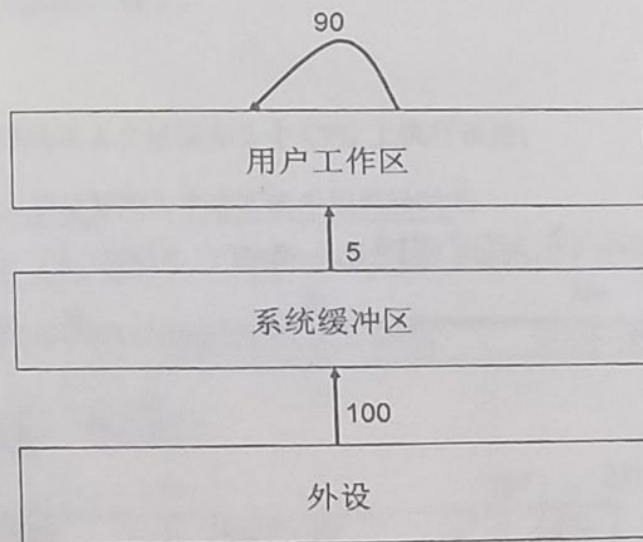
(15:1)

21. 25. 下列选项中会导致进程从执行态变为就绪态的事件是 D

- A. 执行 P(或 wait)操作      B. 申请内存失败  
C. 启动 I/O 设备      D. 被高优先级进程抢占

### 三、 简答(10 points)

1. (5 points) 设系统缓冲区和用户工作区均采用单缓冲区, 从外设读入 1 个数据块到系统缓冲区的时间为 100, 从系统缓冲区读入 1 个数据块到用户工作区的时间为 5, 对用户工作区中的 1 个数据进行分析的时间为 90 (如下图所示), 进程从外设读入并分析 2 个数据块的最短时间是多少, 为什么?



A. 200

~~B. 295~~

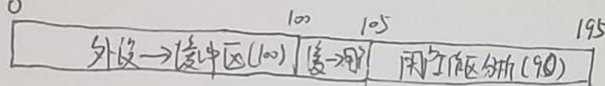
C. 300

D. 390

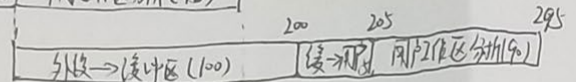
~~选 B. 295~~

如图所示:

第一个数据:



第二个数据:



由图可知, 共占用了 295 的时间。

$$\begin{array}{r} 195 \\ + 100 \\ \hline 295 \end{array}$$

2. (5 points) 在标准的 Reader-Writer 同步互斥问题中, 采用:

- 1) 内核空间中的写互斥二元信号量 `wrt`;
- 2) 用户空间中的读者计数变量 `readcount`;
- 3) 内核空间中控制对 `readcount` 进行互斥访问的二元互斥信号量 `mutex`, 实现读者-写者、写者-写者间对共享数据的互斥访问。

是否可以将 `readcount` 和 `mutex` 整合为 1 个定义在内核空间中、同时具有计数和同步/互斥双重功能的多元计数信号量 (`counting semaphore`) `readcount_semaphore`, 从而只采用 2 个信号量

- 1) 二元信号量 `wrt`
- 2) 多元计数信号量 `readcount_semaphore`,

实现 Reader-Writer 同步互斥问题, 为什么? (2+3 points)

答: 不能, 因为对 `readcount` 计数信号量进行判断时, 不能采用 "==" 来判断是否相等, 因此依然需要一个 `mutex` 控制 `readcount`。



四、 (30 points) 在 1 个在双 CPU 系统中 (不支持超线程 HT), 3 个并发进程的  
执行序列 (CPU burst, I/O burst) 如下:

$P_1$ : computing, 80ms  $\rightarrow$  I/O operation, 100ms  $\rightarrow$  computing, 40ms

$P_2$ : computing, 180ms  $\rightarrow$  I/O operation, 70ms  $\rightarrow$  computing, 20ms

$P_3$ : computing, 130ms  $\rightarrow$  I/O operation, 50ms  $\rightarrow$  computing, 50ms

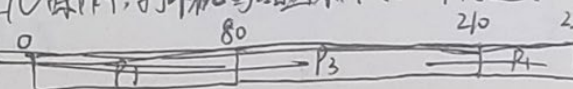
假设: 在这 3 个进程中,  $P_1$ 、 $P_2$  的 I/O 操作均为打印机输出操作, 且只有 1 台打印机;  $P_3$  的 I/O 操作为磁盘访问操作; 3 个进程的 CPU burst 可以任意分配到 2 个 CPU 上执行。

若不考虑调度和切换时间, 合理地安排这 3 个进程的执行步骤, 使得系统总吞吐量 (throughput) 最大。

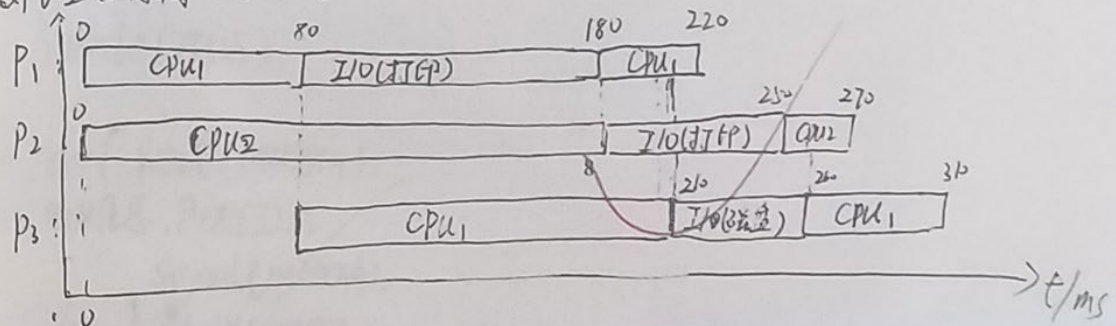
要求:

1. 利用甘特图描述这 3 个进程在 2 个 CPU 上执行轨迹;

2. 计算系统最大吞吐量和 3 个进程的平均周转时间

(1) 如图 ~~修改~~ I/O 操作中, 打印机与磁盘操作可以并行进行  
CPU<sub>1</sub> 甘特图: 

~~CPU2 甘特图~~: 如图所示:





2. 最大吞吐量 =  $\frac{3}{310}$  ~~200/1000~~ (进程/ms)

平均周转时间 =  $\frac{220+270+310}{3} \approx 266.67 \text{ (ms)}$

五、 (29 points) 考虑扩展的生产者-消费者问题。假设有限缓冲区容量为  $M$ ，存在四类并发进程：生产者进程 {Producer}，消费者进程 {consumer}，后续进程  $\{P_3\}$  和  $\{P_4\}$ ，每类进程均有多个。4 类进程的工作流程为：

1. 生产者进程每次向空缓冲区单元写入 1 个数据项；

2. 消费者进程每次从满缓冲单元取出 1 个数据项。

如果此时缓冲区中的数据项总数大于 0 且为 3 的倍数，则通知 1 个第三类进程  $P_3$  开始工作；如果缓冲区中的数据项总数大于 0 且为 5 的倍数，则通知 1 个第四类进程  $P_4$  开始工作；

3. 第三类进程  $P_3$  收到来自消费者进程的通知信息后，开始工作；

4. 第四类进程  $P_4$  收到来自消费者进程的通知信息后，开始工作；

在生产者、消费者访问缓冲区时，允许 1 个生产者、1 个消费者同时进入临界段中访问缓冲区，但不允许多个生产者、多个消费者同时进入缓冲区访问。

用信号量 wait、signal 机制实现 1) 生产者-消费者之间互斥地访问缓冲区，

2) 消费者进程与第三类、第四类进程间的同步

要求：

1) 定义正确的信号量和变量，给出其初值，并解释其含义和作用；

答：Semaphore  $\text{mutex}_1, \text{mutex}_2$  // 生产者、消费者互斥访问缓冲区控制

Semaphore  $\text{empty} = M$ ; // 判断缓冲区状态是否为空

Semaphore  $\text{full} = 0$ ; // 判断缓冲区状态是否为满

int  $\text{Count} = 0$  // 对数据项进行计数

binary semaphore mutex<sub>3</sub>=1, mutex<sub>4</sub>=1; // 用于控制进程 P<sub>3</sub>、P<sub>4</sub> 访问缓冲区  
 binary semaphore mutex-count=1; // 用于控制 count 的互斥访问  
 int count=0; // 即要数据项数目 用于控制 producer consumer 的 count.

2) 描述四类进程的工作流程

解:

Producer

```
do {
  wait(empty);
  wait(mutex1);
  向缓冲区写数据;
  signal(mutex1);
  signal(full);
  wait(mutex-count);
  count++;
  signal(mutex-count);
} while(TRUE);
```

Consumer

```
do {
  wait(full);
  wait(mutex2);
  从缓冲区读数据;
  signal(mutex2);
  signal(empty);
  wait(mutex-count);
  count--;
  signal(mutex-count);
  wait(mutex-count);
  if (count > 0 && count % 5 == 0)
    signal(mutex3);
  else if (count > 0 && count % 5 == 0)
    signal(mutex4);
  signal(mutex-count);
} while(TRUE);
```

P<sub>3</sub>:

```
do {
  wait(mutex3);
  收到消息, P3 进行工作;
  signal(mutex3);
} while(TRUE);
```

P<sub>4</sub>: do { wait(mutex<sub>4</sub>);  
 收到信息, P<sub>4</sub> 进行工作;  
 signal(mutex<sub>4</sub>);  
} while(TRUE);