SYSTEMS PROGRAMMING

(37-023)

Programming in Assembler
Operating System Basics
Computer Architecture

Prof. Gustavo Alonso
Prof. Thomas M. Stricker
Who are we?

Dozenten
• Prof. Gustavo Alonso
• Prof. Thomas M. Stricker

Graduate assistants
• Win Bausch
• Irina Chihaia
• Christian Kurmann
• Cesare Pautasso
• Andrei Popovici
• Felix Rauch

Undergraduate assistants
• Rolf Bruderer
• Beat Fluri
• Patrick Grawehr
• Hampa Hug
• Slavisa Maslic
• Dejan Radovic
• Daniel Wagner
• (tba)
A little quizz ...

• Who has used the Internet?
  ... self taught?

• Who programs regularly?
  ... learned at ETH?
  ... self taught?

Biggest program ... 
  ..1000/10000/100000 LoC?

• Have you built or configured your own computer?
  ... self taught?

• Ever programmed in assembler?
  ... self taught?

• Do you consider yourself a hacker?
General objectives of the course

- Work at the interface between **Hardware and Software**.
  - Basics of computer architecture (instruction set)
  - Basic functionality of an operating system (user/system modes)
- Introduction to the **Art of Systems Programming**.
  - Understand the software hierarchy in a computer
  - Programming tools and techniques
- Introduction to the areas "**System Software**" und "**Computer Architecture**"
  - Operating systems
  - Compilers
  - Libraries (e.g., Communication)
Concrete goals:

- **Introduction to Computer Programming** in C and Assembler
  - Relation between C <-> Assembler

- **RISC Computer Architecture**
  - SPARC V8 Computer Architecture from the point of view of the programmer

- **Introduction to UNIX**
  - Operating System basics
  - Basic concepts of synchronous and asynchronous interrupts
  - Basics of I/O operations

- **General**
  Understand the way computers and complex software systems work and evolve

- **Last but not least** ... **have fun**
Course Program

• Introduction and goals
• System layering
  Operating system
  Compiler, assembler, linker
  Basics of computer architecture
• Programming in C
• SPARC Architecture
• Programming in Assembler
• System internals
  Stacks
  Subroutines
  Traps
  I/O Devices
  Signals
Organization

All you will ever need to know:

http://www.inf.ethz.ch/37-023

Lecture: attendance recommended
(Foils will be available before the lecture on the web page of the lecture)

Exercises:
• ca. 10 Series - 1 or 2 weeks each
• 6 Assistants
• Exercises to be solved in groups of 2-3
• Getting the Testat = see the web pages
• SunSparc in E22, E26.1,E19

Textbook - Manual:


Programming environment:

- SunSPARC Workstation (the real thing)
  - GNU-C-Compiler on UNIX
- Tkisem - SunSPARC V8 - Simulator
  - TKISEM for SUN
  - TKISEM in Linux for WinTel PC (at your own risk).

Exam:

- Written
- Material from Lecture and exercises; programming required (but on paper)

Requirements for the Testat:

- Complete 8 of the 10 exercises
- Exercises 9 are 10 are mandatory
Exercises

- Introduction to UNIX
- Make, compiler
- Programming in C: (1)
- Programming in C: (2)
- Programming in assembler
  - arithmetic expressions
  - control structures
  - copy block
  - Stackframes
  - Traps
  - Sun Serial Port (in C, but low level)
- Programming Competition (how far can you optimize a program or why faster CPUs do not always help)
  - not an exercise
  - volunteers only
  - great prices
Reference Books


