Operating systems I

(1DT044)

Operating systems and process-oriented programming

(1DT096)

Final written exam

Friday 2018-03-16, Fyrishov, 14:00 - 19:00

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

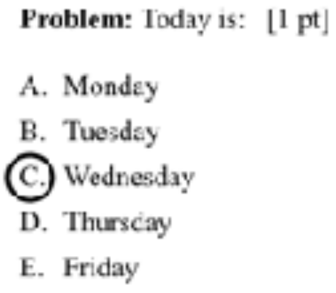
Write your anonymous exam code at the indicated line at the bottom of each page.

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.



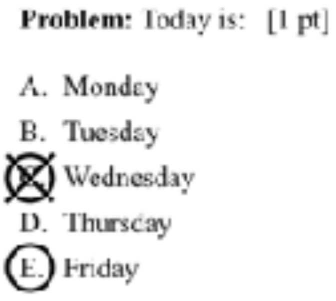
Problem: Today is: [1 pt]

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

A correct answer will result in 1 point. An incorrect answer will result in zero points. Not making a choice will result in zero points. Making more than one choice will result in 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

Written answers

For other types of 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 zero points. Unreadable answers will result in 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

To pass the exam (grade 3) you must score at least 60 % of the total score. For grade 4 you must score at least 75 %. For grade 5 you must score at least 90 % of the total score.

Mixed concepts

Pair each concept with the related statement by filling in the statement label column (L_S) with the related concept label (A, B, C, ..., T) from column L_C. Note that there are 24 statements and 20 concepts. This means that not all of the 24 statements are related to one of the 20 concepts. Each correct answer will result in 0.5 points. If you use a concept label more than once in the statement label column (L_S) this will result in zero points for that concept.

[10 pt]

L_C	Concept
A	System call
B	Critical section
C	Paging
D	TLB
E	Context switch
F	Round Robin
G	SJF
H	Many-to-one
I	fork
J	exec
K	wait
L	Operating systems
M	Message-passing
N	Pipe
O	Signal
P	Race condition
Q	Peterson's solution
R	Throughput
S	Response time
T	External fragmentation

L_S	Statement
	Suspends the execution of the parent process while the child executes.
	A notification sent to a process in order to notify it of an event that occurred.
	A non-preemptive scheduling algorithm.
	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.
	Requesting service from the kernel of the operating system.
	A process creates a copy of itself.
	Total memory space exists to satisfy a request, but it is not contiguous.
	Improves virtual address translation speed.
	Enables multiple processes to share a single CPU and is an essential feature of a multitasking operating system.
	A simplex FIFO communication channel that may be used for one-way interprocess communication (IPC).
	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.
	Requires a priori information.
	Assigns a fixed time unit per process, and cycles through them.
	Runs an executable file in the context of an already existing process.
	Useful for exchanging smaller amounts of data, because no conflicts need be avoided.
	A variation on linked allocation.

Module 1

Fundamental concepts

Problem 1.1: Dual mode operation: [1 pt]

A. makes it possible to execute two processes at the same time.
B. is used to enforce security and protection.
C. is a energy saving technique implemented by all modern CPUs.
D. makes it possible for a system call to return two results at the same time.

Problem 1.2: Exception and interrupts. [1 pt]

A. Exceptions are produced by the CPU control unit while executing instructions and are considered to be asynchronous because the control unit issues them only after terminating the execution of an instruction.
B. Interrupts are external and synchronous.
C. When initiating a system call, a special interrupt is used.
D. Exceptions and interrupts are events that alters the normal sequence of instructions executed by a processor.

Problem 1.3: 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 1.4: An executable: [1 pt]

A. is a passive entity and contains a stack.
B. is an active entity.
C. is a passive entity.
D. is an active entity and contains a page table.

Problem 1.5: System calls. [1 pt]

A. System calls are handled in user mode.
B. Prior to handling a system call the caller places the return address on the stack.
C. Prior to handling a system call the context of the caller must be saved.
D. A timer interrupt is used to initiate a system call.

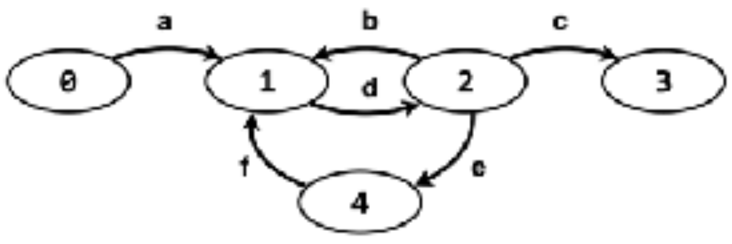
Module 2

The process concept and inter process communication
File descriptors, standard streams, and I/O redirection

Problem 2.1: Individual processes share: [1 pt]

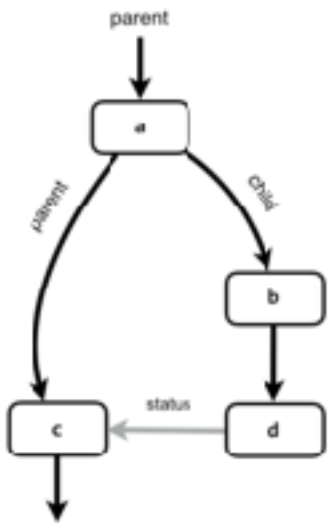
A. signal handlers with each other.
B. heap with each other.
C. heap and stack with each other.
D. neither heap nor stack with each other.
E. only the heap but not the stack with each other.
F. CPU context with each other.

Problem 2:2: In the below figure name the process states 0, 1, 2, 3, and 4. [2 pt]



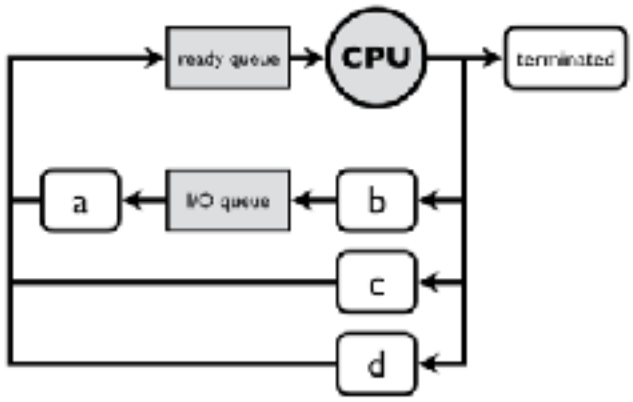
State	Name	
1		(0.5 pt)
2		(0.5 pt)
3		(0.5 pt)
4		(0.5 pt)

Problem 2.3: Name the four system calls a, b, c and d in the figure below. [2 pt]



	System call	
a		(0.5 pt)
b		(0.5 pt)
c		(0.5 pt)
d		(0.5 pt)

Problem 2.4: Name the events a, b, c, and d in the below diagram. [2 pt]



	Event	
a		(0.5 pt)
b		(0.5 pt)
c		(0.5 pt)
d		(0.5 pt)

Problem 2.5: Zombie processes. [1 pt]

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

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

C. A zombie process don’t have an entry in the process table.

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

Problem 2.6: For two child process to be able to communicate using a pipe: [1 pt]

A. The parent must create the pipe before creating the children.

B. The first child must create the pipe.

C. The last child must create the pipe.

D. The parent must create the pipe after creating the children.

Module 3

CPU Scheduling

Problem 3.1: Classification of processes. [1 pt]

A. All batch processes are CPU-bound.

B. All I/O-bound processes are interactive.

C. 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.

D. A CPU bound process is characterised by many short CPU bursts.

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

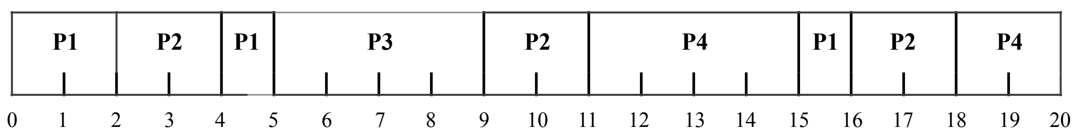
A. Rund robin (RR).

B. Preemptive shortest job first (PSJF).

C. Shortest job first (SJF).

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

Problem 3.3: 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: (P1, 0), (P2, 1), (P3, 4) and (P4, 8). 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.

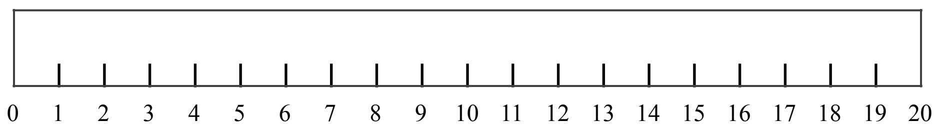
Average response time: ____ (1 pt)

Average waiting time: ____ (1 pt)

Problem 3.4: 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: (P1, 0, 8), (P2, 2, 4), (P3, 5, 3) and (P4, 7, 5). Draw Gantt charts for PSJF and RR with q=3.

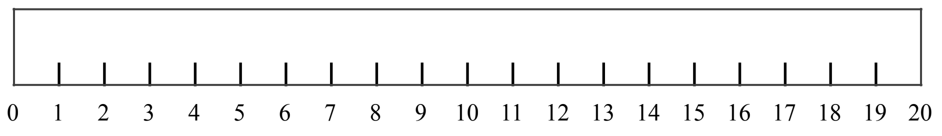
[4 pt]

PSJF



(2 pt)

RR, q = 3

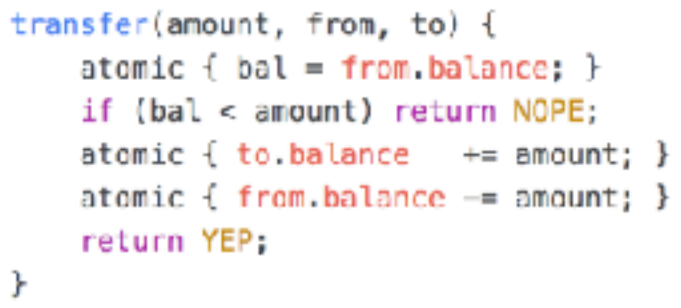


(2 pt)

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) {
    atomic { bal = from.balance; }
    if (bal < amount) return NOPE;
    atomic { to.balance   += amount; }
    atomic { 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: Deadlock prevention: [1 pt]

A. is a dynamic method.

B. requires additional apriori information

C. allows for more concurrency compared to deadlock avoidance.

D. allows for less concurrency compared to deadlock avoidance.

Problem 4.4: In the below table four different implementations (A, B, C and D) of a threaded system are shown. In the INIT section a shared boolean variable lock is initialized. Next code for each of the threads in the system is shown. TAS is the atomic test-and-set instruction.

A	B	C	D
INIT	**INIT**	**INIT**	**INIT**
`lock = TRUE;`	`lock = FALSE;`	`lock = FALSE;`	`lock = TRUE;`
Code for each thread	**Code for each thread**	**Code for each thread**	**Code for each thread**

`if (TAS(&lock)) {`
`   // Critical section`
`   lock = FALSE;`
`}`
	`if (TAS(&lock)) {`
	`   // Critical section`
	`   lock = FALSE;`
	`}`
		`while (TAS(&lock));`
		`// Critical section`
		`lock = FALSE;`
			`while (TAS(&lock));`
			`// Critical section`
			`lock = FALSE;`

Which of the above implementations solves the critical section problem without starvation? [1 pt]

A. Implementation A.

B. Implementation B.

C. Implementation C.

D. Implementation D.

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. [1 pt]

Determine whether the state is safe by showing each step of the Banker's algorithm using the available matrix and choice vector.

[1 pt]

	Allocation				Max				Need				
Task	**A**	**B**	**C**	**D**	**A**	**B**	**C**	**D**	**A**	**B**	**C**	**D**	**Done**
T_0	4	1	0	0	6	5	6	0					
T_1	2	3	6	0	2	5	6	0					
T_2	4	5	3	1	6	5	3	2					
T_3	0	0	0	1	0	5	7	1					
T_4	2	1	0	0	2	1	0	0					

	Available				
Step	**A**	**B**	**C**	**D**	**Choice**
1	0	2	5	1	
2					
3					
4					
5					
-					-

Module 5

Memory management, files and file systems

Problem 5.1: How does paging relate to fragmentation? [1 pt]

A. To minimize external fragmentation the page size must be smaller than the frame size.

B. To minimize internal fragmentation the frame size must be be smaller than the frame size.

C. To prevent external fragmentation the page size must be equal to the frame size.

D. To prevent internal fragmentation the page size must be equal to the frame size.

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

Problem 5.2: While reverse engineering a virtual memory system you have discovered that 8 digit hexadecimal virtual addresses are translated to 7 digit hexadecimal addresses. Further, for all address translations, the three least significant hexadecimal digits are allows identical before and after address translation.

For all of the questions below you must answer using one of the IEC binary prefixes. [3 pt]

Question	Answer	
How large is the virtual address space?		(1 pt)
How large is the physical address space?		(1 pt)
How large is the page/frame size?		(1 pt)

Problem 5.3: The FAT file system is a variation of: [1 pt]

A. linked allocation.

B. indexed allocation.

C. contiguous allocation

D. the Unix iNode.

Problem 5.4: Indexed 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.