Institut for Matematik og Datalogi
Syddansk Universitet

September 4, 2009

Peter Schneider-Kamp

Introduction to Computer Science
E09 – Week 2

Lecture: Tuesday, September 1

Rolf Fagerberg gave an introduction to the course, covering Chapter 0 of the textbook.

Lecture: Wednesday, September 2

Rolf Fagerberg covered Sections 1.1–1.4.

Lecture: Monday, September 7, 12-14 (U20)

Rolf Fagerberg will start and cover Sections 1.5–1.7. Peter Schneider-Kamp will continue with Chapter 2 and cover Sections 2.1–2.4. In particular, Peter will introduce the simple machine model (cf. Appendix C) used in Tuesday's lab session.

Lecture: Thursday, September 10, 10-12 (U20)

Daniel Merkle will lecture on operating systems. He will begin Chapter 3.

Lecture: Monday, September 14, 12-14

Daniel Merkle will lecture on operating systems, i.e., he will continue with Chapter 3.

Lecture: Wednesday, September 16, 14-16

Kim Skak Larsen will lecture. He will give an introduction to databases based on Chapter 9.

Lab: September 8, 14-16 (terminal room above U49)

Make sure you have a login account for the IMADA machines! Discussion will be in groups (only two, or possibly three, people per group, since you will sit at a computer): Two simulators (of varying quality and ease of use) for the Brookshear machine are available from the course homepage:

`http://www.imada.sdu.dk/~petersk/DM526/Simulator.jar`
`http://www.imada.sdu.dk/~petersk/DM526/BrookshearMachine.jar`

You can run the simulators by `java -jar Simulator.jar` and `java -jar BrookshearMachine.jar`.

1. Do problem 1 on page 116 of the textbook. To input the data when using `Simulator`, type `[00] 14 02 34 17 C0 00` in the white field at the top of the window. Click on `Load Data`.

   In the `BrookshearMachine` you can just enter the data directly into the fields. Note that when in `Memory List` view, you need to specify two bytes at a time, i.e, the first memory cell denoted `00` gets `14 02`, the second memory cell denoted `02` gets `34 17` etc.

   Use Appendix C to figure out what should happen. Then, `Single Step` through the execution to see that it does happen.

2. Do problem 2 on page 116. To load the value `B0` into the program counter, for `Simulator` type `[PC] B0` in the `Data Input Window` and click on `Load Data`. For the `BrookshearMachine`, enter `B0B0` at address `00` to jump to `B0`.

   Why does register 3 get the values it does when you step through the program?

3. Do problem 3 on page 117. Note that the operation `B` is usually referred to as a *conditional* branch, and there is usually also an *unconditional* branch instruction, which always causes the program counter to get the specified value (without checking the values of any registers). How is the conditional branch instruction used here to get the effect of an unconditional branch?

4. Do problem 4 on page 117. This is strange in that it is an example of how a program can modify itself when there is no distinction between program and data. Discuss the security implications of this.

Lab: September 10, 14:15-16 (U20)

In this lab you will learn how to typeset documents (e.g. your solutions to the assignments) using LaTeX. It will be based on the following introduction:

`http://imada.sdu.dk/~petersk/DM526/latexintro.pdf`

For a longer introduction to LaTeX, you can use e.g. *The Not So Short Introduction To LaTeX 2e* available at:

`http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf`

Please remember to refresh your Linux command line skills. You are expected to know at least the commands from the introduction available at:

`http://imada.sdu.dk/~kok04/intro/`.

Assignment due 14:15, September 17

Late assignments will not be accepted. Working together is not allowed. You may write this either in English or Danish. Write clearly if you do it by hand. Even better, use LaTeX.

Show how you obtained your answers, and explain what your program does.

You may either do problems 1, 2, 3, or problems 2 and 4 (more challenging).

1. Convert 11001010 from two's complement to its equivalent in base ten.
2. Decode 11001010 from the floating-point representation described in class (and on the first weekly note).
3. Write a program in the machine language from Appendix C which reads values from memory locations `A0` and `A1`, and creates a new value which is the sum of the first four bits of the value in `A0` and the last four bits of the value in `A1`. The result goes to location `A2`. For example if 01010101 is in `A0` and 11001100 is in `A1`, then `A2` should contain 00010001.
4. Write a program in the machine language from Appendix C which reads a value n from memory location `A0`. You may assume $0 \leq n \leq 16$. Add the contents of n memory locations starting with `B0` to n memory locations starting with `C0`. Put the results into n memory locations starting with `D0`. E.g, after the execution, for $n \geq 4$, the value in `D3` should be the sum of the values in `B3` and `C3`.