Norwegian University of Science and Technology

Operating Systems

Rehearsal and Q&A Session 2 – 26.01.2022

Michael Engel

Lectures 1 and 2

- Definitions: what is an operating system?
- History of computing
- Usage models of computers
- Sharing of resources: multiprogramming
- Interactive use of computers

- A simple model of a computer
- Instruction execution
- Memory hierarchy and non-functional properties
- Multiprocessing, NUMA and on-chip buses
- Heterogeneous computing: GPGPUs
- Security, memory mapping and protection

What is an operating system?

- Many different definitions to different extents

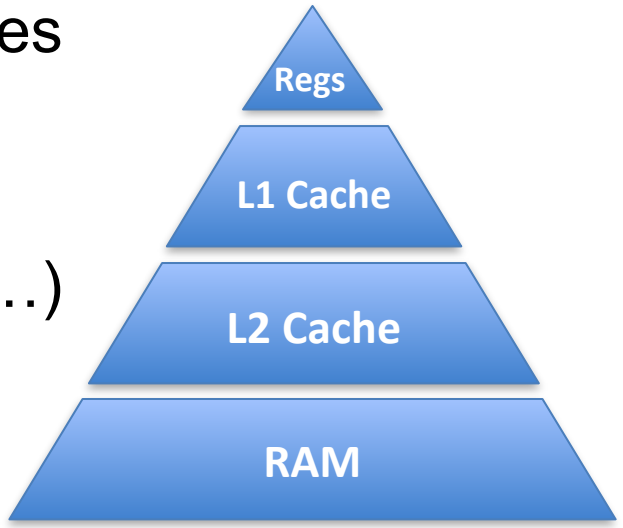
- "It is hard to pin down what an operating system is other than saying it is the software that runs in kernel mode—and even that is not always true."
- "An OS is a program that controls the execution of application programs, and acts as an interface between applications and the computer hardware."
- Criteria for evaluating an OS:
  - Convenience
  - Efficiency
  - Ability to evolve

Usage models of computers

- First: serial processing
  - Non interactive, manual scheduling
  - One user at a time, whole computer for him/herself
  - Often a waste of resources
- Batch systems
  - Reduced frequency of operator interaction
  - First operating systems providing common features
  - I/O bottleneck: processor idle while waiting for I/O
- Multiprogramming
  - Several activities "at the same time"
  - CPU used by other activity when waiting for I/O
- Dialog computing
  - Interactive use of the computer by many users at the same time

A simple model of a computer

- Three main components: CPU, memory, I/O devices
  - Interconnected using buses
  - Asynchronous operation using interrupts
  - Simple model: one machine instruction after the other
- More realistic computer model today
  - Multiple processors or processor cores
- Memory speed does not grow as fast as CPU speed
  - Memory hierarchy (caches, DRAM, …)
  - Memory local to processors: non-unified memory architecture (NUMA)



Heterogeneous computing: GPGPUs

- Processing units are no longer all the same
- Different speed and performance
  - Same instruction set, e.g. ARM big.LITTLE
  - Different instruction sets, e.g. a DSP accelerator
- Different architectures
  - One or more "regular" processors
  - General purpose graphics processing unit(s) (GPGPUs)
  - Generic FPGA accelerators
  - Special function accelerators
    - e.g. for video and image processing, neural networks, …

Security, memory mapping and protection

- Multiple processes share resources of a single computer
  - How can we avoid that one process (accidentally or intentional) accesses resources of another process?
    - e.g. memory, disk space, CPU time
  - Hardware supported **isolation**
    - Virtual memory using the memory management unit (MMU)
    - Memory space of the CPU (e.g. 4 GB on a 32 bit processor) divided into pages of virtual addresses of equal size, e.g. 4 kB
    - Mapping from virtual to physical addresses per page
    - Protection settings configurable per page

Further organization

- First theoretical exercise
  - **Optional**
  - Handout: this Friday (28.1.)
  - Submission deadline: Next Friday (4.2.)

- First practical exercise
  - Mandatory
  - Handout: next week Monday (31.1.)
  - Submission deadline: + 3 weeks: 21.2.

- Hints and questions for the first practical exercise also in the live session next Monday