Information page for written examinations at Linköping University

Examination date	2019-08-21
Room (1)	TER1(9)
Time	8-12
Edu. code	TDDE35
Module	TEN2
Edu. code name **Module name**	Large-Scale Distributed Systems and Networks (Storskaliga distribuerade system och nätverk) Written examination (Skriftlig tentamen)
Department	IDA
Number of questions in the examination	12
Teacher responsible/contact person during the exam time	Niklas Carlsson
Contact number during the exam time	013-282644
Visit to the examination room approximately	ca. 10:00
Name and contact details to the course administrator (name + phone nr + mail)	Veronica Kindeland Gunnarsson veronica.kindeland.gunnarsson@liu.se, 013-285634
Equipment permitted	None (well, please cover page ...).
Other important information	
Number of exams in the bag	

TDDE35/TEN2 – Large-scale distributed systems and networks
Final Examination: 8:00-12:00, Wednesday, Aug. 21, 2019
Time: 240 minutes
Total Marks: 40
Grade Requirements: Three (20/40); four (28/40); and five (36/40).
Assistance: None (closed book, closed notes, and no electronics)
Instructor: Niklas Carlsson

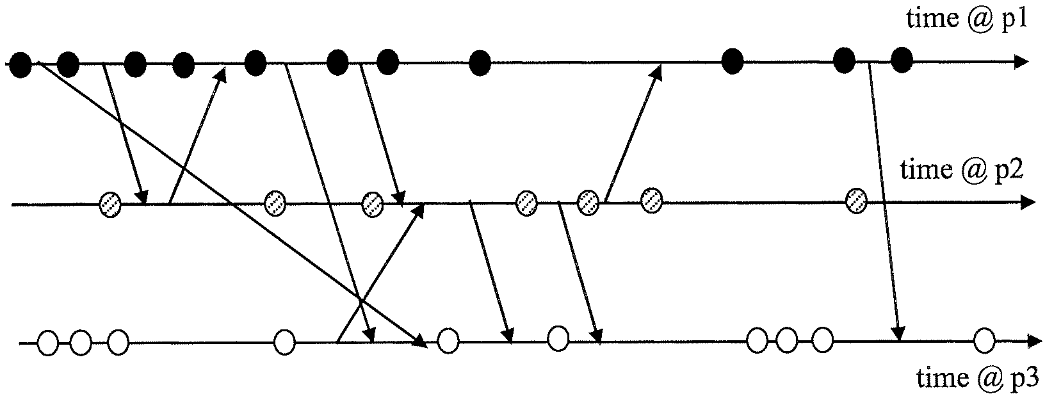
Instructions:

- Read all instructions carefully (including these)!!!! Some questions have multiple tasks/parts. Please make sure to address *all* of these.
- The total possible marks granted for each question are given in parentheses. The entire test will be graded out of 40. This gives you 10 marks per hour, or six minutes per mark, plan your time accordingly.
- This examination consists of a total of 12 questions. Check to ensure that this exam is complete.
- When applicable, please explain how you derived your answers. Your final answers should be clearly stated.
- Write answers legibly; no marks will be given for answers that cannot be read easily.
- Where a discourse or discussion is called for, be concise and precise.
- If necessary, state any assumptions you made in answering a question. However, remember to read the instructions for each question carefully and answer the questions as precisely as possible. Solving the *wrong* question may result in deductions! It is better to solve the *right* question incorrectly, than the *wrong* question correctly.
- Please write your AID number, exam code, page numbers (even if the questions indicate numbers as well), etc. at the top/header of each page. (This ensures that marks always can be accredited to the correct individual, while ensuring that the exam is anonymous.)
- Please answer in English and utilize figures and tables to the largest extent.
- If needed, feel free to bring a dictionary from an official publisher. Hardcopy, not electronic!! Also, your dictionary is not allowed to contain any notes; only the printed text by the publisher.
- Good luck with the exam.

Part A: Distributed Systems

Question 1 (4 points)

Assume that you have three processes p1, p2, and p3 which are implementing Lamport's clocks. There are many events that take place at these processes, including some messages being sent between the processes. In the figure below we use circles and arrows to specify in-processor events and messages being sent between processes, respectively. Please provide the logical timestamps associated with each event. You can assume that all three clocks start at zero, at the left-most point in time. (Also, explain how the processes would adjust their clocks if using Lamport's logical clocks.)



Question 2 (3 points)

In class we talk about mutual exclusion. Please use figures to show the main differences between how this can be implemented using a (i) central coordinator and using (ii) a distributed algorithm (algorithm by Ricart/Agrawala in particular, as discussed in class). For each of the algorithms, please use figures to show all messages involved and the order they take place for some basic example scenario.

Question 3 (3 points)

In class we talked about the bully algorithm. First, use an example use case or system to explain when this algorithm is useful in a distributed system. Second, use a step-by-step example to explain how the algorithm works. Your example should include a figure that clearly shows all messages involved and the order they take place.

Part B: Methodology

Question 4 (4 points)

When designing experiments, it is important to carefully identify the most appropriate factors, levels, and metrics to consider. Consider a researcher wanting to assess the performance of a webserver. The researcher has identified three factors of interest: (i) the request rate, (ii) the job size, and (iii) the processor speed. For each of these factors, the researcher has identified 9, 8 and 4 levels of interest, respectively, including identified a default request rate, job size, and processor speed. Let us call the request rate levels R1, R2, ..., R9; the job size levels S1, S2, ..., S8; and the processor speed levels P1, P2, P3, and P4. Please estimate the number of experiments that the researcher would need to perform if performing (a) one factor experiments with the default scenario as baseline, (b) two factor experiments with the default scenario as baseline, and (c) full factor experiments. Also, please explain which experiments would be performed in each case.

Question 5 (3 points)

Power law distributions (and other related "heavy tailed" distributions) are often observed in nature and large distributed systems. How does "heavy tailed" distributions relate to exponential distributions? Power law relationships are often illustrated using log-log plots. Please use equations to explain why this can be an effective illustration of these distributions. In the context of the Internet topology or a social network, please explain the mapping of the "heavy tails" to routers or people. In other words, which Autonomous Systems (ASes) and people would correspond to the heavy tail?

Question 6 (3 points)

In class we discussed how simulations, analytic models, and experimental instrumentation can be used to evaluate the performance of both large and small computer systems. Please explain the advantages and disadvantages of each approach. Also, please provide a concrete system example (e.g., an L1 cache, a webserver, a server cluster, a datacenter, or a wide-area system) to discuss and illustrate these tradeoffs.

Part C: Multicore and Parallel Programming

Question 7 (3 points)

Questions on parallel computer architecture.

(i) Name and shortly describe an interconnection network topology that is scalable (at reasonable cost) to very large supercomputer clusters, even where the processes of parallel applications exchange data frequently. How does its cost (number of switches and links) and maximum node degree scale with the number of cluster nodes? (1p)

(ii) How does "hardware multithreading" differ from "multicore" technically (be thorough), and what do they have in common from the software point of view? (1.5p)

(iii) What is a heterogeneous multicore system? (0.5p)

Question 8 (4 points)

Questions parallel programming models.

(i) We learned about two different parallel program execution styles (i.e., how to create multiple parallel activities like threads or processes): SPMD style and Fork-Join style. Which style is used in MPI, and which one in Pthreads? Shortly justify your answers. (1p)

(ii) What are "collective communication operations" in MPI? Describe one collective communication operation of your choice: its name, what are its input and output operands, what does it do with the operands (be thorough), and describe how (commented MPI pseudocode and illustrating figure) you would implement it if you were only allowed to use `MPI_Send` and `MPI_Recv` operations. (3p)

Question 9 (3 points)

Questions on theory.

(i) A (parallel) computation model consists of a (parallel) programming model and a (parallel) cost model. What does a parallel cost model describe? What are the general requirements on parallel cost models in the early and later stages of the design and analysis of parallel algorithms? (1p)

(ii) We learned that the computation performed by a parallel program can be described by an acyclic task graph, consisting of n (sequential) tasks and their dependences. The tasks will be scheduled to the $p > 1$ processors of a parallel system for execution. What is a critical path in the task graph, and how does the program's parallel execution time relate to the length of the critical path? (1p)

(iii) A parallel program fulfilling the assumptions of Amdahl's Law takes 100 seconds if run on one processor and 60 seconds if run on 2 processors. What is the upper bound of the relative speedup that this program could achieve for any number of processors? Explain your answer (calculation). (1p)

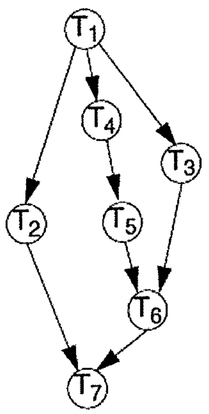
Part D: Embedded Systems

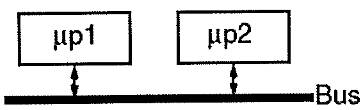
Question 10 (4 points)

Consider an application modelled as the task graph below. Each task, when activated, consumes one message on each input edge and generates, at termination, one message on each output edge. The task graph is executed on the architecture shown in the figure. Execution times of the tasks, when executed on the corresponding processor, are shown in the table. All messages transmitted over the bus, between tasks mapped on different processors, consume 2 time units to reach the destination. Communication between tasks mapped to the same processor is considered to not consume any time.

Propose an efficient task mapping (indicate on which processor each task is executed) and a corresponding static (nonpreemptive) schedule for the application. Illustrate your schedule as a Gantt chart (similar to the way we captured schedules in Lecture 1&2).

Try to achieve a maximum delay (the time interval between the start of T1 and the finishing of T7) of 46 time units!





Task	WCET	
	μp1	μp2
T_1	5	6
T_2	12	15
T_3	10	11
T_4	5	6
T_5	3	4
T_6	17	21
T_7	10	14

Question 11 (3 points)
In the lectures we have particularly emphasized three design steps: architecture selection, task mapping, elaboration of a schedule. Explain, in short, what each step is doing. Illustrate the three steps by a small example.

Question 12 (3 points)
Two questions on task scheduling. (a) What is the basic principle for task scheduling on DVS (variable voltage) processors? (b) What is the problem if we consider particularities, concerning power consumption, of individual tasks?

Good luck!!