If the automobile had followed the same development cycle as the computer, a Rolls-Royce would today cost $100, get a million miles per gallon, and explode once a year, killing everyone inside.

-- Robert X. Cringely
1.1 Functions and Coarse Structure

Operating system (Definition according DIN 44300)
“The programs of a digital computing system which lay - together with the basic properties of the computing system - the foundation for the possible modes of operation and especially control and monitor the execution of programs.”

Main Tasks
• Provision of virtual machine
  • as an abstraction of the computer system
• Resource Management
Tasks

- Adaption of machine structure to user requirements
- Laying the foundation for a controlled concurrency of activities
- Management of data and programs
- Efficient usage of resources
- Support in case of faults and failures
Operating system architecture

• Quotation:
  „The job of a system architect is similar to the one of a witty octopus juggling daily new balls of different size on the back of a jumping dolphin at the shore of Waikiki.“

New applications → Operating system → New Hardware

Market
Quality features:
• Security
• Real-time
• Mobility
• Energy consumption
• Dependability
• ....

Barry Linnert, linnert@inf.fu-berlin.de, Betriebssysteme WS 2016/17
Operating systems for general purpose computers

Operating system: Control

Application

Operating system: Management and operation

Hardware
System architecture

- Complex systems (in all areas) are composed of single components of different types.
- Successful design of a complex system requires the knowledge of different variants of the components and their interplay.
- Example: Buildings (20,000 years of experience)
  - Walls
  - Columns
  - Ceilings
  - Roofs
  - Stairs
  - Doors
  - Windows

Barry Linnert, linnert@inf.fu-berlin.de, Betriebssysteme WS 2016/17
System architecture

• Example Electrical Engineering (ca. 100 years of experience)
  • Resistor
  • Coil
  • Capacitor
  • Diode
  • Tube
  • Transistor

• Example Operating systems (ca. 50 years of experience)
  • Process
  • Signal
  • Address space
  • Channel
  • Interrupt
  • Driver
  • File
  • Capacity
  • Synchronization
  • Type of message transfer

Barry Linnert, linnert@inf.fu-berlin.de, Betriebssysteme WS 2016/17
1.2 Historical Sketch

The Fifties (Early stages)

- One program is being executed by *one* processor.
  - Batch operation
- The Operating system functionality is limited to
  - support of input/output,
  - transformation of number and character representation.
The Sixties (Virtualization)

- The ratio between CPU- and I/O-speed becomes large.
- OS support the interleaving execution of several independent programs (Multiprogramming).
- Real parallelism due to the advent of I/O-processors.
- The notion of a *process* as a *virtual processor* is born.
- Also the memory is „virtualized“ (*virtual memory*).
- The process also becomes an internal mean of structurization for OS.
- Interactive operation by more than one user (Timesharing).
- Prototypes or predecessors of today‘s mainframe OS are developed (OS/360, CTSS/Multics, CP67, VMOS/BS2000).
The Seventies (Software Engineering)

• The beginning of the software crisis: OS become large, complex and error prone.
• Unix is built according to the principle „simple is beautiful“ based on simple hardware (PDP-11).
• The quest for structured system design, maintainability, reliability, protection and security comes up.
• Employment of high level programming languages to implement OS.
• Process becomes a protection domain (context) with a private protected address space and access control (rights, capabilities).
• Quest for support of modular programming abstract data types and object orientation.
• Application of these principles to the operating system itself.
The Eighties (Distributed Systems)

- Workstation computers and personal computers come up.
- Increased communication bandwidth: Ethernet, connected systems.
- For efficient implementation of communication software processes are needed.
- Processes are meanwhile complex entities: A process switch costs several thousand machine instructions. Therefore, address space and process are separated allowing for several processes sharing an address space (*lightweight process, thread*).
- Concepts for parallelism are integrated into program languages.
- Distributed (parallel) computing on networks of workstation computers.
- Workstations provide an ideal means for dissemination of UNIX and UNIX becomes sort of a “standard”.
- Necessity for integration generates pressure for standardization (OSI, TCP/IP, NFS, POSIX, X/OPEN, OMG, ODP).
- OS overcome node boundaries: From communicating computers to distributed systems.
The Nineties

- Due to high production numbers, microprocessors become cheap.
- Connecting thousands of microprocessors achieves (theoretically) higher performance at a lower price compared to supercomputer (e.g. Cray).
- New OS-Functionality needed to support parallel processing.
- PCs and GUIs for OSs become mainstream (Windows 3, Linux).
- Multimedia-applications require support for audio- and video data (real-time capabilities).
- Software in *embedded systems* needs OS-support (e.g. Consumer Electronics).
- Birth and rise of the Web leading to distributed systems in heterogeneous environments (e.g. Corba, Web services).
The 2000s and Today

- Computing technology moves into the everyday while becoming increasingly small and invisible.
- OS support for ubiquitous and pervasive computing and intelligent devices (cf. Internet of things)
- OS platforms for mobile phones with multi-touch user interfaces (e.g. iOS and Android OS)
- Thin clients running web-applications within a browser (e.g. Chrome OS)
- Emulation of other OS-interfaces (i.e. several “OS worlds” on the same computer).
- Converged infrastructures, shared services and the renaissance of virtualization are enabling factors for Cloud computing.
Current topics

- Safety and security
- Robustness and dependability
- Virtualization
- Optimization for multi- and many-core processors (scheduling, locking)
- Energy consumption (mobile devices, data centers)
- User interface
- Database support for file systems
- Cluster-, Grid-, and Cloud-Computing
- Small OS (e.g. for sensor networks)
Further Reading

- Hansen, P.B.: Classic Operating Systems
  Springer, New York, 2001

- The Virtual Museum of Computing
  http://vlmp.museophile.com/computing.html

- ACM Special Interest Group on Operating systems:
  http://www.sigops.org
Example: UNIX

- **1965**: USG/USDL/ATTIS/DSG/USO/USL, MULTICS (Ken Thompson, Dennis Ritchie, Brian Kernighan)
- **1969**: Bell Laboratories Research (AT&T)
- **1973**: Berkeley Software Distribution
- **1976**: UNIX First Edition
- **1977**: Fifth Edition
- **1978**: Sixth Edition
- **1979**: PWB MERT, CB UNIX, UNIX RT
- **1980**: SEVENTH EDITION
- **1980**: XENIX (Microsoft, SCO)

**Timeline**:
- 1965: MULTICS
- 1969: UNIX First Edition
- 1973: Fifth Edition
- 1976: Sixth Edition
- 1977: PWB MERT, CB UNIX
- 1978: UNIX RT
- 1979: SEVENTH EDITION
- 1980: XENIX
- 1980: 3.0.1

**Versions**:
- 3.0
- 3.0.1
- 32V
- 3BSD
- 4.0BSD
- 1BSD
- 2BSD
Example: UNIX

1981
3.0.1

1982
4.0.1
System III

1983
5.0
System V

1984
5.2
System V
Release 2

1986
Chorus Release 3

1987
System V
Release 4

1988
Chorus V3

1989
System V
NeXT Step

XENIX (Microsoft, SCO)

XENIX 3
Eighth Edition

SunOS

SunOS 3
Ninth Edition

XENIX 5

SunOS 4
Tenth Edition

4.0BSD 2BSD

4.1BSD

4.1aBSD

4.1cBSD

2.8BSD

2.9BSD

4.2BSD

4.3BSD

2.10BSD

4.3BSD-Tahoe

2.11BSD

NET/1

Barry Linnert, linnert@inf.fu-berlin.de, Betriebssysteme WS 2016/17
More information ;-)  

- **Halt and Catch Fire**  