



# Försättsblad till skriftlig tentamen vid Linköpings Universitet

(fylls i av ansvarig)

<b>Datum för tentamen</b>	2010-08-25
<b>Sal</b>	TER4
<b>Tid</b>	8-12
<b>Kurskod</b>	TDTS10
<b>Provkod</b>	TEN2
<b>Kursnamn/benämning</b>	Datorarkitektur
<b>Institution</b>	IDA
<b>Antal uppgifter som ingår i tentamen</b>	5
<b>Antal sidor på tentamen (inkl. försättsbladet)</b>	4
<b>Jour/Kursansvarig</b>	Erik Larsson
<b>Telefon under skrivtid</b>	013-286619, 0709-656619
<b>Besöker salen ca kl.</b>	9 och 11
<b>Kursadministratör (namn + tfnr + mailadress)</b>	Madeleine Häger
<b>Tillåtna hjälpmedel</b>	Inga
<b>Övrigt (exempel när resultat kan ses på webben, betygsgränser, visning, övriga salar tentan går i m.m.)</b>	10 arbetsdagar efter tentamen

Exam  
**Computer Architecture**  
TDTS10

August 26, 2010 - 08:00-12:00

**Jour:** Erik Larsson(0709-656619, 013-286619)

**Hjälpmedel/Admitted material:**

- Engelsk ordbok
- Dictionary from English to your native language

**General instructions:**

- This exam has 5 assignments and 2 pages, including this one.
- Read all assignments carefully and completely before you begin.
- Use a new sheet for each assignment.
- You may answer in either English or 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.
- Grading: U, 3, 4, 5. The preliminary threshold for passing is 22 points.
  
- For ECTS, LiU make use of: 5=A, 4=B, 3=C, and UK=Fx.

## 1. Execution (5 points)

- Detail what a CPU does when it is executing a branch instruction?
- How is the CPU computing where to find the data when register indirect addressing is used?

## 2. Processor Design (10 points)

- Detail (compare) RISC and CISC
- For the instruction set design - does it matter and if so how if the processor uses memory-mapped I/O or isolated I/O?
- Explain how to use a stack to enable procedure calls
- Explain how to use a large register file to enable procedure calls
- Of the techniques above, which is the most efficient (in terms of performance/speed)?

## 3. Memory System (10 points)

- Why do computer systems in general have both primary memory and secondary memory?
- Compare characteristics of a primary memory and a secondary memory.
- Why is fragmentation a problem for memory systems?
- For a secondary memory like a hard disc, detail where and how fragmentation may occur.
- Detail (propose) two ways to organize (store) files on a hard disc. Describe your schemes using a file of 4Kbytes where the block size is 512 bytes.
- Given your two schemes from above, compare (discuss) them in (1) time for reading, (2) time for writing and (3) fragmentation.

## 4. Operating system (10 points)

- A process may be in different states (such as running); list and explain the states in which a process can be, and explain how, when and why a process moves between states.
- What is time sharing used for?
- What is a context switch?
- What is a process control block and what is it used for in general and in particular at a context switch?

## 5. I/O (5 points)

- In programmed I/O, the programmer controls the checking of a resource. How does it work in interrupt driven I/O? Hint, think of a program that is executing on the CPU and an interrupt occurs.