GÖTEBORG UNIVERSITY
The Board of the IT Faculty

DIT681, Unix internals, 7,5 higher education credits

Second Cycle/A1N

This syllabus in English is a binding document.

1. Confirmation

The syllabus was confirmed by the Faculty Board of IT Faculty/The Dean on 2006-11-17 revised on 2008-12-10 and 2009-10-15 to be valid from spring semester, 2010.

Field of education: Sciences.

Responsible department: Computer Science and Engineering.

2. Position in the educational system

The course is a part of the Computer Science Master´s programme and a single subject course at the University of Gothenburg.

3. Entrance qualifications

The requirement for the course is to have successfully completed a first year within Computer Science or equivalent. Specifically students should have completed the course DIT400 Operating systems or equivalent.

4. Course content

The course is intended to give in-depth knowledge on how Unix is designed and implemented. The course will describe the concepts, data structures and algorithms used in implementing FreeBSD´s system facilities. A description of process management lays the foundation for later parts by describing the structure of a process, the scheduling algorithms and the synchronization mechanisms used to ensure consistent access to kernel-resident data structures. The virtual-memory system is discussed in detail. The structure of the Input/Output system is described including the buffer management for the file systems and the interface to the device drivers. Communication within Unix and to and from the computer is described. The implantation of computer communication protocols and operating system bootstrapping is also included.

The course consists of two parts: One part is a lecture series and the other one is a number of lab exercises which explore the facilities in modern operating systems.

5. Learning outcomes

To give in-depth knowledge in how a modern operating system(Unix) is designed regarding its internal algorithms, how internal and external communication is realized an how and dynamic modules are implemented.

After completing the course the student shall be able to describe how central components in the Unix system are designed and implemented. Therefore he/she will be able to give input to the design of advanced computer based services in order to use the computer system as efficient as possible.

After completing the course the student is expected to be able to:

- Describe the implementation of Unix process management and scheduling
- Describe the architecture if a Unix I/O system.
- Describe the implementation of a virtual memory system.
- Describe the implementation of some local and distributed file system.
- Describe the implementation of socket based interprocess communication.
- Describe the implementation of the internet protocols TCP and IP.

6. Required reading

See separate literature list.

7. Assessment

The student is evaluated through Lab work and a final written exam.

8. Grading scale

The grades are Pass, Pass with distinction or Fail.

9. Course evaluation

The course is evaluated through meetings both during and after the course between teachers and student representatives. Further, an anonymous questionnaire can be used to ensure written information. The outcome of the evaluations serves to improve the course by indicating which parts could be added, improved, changed or removed.

10. Additional information

The course is given in English.