Försättsblad till skriftlig tentamen vid Linköpings universitet

	(fylls i av ansvarig)
Datum för tentamen	2009-08-22
Sal	KÅRA
Tid	8-12
Kurskod	TDTS10
Provkod	TEN1
Kursnamn/benämning	Datorarkitektur
Institution	*IDA*
Antal uppgifter som ingår i tentamen	6
Antal sidor på tentamen (inkl. försättsbladet)	3
Jour/Kursansvarig	Erik Larsson
Telefon under skrivtid	0709-656619
Besöker salen ca kl.	10
Kursadministratör (namn + tfnnr + mailadress)	Madeleine Häger Dahlqvist 282360, madha@ida.liu.se
Tillåtna hjälpmedel	
Övrigt (exempel när resultat kan ses på webben, betygsgränser, visning, övriga salar tentan går i m.m.)	10 arbetsdagar efter tentamen
Vilken typ av papper ska användas, rutigt eller linjerat	
Antal exemplar i påsen	

Linköping University
Department of Computer Science
Erik Larsson

August 18, 2009

Exam
Computer Architecture
TDTS10
August 22, 2009 - 08:00-12:00

Jour: Erik Larsson(0709-656619)

Hjälpmedel/Admitted material:

- Engelsk ordbok
- Dictionary from English to your native language

General instructions:

- This exam has 6 assignments and 3 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 30 points. The case study and laboratory work may add on up to 15 points. The maximum is 40 points.
- Grading: U, 3, 4, 5. The preliminary threshold for passing is 22 points (including points from case study).
- For ECTS, LiU make use of: 5=A, 4=B, 3=C, and UK=Fx.

1. Operating system (5 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.

2. Pipelining (5 points)

   - How many cycles would the following sequence of instructions take if they are executed in a 6-stage pipeline (assume one cycle per stage in the pipeline (FI-fetch instruction, DI-decode instruction, CO-calculate operand, FO-fetch operands, EI-execute instruction, WO-write operand) (2p).

```
Instructions
ADD R1, (R2)
ADD R2, R1
```

3. Execution (5 points)

   - What would the program counter, the zero status register, R1 and R2 contain after execution of the program below?

```
Address         Instruction/Data
0               LOAD R2, #10
1               LOAD R1, #0
2               ADD  R1,(R2)
3               ADD R1, R2
4               BR  6
5               MUL R2, R1
6               HLT
7               ADD R1, R2
8               SUB R2, #1
9               HLT
10             4
11             5
12             8
```

```
The instructions are:
    LOAD=load, SUB=subtraction,
    ADD=addition, BR=unconditional branch,
    HLT=halt
    number/data are given in decimal numbers
```

4. Memory system (5 points)

   - For a memory system, explain what memory fragmentation is, what types of fragmentation there can be, and discuss how to address memory fragmentation.
   - Explain the difference between a virtual, logic and physical address
   - Discuss performance when the page table is placed in the main memory
   - Discuss the relation between trashing and demand paging

5. Cache (5 points)

   - Locality of reference is an important feature of programs, in the context of memory hierarchies. Explain what locality of reference means, and why it is important
   - Assume a main memory of size 64 bytes and a cache memory of size 16 bytes. The cache memory is organized as direct mapping and a cache line is of size 4 bytes. Explain what happens (decoding and memory read) when the CPU makes the following memory requests: (a) read at address 000000, (b) read at address 100111, and (3) read at address 111001.

Cache line	**Tag**	**Byte address**			
		00	01	10	11
00	00	A	B	C	D
01	10	E	F	G	H
10	01	I	J	K	L
11	00	M	N	O	P

6. I/O (5 points)

   - Programmed I/O is one alternative to handle I/O operations. How does programmed I/O work? What alternatives exists (discuss and explain)?