Concept

Educational Objectives

• an excursus on modern operating-system design and implementation
  – focusing on system-software flexibility, portability, and scalability

• the use of software-engineering techniques in system-software design
  – feature modeling [3]
  – program families [5]
  – object orientation [8]
  – aspect-oriented programming [4]

• an inauguration into the secrets and a rationale of PURE [1]
Subject Matter

introduction .................................................. a first problem analysis
fundamentals ...... domain analysis, program families, and object orientation
case study ...... design and development of TAL, a threads abstraction layer
discussion ............... things nice to have and not to have
revision ........................ aspect orientation
conclusion ................ lessons learned

Prerequisites

• structured computer organization, operating systems
• C/C++, assembler
• enjoy system-level programming
• no fear of stuff hard to digest
• some sort of staying power
Syllabus

- one **lecture** per week, two hours each ................................. 2 SWS
  - subject presentation

- one **seminar** per week, two hours each ................................. 2 SWS
  - subject consolidation
  - practice discussion

- **computer practice** $N$ hours per week, $0 < N \leq 164$ .................. 0 SWS

---

Achievement Control

- **practice** or
  - **pass** ........................ in case of successful elaboration of all exercises
  - **consultation** ................. in case of unsuccessful elaboration of one exercise
  - **fail** ................................ otherwise, or enjoy...

- **examination** on lecture and seminar stuff
Operating-System Engineering — Concept
6

Suggested Reading


