Operating systems
Operativsystem

7.5 credits

Course Code: 5DV171
Syllabus valid from: 2016, week 20
Responsible Department: Department of Computing Science
SCB Subject: Computer Science

Main Field of Study and progress level: Computing Science: first cycle, has at least 60 credits in first-cycle course/s as entry requirements (G2F)
Grading System: 5 Pass with distinction, 4 Pass with merit, 3 Pass, VG Pass with distinction, G Pass, U Fail
Level of Education: Basic level

Requirements
To be admitted you must have 60 ECTS-credits in Computing Science or 2 years of completed studies, in both cases including good knowledge and experience of the programming language C, including working with threads. You should also have knowledge and experiences of programming directly with/to an operating system. These requirements are fulfilled by the course C-programming and Unix or equivalent.

Learning Outcomes

Knowledge and Understanding
After finished course the student should be able to:

- demonstrate knowledge and understanding of the different modules in a modern Operating System (OS) in general. (FSR 1)
- exemplify and explain how the kernel of an OS is designed, and how it works including being able to explain what a process is, the interaction between the kernel and the hardware, user mode vs. kernel mode and process management (FSR 2)
- demonstrate knowledge and understanding of how concurrency in OS is handled including thread abstraction, synchronizing access to shared objects, resources and scheduling in the OS (FSR 3)
- demonstrate knowledge and understanding of how memory is managed in an OS. This includes address translation, caching, virtual memory, cache misses and replacement policies. (FSR 4)
- explain how persistent storage is handled by the OS (FSR 5)

Skills and Abilities
After finished course the student should be able to:

- demonstrate practical skills in kernel hacking and developing OS modules for the kernel. (FSR 6)
- solve problems in groups and adapting to changes in the group setting (FSR 7)
- demonstrate an ability to deal with large and complex code bases for complex softwares. (FSR 8)

Values and attitudes
After finished course the student should be able to:

- during problem solving, identify areas where the student needs further knowledge, find relevant information and apply it to the problem (FSR 9)

Contents
The goal of the course is to introduce the students to modern operating systems design. Both practical and theoretical aspects of operating Systems will be studied. The course will cover 5 main topics, namely,
Module 1: Introduction
This module will provide a top level description of modern operating systems.

Module 2: Kernels and Processes
Introduces the students to the design and implementation of different operating system kernels in some detail.

Module 3: Concurrency, Synchronization and Scheduling
The module covers threads and the semantics of concurrency, synchronization of multi-threaded programs, deadlocks, race conditions, processor scheduling and other related topics.

Module 4: Memory Management
The fourth module covers topics related to memory such as hardware and software address translation mechanisms, caching and virtual memory.

Module 5: I/O and File Systems
This module introduces the file system abstraction, different storage devices and filesystems

Instruction
Instruction consists of lectures, group exercises, and mandatory assignments. In addition to scheduled activities, individual work with the material is also required.

Examination
The student is assessed in three different ways: three assignments, a project done in groups of students and a written exam. On the whole course one of the grades Fail (U), Pass (3), Pass with merit (4) or Pass with distinction (5) is given.

In order to get the grade Pass (3) on the course a student must pass the project and pass the last of the three assignments.
For the higher grades Pass with merit (4) and Pass with distinction (5) a summary assessment of the results of all parts of the examination is done.

For all students who do not pass the written examination and/or the assignments there are more opportunities to do the examination.

If a student (or a group of students) has actively participated in the majority of the project work but not finished it by the end of the course he/she can get some extra time or an complementary assignment to solve within a given time frame. If that is done satisfactory, the student will pass the project part of the course. If a student has not participated in the majority of the project work during the course, he/she cannot be examined on the project part until the next time the course is given. The student does not have the right to continue with the same project the next time (s)he attend the course, and will need to start over with the project work in collaboration with a new student group and with a new topic.

A student who has passed the course and received a grade may not be re-examined. A student who has taken two tests for a course or segment of a course, without passing, has the right to have another examiner appointed, unless there exist special reasons (Higher Education Ordinance Chapter 6, section 22). Requests for new examiners are made to the head of the Department of Computing Science.

Examination based on this syllabus is guaranteed for two years after the first registration on the course. This applies even if the course is closed down and this syllabus ceased to be valid.

TRANSFER OF CREDITS
Students have the right to be tried on prior education or equivalent knowledge and skills acquired in the profession can be credited for the same education at Umeå University. Application for credit is submitted to the Student Services / Degree. For more information on credit transfer available at Umeå University's student web, www.student.umu.se, and the Higher Education Ordinance (Chapter 6). A refusal of crediting can be appealed (Higher Education chapter 12) to the University Appeals Board. This applies to the whole as part of the application for credit transfer is rejected.

Other Directives
This course may not be used towards a degree, in whole or in part, simultaneously with another course of similar content. If in doubt, consult the student counselors at the Department of Computing Science and / or program director of your program.

Course Literature

Reading list valid from: 2016, week 25

*Anderson, Thomas; Dahlin, Michael*

**Operating systems : principles and practice**


ISBN:9780985673529  LIBRIS-ID:19430560