

Freie Universität Berlin



Freie Universität Berlin
Erasmus Program

10 ECTS

Systems Software

Professor:
Prof. Barry Linnert

Autor:
Filippo Ghirardini

Winter Semester 2024-2025

Contents

1	Introduction	3
1.1	History	3
1.1.1	1950	3
1.1.2	1960	3
1.1.3	1970	3
1.1.4	1980	3
1.1.5	1990	3
1.1.6	2000	3

Systems Software

Author: Ghirardini Filippo

Winter Semester 2024-2025

1 Introduction

Definition 1.0.1 (Operating system). *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.*

The main tasks of an OS are:

- Provision of virtual machine as an abstraction of the computer system
- Resource management
- Adaptation of machine structure to user requirements
- Foundations for a controlled concurrency activities
- Management of data and programs
- Efficient usage of resources
- Support in case of faults and failure

All of these tasks must be taken care of by the OS architect while new hardware and new applications come out in a complex market.

Every complex system in every area (e.g. buildings) is 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.

1.1 History

1.1.1 1950

In the fifties only one program was executed by one processor. The OS functionality is limited to support for input/output and transformation of number and character representation.

1.1.2 1960

The CPU and I/O speed become **faster**. Now the OS supports multiple applications running at the same time with real **parallelism**. The notion of process as a virtual processor is born and the memory is now virtualized. Interactive operation by more than one user (**timesharing**).

1.1.3 1970

Software crisis: OS become large, complex and error prone. Now design, maintainability and security are important (software engineering). High level programming languages are now used to program OS. The need for modular programming, abstract data types and object orientation come up.

1.1.4 1980

Personal computers are born. The systems can now be **connected**, for example via Ethernet. Processes are now fundamental and complex: since a context switch is very CPU intensive, address space and processes are now separated to allow sharing of the address space (**threads**).

Parallelism becomes a part of programming languages. There is a need to create standards and protocols.

1.1.5 1990

Due to the popularity, microprocessors become **cheap**. Now multiple chips can work together. GUIs are born and with them **audio and video data**. Birth of the **Web**.

1.1.6 2000

Now everything is a computer: it's ubiquitous and pervasive. The cloud computing is born. Today the main topics are:

- Safety and security
- Robustness and dependability
- Virtualization
- Optimization for multi core processors (scheduling, locking)
- Energy consumption
- User interface
- Database support for file systems
- Cluster-Computing, Grid-Computing and Cloud-Computing
- Small OS for small devices