Operating systems
- An overview -
+
UNIX/Linux Introduction

Prof. Răzvan Zota

Operating systems

Grading rules

- 30% - seminar activity (10% - test1 + 10% - test2 + 10% - project)
- 70% - result from the final examen (test using online.ase.ro)

Conditions to pass the exam: you must achieve at least 50% from the total points at the seminar and at the final exam

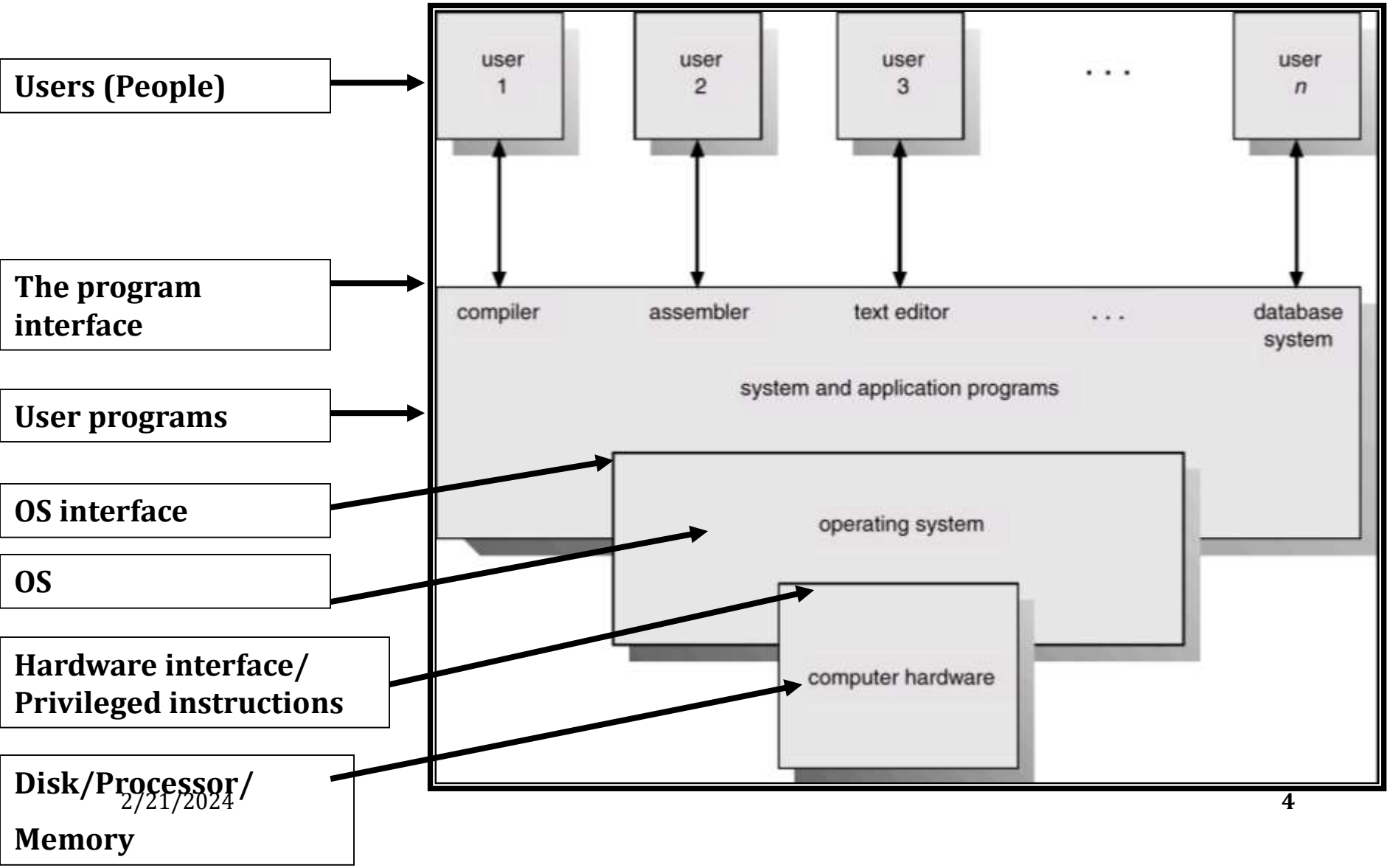
Operating systems Introduction

What is an operating system?

- A set of programs for managing the computer's resources
- An interface between users and hardware – a medium "architecture"
- Enables convenient data storage, hiding details from the user
- Enables the efficient usage of the system, parallel execution of multiple activities, not wasting the clock cycles
- Enables the information protection
- Giving each user a "slice" from the total resources of the system
- It actions like a control program.

Operating systems
An overview

OS position



Operating systems
An overview

Components

As a big picture, an OS can be seen as:

- A mechanism used for jobs' and processes' scheduling. The scheduling can be as simple as running the next process from a waiting queue or can be more complex by using more complicated rules for choosing next process to run

- A way for simultaneous execution for multiple CPUs and I/O management.

Operating systems An overview

Components

CPU activity is wasted if a job waits for an I/O operation. This lead to:

- **Multiprogramming**. While a job is waiting for a resource to be free the CPU can run another job. This means that multiple jobs are ready (in the same time) to run and are waiting for the CPU to continue.

CPU scheduling is an important part in the OS study.

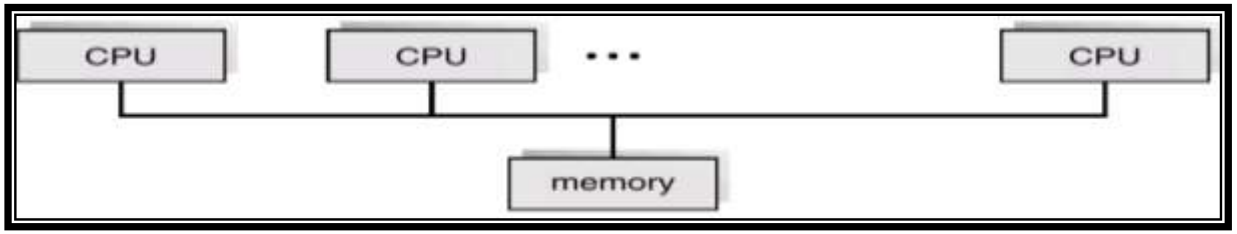
Al these are leading also to:

- **memory and process management**
- **resource planning**
- **deadlock protection**

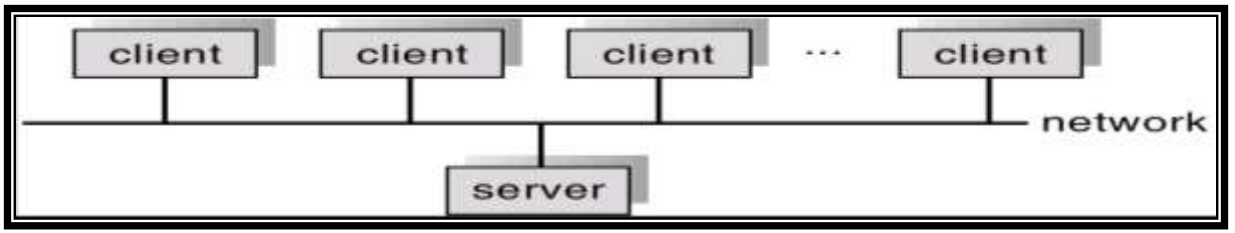
Operating systems
An overview

Characteristics

- **Time sharing** – the multiprogramming medium is, also, interactive
- **Multiprocessing** - shared memory collaboration and communication. Used especially for improving speed by putting to work more processors in the same time.



- **Distributed systems** – Long distance connected systems communicating by message transfer. Advantages: resource sharing, more speed, reliability, communication.



- **Real time systems** – The main characteristic is the quick answer. Used for controlling applications where a quick answer is essential.

OS types

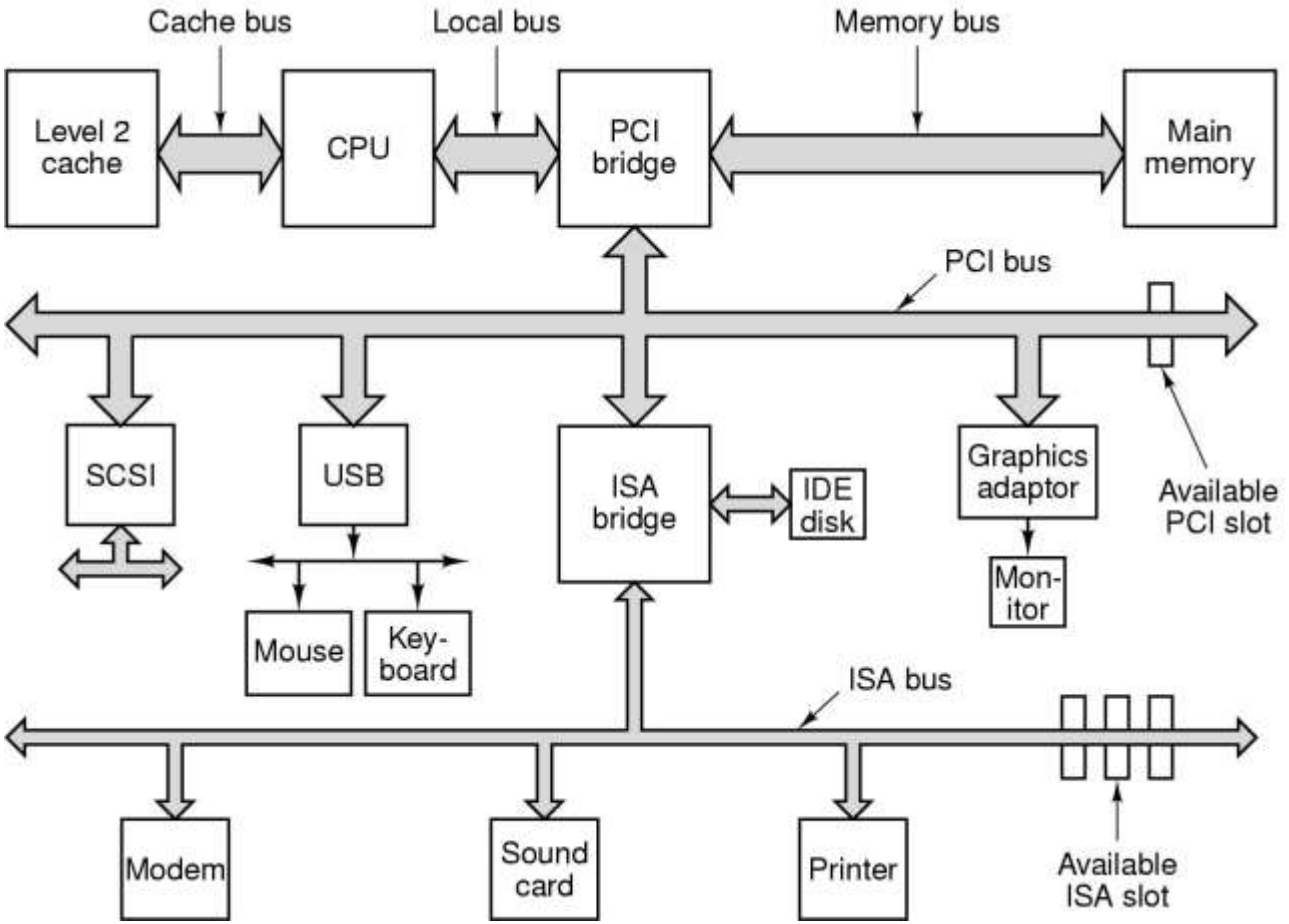
- **Smart card OS-s –** have basic functions like: secure access to card storage, authentication and cryptography(the most usual are JavaCard and MULTOS).
- **Embedded OS –** there are OS-s incorporated in mobile phones, TVs, etc. Examples: Android, iOS, Windows Phone.
- **Real time OS-s (RTOS – Real Time OS) –** used in scientific applications (space ships, etc.), industry (auto – car infotainment systems, robots, etc.), medicine (medical equipment). Examples: RTLinux, QNX. The main characteristic is the response time. Two categories: **hard RT** – where the time constraints are essential and **soft RT** – where these constraints are not so important (for example, in case of dedicated multimedia systems).
- **Desktop OS-s** – the Windows family (7,8,10), Linux (several distros), Mac OSX (El Capitan), macOS High Sierra.
- **Server OS-s** - Unix/Linux (RHEL, SLES – Suse Linux Enterprise Server), Windows Server 2016.
- **Mainframe OS-s** – IBM z /OS (z10), Linux, OpenSolaris.

Operating systems
An overview

Hardware support

One system is made by several devices

These devices may generate an electrical signal (called interrupt) which catch the attention of the CPU.



Operating systems
An overview

Hardware support

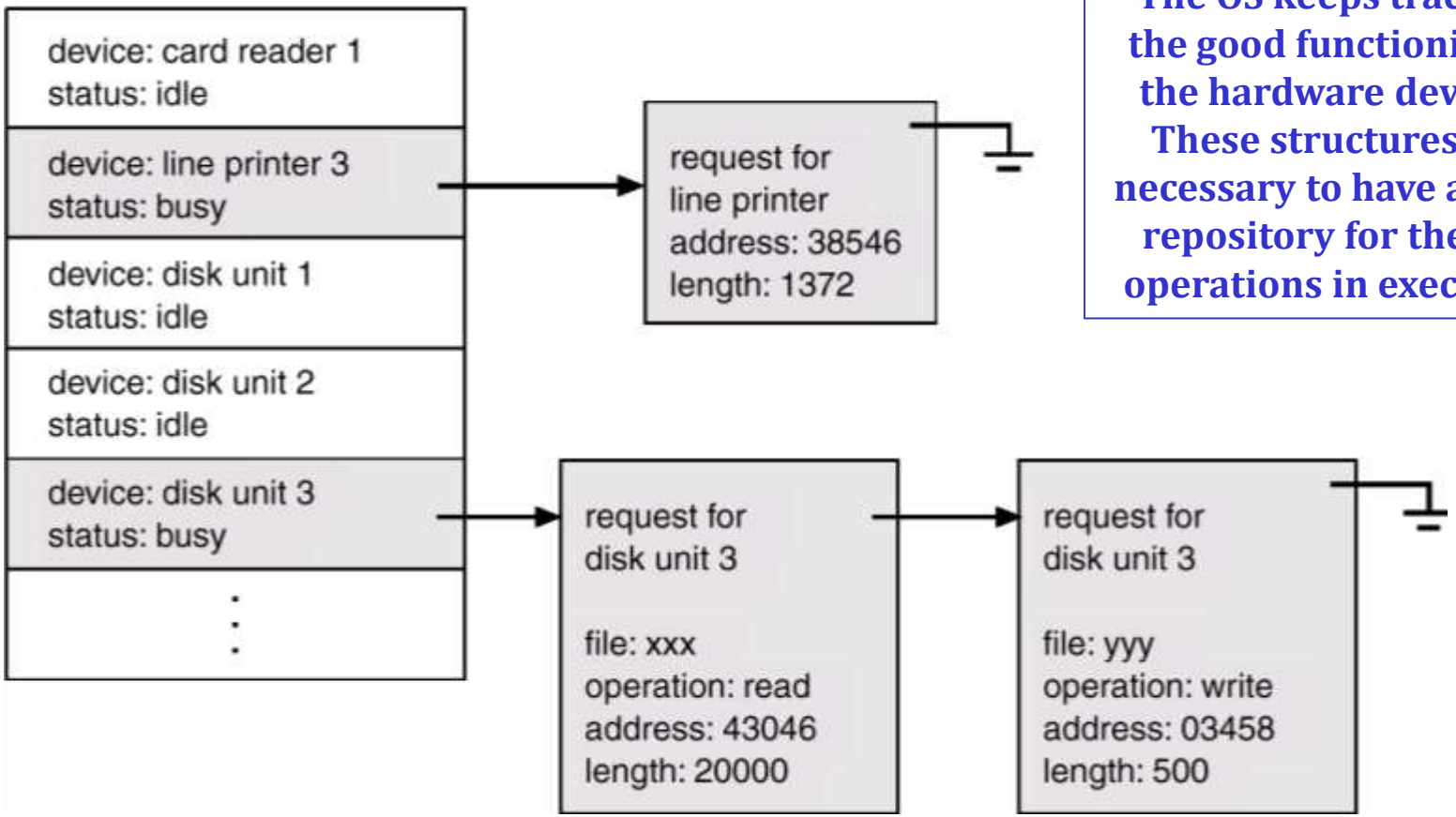
Interrupts - a device is sending "a disturbing" signal to the CPU to get a service. Different from the action of **polling**.

- Depending by the interrupt it is determined the next step to be done.
- There are *soft interrupts* and *hard interrupts*.
- The interrupt manager choose the code to be run on each device.

DMA I/O Controllers (Direct Memory Access) have direct access to the memory, without "asking" CPU.

Operating systems
An overview

Hardware support



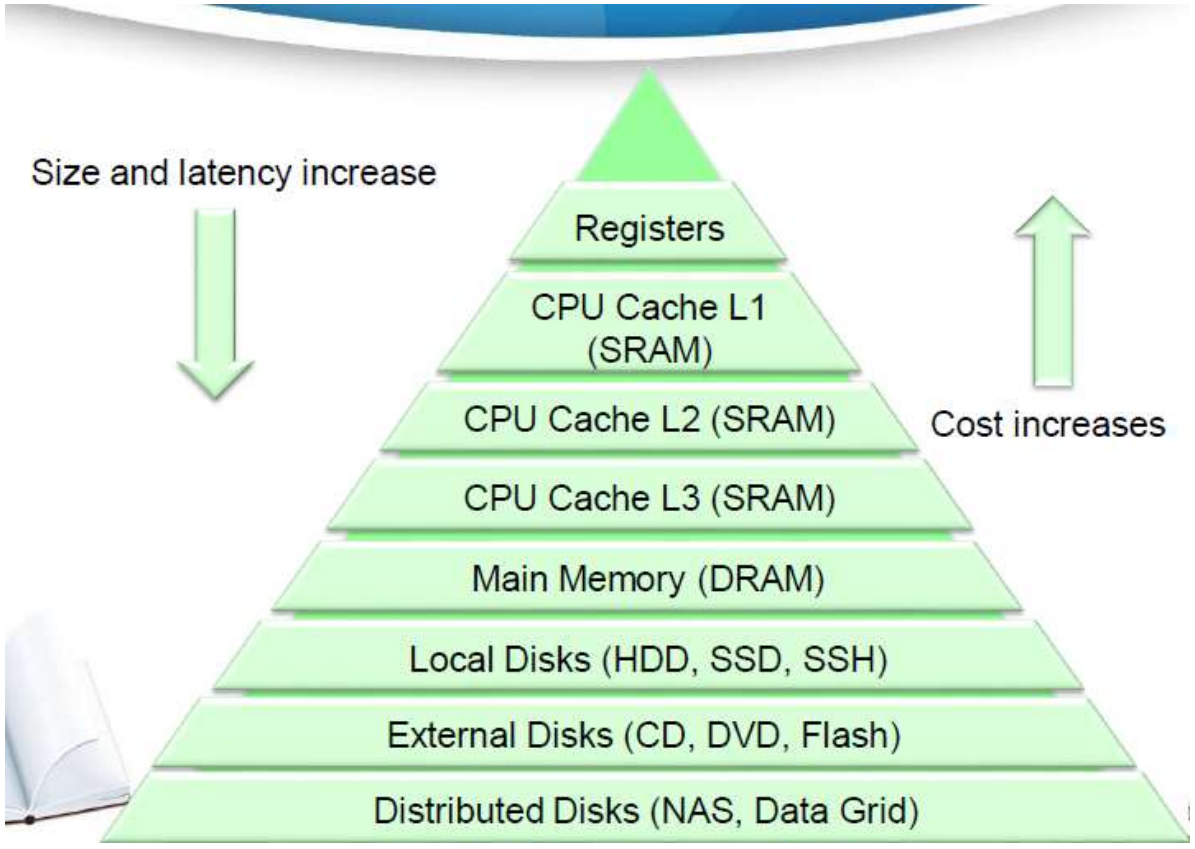
The OS keeps tracking the good functioning of the hardware devices. These structures are necessary to have a good repository for the I/O operations in execution

Operating systems
An overview

Memory hierarchy

Ultra-fast memory is expensive.

OS manages the memory hierarchy in order to gain a better resource utilization.



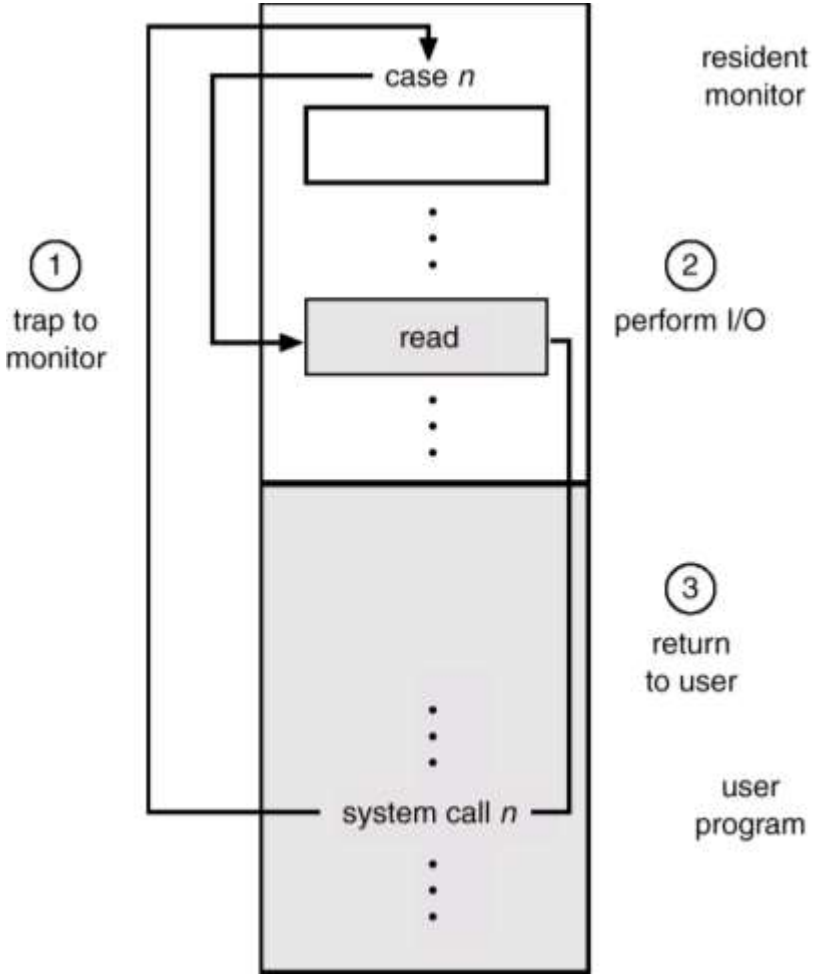
Operating systems
An overview

Protection

The main goal is to protect the OS and devices from malicious /ignorant users.

There are special instructions in **user mode (normal instructions)** or **supervisor mode (privileged instructions**).

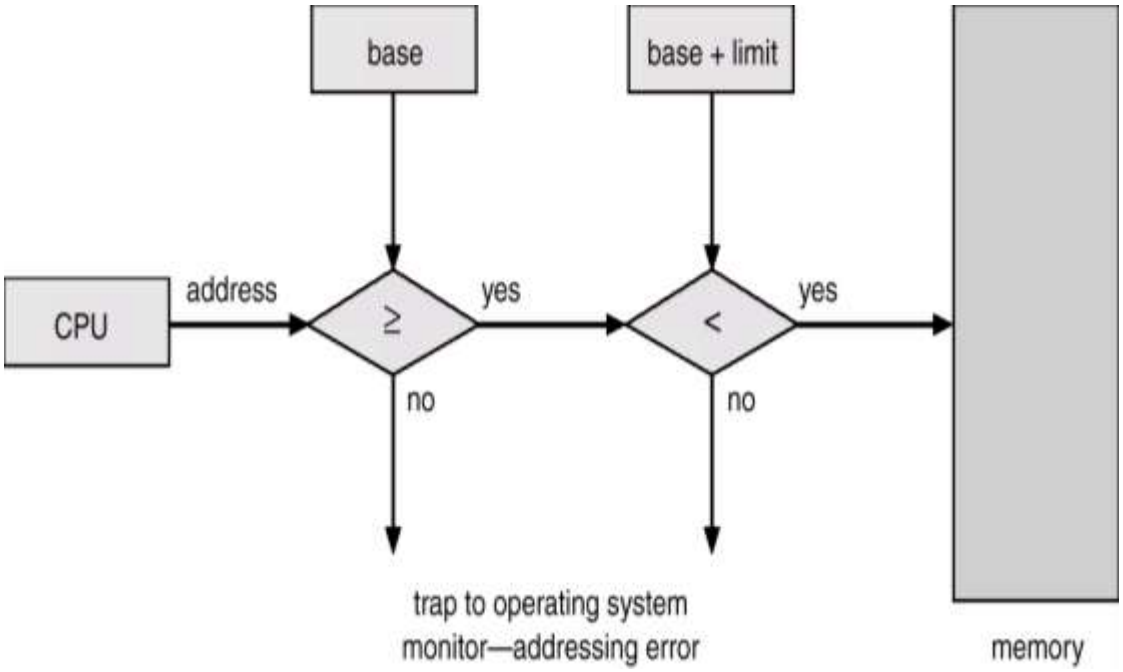
Concurrent threads may interfere one with another. This leads to the necessity for resource protection by using **user/supervisor modes**. For example, the I/O instructions are privileged, it can be run only in supervisor mode. The system calls are making the transfer from user mode into supervisor mode.



Operating systems
An overview

Protection

Memory A user program may access only it's logical memory. For example, it cannot modify supervisor code. It depends by a translation address schema.



Operating systems
An overview

Protection and security

CPU The system clock is preventing programs to use all the CPU time. This clock generates an interrupt that enables the OS to take control from an user program.

For multiprocessor and distributed systems the protection must include :

- **Resource sharing**
- **Multiprocessor architectures**
- **Cluster systems**

The systems are known as “distributed operating systems”.

UNIX – A (very) short history

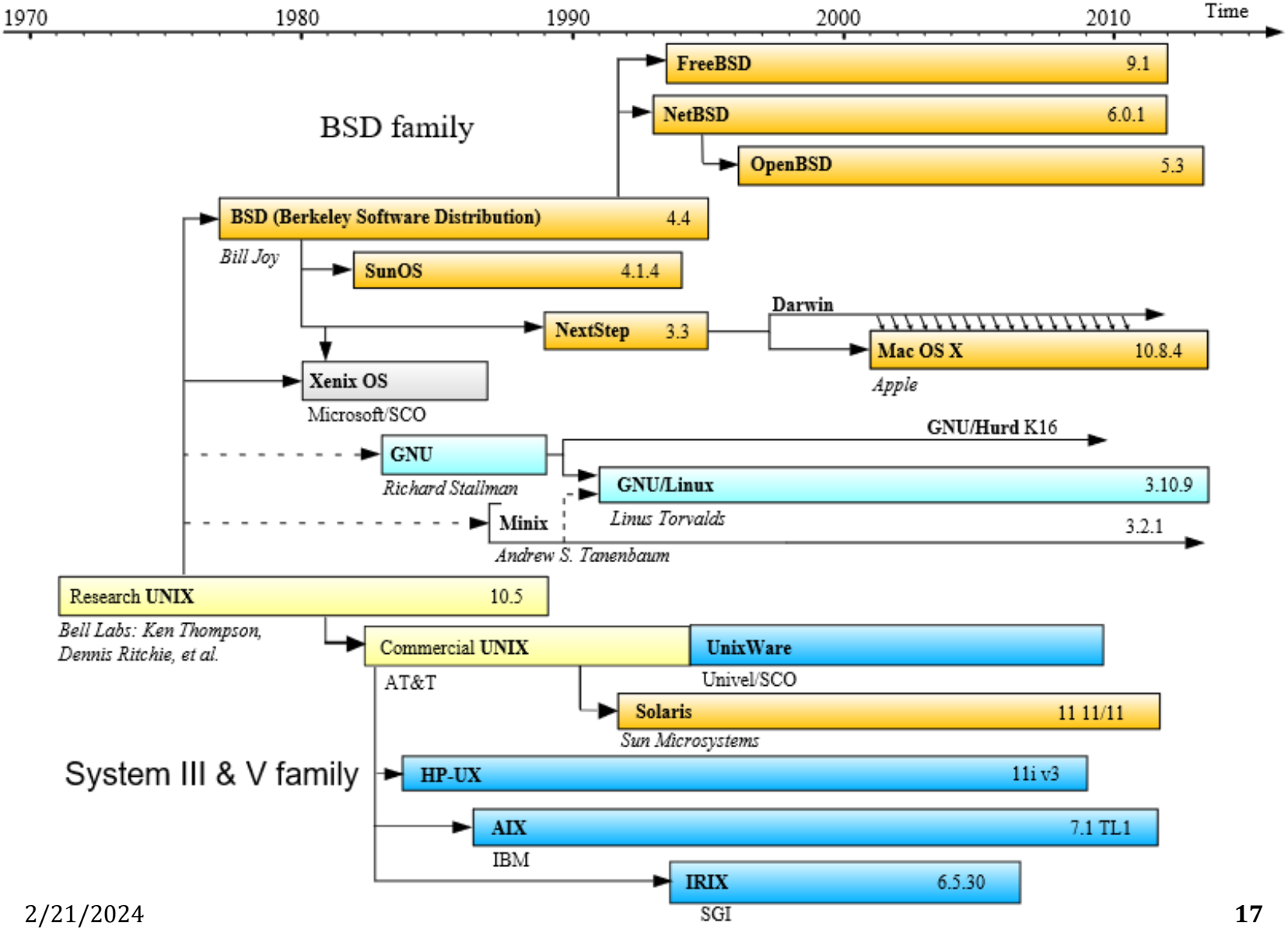
UNIX origins: AT&T Bell Laboratories + GE + MIT – SO Multics

Ken Thompson
Dennis Ritchie (+ Brian Kernighan)

They wrote together a travel space game running on a DEC PDP-7 machine. In 1969 they have decided to write an OS for PDP-7, named UNICS (Uniplexed Information and Computing Service).
1971 – C
1973 – UNIX written in C -> **portable system**

BSD UNIX – adding networking software
SunOS – BSD UNIX Version 4.2
AT&T System V (five)
1988 SunOS, AT&T System V and XENIX -> System V Release 4 (SVR4)

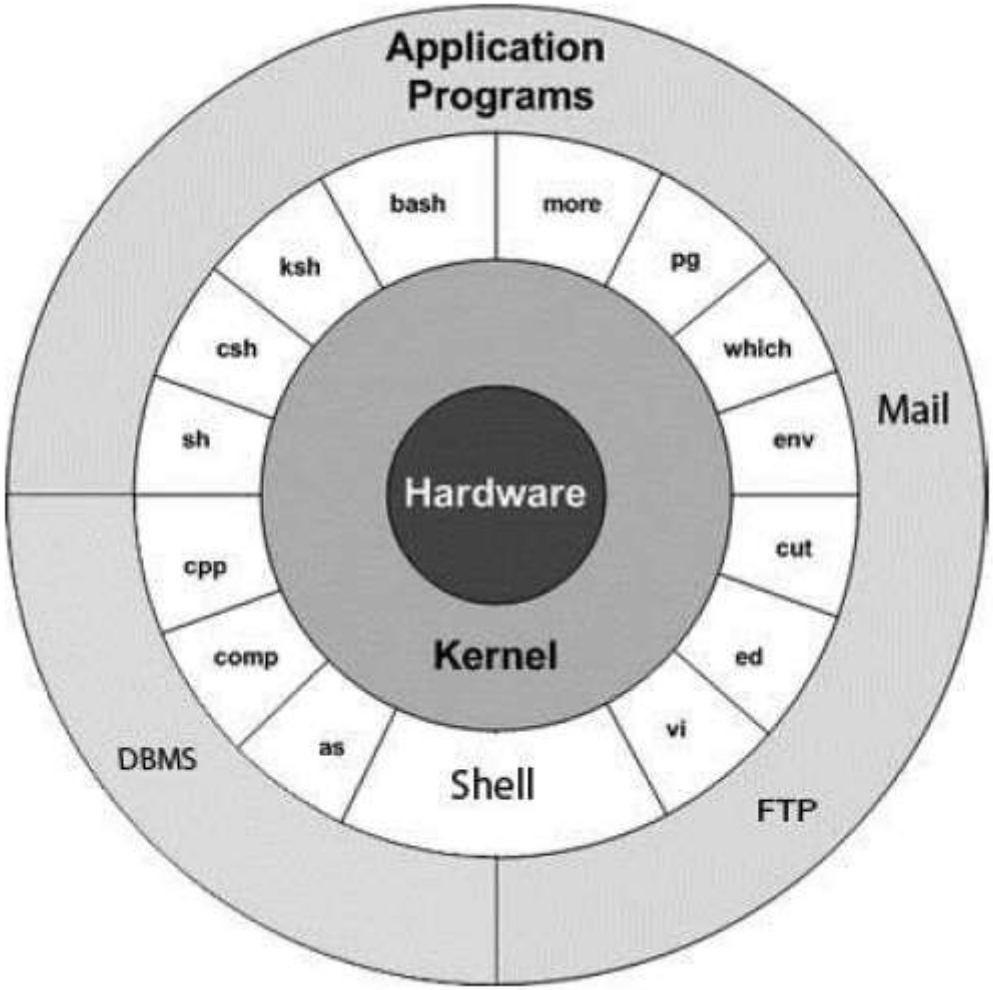
Unix “families” evolution



UNIX

UNIX strong points:
-Standards based
-Powerful, flexible, scalable, secured
-Good support from the hardware manufacturers
-Mature and stable OS
-Good integration with the TCP/IP network protocol stack
-Used on a large scale for critical applications

Key components:
-Kernel
-Shell
-File system
-Commands


Application Programs
bash
more
ksh
pg
csh
which
sh
env
Hardware
cut
cpp
comp
ed
Kernel
as
vi
Shell
DBMS
FTP
Mail

UNIX

- **Kernel:** The kernel is the heart of the operating system. It interacts with hardware and most of the tasks like memory management, tash scheduling and file management.

- **Shell:** The shell is the utility that processes your requests. When you type in a command at your terminal, the shell interprets the command and calls the program that you want. The shell uses standard syntax for all commands. C Shell, Bourne Shell and Korn Shell are most famous shells which are available with most of the Unix variants.

- **Commands and Utilities:** There are various command and utilities which you would use in your day to day activities. **cp**, **mv**, **cat** and **grep** etc. are few examples of commands and utilities. There are over 250 standard commands plus numerous others provided through 3rd party software. All the commands come along with various optional options.

- **Files and Directories:** All data in UNIX is organized into files. All files are organized into directories. These directories are organized into a tree-like structure called the filesystem.

UNIX

The kernel:

-The center of the OS- provides the capabilities for normal functioning of the computer

-It is an executable file loaded into memory when the computer is booting and it is called *unix* (System V) or *vmunix* (BSD) or … *linux !*

-After loading in memory, the kernel has the following main functions:

-Managing devices, memory, processes

-Controlling the information transmission between system programs and the hardware

UNIX

The kernel:

-Managing functions about:

-The *swap space*– very important forUnix, reserved for virtual memory space

-*Daemons*– there are programs (processes) with a specific function or they monitor the programs execution or device functioning. Daemons are special processes loaded into memory with the kernel and waiting for an event to happen. These processes help for a good functioning of the OS, offering various services. Daemons can be started or stopped anytime it is necessary. Equivalent with the services from Windows and NLM – Novell Netware.

-*File systems* – files' hierarchy, directories and sub-directories for structuring and managing the information on the HD.

UNIX

Shells: Bourne, Korn, C, Bash, TC (the *ps* command or *echo $SHELL*).

-Bourne /bin/sh – Stephen Bourne AT&T System V.2 UNIX (prompter: $)

-Korn /bin/ksh – Bell Labs (prompter: $)

-C Shell /bin/csh - Bill Joy from University of California at Berkeley (prompter: %)

-Bash – GNU (www.gnu.org)

File system

/bin – UNIX commands

/usr/bin – commands, system administration utilities, library routines

/usr/ucb – original commands for the BSD variant

/opt – optional applications or from another producers

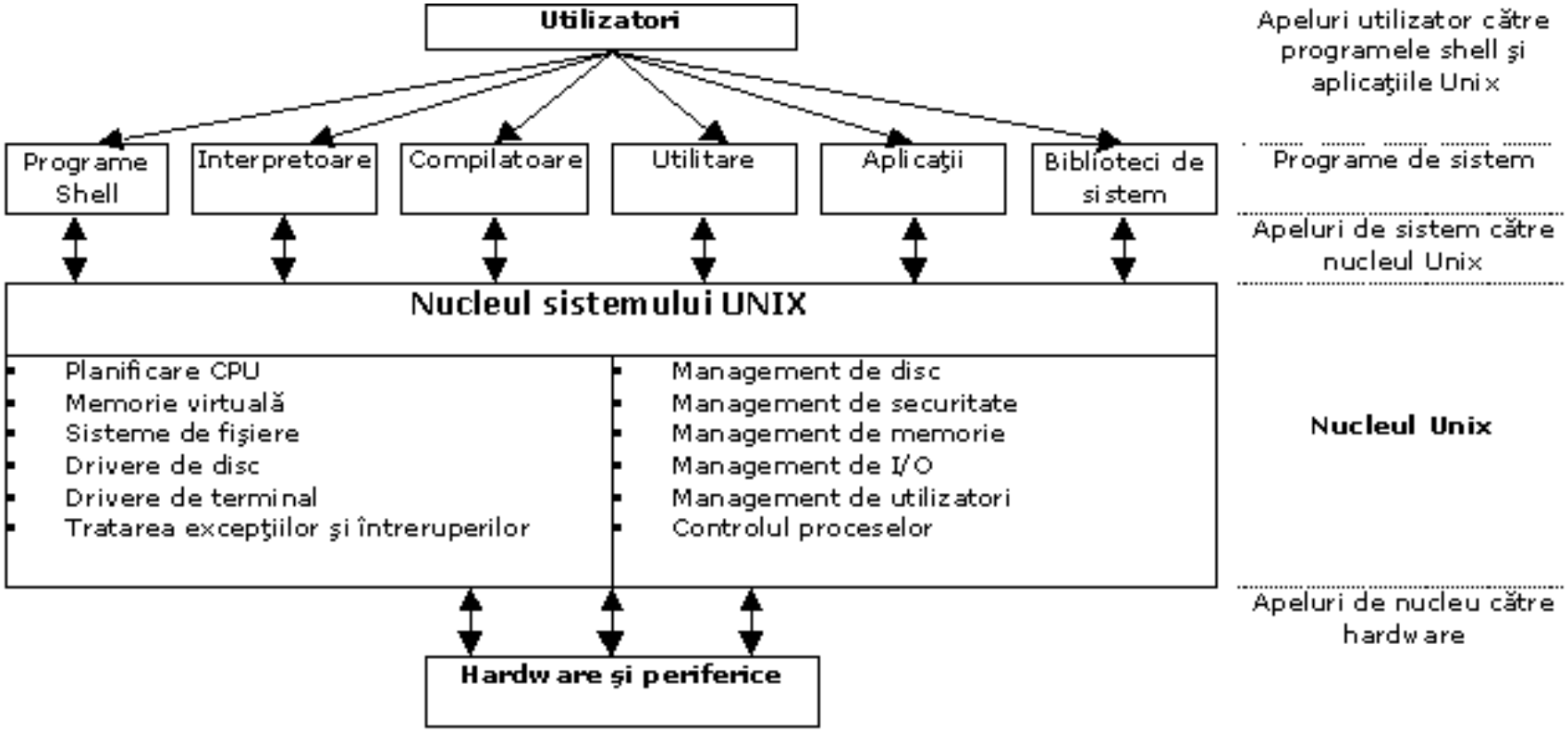
/etc – system administration files (see /etc/passwd)

UNIX

/dev – files pointing to names of equipment
(Solaris)/kernel – contains the basic files of the OS
/sbin – the basic executables used for booting and recovery + administration programs
/tmp – users' temporary files
/var – location for printers' jobs (print spooling) and error messages for mail system.

Commands – around 350-400 commands and utilities

UNIX General Architecture



Bibliography

- Sisteme de operare - R. Zota, A. Vasilescu, Ed. ASE, 2015
- Sistemul de operare Unix – Utilizare si programare shell, R. Zota, Ed. ASE, 2003
- Unix – R. Zota, Ed. ASE, 2004
- Other materials posted on http://zota.ase.ro/os