Information page for written examinations at Linköping University

Examination date	2019-06-11
Room (1)	TER1(51)
Time	14-18
Edu. code	TDDB68
Module	TEN1
Edu. code name **Module name**	Concurrent Programming and Operating Systems (Processprogrammering och operativsystem) Examination (Tentamen)
Department	IDA
Number of questions in the examination	6
Teacher responsible/contact person during the exam time	Mikael Asplund
Contact number during the exam time	0700 895 827
Visit to the examination room approximately	15-16
Name and contact details to the course administrator (name + phone nr + mail)	Veronica Kindeland Gunnarsson 013-28 56 34 adm-gu@ida.liu.se
Equipment permitted	Engelsk ordbok / english dictionary, Miniräknare / pocket calculator
Other important information	
Number of exams in the bag	

TENTAMEN / *EXAM*

TDDB68

Processprogrammering och operativsystem / *Concurrent programming and operating systems*

2019-06-11

Examiner: Mikael Asplund (0700895827)

Hjälpmedel / *Admitted material:*

– Engelsk ordbok / *Dictionary from English to your native language*;
– Miniräknare / *Pocket calculator*

General instructions

- This exam has 6 assignments and 7 pages, including this one.
  Read all assignments carefully and completely before you begin.
- Please **use a new sheet of paper for each assignment**, because they will be corrected by different persons.
  Sort the pages by assignments and number them consecutively.
- You may answer in either English or Swedish. English is preferred because not all correcting assistants understand Swedish.
- Write clearly. Unreadable text will be ignored.
- Be precise in your statements. Unprecise formulations may lead to a reduction of points.
- **Motivate clearly** all statements and reasoning.
- Explain calculations and solution procedures.
- The assignments are *not* ordered according to difficulty.
- The exam is designed for 40 points. You may thus plan about 5 minutes per point.
- Preliminary grading thresholds:
  - 3: at least 20p
  - 4: at least 27p
  - 5: at least 34p

Good luck!

1. **Multiple choice quesions** (10p)

Below are 10 multiple choice questions. Please answer them by removing the last page of the exam, fill the appropriate boxes, and hand it in together with the rest of your answers. Please note that there might be more than one correct option. Each question below can give 0 or 1 point(s), and to get 1 point you must have identified exactly the right set of choices.

(a) Consider the program below, called *fork-test.* (1p)

```
#include <stdio.h>
#include <unistd.h>

int main(void) {
  int pid1 = 0;
  pid1 = fork();
  int pid2 = 0;
  pid2 = fork();
  printf("PID1: %d, PID2: %d\n", pid1, pid2);
  return 0;
}
```

Which of the lines below could have been produced by the *first* process that was started when running fork-test?

A) PID1: 0, PID2: 0

B) PID1: 0, PID2: 28252

C) PID1: 28250, PID2: 0

D) PID1: 28250, PID2: 28251

(b) Which of the following statements are true regarding interrupts and operating systems? (1p)

A) Interrupts happen only when a user program invokes a system call.

B) When an interrupt is triggered the processor changes its mode so that the code in the interrupt service routine runs in kernel mode.

C) One method to pass parameters to a system call is to store them in main memory and put a pointer to that memory block in a specific register.

D) "printf" is an example of a system call provided in unix-style operating systems such as Linux.

(c) Which of the following requirements are included in the Critical section problem as defined by the course book. (1p)

A) Mutual Exclusion

B) Hold and Wait

C) First come, first served

D) Progress

(d) Which of the following statements are true about deadlocks: (1p)

A) If there is only a single instance of every resource, a cycle in the resource allocation graph means that there is a deadlock.

B) All four Coffman conditions must me met for there to be a deadlock.

C) Banker's algorithm is used to detect and remove deadlocks.

D) Banker's algorithm guarantees freedom from starvation.

(e) Which of the following statements (relating to memory management) are true? (1p)

A) TLB is short for Transitive Local Buffer.

B) A page fault is a critical failure that the OS cannot recover from.

C) Paging and segmentation can be combined.

D) Belady's page replacement algorithm requires knowing the future memory access pattern.

(f) Which of these parameters will affect the effective access time (EAT) of demand paging? (1p)

A) Page fault rate (probability of page fault)

B) Segment table length

C) Time to swap page in from disk

D) Disk size

(g) Which of the following properties are true for semaphores? (1p)

A) The wait and signal operations are atomic.

B) They guarantee freedom from deadlock.

C) They can be used to solve the critical section problem.

D) They can only be implemented on a system with special hardware.

(h) Which of the following statements are true? (1p)

A) Hard links in a file system work well across different partitions.

B) Memory-mapped files remove the need to synchronize file system operations.

C) Contiguous allocation for secondary storage works well for read-only media.

D) Journalling file systems allows interrupted disk operations to be rolled-back, thereby avoiding inconsistencies on disk even in case of sudden power failure.

(i) Which of the following are possible states for a system process (as described in the course literature): (1p)

A) Waiting

B) Ready

C) Forking

D) Controlling

(j) Which of the following are algorithms that can be used for process scheduling? (1p)

A) Shortest-job-first

B) Round robin

C) Priority inversion

D) First-come, first-served

2. **Synchronization** (8p)

Consider a shared stack of positive integers, to be implemented as a shared array of sufficiently large size and a shared stack pointer. The stack operations `push` and `pop` can be called concurrently by multiple threads. When the stack is found to be empty, operation `pop` should return an error code (-1).

(a) Write (unprotected) code for the stack initialization and for the operations `push` and `pop`. (Use C, Java or pseudocode.) (2p)

(b) Describe a contrived scenario with two threads where the concurrent execution of these unprotected operations leads to an unexpected result (i.e., a race condition). Identify precisely the variables and statements that could potentially be involved in any race condition (that is, describe the critical section(s)). (2p)

(c) You are given a single-processor system where multi-tasking is implemented by a preemptive scheduling algorithm, and where an atomic test-and-set operation is available. Here, there exist actually two different hardware-supported mechanisms to protect critical sections. Name and explain these two variants, and show for each case how your code of part (a) is to be modified to protect its critical section(s) against race conditions. (2p)

(d) Write a *monitor* solution for the shared stack (based on your code above). Use C or pseudocode notation with appropriate keywords to identify the monitor components, and explain your code. (2p) (2p)

3. **Deadlocks** (5p)

(a) Consider the Dining Philosophers problem as discussed in the lectures, where each philosopher first tries to obtain the chopstick to her left. Describe how each of the four conditions applies in this scenario, thereby showing that a deadlock could occur. (1p)

(b) Consider the following resource allocation problem in a system with 3 resources (R1-R3), and 4 processes (P1-P4). The table indicates the currently allocated resources and in parenthesis the maximum possible demand. (4p)

	R1	R2	R3
P1	0 (9)	1 (3)	1 (1)
P2	3 (3)	0 (2)	2 (2)
P3	1 (1)	1 (4)	0 (1)
P4	0 (6)	1 (5)	0 (3)

The currently available resources are: [5, 3, 0]. Use Banker's algorithm to determine if the request [0, 1, 0] from Process P2 should be granted.

4. **Processes and scheduling** (7p)

(a) Define the terms *process*, *kernel-level thread* and *user-level thread*, and explain the differences between them. (3p)

(b) Two main methods for inter-process communication in a computer are *shared memory* and *message passing*. (4p)

For each of them, give a short explanation of how it works and how the operating system is involved, i.e., which important system calls are to be used and what they do.

Which of the two methods is likely to have less overhead if two processes communicate frequently with each other, and why?

5. **Memory management** (5p)

(a) Paging for large virtual address spaces and small page sizes leads to the large-page-table problem. Explain *one* technique how this problem can be solved. Be thorough! (3p)

(b) How does a simple segmented memory management system work, and what hardware support should be provided by the memory management unit (MMU)? (2p)

6. **Security** (5p)

(a) Describe the difference between a vulnerability and an attack. Explain using a realistic example. (2p)

(b) Explain the role of authentication in the context of operating systems. Moreover, explain consequences and requirements relating to authentication when sharing data over multiple computers using a distributed file system. (3p)

Intentionally empty page.

Multiple choice form for answering question 1. Please put X:s in the appropriate cells:

	A	B	C	D
1 a)				
1 b)				
1 c)				
1 d)				
1 e)				
1 f)				
1 g)				
1 h)				
1 i)				
1 j)				

Optional: if you feel you need to clarify your interpretation of the question you can do so here. This is not needed if your answer is correct.

1 a)	
1 b)	
1 c)	
1 d)	
1 e)	
1 f)	
1 g)	
1 h)	
1 i)	
1 j)	