Operating systems I

(1DT044)

Operating systems and process-oriented programming

(1DT096)

Written retake exam

Thursday 2018-06-07

Bergsbrunnagatan 15, sal 2

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.

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.

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.

Problem: Today is: [1 pt]

A. Monday
B. Tuesday
[✓ C.] Wednesday
D. Thursday
E. Friday

Problem: Today is: [1 pt]

A. Monday
B. Tuesday
~~C.~~ Wednesday
D. Thursday
[✓ E.] Friday

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 the concept number in the concept column to the left of the statement 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. Each correct pairing will result in 0.5 points. If you use the same concept number more than once in the statement column this will result in zero points for that concept.

[5 pt]

	Concept
1	SJF
2	Many-to-one
3	Paging
4	Signal
5	Context switch
6	TLB
7	Pipe
8	Throughput
9	Critical section
10	External fragmentation

Concept	Statement
	A simplex FIFO communication channel that may be used for one-way interprocess communication (IPC).
	A notification sent to a process in order to notify it of an event that occurred.
	Requires a priori information.
	Sits between the main memory and the CPU registers.
	A wrapper around the command interpreter that adds useful features that makes it easer to enter commands.
	Entire process will block if a thread makes a blocking system call.
	Controls the hardware and coordinates its use among the various application programs for the various user.
	Requires mutual exclusion.
	A concurrent programming algorithm for mutual exclusion.
	Solves the problem with external fragmentation.
	Amount of time it takes from when a request was submitted until the first response is produced.
	A variation on linked allocation.
	Improves virtual address translation speed.
	Enables multiple processes to share a single CPU and is an essential feature of a multitasking operating system.
	Suspends the execution of the parent process while the child executes.
	A non-preemptive scheduling algorithm.
	Number of processes that complete their execution per time unit.
	Behaviour of an electronic, software or other system where the output is dependent on the sequence or timing of other uncontrollable events.
	Total memory space exists to satisfy a request, but it is not contiguous.
	Assigns a fixed time unit per process, and cycles through them.

Module 1

Fundamental concepts

Problem 1.1: Dual mode operation: [1 pt]

A. makes it possible to seemingly execute multiple processes at the same time.

B. makes it possible for the CPU to execute two threads or processes in parallell at the same time.

C. place restrictions on the type and scope of operations that can be executed by the CPU

D. is a energy saving technique implemented by all modern CPUs.

Problem 1.2: The CPU context: [1 pt]

A. defines whether the CPU is executing in user mode or kernel mode.

B. is saved when transition from user space to kernel space.

C. is used to enforce security and protection.

D. includes the stack and the heap.

Problem 1.3: 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 1.4: Multiprogramming: [1 pt]

A. allows for a single job to get stuck in an infinite loop and block all other jobs from executing.

B. is an extension of multitasking for systems with more than one CPU or CPU cores.

C. gives each job and equal share of CPU time.

D. uses exceptions to notify the system of important events such as the completion of an I/O request.

Problem 1.5: A program: [1 pt]

A. contains a stack pointer.

B. is an active entity.

C. resides in secondary storage.

D. contains a program counter.

Problem 1.6: System calls. [1 pt]

A. An exception is used to initiate a system call.

B. Prior to handling a system call the caller places the return address on the stack.

C. System calls are implemented using a special function call from user space to kernel space.

D. An interrupt is used to initiate a system call.

Module 2

The process concept and inter process communication

File descriptors, standard streams, and I/O redirection

Problem 2.1: The process control block (PCB): [1 pt]

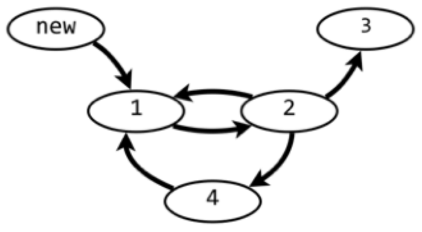
A. serves as the repository for any information that may vary from process to process.

B. is a data structure in user space containing the information needed to manage a particular 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 2:2: A process can transition between various states as depicted in the below diagram. When waiting for I/O, a process transitions: [1 pt]



A. from state 2 to state 1.

B. from state 2 to state 3.

C. from state 2 to state 4.

D. from state 4 to state 1.

Problem 2.3: In Unix-like operating systems: [1 pt]

A. the parent process can use exit() to terminate a child process.

B. fork() creates a copy of the calling process.

C. exec() is used to create a new process.

D. a child process can use wait() to wait for the parent process to terminate.

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

A. returns 0 (zero).

B. returns the pid.

C. returns twice.

D. never returns.

Problem 2.5: Zombie processes. [1 pt]

A. A child process that terminates always first becomes a zombie before being removed from the process table.

B. A zombie process don't have an entry in the process table.

C. A child process becomes a zombie when the parent process terminates before the child.

D. A child process must be an orphan process to become a zombie process.

Problem 2.6: A file descriptor: [1 pt]

A. is a pointer.

B. can only have the values 0 (stdin), 1 (stdout) or 2 (stderr).

C. is not part of the POSIX application programming interface.

D. is a non-negative integer.

Problem 2.7: Signals: [1 pt]

A. can be synchronous or asynchronous.
B. are always synchronous.
C. are always asynchronous.
D. are used to send the exit status from a child back to the parent.

Problem 2.8: A parent process creates a pipe and then calls fork(). Now: [1 pt]

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

Problem 2.9: The dup2() system call: [1 pt]

A. copies the file discriptor table from one process to another process.
B. copies one file descriptor entry to another entry in the file descriptor table.
C. creates a copy of a file descriptor, using the lowest-numbered unused descriptor for the new descriptor.
D. redirects file discriptor 2 (stdout) to a file.

Module 3

CPU Scheduling

Problem 3.1: Classification of processes. [1 pt]

A. In order to offer good response time to interactive processes, Linux (like all Unix kernels) implicitly favours I/O-bound processes over CPU-bound ones.
B. All batch processes are CPU-bound.
C. All I/O-bound processes are interactive.
D. A CPU bound process is characterised by many short CPU bursts.

Problem 3.2: Multiprogramming: [1 pt]

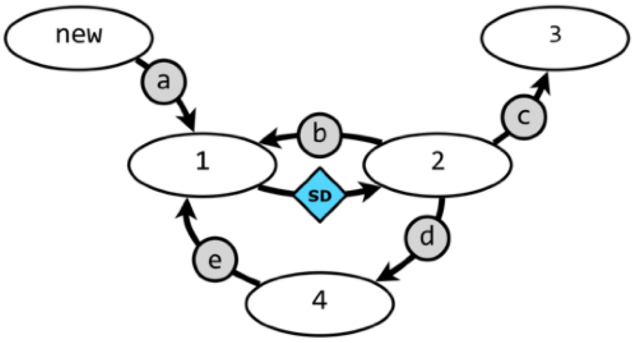
A. maximises the average response time.

B. minimises the average response time.

C. maximises CPU utilisation.

D. minimises CPU utilisation.

Problem 3.3: 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]



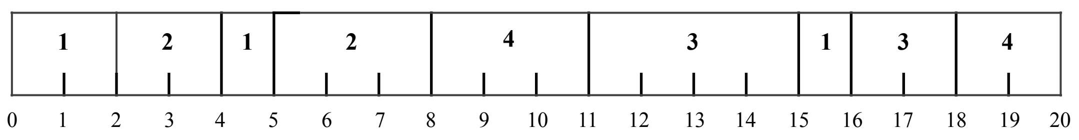
A. All transitions.

B. Transitions a, b and e.

C. Transitions c and d.

D. Only transition b.

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.

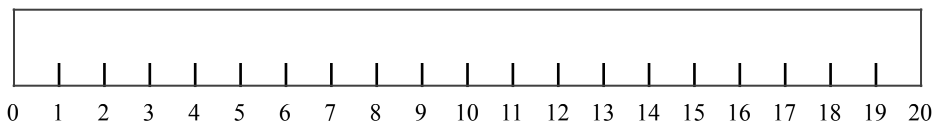
Average response time: ____ (1 pt)

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.

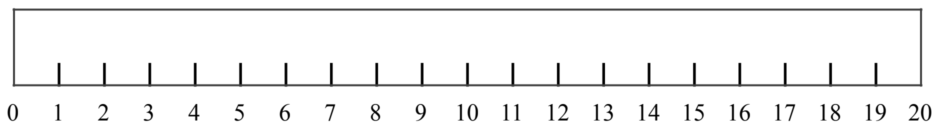
[4 pt]

PSJF



(2 pt)

RR, q = 3



(2 pt)

Module 4

Threads, synchronisation and deadlock

Problem 4.1: 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 have: [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 4.2: Name the four necessary conditions for deadlock. [2 pt]

	Condition	
1		(0.5 pt)
2		(0.5 pt)
3		(0.5 pt)
4		(0.5 pt)

Problem 4.3: Imposing a total ordering of all resource types, and require that each process requests resources in an increasing order of enumeration: [1 pt]

A. is an example of deadlock prevention

B. is an example of deadlock avoidance

C. does not prevent nor avoids deadlocks.

D. does not allow hold and wait.

Problem 4.4: 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 = ⓐ ;
   ⓑ (ⓒ == TRUE) SWAP(&lock, &key);
   // Critical section
   ⓓ = FALSE;
   // Remainder section
} while (TRUE);
```

ⓐ ______________________

ⓑ ______________________

ⓒ ______________________

ⓓ ______________________

Problem 4.5: 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: In the below table, two statements about a MMU using contagious allocation is shown, each which is either true or false. Which combination of the two statements is correct? [1 pt]

	Solves the problem with external fragmentation ...	
... and protects processes from each other	**TRUE**	**FALSE**
TRUE	A	B
FALSE	C	D

A. Combination A

B. Combination B

C. Combination C

D. Combination D

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. suffer from fragmentation and allow random access.

B. do not suffer from fragmentation and allow random access.

C. do not suffer from fragmentation and do not allow random access.

D. suffer from fragmentation and do not allow random access.