Operating systems I

(1DT044)

Operating systems and process-oriented programming

(1DT096)

Written retake exam

Wednesday 2018-08-22

Bergsbrunnagatan 15, sal 1

08:00 - 13:00

The student must fill in the following information.

Anonymous exam code			-				

Table number	

When you hand in the exam, the following must be filled in by the staff in the exam hall.

Time for turning in the exam	

Instructions

Anonymous exam code

Write your anonymous exam code at the indicated line at the bottom of each page.

Closed book exam

This is a closed book exam. You are not allowed to bring any books, notes, calculators, computers, cell phones or other devices.

Multiple choice questions

The majority of the problems in this exam will be multiple choice problems. For each such problem there is a single correct answer or a single true statement. To answer a multiple choice problem, clearly circle one of the letters A, B, C, ... used for the listed options as shown in the figure to the right.

Problem: Today is: [1 pt]

A. Monday
B. Tuesday
(C.) Wednesday
D. Thursday
E. Friday

A correct answer will result in 1 (one) point if not stated otherwise. An incorrect answer will result in 0 (zero) points. Not making a choice will result in 0 (zero) points. Making more than one choice will result in 0 (zero) points.

Problem: Today is: [1 pt]

A. Monday
B. Tuesday
(⊗) Wednesday
D. Thursday
(E.) Friday

Tip: If you think there are more than one correct answer, circle the best answer. Based on the context and what has been presented during the course, use your judgement to select the best answer.

If you use ink and want to change your answer, cross over the circled option(s) you no longer want to circle as shown in the figure to the right.

Written answers

For a few questions where you must answer in writing, you may answer in English or Swedish. Do your best to write short and clear answers. Ambiguous answers will result in 0 (zero) points. Unreadable answers will result in 0 (zero) points. Only give one answer to each problem if not explicitly asked to provide more than one answer.

State any assumptions

If you suspect that something in a question is wrong, make a reasonable assumption of what is wrong or missing and write down this assumption as close to the question as possible.

Grading

The maximum score is 45.0 points. To pass the exam (grade 3) you must score at least 60 % (27 pts). For grade 4 you must score at least 75 % (33.5 pts). For grade 5 you must score at least 90 % (40.5 pts).

Mixed concepts

Pair each of the 10 numbered concepts with one of the statements by filling in one of the statement labels (A, B, C, ... or T) in the empty statement column to the right of the concept column. Note that there are 10 concepts and 20 statements. This means that there is only 10 statements that can be correctly paired with a concept. If you think there is more than one statement that describes a concept, choose the statement that best describes the concept. Each correct pairing will result in 0.5 points. If you pair more than one concept with the same statement label, this will result in zero points for all concepts paired with the same statement label.

[5 pt]

	Concept	Statement
1	System calls	
2	Round Robin	
3	Race condition	
4	Peterson's solution	
5	Response time	
6	TLB	
7	Paging	
8	Operating system	
9	Critical section	
10	External fragmentation	

	Statement
A	A notification sent to a process in order to notify it of an event that occurred.
B	Requires a priori information.
C	Controls the hardware and coordinates its use among the various application programs for the various user.
D	Sits between the main memory and the CPU registers.
E	A wrapper around the command interpreter that adds useful features that makes it easer to enter commands.
F	Entire process will block if a thread makes a blocking system call.
G	Requires mutual exclusion.
H	A variation on linked allocation.
I	A concurrent programming algorithm for mutual exclusion.
J	Solves the problem with external fragmentation.
K	Interface for requesting services provided by the operating system.
L	Enables multiple processes to share a single CPU and is an essential feature of a multitasking operating system.
M	Improves virtual address translation speed.
N	Amount of time it takes from when a request was submitted until the first response is produced.
O	Number of processes that complete their execution per time unit.
P	A non-preemptive scheduling algorithm.
Q	Suspends the execution of the parent process while the child executes.
R	Behaviour of an electronic, software or other system where the output is dependent on the sequence or timing of other uncontrollable events.
S	Total memory space exists to satisfy a request, but it is not contiguous.
T	Assigns a fixed time unit per process, and cycles through them.

Module 1

Fundamental concepts

Problem 1.1: An executing process: [1 pt]

A. is the result of compiling a program.
B. resides is in memory.
C. is a passive entity stored on secondary storage.
D. is an active entity stored on secondary storage.

Problem 1.2: In order to allow the operating system kernel to execute with more privileges than user application processes:

[1 pt]

A. all interrupts are turned off while executing in kernel space.
B. dual mode operation is used.
C. the kernel uses a separate stack.
D. a special bit vector is used.

Problem 1.3: When transitioning from kernel space to user space: [1 pt]

A. the values in all CPU registers must be restored from the stack.
B. the values in all CPU registers must be restored from a the CPU context of the resumed process.
C. a special system call exception is used.
D. a special system call interrupt is used.

Problem 1.4: Exceptions are: [1 pt]

A. used to notify the system of input events.
B. used to implement basic output.
C. considered to be asynchronous because the control unit issues them only after terminating the execution of an instruction.
D. considered to be synchronous because the control unit issues them only after terminating the execution of an instruction.

Problem 1.5: To initiate a system call: [1 pt]

A. the caller places the return address on the stack.
B. a special function call from user space to kernel space is used.
C. an interrupt is used.
D. an exception is used.

Problem 1.6: Multitasking: [1 pt]

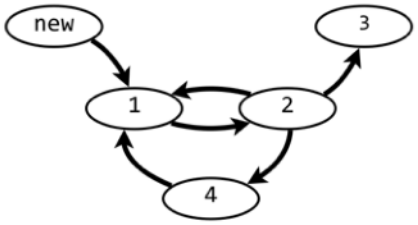
A. allows for a single job to get stuck in an infinite loop and block all other jobs from executing.
B. maximises the CPU utilisation.
C. is an extension of multiprogramming.
D. requires multiple CPU or CPU cores.

Module 2

The process concept and inter process communication

File descriptors, standard streams, and I/O redirection

Problem 2:1: A process can transition between various states as depicted in the below diagram. When the time quantum of a process expires, the process transitions: [1 pt]



A. from state 4 to state 1.
B. from state 1 to state 2.
C. from state 2 to state 1.
D. from state 2 to state 4.

Problem 2.2: A parent process creates a pipe and then calls fork(). Now: [1 pt]

A. both child and parent can read and write to the pipe.
B. only the parent can write to the pipe and only the child can read from the pipe.
C. only the parent can read from the pipe and only the child can write to the pipe.
D. only the parent can access the pipe.

Problem 2.3: Forgetting to close a pipe read file descriptor may cause: [1 pt]

A. EOF to be returned.
B. a SIGPIPE signal to be sent to a writer.
C. a writer to block.
D. a SIGPIPE signal to be sent to all readers.

Problem 2.4: On success, the exec() system call: [1 pt]

A. returns once in the parent and once in the child process.
B. returns a positive value.
C. only returns in the child.
D. never returns.

Problem 2.5: A process whose parent process has terminated: [1 pt]

A. becomes a zombie process.
B. will get a new parent process.
C. will also terminate.
D. will receive a special signal.

Problem 2.6: The dup() system call: [1 pt]

A. can be used to implement I/O redirection.
B. creates a copy of the process file descriptor table.
C. creates a new child process with the same file descriptor table as the parent.
D. creates an alias for an existing file descriptor.

Problem 2.7: The process control block (PCB): [1 pt]

A. is a data structure in user space containing the information needed to manage a particular process.
B. serves as the repository for any information that may vary from process to process.
C. makes is possible for one process to control another process.
D. must be saved on the stack when transition from user mode to kernel mode.

Problem 2.8: When a process terminates the exit status: [1 pt]

A. is sent to the parent using a signal.
B. is saved to the PCB.
C. is written to a special register.
D. can no longer be obtained by the parent.

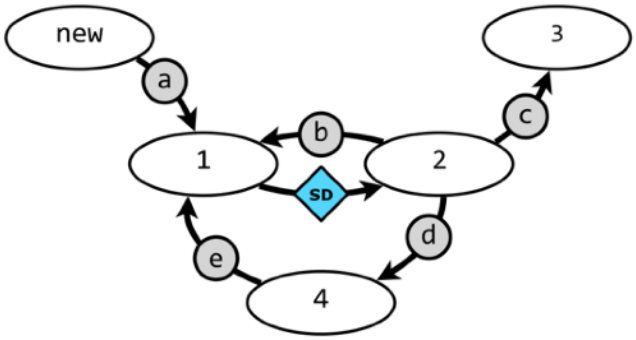
Problem 2.9: Asynchronous signals: [1 pt]

A. are generated by an event external to an executing process.
B. are delivered to the same process that performed the operation that caused the signal.
C. can not have their default signal handler overridden.
D. can be generated by an illegal memory access.

Module 3

CPU Scheduling

Problem 3.1: A process can transition between various states as depicted in the below diagram. The state transitions are labeled a, b, c, d, and e. Scheduler dispatch is marked with SD. Which state transitions may cause a preemptive scheduler dispatch? [1 pt]



A. All transitions.

B. Transitions c and d.

C. Only transition b.

D. Transitions a, b and e.

Problem 3.2: In which CPU scheduling algorithm can the convoy effect be observed? [1 pt]

A. Rund robin (RR).

B. Preemptive shortest job first (PSJF).

C. Shortest job first (SJF).

D. First-come, first-served (FCFS).

Problem 3.3: The medium term scheduler: [1 pt]

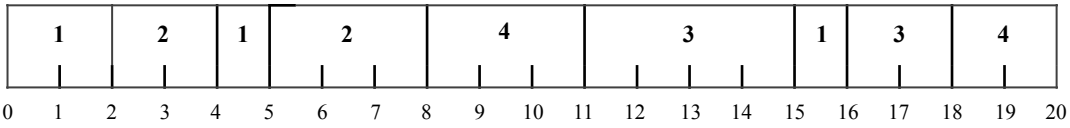
A. temporarily removes processes from main memory and places them in secondary storage and vice versa.

B. decides whether a new process should be brought into the ready queue in main memory or delayed.

C. performs scheduler dispatch.

D. handles the scheduling of all processes with medium priority.

Problem 3.4: A tuple (PID, A) denotes a process with process identity PID that arrives to the ready queue at time A. In a system the ready queue holds the following processes: (1, 0), (2, 1), (3, 4) and (4, 7). An unknown scheduling algorithm results in the following Gantt chart.



From the above Gantt chart, calculate the average waiting time and the average response time. [2 pt]

Tip: Remember that 1/2 = 0.5, 1/4 = 0.25, 3/4 = 0.75.

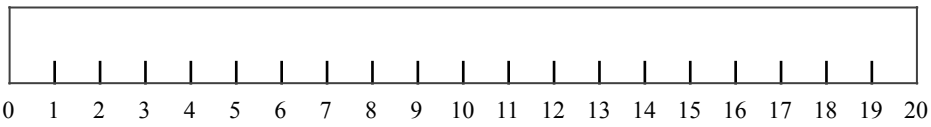
Average response time: _____ (1 pt)

Average waiting time: _____ (1 pt)

Problem 3.5: A triple (PID, A, B) denotes a process with process identity PID that arrives to the ready queue at time A with CPU burst time B. In a system the ready queue holds the following processes: (1, 0, 6), (2, 2, 2), (3, 4, 3), (4, 6, 7) and (5, 15, 2). Draw Gantt charts for PSJF and RR with q=3.

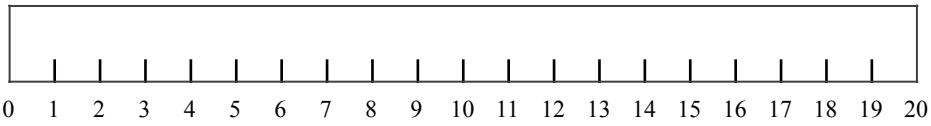
[4 pt]

PSJF



(2 pt)

RR, q = 3



(2 pt)

Module 4

Threads, synchronisation and deadlock

Problem 4.1: Threads within a process: [1 pt]

A. do not share heap.

B. share heap and stack.

C. share both text segment and heap.

D. share text segment but do not share heap.

Problem 4.2: During a code review in a project using a Java-like language you have found the transfer method.

```
transfer(amount, from, to) {
    if (from.balance < amount) return NOPE;
    to.balance   += amount;
    from.balance -= amount;
    return YEP;
}
```

The transfer method has: [1 pt]

A. both data races and race conditions.

B. no data races but race conditions.

C. no data races and no race conditions.

D. data races but no race conditions

Problem 4.3: A mutex lock: [1 pt]

A. can always replace a counting semaphore.

B. can always replace a binary semaphore.

C. is similar to a binary semaphore initialised to 0.

D. is similar to a binary semaphore initialised to 1.

Problem 4.4: Imposing a total ordering of all resource types, and require that each process requests resources in an increasing order of enumeration: [1 pt]

A. does not prevent nor avoids deadlocks.

B. does not allow hold and wait.

C. is an example of deadlock avoidance.

D. is an example of deadlock prevention.

Problem 4.5: During initialisation, a shared boolean variable named lock is set to FALSE. After initialisation, each task in the system will execute according to the below pseudo code. Each task has a local boolean variable named key. You must complete the pseudo code in order to guarantee mutual exclusion to the critical section using the atomic SWAP operation. Fill in the missing code at the labels a, b, c and d.

[2 pt]

```
do {
 // Entry section
 key = (a);
   (b) ((c) == TRUE) SWAP(&lock, &key);
   // Critical section
   (d) = FALSE;
   // Remainder section
} while (TRUE);
```

(a) _____

(b) _____

(c) _____

(d) _____

Problem 4.6: The state of a system is defined by the allocation and max matrices together with the available vector below shown below. Fill in the need matrix (0.5 pt). Show each step of the Banker's algorithm using the available matrix and the done and choice vectors (2 pt). [3 pt]

	Allocation				Max				Need				
Task	**A**	**B**	**C**	**D**	**A**	**B**	**C**	**D**	**A**	**B**	**C**	**D**	**Done**
T_0	2	0	1	2	4	1	1	6					
T_1	0	1	2	0	4	3	4	0					
T_2	5	1	1	0	5	2	2	1					
T_3	0	2	3	1	2	3	6	2					

	Available				
Step	**A**	**B**	**C**	**D**	**Choice**
1	1	1	1	2	
2					
3					
4					
5					-

According to the result of Banker's algorithm from above, is the state safe (0.5 pt)? You will only be rewarded for a correct answer if you have shown all the steps of the Banker's algorithm above correctly.

A. Yes.

B. No

Module 5

Memory management, files and file systems

Problem 5.1: The process memory image is divided into four segments, name these segments. [2 pt]

A. _____

B. _____

C. _____

D. _____

Problem 5.2: A MMU using contiguous allocation: [1 pt]

A. solves the problem with external fragmentation and protects processes from each other.

B. solves the problem with external fragmentation and does not protect processes from each other.

C. does not solve the problem with external fragmentation and protects processes from each other.

D. does not solve the problem with external fragmentation and does not protect processes from each other.

IEC binary prefixes: 1 KiB = 2^{10} Byte, 1 MiB = 2^{20} Byte and 1 GiB = 2^{30} Byte.

Problem 5.3: In a paged system with equally sized logical and physical address spaces of 16 bits and 8K byte pages, a process in the system gets the following page table.

page	0	1	2	3	4	5	6	7
frame	6	0	7	1	2	3	5	4

Compute the physical addresses for the following logical addresses 0xA619 and give the answer in hexadecimal.

The logical address 0xA619 is translated to the physical address _____ [2 pt]

Problem 5.4: The Unix inode: [1 pt]

A. is a variation on linked allocation.
B. uses a combination of direct index blocks and indirect index blocks.
C. is a variation of FAT.
D. uses a combination of contiguous allocation and index blocks.

Problem 5.5: Linked file block allocation: [1 pt]

A. suffers from fragmentation and allows random access.
B. does not suffer from fragmentation and allows random access.
C. does not suffer from fragmentation and does not allow random access.
D. suffers from fragmentation and does not allow random access.