Chapter 1: Introduction

Operating system

@ptit.edu.vn

Posts and Telecommunications
Institute of Technology
Faculty of Information Technology 1

IT1 Faculty
PTIT

August 15, 2023

Contents

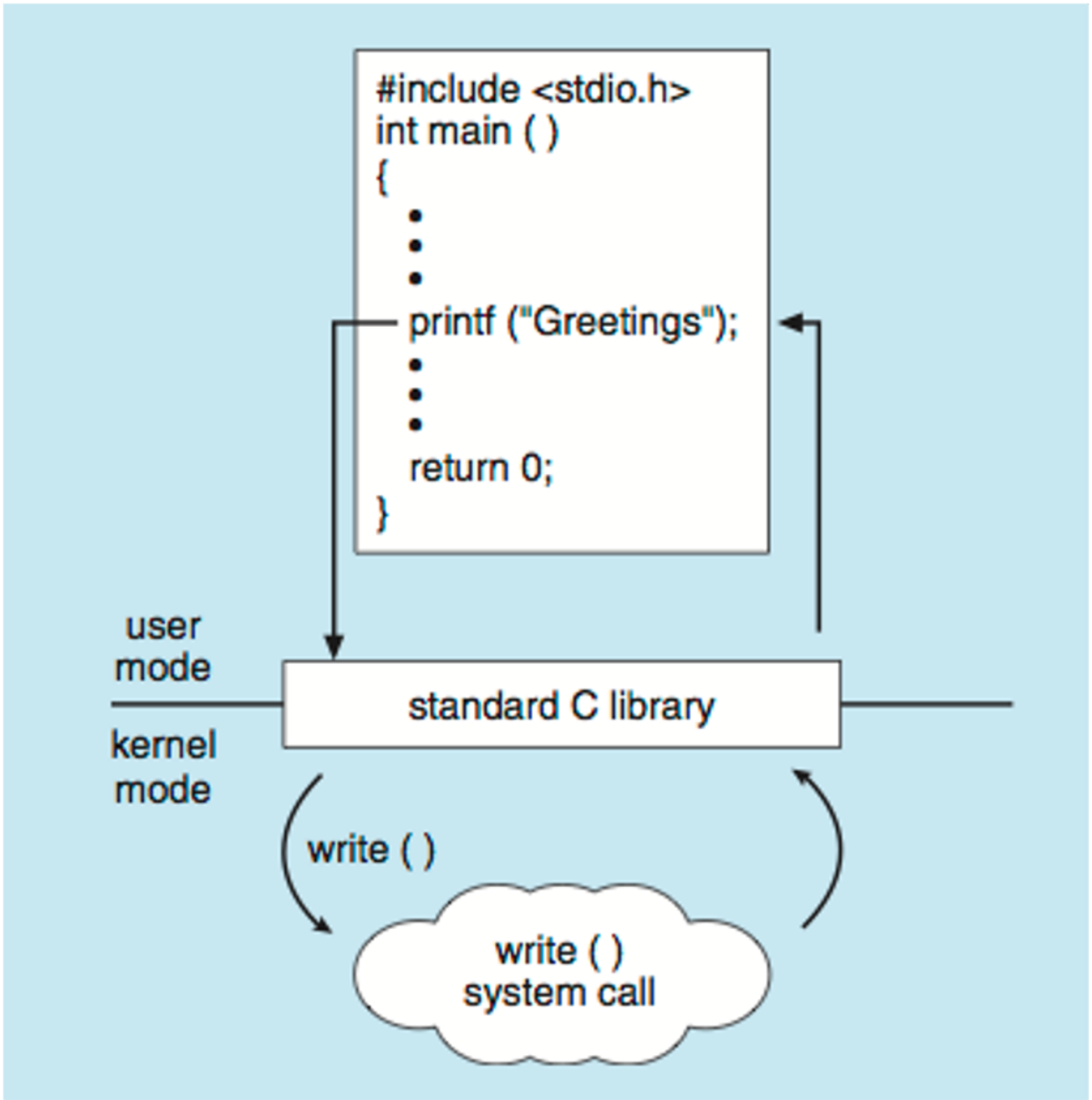
Target

1. Components of computer systems
2. Operating system concept
3. Services provided by the OS
4. Programming interface of the OS
5. Development process and some important concepts
6. OS structure
7. Some specific operating systems

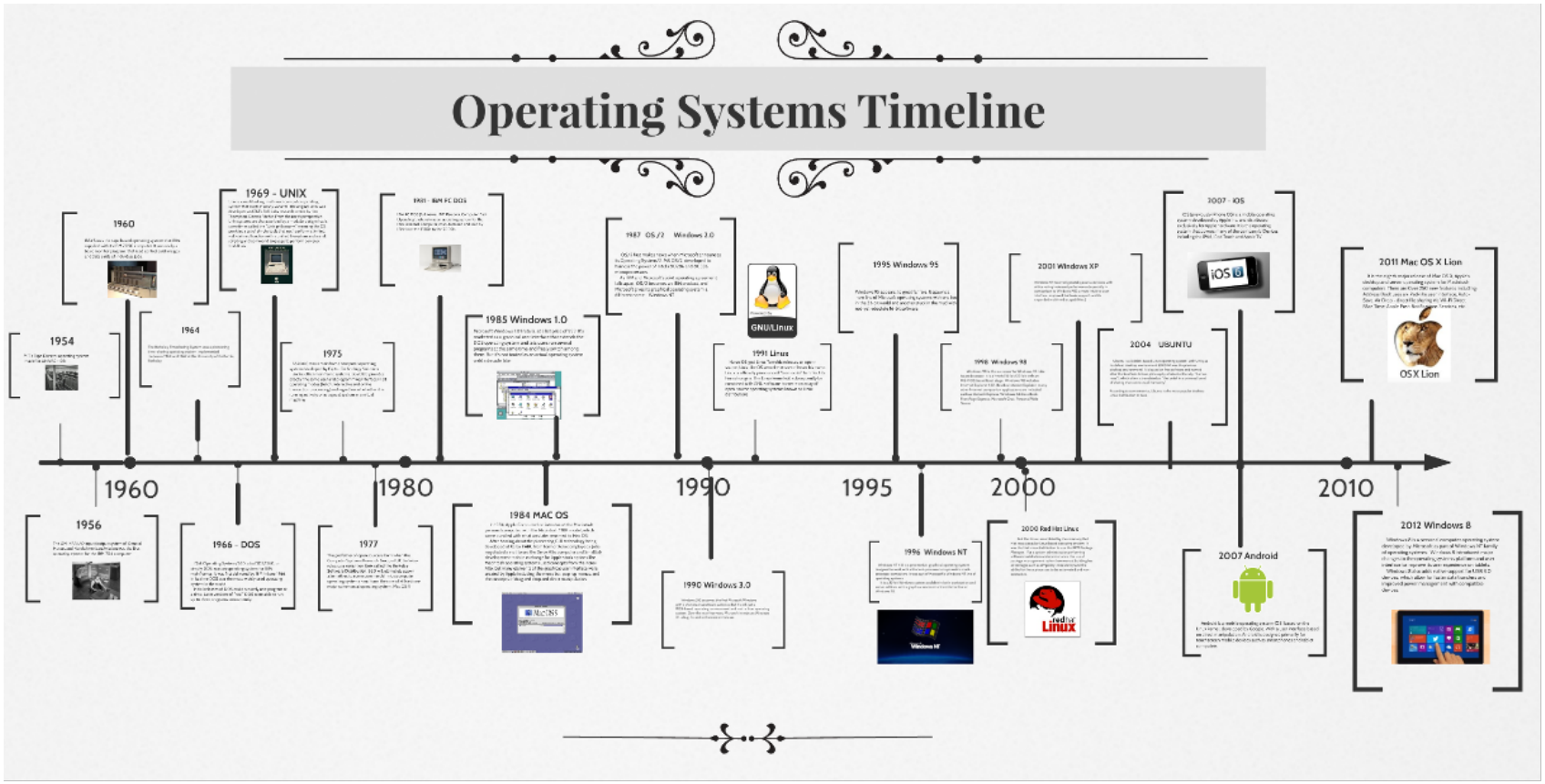
Programming interface of the OS

- In order for programs to use services, the Operating System provides a programming interface.
- This interface includes system calls that the program uses to request a service from the operating system.
- System calls: special commands that Application calls when it needs to ask the Operating System to do something.
- System calls are made through function libraries called system libraries. These functions will help the programmer call the corresponding system calls of the operating system.

Programming interface of the OS (cont.)



Development process



Operating systems timeline

Development process (cont.)

- Simple systems (40s-50s of the last century): computer processing speed was very low, input/output was done manually and was difficult.
- Program loading is done using switches, pre-soldered circuits, and punched covers. During this period, programmers interacted directly with the hardware.

Computers in this period did not have an operating system.

- Batch processing:
  - The program is divided into batches: including programs with similar requirements
  - The entire batch is loaded onto the magnetic tape and loaded into the machine to be processed one after another
- Monitor program: every time a batch program ends, the monitoring program automatically loads the next program into the machine and allows it to run

Development process (cont.)

Significantly reduces changeover time between two programs in the same batch

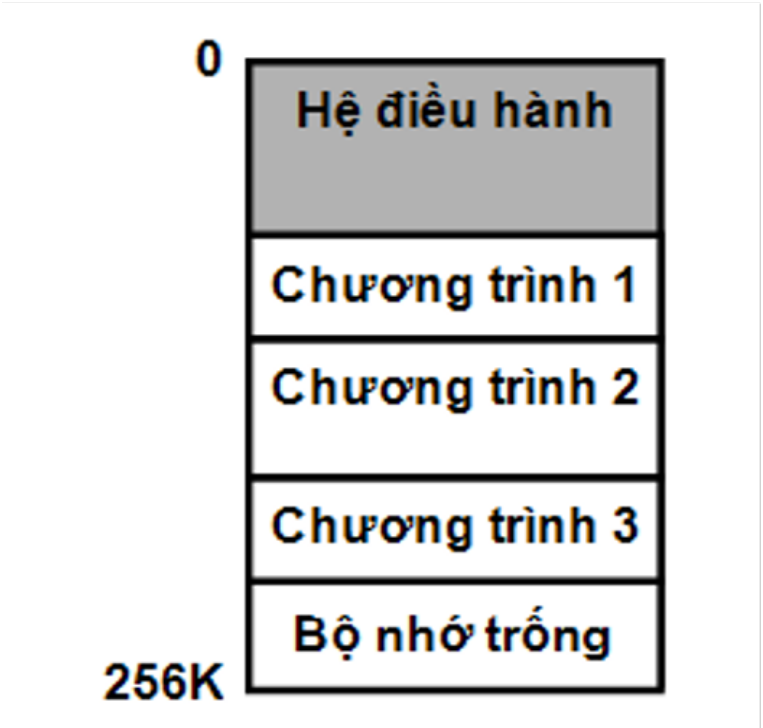
- Supervisor is the simplest form of Operating System :
  - Disadvantage: Every time there is an I/O request, the CPU must stop processing data to wait for the I/O process to finish. Because the I/O speed is much lower than the CPU speed, the CPU often has to wait for a long time.

  Lead to *low CPU performance.*

Development process (cont.)

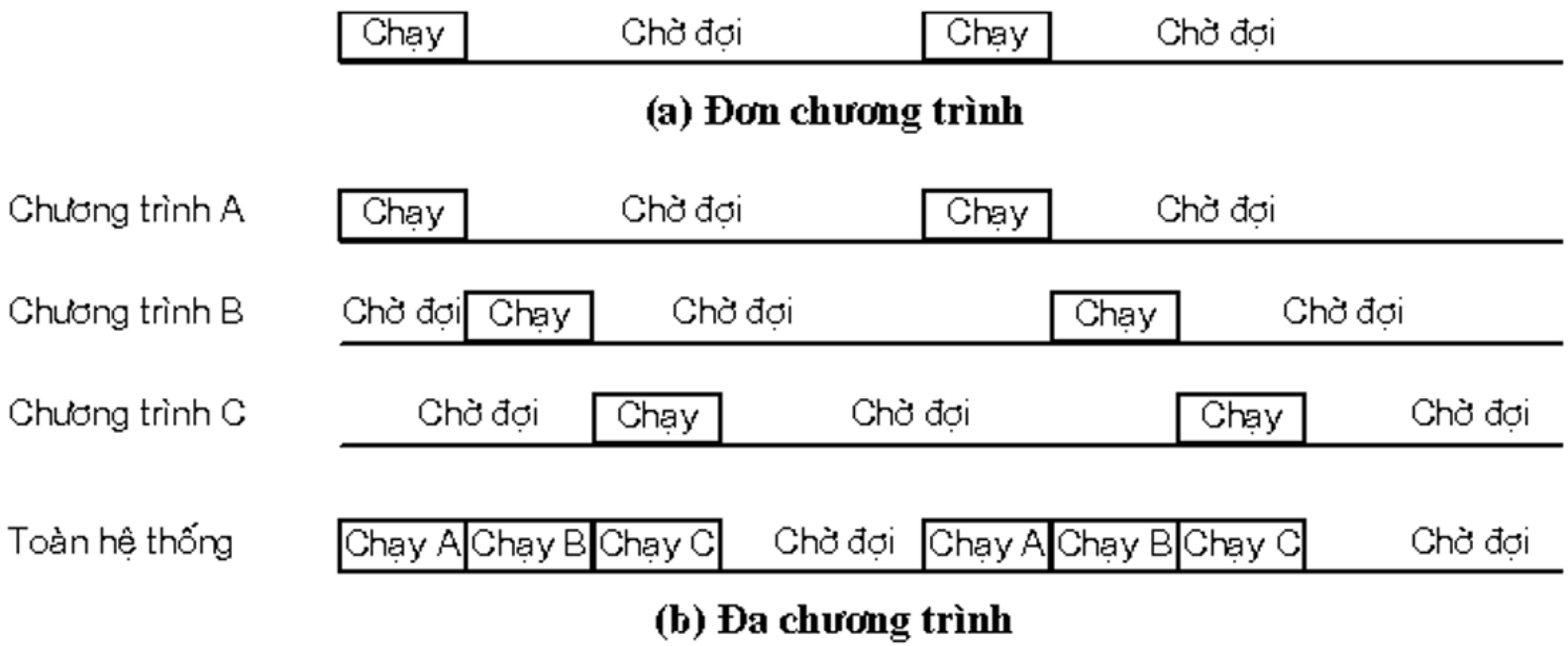
- Multi-program:
  - The system holds multiple programs in memory simultaneously
  - When a program must stop to perform I/O, the operating system switches the CPU to execute another program

Reduce CPU idling time



Development process (cont.)

- Multi-program:



- The waiting time of the CPU in multiprogram mode is significantly reduced compared to the single program case
- Operating systems are much more complex than single-program operating systems

Development process (cont.)

- Requires hardware support, especially interrupt I/O capabilities and DMA mechanisms

- Limitations of Multiprogramming:
  - Although multiprogramming allows efficient use of the CPU and other system resources, this technique does not allow users to interact with the system.
  - Later generation computers allowed the computer and the user to work directly through the screen and keyboard.
  - For these systems, the time from when the user types a command until the computer responds is relatively small.
  - Multiprogramming techniques cannot guarantee such short response times.

Development process (cont.)

Time Sharing (Multitasking):

- Time sharing can be considered as innovative multiprogramming
- The CPU takes turns performing different tasks in short periods of time called time quanta
- Switching between jobs takes place with high frequency and high CPU speed

All users have the feeling that the computer only executes its program
The CPU is shared among different users who interact directly with the system

Summary of Session 2

Chapter 1

- Programming interface of the OS development process

Next

- Operating System Components
- Kernel Operating System
- Some types of Operating System structures
- Some specific Operating Systems

An operating system is a complex software system made up of many components that perform different tasks or provide different services. The components perform the following tasks:

- Process management
- Memory management
- Input and output management
- Manage files and folders
- Network and distributed processing support
- User interface
- Utility programs and applications
- **Process management:**
  - A program in progress is called a process.

Programme	Process
- Is a static entity - Recorded as bits, bytes on disk	- Is a dynamic entity - Processing is in progress, calculation, given some resources: CPU time, memory

- **Process management:**
  - Create and delete processes (including user processes and system processes)
  - Suspend and restore suspended processes
  - Synchronization of processes (process scheduling, etc.)
  - Creates a communication mechanism between processes
  - Resolve deadlocks, for example when there are resource conflicts
  - Deadlock: is a program that needs resources but it is waiting and is not provided.
- **Memory management:**

- Management, supply and release
- Provide and release memory as required by processes
- Manage allocated memory space and free space
- Manage memory distribution between processes => ensure parallel running between multiple programs
- Creates virtual memory and maps virtual memory addresses to real memory

- **Managing input and output systems:**
  - Management through control programs
  - Simplify and increase the efficiency of information exchange between processes and input and output devices
- **Manage files and folders:**
  - Create and delete files and directories
  - Read and write files

- Map files and folders to external storage

- **Network and distributed processing support:**
  - Manage network devices
  - Supports communication protocols
  - Communication management, load balancing *Through control components, network communication.*

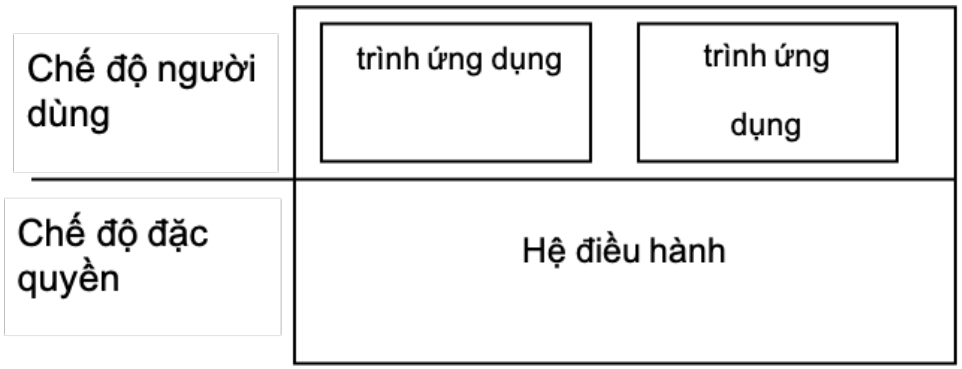
- **User interface:**
  - It is a command interpretation system
  - Helps computers understand and process user instructions and commands.
  - For example: bash for Linux, command for windows

- **Utility programs and applications**

- The OS consists of many components, but the importance of each component is different. There are indispensable components that are the basis for the entire system to operate, and some components of the OS provide less important functions.
  only loads important and indispensable components into memory called the kernel.
- The kernel is the core part, performing the most basic and important functions of the operating system and is constantly kept in memory.
- Kernel is responsible for managing system resources (communication between hardware and software components)

- Modern computers are often designed with two program execution modes.
  - The kernel runs in privileged mode - kernel mode: is the mode in which the program executes in which it has full access and control rights to the computer hardware.
  - User mode: programs executing in user mode have severely restricted access and use of the hardware.
- The distinction between kernel mode and user mode is intended to prevent Application from accidentally or intentionally performing operations that affect the system.

- **Monolithic structure**
  - All programs and data of the operating system share the same memory space. Therefore, it can be considered as a single block.
  - The operating system becomes a collection of procedures or subprograms
  - Advantage: fast, no time wasted between memory spaces
  - Disadvantages: Not safe: when any component has a problem, the entire system will not work; Not flexible and difficult to modify, adding or removing any city will affect the entire system. When there is an error, it is difficult to determine which city caused the error.

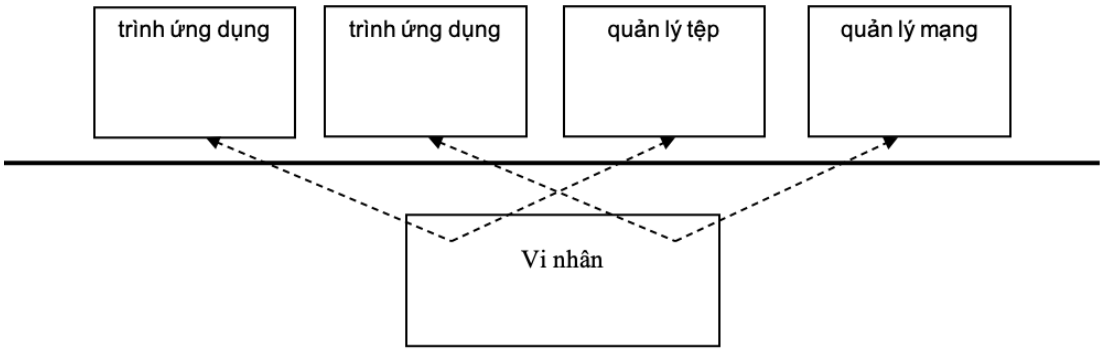


Hình :Cấu trúc nguyên khối

- **Layered structure**
  - Components are divided into layers that overlap each other
  - Each layer can only communicate with the layer above and below it
  - Each layer can only use services provided by the layer immediately below
  - Advantages: divided functions, easy to use, easy to fix errors
  - Disadvantages: difficult to design (determining the number of layers, dividing the functions of each layer), slower speed than monolithic structures

- **Micronucleus structure (micro kernel)**
  - The nucleus is small in size, containing only the most important functions
  - The remaining functions are placed in separate modules: run in privileged or user mode. When there is a request from the application, the kernel will pass it to the corresponding module to process and receive the results back. The kernel mainly acts as a communication intermediary.
  - Advantages: flexible, safe

- Disadvantage: slower speed compared to monolithic structure



Hình 1.5 Cấu trúc vi nhân

Some specific Operating Systems

- UNIX
  - is a multitasking operating system, first developed by Ken Thompson, Dennis Ritchie and Douglas McIlroy at AT & T Bell.
  - UNIX was researched in laboratories in 1969 and gradually improved, developed and became popular. Unix was first reprogrammed by Ken Thompson in the C language in 1973.
  - Creates high-level languages in operating systems
  - Creates a hierarchical file system
  - The Unix shell has inspired many subsequent command-line interpreters.
  - Helps the C programming language become more popular
  - Contributed to the launch of the free software movement

Some specific Operating Systems (cont.)

- MINIX (Từ mini-Unix)
  - is a Unix-like computer operating system based on micro-kernel architecture.
  - The initial version of MINIX was created by Andrew S. Tanenbaum for educational purposes such as illustration, training, and is available for free use.
  - MINIX is now developed as open source software.
- LINUX
  - In 1991 while studying in Helsinki - Finland, Linus Torvalds began to have the idea of an operating system.
  - Because he also noticed a limitation in Minix's license - it only allows the use of Minix in education only. He started writing the LINUX operating system developed from MINIX.

Some specific Operating Systems (cont.)

- MS-DOS
  - It is a product of Microsoft and was equipped with IBM's first PCs
  - To be able to run on PCs with limited resources, MS-DOS was built simpler and less functional
  - Many technical solutions in MS-DOS originated from UNIS such as the programming interface (system calls), directory hierarchy, command interpreter
  - Does not have functions such as security, network support, or multi-process support

Some specific Operating Systems (cont.)

- Windows NT (NT-new technology)
  - is a member of the new generation operating family such as Windows 2000, XP, Vista,7.
  - The first version was released in 1993 is an operating system that uses many advanced techniques in the field of operating systems that have been developed up to this point, including solutions taken from UNIX.
  - is a multitasking operating system that supports networks, has security functions, has a graphical interface in the form of a window and is used for PCs that require high stability.

Summary of Session 3

Chapter 1

- Operating System Components
- Operating System Kernel
- Some types of Operating System structures
- Some specific Operating Systems

Continued Chapter 2

- File concept
- File structure
- File access method
- Operations with files and directories

Questions and exercises

1. **Question 1**: What are the main functions of an Operating System?
2. **Question 2**: Based on the definition of operating system, can a Web browser be a component of an operating system?
3. **Question 3**: Does any computer system have an operating system? Why? Here, a computer system is broadly understood as any system with a processor and memory.
4. **Question 4**:One of the requirements for computing systems is the requirement for safety, that is, ensuring that processes do not violate resources without permission. Can a system achieve security requirements without distinguishing between user mode and privileged mode (kernel mode)? Please explain your answer by giving an example.