班级：_____ 学号：_____ 班内序号_____ 姓名：_____

----------装---------订---------线----------

北京邮电大学 2005——2006 学年第 1 学期

《操作系统》期中考试试题

考试注意事项	一、学生参加考试须带学生证或学院证明，未带者不准进入考场。学生必须按照监考教师指定座位就坐。 二、书本、参考资料、书包等与考试无关的东西一律放到考场指定位置。 三、学生不得另行携带、使用稿纸，要遵守《北京邮电大学考场规则》，有考场违纪或作弊行为者，按相应规定严肃处理。 四、学生必须将答题内容做在试卷上，做在草稿纸上一律无效。

题号	一	二	三	四	五	总分
得分						

一、FILL IN BLANKS (1 point * 20)

1. Programming interface provided by operating system is __system call_____.
2. Privilege instruction refers to __the instructions that can obly be executed by operating system__。
3. The 5 basic states of processes are _____new_____, _____ready_____, _____running_____, _____waiting_____, and _____terminated_____。.
4. In a system there are 10 tape-drivers shared by M processes, each process needs 3 tape-drivers at most, then if M __<=4_____, the system can be deadlock free?
5. There are 3 jobs, their running time are 2, 5, and 3 hours. Assume they arrive at the same time, running on the same processor in single programming method; running sequence __J1, J3, J2__ will have the least average turnaround time.
6. The necessary and sufficient condition for deadlock are___mutual exclusion_____, ___hold and wait____, ____no preemption____, and ____circular wait_____.
7. The value of a semaphore specifies some meaning, if it is greater than or equal to zero, the value stands for ___the number of resources available__, if it is less than zero, its absolute value stands for ___the number of processes waiting for this type of resource_____.
8. Two communication methods between processes are _____shared memory_____ and _____message passing_____。
9. Programs loaded into and running in memory refers to _____processes_____.
10. 3 conditions that a good solution for critical section problems should satisfy are Mutual Exclusion，___progress__, and_____bounded waiting__。

二、Select the best answer for each blank (1 point * 10)

1. Contents of interrupt vector are_____B_____.
   A. begin address of sub-programs
   B. begin addresses of interrupt handling programs
   C. the address of begin addresses of interrupt handling programs
   D. begin address of handling programs
2. Deadlock avoidance is implemented by _____D_____.
   A. providing sufficient resources
   B. controlling proper sequence of processes progress
   C. destroying one of the 4 necessary and sufficient conditions
   D. preventing system enter into unsafe state
3. In multiprogramming system, in order to guarantee the integrality of shared variable, processes should enter their critical section mutual exclusively. Critical section refers to _____D_____.
   A. a buffer　　B. a data segment
   C. synchronous mechanism　　D. a code segment
4. There are 4 same type of resources shared by 3 processes, these resources can only be allocated or released one at a time. Each process needs 2 resources at most, so this system is_____D_____.
   A. some processes can never gain resources.
   B. deadlock consequentially
   C. resource requesting from process can be satisfied immediately
   D. deadlock free consequentially
5. User process creates a new process by calling system call fork(), before calling fork(), the user process is running in _____B_____; during running fork(), the user process is running in _____A_____.
   A. kernel mode　　B. user mode
   C. kernel mode or user mode　　D. internal mode
6. In multiprogramming systems, several processes can be running concurrently in memory and does not interfere each other. This is implemented by using ____B_____.
   A. memory allocation　　B. memory protection
   C. memory extension　　D. address mapping
7. Deadlock is mainly caused by _____B___and wrong sequence of processes progress.

A. Improper resource allocation B. shortage of system resource
C. improper job scheduling D. improper process scheduling

8. (C) Which of the following migrations is impossible?
A. running→ready B. running→waiting
C. waiting→running D. running→terminate

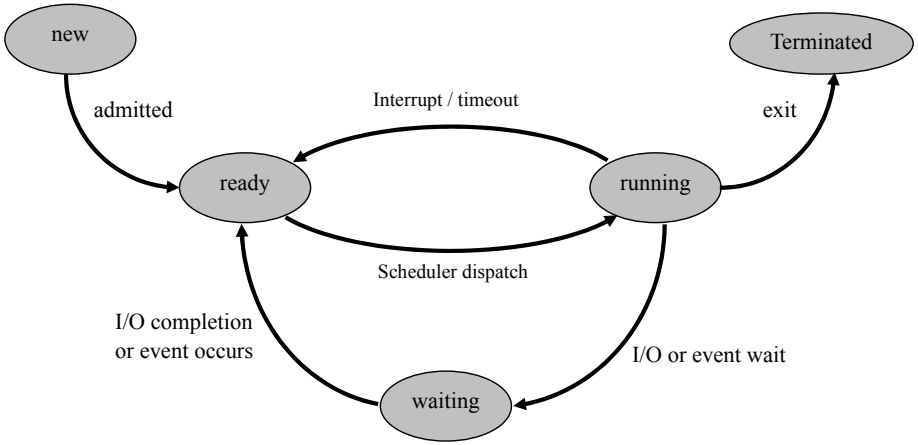
9. (C) Which of the following is not included in the context of process?
A. code B. PCB C. interrupt vector D. kernel stack

三、Judge the following statements, if right tick √, or X (1 point * 10)

1. (√) Operating System is driven by interrupt.
2. (X) Spooling technology can be used to increase the speed of slow peripheral equipments.
3. (X) The program stored in boot control block is initialization program for OS.
4. (X) Switch between threads can not cause the switch between processes.
5. (X) The cycle in process resource-allocation graph means there is deadlock in the system.
6. (X) Deadlock means that all processes in the system are in waiting state.
7. (X) When operation WAIT and SIGNAL are used to realize processes synchronization or mutual exclusion, the sequence of WAIT and SIGNAL must be right, or deadlock will be caused.
8. (X) The efficiency of semaphore is higher than that of monitor, but using semaphores is easy to lead to deadlock.
9. (√) A waiting process can not be waked up by itself.
10. (√) Programs running in concurrent system has the feature discontinuity.

四、Essay question (20 points)

1. (10 points) Please give the migration diagram of process with 5 states, and indicate the migration reasons.



2. (10 points)What is critical resources, and what is critical section? What conditions should be satisfied for a good solution to critical section problem?

ANSWER:

Critical resource is one that can be used by only one process at a time.

Critical section is a program code segment in which the critical resource is accessed.

A good solution to critical section problem should satisfy 3 conditions: mutual exclusion, progress, bounded waiting

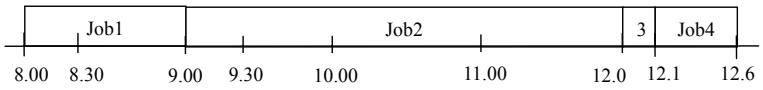
五、Calculation (40 points)

1. (10 points) Given jobs as following:

	Arrival time	CPU burst time
Job1	8.00	1.00
Job2	8.30	3.00
Job3	9.00	0.10
Job4	9.30	0.50

What is the average turnaround time for these processes with the FCFS scheduling algorithm and SJF scheduling algorithm?

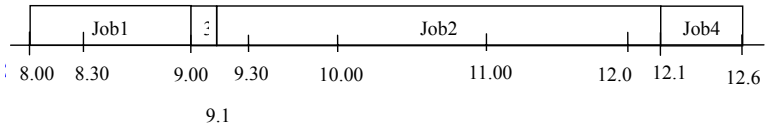
Answer:



FCFS:

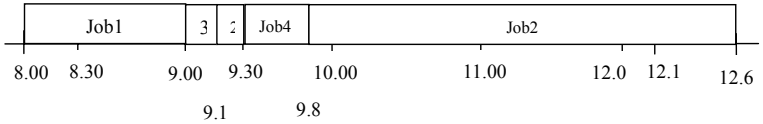
T1=9.0-8.0=1 T2=12.0-8.3=3.7 T3=12.1-9.0=3.1 T4=12.6-9.3=3.3
T=(T1+T2+T3+T4)/4=(1+3.7+3.1+3.3)/4=11.1/4=2.775

SJF
1) NON-PREEMPT



T1=9.0-8.0=1 T2=12.1-8.3=3.8 T3=9.1-9.0=0.1 T4=12.6-9.3=3.3
T=(T1+T2+T3+T4)/4=(1+3.8+0.1+3.3)/4=8.2/4=2.05

2) PREEMPT



T1=9.0-8.0=1 T2=12.6-8.3=4.3 T3=9.1-9.0=0.1 T4=9.8-9.3=0.5
T=(T1+T2+T3+T4)/4=(1+4.3+0.1+0.5)/4=5.9/4=1.475

2. (20 points) There is a plate on the table, only one fruit is allowed to be put into it at a time. Father puts one apple into the plate every time, mother puts one orange into the plate every time, the daughter takes the apple from plate to eat, and the son takes the orange from plate to eat. Please design processes for father, mother, daughter, and son by using semaphores.

Answer:

```
VAR
    Semaphore empty:=0;  apple:=0       orange:=0;
BEGIN
    parbegin
        Father: begin
                while (1) {
                    wait (empty);
                    puts apple into plate;
                    signal (apple) }
                end;
        Mother: begin
                while (1) {
                    wait (empty);
                    puts orange into plate;
                    signal (orange) }
                end;
        Son: begin
                while (1) {
                    wait (orange);
                    takes orange from plate;
                    signal (empty);
                    Eats orange }
                end;
        Daughter: begin
                    while (1) {
                        wait (apple);
                        takes apple from plate;
                        signal (empty);
                        Eats apple }
                    end;
    parend;
END
```

3. (15 points) Consider the following snapshot of a system, answer the following questions according to banker's algorithm.

	MAX				ALLOCATION				AVAILABLE			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	1	2	0	0	1	2	2	1	0	0
P1	2	7	5	0	2	0	0	0				
P2	6	6	5	6	0	0	3	4				
P3	4	3	5	6	2	3	5	4				
P4	0	6	5	2	0	3	3	2				

1) Calculate matrix NEED.
2) Now, is the system in safe state? Why?
3) If process P2 requests more resources, request[2]=(0,1,0,0), can this request be satisfied immediately? Why?

ANSWER:

1) NEED:

	A	B	C	D
P0	0	0	0	0
P1	0	7	5	0
P2	6	6	2	2
P3	2	0	0	2
P4	0	3	2	0

2) Yes.

PROCESS	WORK
	2 1 0 0
P0	2 1 1 2
P3	4 4 6 6
P4	4 7 9 8
P1	6 7 9 8
P2	6 7 12 12

There is a safe sequence of process (p0, p3, p4, p1, p2), so now the system is safe.

3) No.

request[2]=(0, 1, 0, 0) < Need (6, 6, 2, 2)

request[2]=(0, 1, 0, 0) < available (2, 1, 0, 0)

If allocate resources to P2,then the system state will be:

	MAX				ALLOCATION				AVAILABLE			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	1	2	0	0	1	2	2	0	0	0
P1	2	7	5	0	2	0	0	0				
P2	6	6	5	6	0	1	3	4				
P3	4	3	5	6	2	3	5	4				
P4	0	6	5	2	0	3	3	2				

Need:

	A	B	C	D
P0	0	0	0	0
P1	0	7	5	0
P2	6	5	2	2
P3	2	0	0	2
P4	0	3	2	0

Process	work
	2 0 0 0
P0	2 0 1 2
P3	4 3 6 6
P4	4 6 9 8

P1 and p2 can not finish, so there is not a safe sequence of processes, the system is in unsafe state, so request from P2 can not be satisfied immediately.