



COURSE SYLLABUS

Programmering i UNIX-miljö Programming in UNIX Environment 6 credits (6 högskolepoäng)

Course code: DVI578

Main field of study: Computer Science

Disciplinary domain: Technology

Education level: First cycle

Specialization: G2F - First cycle, has at least 60 credits in first cycle course/s as entry requirements

Subject area: Computer Technology

Language of instruction: English

Applies from: 2020-01-20

Approved: 2019-12-17

1. Decision

This course is established by Dean 2017-06-02. The course syllabus is approved by Head of Department of Computer Science 2019-12-17 and applies from 2020-01-20.

2. Entry requirements

For admission to the course, the student must have completed courses in Programming 12 credits, Algorithms and data structures 6 credits. Taken courses in Operating Systems 6 credits, Computer Communication or Local Network at least 4 credits.

3. Objective and content

3.1 Objective

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.

3.2 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.

- A general understanding of 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.

4. Learning outcomes

The following learning outcomes are examined in the course:

4.1 Knowledge and understanding

On completion of the course, the student will 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

4.2 Competence and skills

On completion of the course, the student will 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

4.3 Judgement and approach

On completion of the course, the student will 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

5. Learning activities

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.

6. Assessment and grading

Modes of examinations of the course

Code	Module	Credits	Grade
2005	Written assignment 1	2.5 credits	AF
2015	Written assignment 2	2.5 credits	AF
2025	Written assignment 3	1 credits	AF

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.

The course-PM for each course revision should include the assessment criteria and make explicit in which modes of examination that the learning outcomes are assessed.

An examiner can, after consulting the Disability Advisor at BTH, decide on a customized examination form for a student with a long-term disability to be provided with an examination equivalent to one given to a student who is not disabled.

7. Course evaluation

The course evaluation should be carried out in line with BTH:s course evaluation template and process.

8. Restrictions regarding degree

The course can form part of a degree but not together with another course the content of which completely or partly corresponds with the contents of this course.

9. Course literature and other materials of instruction

10. Additional information

This course replaces the course DVI457

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.