



COURSE SYLLABUS

Programmering i UNIX-miljö

Programming in UNIX Environment

7,5 ECTS credit points (7,5 högskolepoäng)

Course code: DV1457

Educational level: First cycle

Course level: G2F

Field of education: Technology

Subject group: Computer Technology

Subject area: Computer Science, Software Engineering

Version: 11

Applies from: 2013-09-02

Approved: 2013-05-21

1 Course title and credit points

The course is titled Programming in UNIX Environment/Programmering i UNIX-miljö and awards 7,5 ECTS credits. One credit point (högskolepoäng) corresponds to one credit point in the European Credit Transfer System (ECTS).

2 Decision and approval

This course is established by School of Computing 2013-05-21. The course syllabus was revised by School of Computing and applies from 2013-09-02. Reg.no: BTH 4.1.1-0274-2013

3 Objectives

The aim of the course is to enable students to acquire specialised understanding of UNIX-based systems and for programming close to the operating system. This involves, among other things, being able to program at the most abstract level of the operating system, close to the user, and down through the levels of abstraction to the lowest level, system calls. The course trains students in designing software that interacts with the computer through the operating system UNIX (and operating systems similar to or based on UNIX such as Linux and MacOS). It provides a foundation for continued studies in other fields (such as computer security) demanding a practical understanding of the technology involved. The objective of the course is achieved by the student through gaining practical experience of developing programs in a UNIX environment in a series of laboratory exercises.

4 Content

The course covers the following specific levels of abstraction in UNIX programming in greater detail:

- **Commands and scripting language**

The scripting language is a way of making what is normally perceived as user interaction automatic, i. e. interactively commanding the computer to load files, sort the contents and print them. Scripting

languages (exemplified with Bourne shell) are thus very powerful but also less general than ordinary programming languages.

- **Program development in C**

C is the mainstay in programming of UNIX applications and major parts of UNIX systems are often written in C. Advanced UNIX programming, such as for the implementation of network services, is often executed in C with the help of system and library calls to the operating system.

- **Program development in the assembly language**

High-level programming languages such as C are sometimes not sufficiently expressive or include undesired or unnecessary functions that can affect performance. Such cases can make it necessary/important to communicate "directly" with the computer in its own language, i. e. the assembly language. An important advantage of being able to program close to the machine is an increased understanding of how the computer works.

5 Aims and learning outcomes

Knowledge and understanding

On completion of the course, the students shall be able to:

- demonstrate understanding of the main components of the UNIX system and their connections
- demonstrate basic knowledge of programming in scripting languages
- demonstrate basic knowledge of the interaction between programs written in C and the operating system in a UNIX environment
- demonstrate basic knowledge of programming in the assembly language and interaction of software and hardware at the level of the machine

Competence and skills

On completion of the course, the students shall be able to:

- use the command line interface of the UNIX system

- write programs with the building blocks of the UNIX programming interface
- develop programs in C and an assembly language that can be run in a UNIX environment

Judgement and approach

On completion of the course, the students shall be able to:

- determine the appropriate abstraction level of programming for a specific assignment
- justify, discuss and assess their own solutions in speech and writing

6 Generic skills

7 Learning and teaching

The theoretical foundations of the course is presented in lectures and / or exercises. The student is also expected to independently assimilate theoretical knowledge through self study of relevant literature. The theoretical knowledge is applied and deepened in practical laboratory work / project assignments in which topic-related problems should be solved through the implementation of the current system. Each exercise is presented in writing and orally.

English

8 Assessment and grading

Examination of the course

Code	Module	Credit	Grade
1310	Shell script programming	2.5 ECTS	A-F
1320	UNIX Programming in C	2.5 ECTS	A-F
1330	Assembly Programming	2.5 ECTS	A-F

The course will be graded A Excellent, B Very good, C Good, D Satisfactory, E Sufficient, FX Fail, supplementation required, F Fail. The final grade is based on a weighted average. Rounding occurs downwards.

9 Course evaluation

The course coordinator is responsible for systematically gathering feedback from the students in course evaluations and making sure that the results of these feed back into the development of the course.

10 Prerequisites

To be admitted, students must have completed courses in Programming, 15 ECTS, Algorithms and Data Structures, 6 ECTS, operating systems, 6 ECTS and data alt local networks, 4 ECTS.

11 Field of education and subject area

The course is part of the field of education and is included in the subject area Computer Science and the subject area Software Engineering.

12 Restrictions regarding degree

The course cannot form part of a degree with another course, the content of which completely or partly corresponds with the contents of this course.

13 Additional information

The main programming languages are C and Bourne Shell. Students are expected to have learnt these programming languages earlier or be able to learn them on their own during the course.

Replaces DV1305 and DV1425.

14 Course literature and other teaching material

■