1st Slide Set
Operating Systems

Prof. Dr. Christian Baun

Frankfurt University of Applied Sciences
(1971–2014: Fachhochschule Frankfurt am Main)
Faculty of Computer Science and Engineering
christianbaun@fb2.fra-uas.de

Organizational Information

- **E-Mail:** `christianbaun@fb2.fra-uas.de`

!!! Tell me when problems problems exist at an early stage !!!

- **Homepage:** `http://www.christianbaun.de`

!!! Check the course page regularly !!!

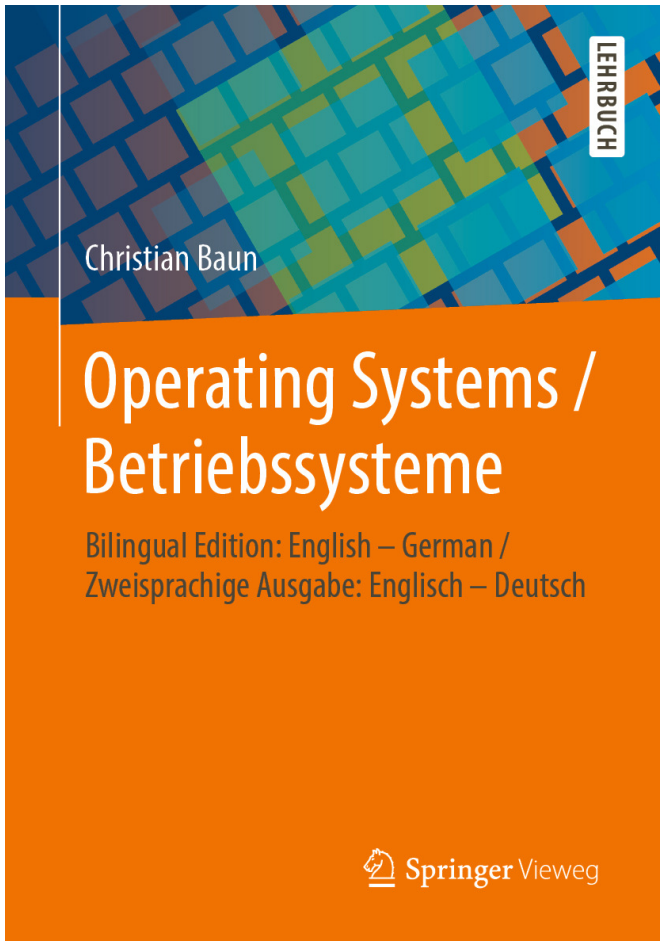
- The homepage contains among others the **lecture notes**
  - **Presentation slides** in English and German language
  - **Exercise sheets** in English and German language
  - **Sample solutions** of the exercise sheers
  - Old **exams** and their **sample solutions**

What is the password?

There is no password!

The content of the English and German slides is identical, but please use the English slides for the exam preparation to become familiar with the technical terms

Literature



- My slide sets were the basis for these books
- The two-column layout (English/German) of the bilingual book is quite useful for this course

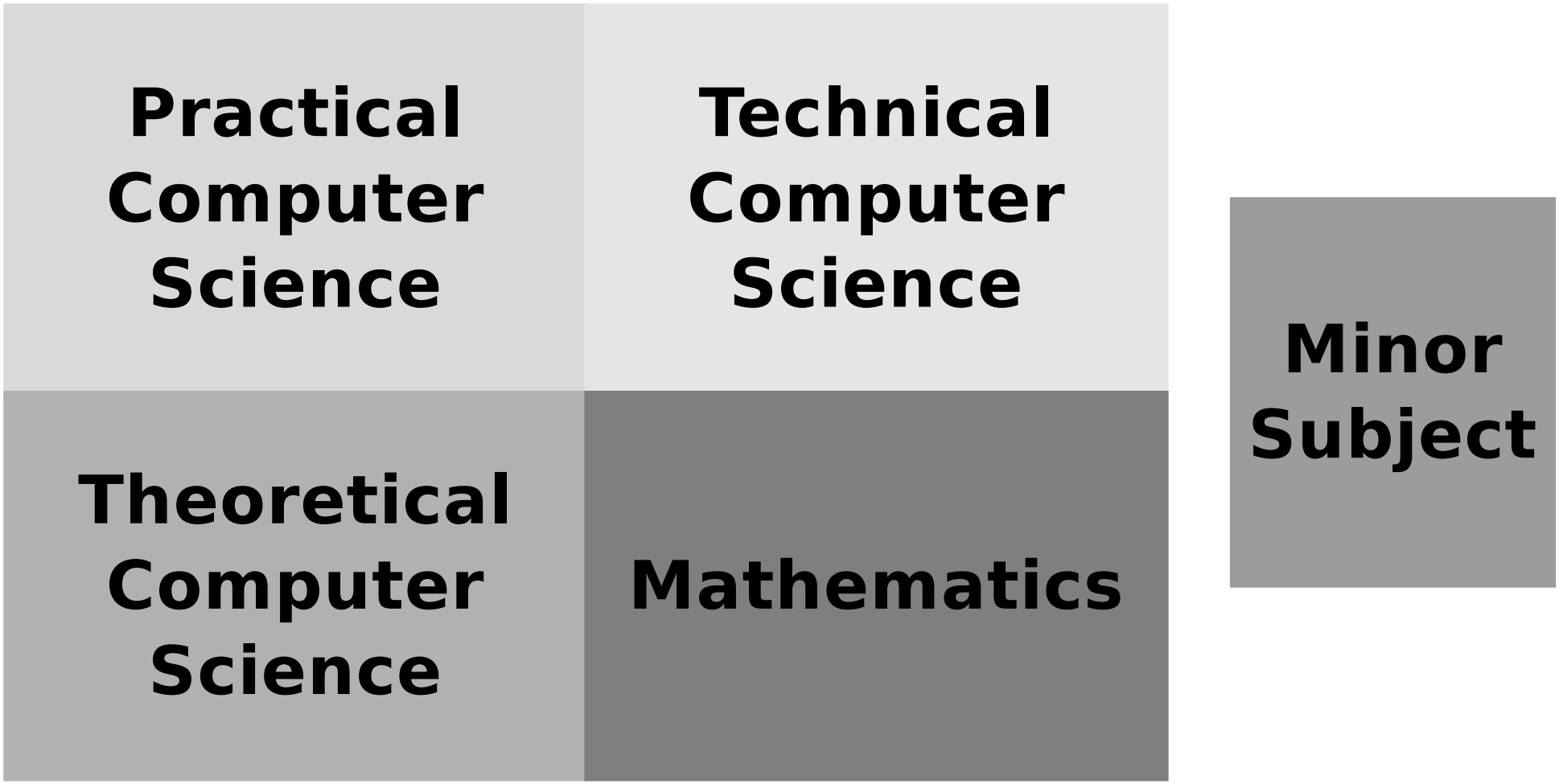
You can download both books for free via the FRA-UAS library from the intranet

Learning Objectives of this Slide Set

- At the end of this slide set You know/understand...
  - to which computer science branches the **operating systems** belong to
  - how the evolution of the hardware influenced the **evolution of the operating systems**
    - Batch processing
    - Singletasking
    - Multitasking
    - Time Sharing
  - **Core functionalities** of operating systems:
    - Memory management
    - File systems
    - System calls to manage access operations to the hardware
    - Process management
    - Interprocess communication
    - Synchronization of processes

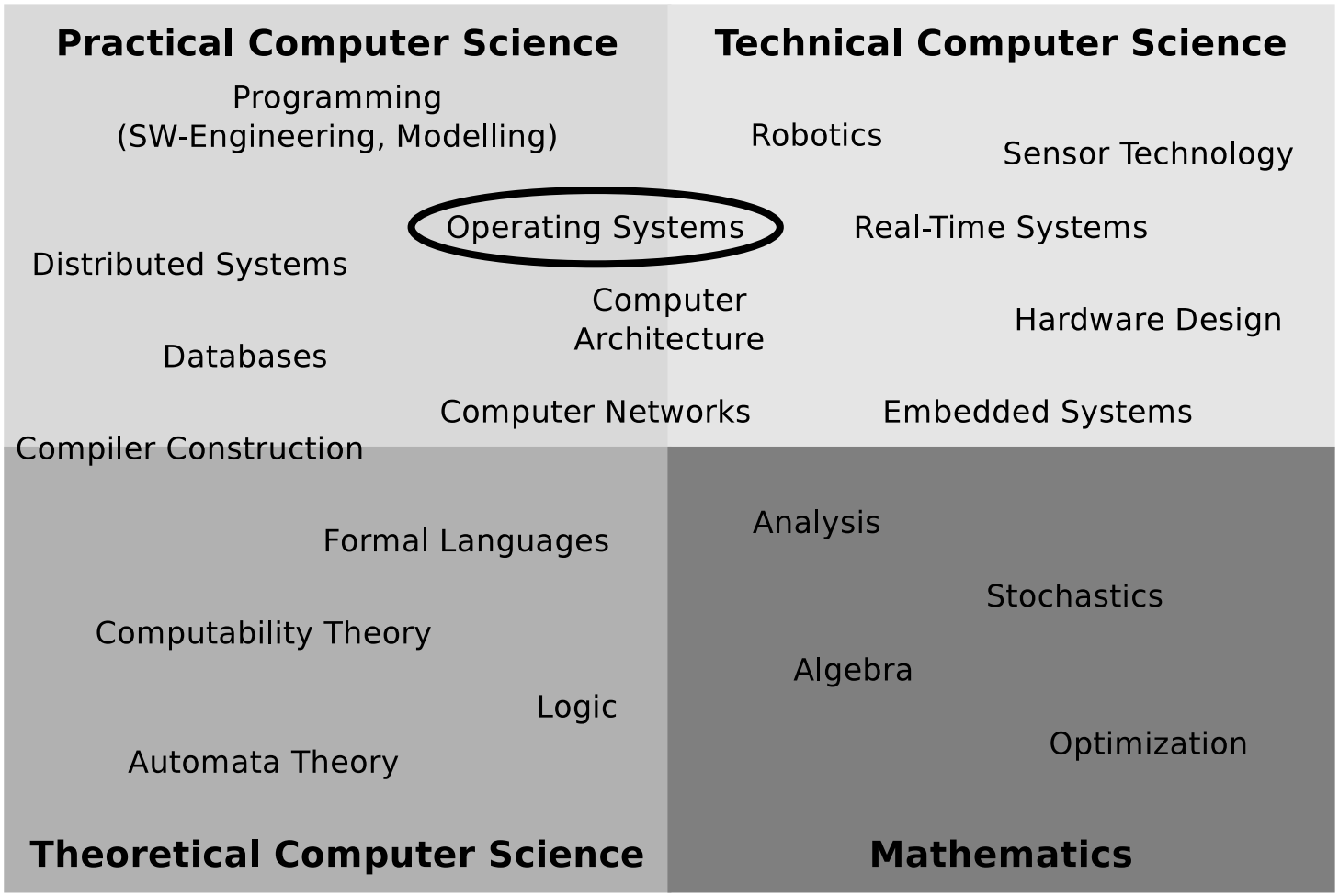
Exercise sheet 1 repeats the contents of this slide set which are relevant for these learning objectives

Operating Systems in Computer Science (1/2)



Where would you place the operating systems?

Operating Systems in Computer Science (2/2)



Minor Subject

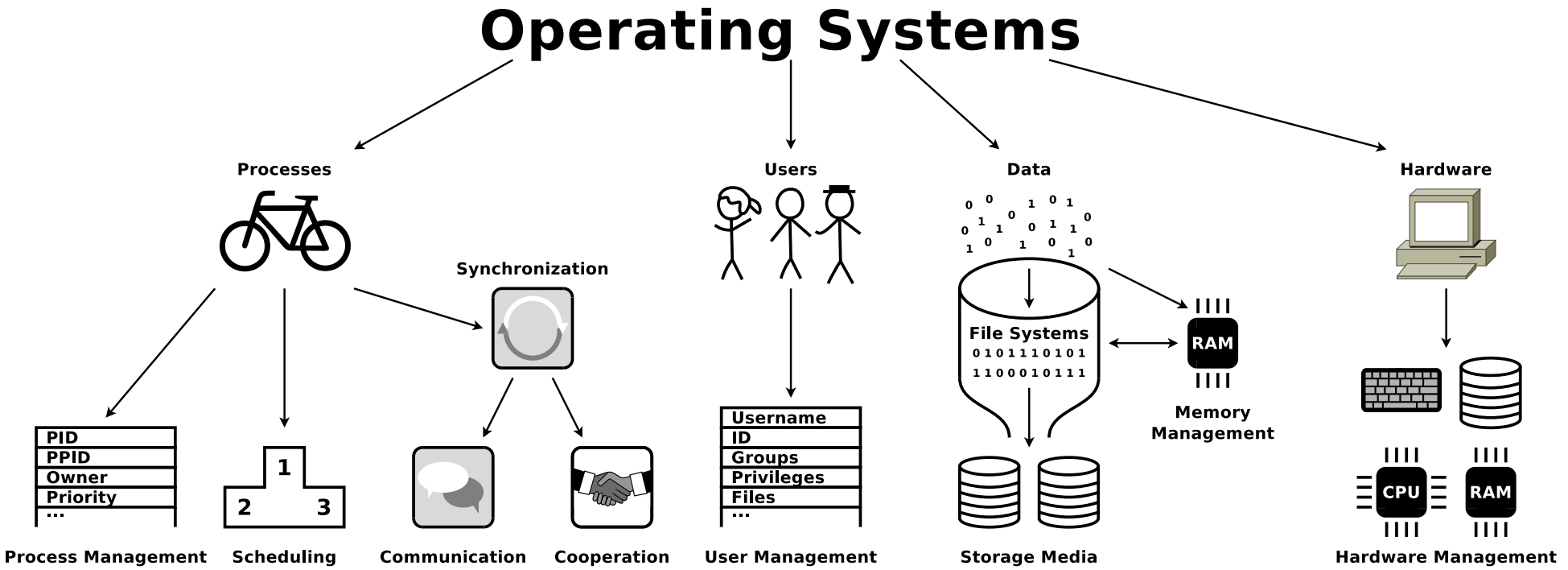
Electrical Engineering

Business Administration/ Economics

Medicine

Operating systems belong to practical computer science and technical computer science

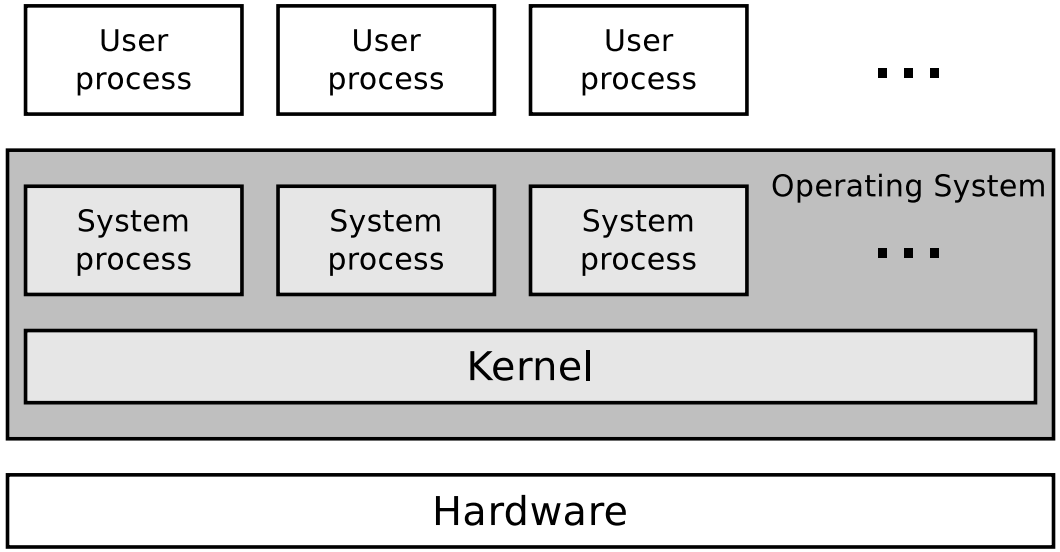
Core Functionalities of Operating Systems



At the end of the semester You...

- know and understand the **functioning** of the **core functionalities** of operating systems
- unterstand the **functioning** of the most important hardware components
- have basic skills in working with **Linux**
- have basic skills in **shell scripting**

Basic Structure of an Operating System



- **User processes** process the users' jobs
- **System processes** provide services of the operating system
- The operating system core ($\Longrightarrow$ **kernel**) contains all components of the operating system, which are not implemented as system processes
  - We will talk a lot about operating system kernels in slide set 2

Operating Systems are Part of the System Software

System software controls the operation of a computer, assists users and their applications in making use of the hardware and controls the use and allocation of the available hardware resources

Generations of Computer Systems and Operating Systems

The next slides deal with these questions:. . .

- What operating systems exist?
- Since when do operating systems exist?
- How did the evolution of the hardware influence the evolution of the operating systems?

Generation	Time period	Technological progress
0	until 1940	(Electro-)mechanical calculating machines $\Longrightarrow$ no software!
1	1940 – 1955	Electron tubes, relays, jack panels
2	1955 – 1965	Transistors, batch processing
3	1965 – 1980	Integrated circuits, time sharing
4	1980 – 2000	Very large-scale integration, microprocessors, PCs/Workstations
5	2000 until ?	Distributed systems, *the network is the computer*, virtualization

Quote from the magazine *Popular Mechanics* (1949)

„In the future, computers may weigh no more than 1.5 tonnes.“

1st Generation (1940 – 1955)

- The 1st generation of computer systems was constructed during WW2
  $\Longrightarrow$ Konrad Zuse, John von Neumann
- Requirements, a universal computer must satisfy:
  - Stored program
  - Conditional jump (GOTO)
  - Separation of memory and CPU
- Computers were machines with partially $>$ 10,000 tubes or relays, which worked slow and error prone
- **No operating systems and programming languages in this generation**
- Programs were implemented via circuits in patch bays
  - The user/programmer launches **one** program, which directly accesses the hardware

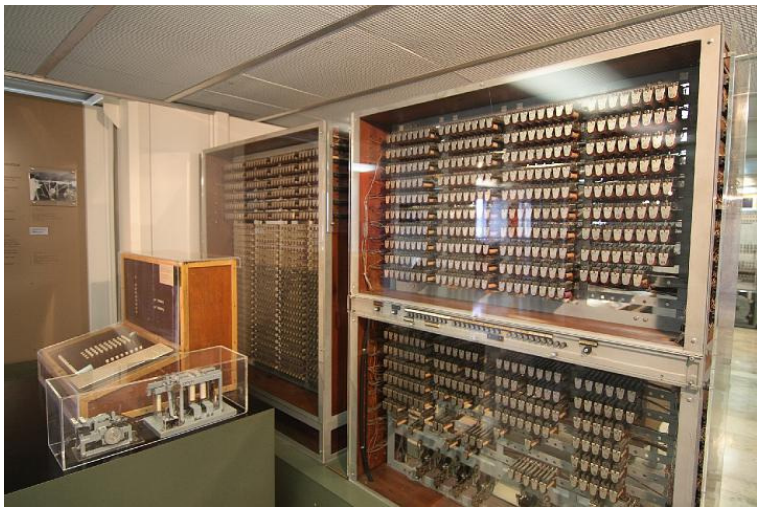
Some systems of the 1st Generation

Image Source: Own work (12.12.2008)

Computer	Development	Storage/CPU separated	Conditional jumps	Program-ming	Internal encoding	Number representations	Technology
Z1 / Z3	1936-1941	yes	no	SW	binary	floating point	mechanical (Relays)
ABC	1938-1942	yes	no	HW	binary	fixed-point	electronic
Harvard Mark 1	1939-1944	no	no	SW	decimal	fixed-point	electronic
ENIAC	1943-1945	no	partially	HW	decimal	fixed-point	electronic
Manchester	1946-1948	yes	yes	SW	binary	fixed-point	electronic
EDSAC	1946-1948	yes	yes	SW	binary	fixed-point	electronic

Computers that operate according to the decimal system?

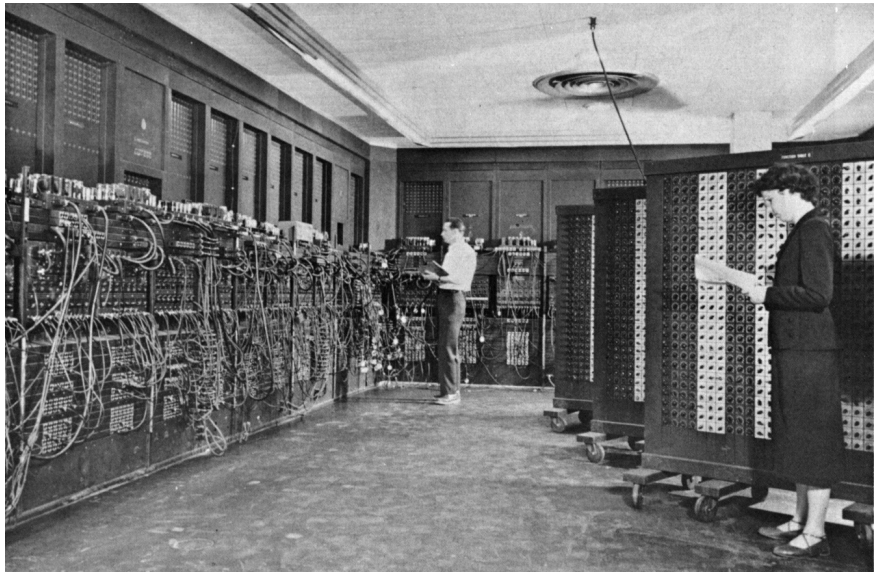
Detailed description of the structure: `http://computer-modell-katalog.de/eniac.htm`

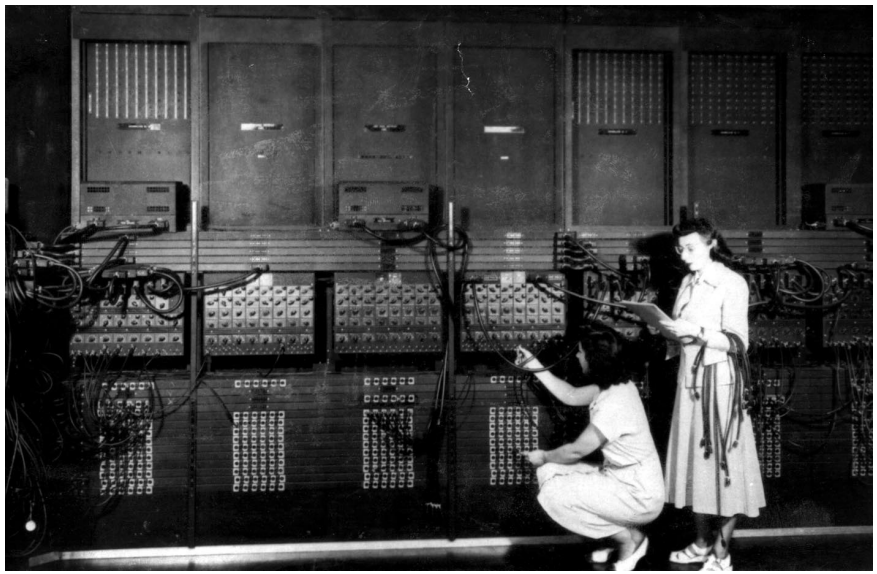


- Image: Zuse Z3 (1941)
- The world's first working programmable, digital computer (based on relay technology)
- First computer, which implemented the binary system

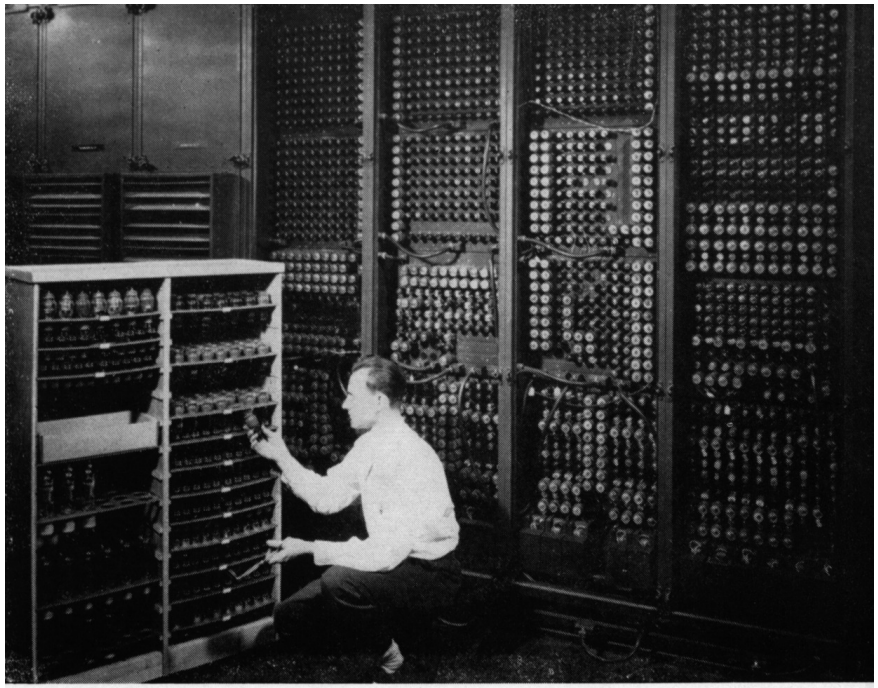
1st Generation: ENIAC (1944)

Image Source: US Army (Public Domain)





- Electronic Numerical Integrator and Computer (ENIAC)
- First electronic general-purpose computer (with electron tubes)



Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.

2nd Generation (1955 – 1965)

Image Source: Flickr (born1945, CC-BY-2.0)

- Early 1950s: Punch cards replace the patchbays
- Mid-1950s: Introduction of the transistors:
  $\Longrightarrow$ Computer systems become more reliable

- FORTRAN or COBOL programs were. . .
  - written down by the programmer on form sheets,
  - punched from coders into punch cards
  - and handed over to the **operator** (**administrator**)
- The operator. . .
  - coordinates the order (**schedule**) of programs (**jobs**)
  - equips the computer with the punch cards
  - loads the compiler from the magnetic tape
  - hands over the printed out computation result
    $\Longrightarrow$ Inefficient method
- Later, for efficiency reasons, programs were collected, stored on magnetic tape and then processed in the machine room

2nd Generation: Batch Processing (1/4)

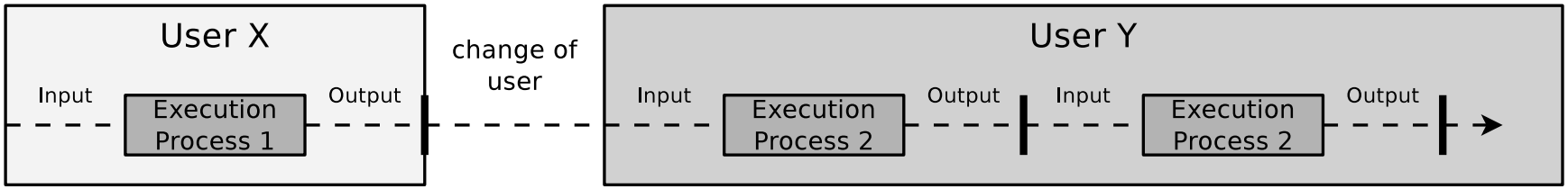
- Operating systems of this generation were all **batch processing operating systems**
- Objective: **Maximize CPU utilization**

- Each program needs to be **provided completely** (with all input data!) before the execution may begin
- Batch processing is well suited for the execution of **routine tasks**
- Today's systems still allow to process program sequences automatically (e.g. non-interactive batch files and shell scripts)

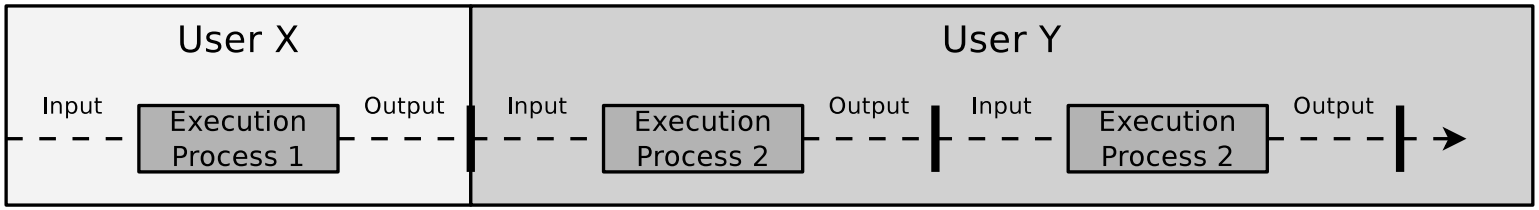
Image Source: IBM (the image shows an IBM 7090 from 1959)
`http://www.computer-history.info/Page4.dir/pages/IBM.7090.dir/images/ibm.7090.jpg`

2nd Generation: Batch Processing (2/4)

Single user mode with singletasking without batch processing



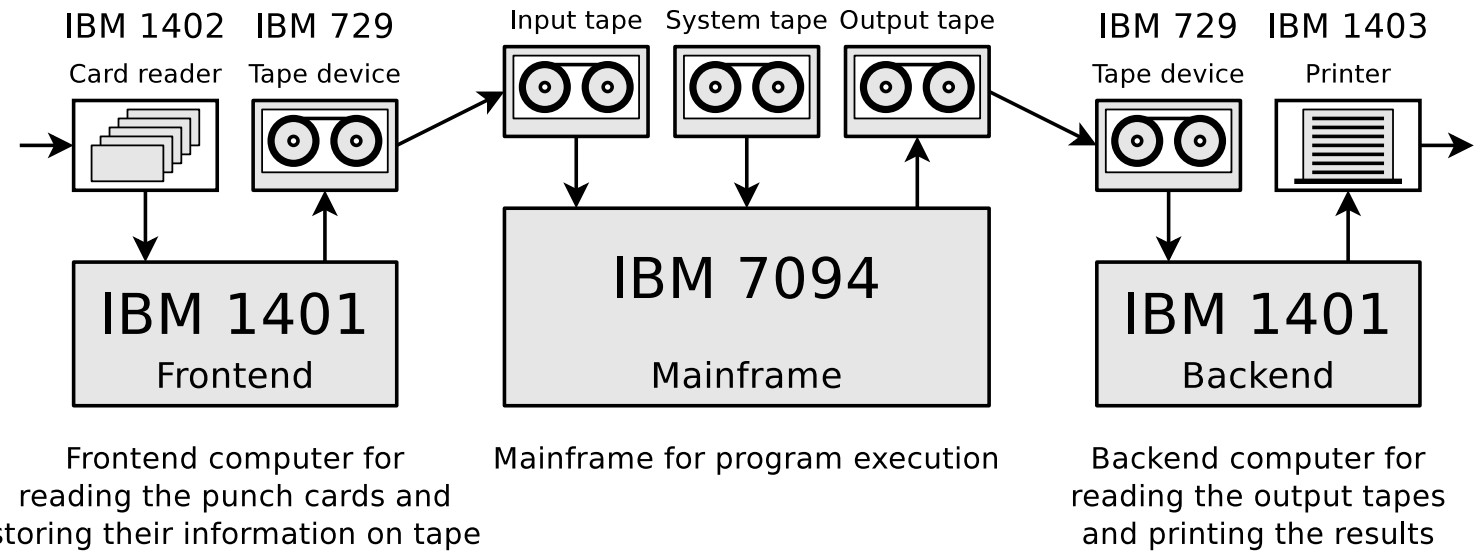
Batch processing



Time

- Batch Processing $\Longrightarrow$ **Acceleration via automation**
- Drawback: The CPU is still not utilized in an optimal way
  - During input/output operations the CPU is idle

2nd Generation: Batch Processing (3/4)



- Frontend/backend computers free the mainframe from slow I/O operation
  - Data can be read from tape much faster than from punch cards and data can be stored on tape much faster than printed out
- **Spooling** removes I/O workload from the CPU by using additional HW
  - I/O is carried out concurrently with the processing of other jobs

Today, computers have in addition to the CPU, specific I/O processors with DMA capability (*Direct Memory Access*)

These write data directly into the main memory and fetch the results from there

2nd Generation: Batch Processing (4/4)

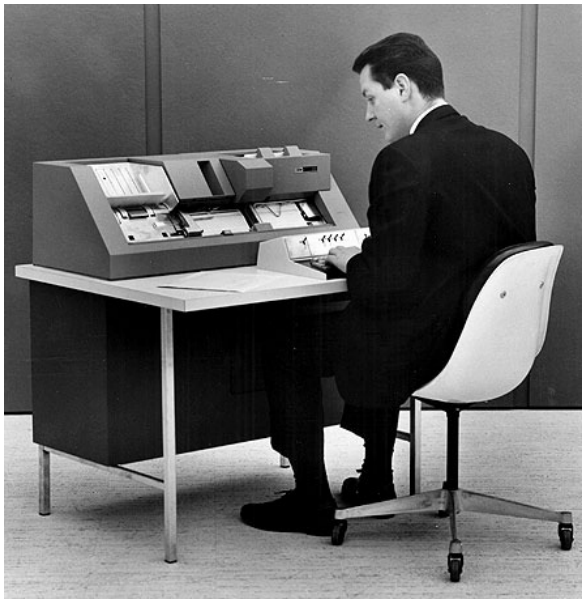


Image source: IBM Archives
`https://onfoss.com/a-timeline-of-computer-interface-technology/`

- Spooling is still used today
  - e.g. spooling processes for printing
- Batch processing is usually **non-interactive**
  - A started process is executed without any user interaction until it terminates or an error occurs

Batch processing is not obsolete today!

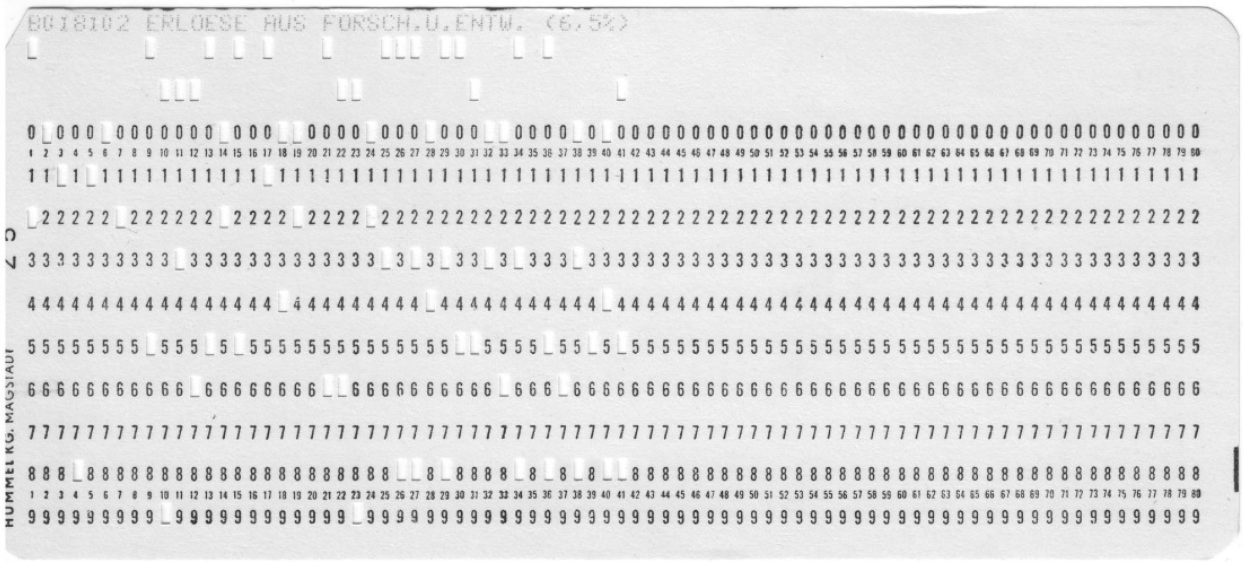
compute-intensive programs in distributed systems are usually non-interactive batch programs
$\Longrightarrow$ Distributed computing and so-called number crunching

- Batch processing operating systems of the 2nd generation only implement **singletasking** ($\Longrightarrow$ slide set 2)
  - The operating system allows only the execution of one program at once
  - Starting a second program is only possible after the first one has finished

Some Operating Systems of the 2nd Generation

Atlas Supervisor, GM-NAA I/O, UMES, SHARE, IBSYS

2nd Generation: Punch Cards



- Each punch card usually represents a single line of text with 80 characters or a corresponding number of binary data
  - The standard line size of ≤ 80 characters in E-mails and text files dates back to the punch card
- 12 punch hole positions for the encoding of each character
  - Digits are encoded with a single hole in the corresponding row
  - Letters and special characters are encoded by punching multiple holes in the column

3rd Generation (1960 – 1980)

- Early 1960s: Integrated circuits are available
  $\Longrightarrow$ More powerful, smaller and less expensive computers
- 1960s:
  - Improvement of the batch processing systems to allow the execution of multiple jobs during the same period of time $\Longrightarrow$ **multitasking**
  - First simple **memory management** (*fixed partitions*) $\Longrightarrow$ slide set 5
- 1970s: **Time-sharing** (*interactive mode*)
  - One central unit, multiple terminals
  - Each user gets a user process when logging in
- End of the 1970s: Development of the microprocessor
  $\Longrightarrow$ Development of the home computer / personal computer (PC)
  - 1977: Apple II. First home computer
  - 1981: IBM PC. Top selling computer architecture (Intel 80x86)

Some Operating Systems of the 3rd Generation

BESYS, CTSS, OS/360, CP/CMS, Multics, Unics (later Unix), DEC DOS-11, DEC RT-11, Version 6/7 Unix, DEC CP/M, Cray Operating System, DEC VMS

Some systems of the 3rd Generation

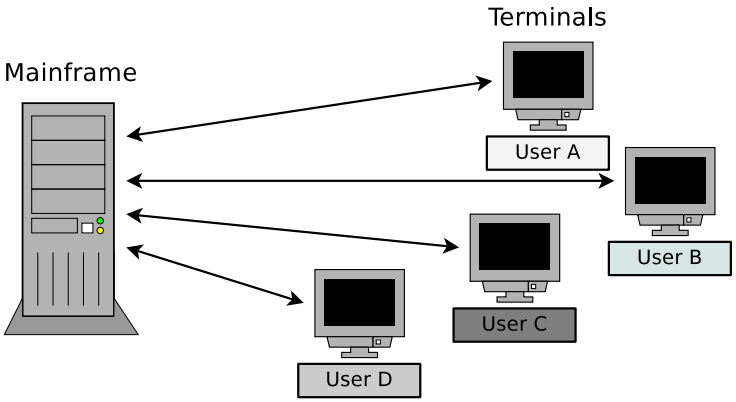
Image Source: Clemens Pfeiffer (CC-BY-2.5)

Computer	Development	Special features
CDC 6600	1964	First supercomputer
IBM System/360	1964	8-bit character size. Flexible architecture
PDP-8	1965	First commercial minicomputer from DEC
ILLIAC IV	1969	First multiprocessor computer
CRAY 1	1976	Supercomputer

This generation includes also. . .

- first decentralized computer network (ARPANET)
- computer networks to connect terminals with mainframe computers via serial lines (e.g. IBM Systems Network Architecture)
- proprietary interconnection networks (e.g. DECnet)

3rd Generation: Time-sharing (1/2)



- **Multiple users** work with a single computer in a **simultaneous and competitive** way by sharing the available computing time of the CPU
  - Objective: Fair distribution of the computing time
- The computing time is distributed via **time slices**
  - The distribution can carried out according to different strategies
- **Multiple users** can work **interactively** and **simultaneously** with a computer via terminals $\Longrightarrow$ **Multi-user operation** ($\Longrightarrow$ slide set 2)
- The programs of the individual users are independent of each other
- The quasi-parallel program or process execution is called **multitasking** ($\Longrightarrow$ slide set 2)
  - Objective: Minimizing the response time

3rd Generation: Time-sharing (2/2)

- Because of time-sharing, new concepts were required:
  - **Memory protection**: The memory is split and running programs are separated from each other
    - This way, a bug or crash of a single program does not affect the stability of other programs and the total system

    $\Longrightarrow$ slide set 5
  - **File systems**, which allow quasi-simultaneous file access

    $\Longrightarrow$ slide set 6
  - **Swapping**: Process of storing and removing data to/from main memory from/into background memory (HDDs/SSDs)

    $\Longrightarrow$ slide set 7
  - **Scheduling**: Automatic creation of an execution plan (*schedule*), which is used to allocate time limited resources to users or their processes

    $\Longrightarrow$ slide set 8

4th Generation (1980 – 2000)

- This generation provides highly integrated circuits and an exponentially growing integration density of electronic components
  - CPUs become more powerful and cheaper
  - The main memory capacity rises
- High computing power can be installed on every workplace
  - Workstations become standard in the in the professional sector
  - Popularity of home computers and personal computers (PC) rises
    - Main objective of operating systems: **Intuitive user interfaces** for users who do not want to know anything about the underlying hardware

Some Operating Systems of the 4th Generation

QDOS, Xenix, MS-DOS, PC-DOS, QNX, GNU project, SunOS, MacOS, AmigaOS, Atari TOS, Windows, IBM AIX, GEOS, SGI IRIX, MINIX, OS/2, NeXTSTEP, SCO UNIX, Linux, BeOS, Haiku, Google Fuchsia

- Computer networks with open standards became popular
  - Ethernet, Token Ring, WLAN ($\Longrightarrow$ computer networks course)

5th Generation (2000 – ????)

- Some key words from the 5th generation:
  - *The network is the computer*
  - Distributed systems $\Longrightarrow$ Cluster-, Cloud-, Grid-, P2P-Computing
  - Multicore processors and parallel applications
  - Virtualization $\Longrightarrow$ **VMware, XEN, KVM, Docker...**
  - Free Software (OpenSource) $\Longrightarrow$ **Linux (Android), BSD**,...
  - Communication everywhere $\Longrightarrow$ mobile systems
  - New ways of working $\Longrightarrow$ e-Science, e-Learning, e-Business,...
  - Services $\Longrightarrow$ web services (REST, SOAP, JSON)
  - Resources are requested and rent when needed $\Longrightarrow$ on demand
  - Personal Computing vs. Parental Computing (e.g. iOS)
  - Artificial Intelligence (AI)
- Keywords for later generations:
  - Quantum computers (maybe 6th or 7th generation)