操作系统模拟卷

Okay, here is the restored content from the PDF document:

Page 1

课程代码	课程名称	考试学期	
BJSL0080	操作系统(双语)	考试形式	闭卷
BJSL0081	操作系统(全英文)	考试时长	120分钟
适用专业	计算机大类		
(开卷、半开卷	请在此写明考试可带哪些资料)		

题目	一	二	三	四	五	六	七	八	总分
得分									
批阅人									

Answer Sheet

1. True or False Questions [20 marks]

a	b	c	d	e	f	g	h	i	j

k	l	m	n	o	p	q	r	s	t

2. Short Answer Questions [30 marks]

(a)

Page 2

(d)

(e)

(f)

1. (20 points) True or False Questions.

Mark "T" or "F" on the ANSWER SHEET.

(a) The operating system is a program that acts as an intermediary between the application programs and the computer hardware.

(b) "Concurrency" implies a system can perform more than one task simultaneously, while "parallelism" supports more than one task making progress.

(c) A process can change from the Waiting state to the Running state directly if the required events occur.

(d) A thread shares code section, data section, stack, and other OS resources, such as open files and signals with other threads belonging to the same process.

(e) A blocking kernel-scheduled thread blocks all threads in the process.

(f) The disadvantage of spin locks include busy waiting and wasting CPU cycles when multiple processes share the single CPU.

(g) Starvation means deadlock. It happens when a process is never removed from the semaphore queue in which it is suspended.

(h) If there is a cycle in resource-allocation graph, then the system may or may not be in a deadlocked state.

(i) The shortest-job-first scheduling algorithm is the best choice in practice.

(j) If x is a condition variable, x.signal() always changes the value of x.

(k) The execution-time address binding scheme results in identical logical and physical addresses.

(l) Fixed-sized contiguous memory allocation suffers from internal fragmentation.

(m) A smaller page size leads to smaller page tables.

Page 3

(n) After a page-fault trap, it is necessary to restart the instruction which caused the trap.

(o) Assuming a thread is in the Running state, the thread will change state if it incurs a page fault.

(p) Belady's anomaly is the situation in which adding more physical memory to a system results in a higher number of page faults.

(q) As a process executes, it moves from locality to locality. A program is generally composed of several different localities.

(r) RAID 1+0 provides better fault tolerance in most cases than RAID 0+1.

(s) Polling is the only way that the OS can know whether an I/O device is busy.

(t) When a page is swapped into main memory, it is always brought in from the swap area of the disk.

2. (30 points) Short Answer Questions.

Provide your answers on the ANSWER SHEET.

(a) (6 points) Please compare internal fragmentation and external fragmentation, analyze their causes and solutions.

(b) (6 points) Please list at least 4 issues considered in deciding the page size, and illustrate how it affects.

(c) (6 points) Compare the concepts of contiguous allocation, linked allocation and indexed allocation.

(d) (6 points) Compare the concepts of long-term scheduling, medium-term scheduling and short-term scheduling.

(e) (6 points) Suppose that a disk drive has 100 cylinders, numbered 0 to 99. The drive is currently serving a request at cylinder 42, and the previous request was at cylinder 13. The queue of pending requests in FIFO order is 40, 3, 25, 10, 96, 45. Two disk-scheduling algorithms (SSTF: Shortest-seek-time-first, SCAN) are considered.

Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the requests for each of the two algorithms respectively?

Answer Other Questions (3-7) in the spaces provided on the QUESTION SHEET.

3. (8 points) CPU Scheduling.

Draw a Gantt chart for the CPU schedule and compute the average turnaround time for the set of processes, when (i) Preemptive Shortest Job First scheduling (ii) Round-Robin with time quantum q=3 is used for scheduling.

Process	Burst Time	Arrival Time	Priority
P1	9	0	3
P2	2	1	3
P3	7	3	3
P4	5	5	3
P5	3	7	3

Page 4

4. (8 points) Process Synchronization.

Consider a concurrent program that includes two functions, called funcA and funcB.

This program has the following synchronization requirements, both of which must be satisfied.

Requirement 1: At most one thread at a time may be running funcB.

Requirement 2: At most two threads at a time may be running any combination of funcA or funcB.

These requirements are summarized in the table, which shows which combinations of funcA and funcB may be executed concurrently.

Note that it is never OK for more than two threads to be running any combination of these functions concurrently.1

	funcA	**funcB**
funcA	OK	OK
funcB	OK	NO

Your task is to determine how to enforce these synchronization requirements using semaphores.

Your solution should not be more restrictive than necessary, and it should ensure that deadlock is not possible.

(a) List the semaphores that you will use in your solution in 1. For each semaphore, state what its initial value should be.

(b) Add semaphore operations (wait and signal) 2-5 to the skeleton code that threads should perform before and after each call to funcA and funcB to enforce the synchronization requirements.

Semaphores:①	
② //synchronizations `funcA();` ③ //synchronizations	④ //synchronizations `funcB();` ⑤ //synchronizations

Page 5

5. (12 points) Deadlock.

Consider the following system snapshot using the data structures in the Banker's algorithm, with resources A, B C and D, and processes P0 to P4:

Process	**MAX (A,B,C,D)**	**Allocation (A,B,C,D)**	**Need (A,B,C,D)**	**Available (A,B,C,D)**
P0	5, 1, 1, 7	3, 0, 1, 4		1, 0, 0, 3
P1	3, 2, 1, 1	2, 2, 1, 0		1, 0, 0, 3
P2	3, 3, 2, 1	3, 1, 2, 1		1, 0, 0, 3
P3	4, 6, 1, 2	0, 5, 1, 0		1, 0, 0, 3
P4	6, 3, 2, 5	4, 2, 1, 2		1, 0, 0, 3

Using the Banker's algorithm, determine whether each of the following states is safe or not.

If the system is safe, specify one execution sequence in that all the processes may complete.

Otherwise, briefly justify why the state is unsafe.

(a) Describe the four necessary conditions of a deadlock.

(b) How many resources of type A, B, C, and D are there? What are the contents of the Need matrix?

Is the system in a safe state? Why?

(c) If Available = (1, 0, 0, 3), and if a request (1, 0, 0, 1) from process P1 arrives, can this request be granted immediately?

(d) If Available = (1, 0, 0, 3), and if a request (1, 0, 0, 1) from process P4 arrives, can this request be granted immediately?

6. (10 points) Memory Management.

A computer system is byte-addressed and employs a two-level page table paging mechanism.

The virtual address format is as follows:

Outer page	**Inner page**	**Offset**	
10	10	12	

Please answer the following questions:

(a) What is the size of each page and page frame in bytes?

How many pages are there in the virtual address space of a process?

(b) If each outer page entry and inner page entry occupies 4 bytes, what are the MAXIMUM numbers of pages that can be used by the outer page table and inner page tables for a process?

Provide your calculation.

(c) If during one instruction cycle, the TLB (translation look-aside buffer) is empty and the accessed virtual addresses are 0x01000000 and 0x01112048, how many pages are accessed?

Explain the reason.

7. (12 points) Memory Management.

Consider the reference page sequence is 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5, 1, 2, 3. The number of page frame is 4.

(a) How many page faults for FIFO algorithm?

Fill your answer in the blank, and complete the following diagram to show how the page faults are generated.

Page 6

(a) How many page faults for FIFO algorithm? Fill your answer in the blank, and complete the following diagram to show how the page faults are generated.

Ref String	1	2	3	4	1	2	5
Frame1							
Frame2							
Frame3							
Frame4							
Fault?							

(b) How many page faults for OPT algorithm? Fill your answer in the blank, and complete the following diagram to show how the page faults are generated.

Ref String	1	2	3	4	1	2	5
Frame1							
Frame2							
Frame3							
Frame4							
Fault?							