北京邮电大学 20222——2023 学年第一学期

《操作系统》期中考试试题

1. CHOICE (16 points)

(1) Which of the following statement is correct? __________

A. More instructions can be executed in user mode than in kernel mode

B. We can switch from user mode to kernel mode through software interrupt traps

C. The dual mode can prevent the process from entering the loop state

D. Privileged instructions refer to the instructions that can be used by user mode processes

B

(2) Which of the following system has the strict time constraint? ___________

A. distributed system B. time-sharing system

C. real time system D. interactive system

C

(3) If a system can deal with 4 interactive processes, 5 real-time processes, 6 batch processes in 250 ms, then the throughput of this system is___________

A. 10/s B. 15/s C. 30/s D. 60/s

D

(4) Modern operating systems are _________ driven. If there are no processes to execute, no I/O devices to service, and no users to whom to respond, an operating system will sit quietly, waiting for something to happen.

A. user B. interrupt C. device D. hardware

B

(5) The interrupt mechanism is used by operating system to handle the following events except ______.

A. I/O completion B. dividing by zero

C. virtual memory paging D. calling a procedure

D

(6) __________ is not included in the context of process?

A.PCB B. code C. kernel stack D. interrupt vector

D

(7) When does a process migrate from running state to ready state?_________

A. the process invokes a system call.

B. the process is selected by scheduler.

C. its time slice is used up.

D. event that the process is waiting for occurs.

C

(8) Among the following migrations, _________is impossible?

A. ready→running B. ready→waiting

C. running→ready D. running→waiting

B

(9) __________is the amount of time a process spent waiting in the ready queue.

A. Throughput B. Response time C. Waiting time D. Turnaround time

C

(10) ________is the interval from the time of submission of a process to the time of completion.

A. Waiting time B. Turnaround time C. Response time D. Throughput

B

(11) A starvation-free scheduling policy guarantees that no process waits indefinitely for service. Which of the following scheduling policies is starvation free? _________

A. Shortest Job First B. Priority C. Round Robin D. None of the above

C

(12) In multiprogramming system, in order to guarantee the integrality of shared variable, processes should enter their critical section mutual exclusively. Critical section refers to ___________.

A. synchronous mechanism B. a data segment

C. a code segment D. a buffer

C

(13) A deadlock situation can arise if the four necessary conditions hold simultaneously in a system. Which one of the following is not the necessary conditions? ________

A. mutual exclusion B. hold and wait C. preemption D. circular wait

C

(14) Which handling procedures of the following situations will not switch into kernel mode?______

A. procedure calls B. system calls C. interrupts D. traps

A

(15) In operating systems, the semaphore stands for instances of resource, it is a integer variable relevant to a queue， its value can only be changed by operation wait() and signal(). If a semaphore S is initialized to 10, now it's value is 2, how many processes is or are waiting in the queue relevant to S. ________

A. 8 B. 2 C. 1 D. 0

D

(16) In the following comments on processes and threads, only _______ is correct.

① As the result of I/O completion, a process switches from the waiting state to the ready state, and the CPU scheduling may take place.

② In a system with the operating system supporting kernel-level threads, the process is the basic unit for resources allocation, while the thread is the basic unit for CPU scheduling.

③ For several threads created by one process, one thread's stack and registers can be shared by the others.

④ The round robin algorithm will never result in process starvation.

A. ① ② ③ B. ① ② ④ C. ① ③ ④ D. ② ③ ④

B

2. (9 points) In a system with 8 processors, there are 10 concurrent processes sharing a type of resource based on a semaphore *S*, if at most 5 processes are allowed to enter their critical sections to use the resource, then answer the following questions.

(1) what are the maximum, and minimum numbers of processes in ready, running, and waiting state?

number of processes	ready	running	waiting
maximum	2	8	10
minimum	0	0	0

(2) what are the initial, maximum, and minimum values for the semaphore *S* respectively?

initial value	maximum value	minimum value
5	5	-5

3.（20 points）In a computer system with only one CPU, one input device and one printer. Processes A and B enter the system sequentially at time 0 and time 10, and B has a higher priority. The execution tracks of A and B are as follows:

A: CPU burst lasting 15ms, then I/O burst of 20ms on the input device, and then CPU burst lasting 40ms, then I/O burst of 30ms on the printer, and then CPU burst lasting 20ms, exiting.

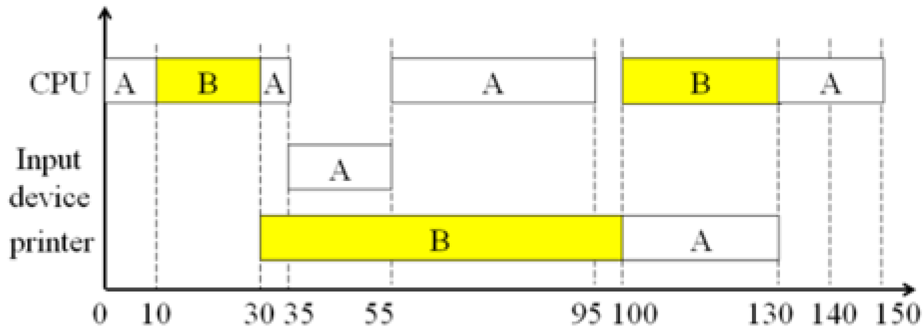
B: CPU burst lasting 20ms, then I/O burst of 70ms on the printer, and then CPU burst lasting 30ms, exiting.

(a) Suppose that preemptive priority scheduling algorithm is employed, draw the Gantt chart to describe the resource usage of A and B on the CPU, the input device and the printer.

(b) Calculate the waiting time and turnaround time for process A and B respectively.

16

(a)



4

(b) A: WT=20ms TT=150ms
B: WT=0ms TT=120ms

4. (20 points) Consider the following set of processes, their arrival time, CPU burst time, and priority numbers are as following. The length of the CPU burst given in milliseconds, and larger priority numbers imply higher priority.

Process	Arrival Time	CPU Burst Time	Priority number
P1	8	10	1
P2	9	10	3
P3	10	7	7
P4	11	5	5

(1) Suppose that preemptive SJF scheduling algorithm is employed,

(a) Draw a Gantt chart illustrating the execution of these processes;

(b) Calculate the average waiting time.

7

3

Waiting time: T=12, T2=21, T3=5, T4=0;

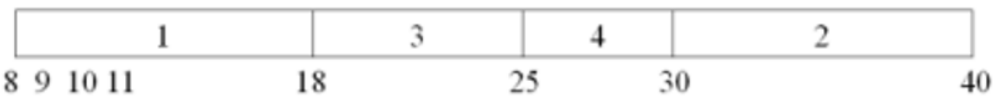
avg. (12+21+5+0)/4=9.5

(2) Suppose that priority-based non-preemptive scheduling is employed,

(a) Draw a Gantt chart illustrating the execution of these processes;

(b) Calculate the average turnaround time.

7



3

average turnaround time: (10+31+15+19)/4=18.75 ms

5. (15 points) Implement the Readers-Writers problem using semaphores.

Readers: only read the content of the shared data.

Writers: may update the content of the shared data.

Writers access the shared data mutually exclusive.

Readers and writers access the shared data in the order they request, no reader will be kept waiting unless a writer has requested access to the shared data.

```
//  3 points:
    int   r_count=0;   //当前读者数量
    Semaphore
       wrt=1;            //读者与写者互斥访问 shared data
       rw_queue=1;      //读者、写者排队访问 shared data F
       r_mutex=1;       //读者互斥访问变量 r_count

//  4 points:
    writer(){
        wait(rw_queue);
        wait(wrt);
           writing the shared data;
        signal(wrt);
        signal(rw_queue);
    }

//  8 points:
    reader(){
        wait(rw_queue);
        wait(r_mutex);
            r_count++;
            if (r_count==1)   wait(wrt);   // 当前第一个读者申请访问文件 F
        signal(r_mutex);
        signal(rw_queue);
            reading the shared data;
        wait(r_mutex);
           r_count--;
           if (r_count==0)   signal(wrt);   // 最后一个读者离开，释放文件 F
        signal(r_mutex);
    }
```

6. (20 points) Consider a system with 3 resources types (A, B, and C) and 5 processes (P1, P2, P3, P4, and P5). Resource type A has 17 instances, B has 5 instances and C has 20 instances。 The snapshot at time T0 is showed in the following table.

Process	Max requirement			Allocation			Available		
	A	B	C	A	B	C	A	B	C
P1	5	5	9	2	1	2	2	3	3
P2	5	3	6	4	0	2			
P3	4	0	11	4	0	5			
P4	4	2	5	2	0	4			
P5	4	2	4	3	1	4			

Assume the system uses Banker's algorithm to avoid deadlock.

3

(1) Calculate matrix NEED.

	A	B	C
P1	3	4	7
P2	1	3	4
P3	0	0	6
P4	2	2	1
P5	1	1	0

6

(2) Is the state at T0 safe? If yes, give the safety process sequence.

The system at time T0 is safe,
because there is a safe sequence existing (P4, P5, P1, P2, P3).

PROCESS	WORK		
	2	3	3
P4	4	3	7
P5	7	4	11
P1	9	5	13
P2	13	5	15
P3	17	5	20

4

(3) At time T0, can the request for resources (0, 3, 4) by P2 be granted? Why?

This request can not be granted,
because the resources available are not enough.

Request2=(0, 3, 4) , available=(2, 3, 3)

4

(4) Then based on (3), can the request for resources (2, 0, 1) by P4 be granted? Why?

This request can be granted,

because after allocation, resources available are (0, 3, 2), there still exists a safe sequence (P4, P5, P1, P2, P3).

Request4=(2, 0, 1) < need4=(2, 2, 1), available=(2, 3, 3)

after allocation:

Process	Need			Allocation			Available		
	A	B	C	A	B	C	A	B	C
P1	3	4	7	2	1	2	0	3	2
P2	1	3	4	4	0	2			
P3	0	0	6	4	0	5			
P4	0	2	0	4	0	5			
P5	1	1	0	3	1	4			

3

(5) Then, based on (4), can the request for resources (0, 2, 0) by P1 be granted? Why?

This request cannot be granted,

because after (4), the resources left are (0, 3, 2),.

Request1=(0, 2, 0) < need1=(3, 4, 7), available=(0, 3, 2)

If granted, then after allocation:

Process	Need			Allocation			Available		
	A	B	C	A	B	C	A	B	C
P1	3	2	7	2	3	2	0	1	2
P2	1	3	4	4	0	2			
P3	0	0	6	4	0	5			
P4	0	2	0	4	0	5			
P5	1	1	0	3	1	4			

Any process's need cannot be satisfied. this state is unsafe.