Міністерство освіти і науки України
Національний університет водного господарства та природокористування
Навчально-науковий інститут кібернетики, інформаційних технологій та інженерії
Кафедра обчислювальної техніки

04-04-284М

METHODICAL GUIDELINES

for laboratory classes in the course
«Computer architecture» (part 2)
for higher education students of the first (Bachelor's) level
in the educational degree programme «Computer Engineering»
of the field of study 123 «Computer Engineering»
of full-time and part-time forms of study

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ

до лабораторних робіт з навчальної дисципліни
«Архітектура комп'ютерів» (частина 2)
для здобувачів вищої освіти першого (бакалаврського) рівня за освітньо-професійною програмою «Комп'ютерна інженерія» спеціальності **123 «Комп'ютерна інженерія»**
денної та заочної форм навчання

Рекомендовано науково-методичною радою з якості ННІКІТІ
Протокол № 4 від 24.02.2025 р.

Рівне – 2025

Methodical guidelines for laboratory classes in the course «Computer architecture» (part 2) for higher education students of the first (Bachelor's) level in the educational degree programme «Computer Engineering» of the field of study 123 «Computer Engineering» of full-time and part-time forms of study. [Electronic edition] / Shatna A. V., Shatnyi S. V., Boichura M. V., Sydor A. I. – Rivne : NUWEE, 2025. – 40 p.

Методичні вказівки до лабораторних робіт з навчальної дисципліни «Архітектура комп'ютерів» для здобувачів вищої освіти першого (бакалаврського) рівня за освітньо-професійною програмою «Комп'ютерна інженерія» (частина 2) спеціальності 123 «Комп'ютерна інженерія» денної та заочної форм навчання. [Електронне видання] / Шатна А. В., Шатний С. В., Бойчура М. В., Сидор А.І. – Рівне : НУВГП, 2025. – 40 с.

Compilers: Shatna A. V., Senior Lecturer of the Department of Computer Engineering; Shatnyi S. V., PhD, Associate Professor of the Department of Computer Engineering; Boichura M. V., PhD, Associate Professor of the Department of Computer Engineering; Sydor A. I., PhD, Acting Head of the Department of Computer Engineering.

Укладачі: Шатна А. В., ст. викладач кафедри обчислювальної техніки; Шатний С. В., к.т.н., доцент кафедри обчислювальної техніки; Бойчура М. В., к.т.н., доцент кафедри обчислювальної техніки; Сидор А. І., к.т.н., в.о. завідувача кафедри обчислювальної техніки.

Responsible for the publication: Sydor A. I., PhD, Acting Head of the Department of Computer Engineering.
Відповідальний за випуск: Сидор А. І., к.т.н., в.о. завідувача кафедри обчислювальної техніки.

Head of the support group in the field of study
123 «Computer Engineering» Sydor A. I.
Керівник групи забезпечення спеціальності
123 «Комп'ютерна інженерія» Сидор А. І.

CONTENT

Introduction

Computer architecture is a science that focuses on the study of the design and organization of the components of computer systems. It involves examining the interaction between hardware and software, which enables the execution of various tasks, as well as the principles of structuring these elements to achieve high performance, efficiency, and reliability. Key components studied in this field include the central processing unit (CPU), memory hierarchy, input/output systems, and data processing systems.

The main goal of computer architecture is to create systems capable of efficiently executing instructions, processing information, and interacting with peripheral devices. It combines theoretical foundations and practical design principles that help understand the operation of modern computing systems. Different types of architectures, such as the von Neumann and Harvard architectures, are analyzed for their impact on performance and the selection of optimal solutions during the design process.

As technology advances, computer systems are becoming increasingly complex. The emergence of multi-core processors, parallel computing, and specialized hardware accelerators is driving the rapid evolution of computer architecture, requiring specialists not only to understand classical principles but also to adapt to the latest innovations. Studying this discipline will provide a solid understanding of the basic concepts, design principles, and current trends in computer architecture, enabling the development and analysis of efficient computing systems for various applications.

Вступ

Архітектура комп'ютерів є наукою, що займається дослідженням проєктування та організації компонентів комп'ютерних систем. Вона охоплює вивчення взаємодії між апаратним і програмним забезпеченням, що забезпечує виконання різноманітних завдань, а також принципи структурування цих елементів для досягнення високої продуктивності, ефективності та надійності. Основними об'єктами вивчення є такі компоненти, як центральний процесор (ЦП), ієрархія пам'яті, системи вводу/виводу та обробки даних.

Головною метою архітектури комп'ютерів є створення систем, здатних ефективно виконувати інструкції, обробляти інформацію та взаємодіяти з периферійними пристроями. Це поєднання теоретичних основ і практичних принципів проєктування, що дозволяють зрозуміти роботу сучасних обчислювальних систем. Аналізуються різні типи архітектур, такі як архітектура фон Неймана і Гарварда, та їх вплив на ефективність і вибір оптимальних рішень під час проєктування.

Зі зростанням технологій комп'ютерні системи стають дедалі складнішими. Виникнення багатоядерних процесорів, паралельних обчислень і спеціалізованих апаратних акселераторів веде до швидкої еволюції архітектури комп'ютерів, що вимагає від фахівців не лише знання класичних принципів, а й здатності адаптуватися до новітніх досягнень. Вивчення цієї дисципліни допоможе освоїти основні концепції, принципи проєктування та сучасні тенденції в архітектурі комп'ютерів, що дозволить розробляти та аналізувати ефективні обчислювальні системи для різноманітних застосувань.

Laboratory work 6

Study and installation of a virtual machine

Goals

Getting acquainted with the principles of virtualization, exploring the key functions of a virtual machine, and gaining hands-on experience in configuring and installing it on various operating systems to optimize resource usage and ensure environment isolation.

Procedure:

1. Familiarize yourself with the theoretical information.
2. Complete the practical tasks.
3. Prepare a report on the laboratory work.
4. Answer the control questions.

Theoretical part

During testing, virtual machines and emulators are often essential, as they greatly simplify the process by enabling the testing of applications and websites across various devices and operating systems.

They are especially crucial when testing on a large variety of devices, as it is impractical to have all possible device variations, and it is often not cost-effective to constantly maintain even the available devices. Using virtual machines and emulators allows potentially risky actions to be performed safely. Additionally, they help reduce time costs, streamline the setup and backup of different testing environments, and make managing them more convenient. The ease of using virtual machines is also evident when teams in different locations collaborate, as they can simply access a virtual machine instead of needing a physical device.

However, there are some challenges and potential drawbacks to using virtual machines and emulators. Emulation does not fully replicate real devices, so real devices are still necessary for precise final testing, especially for key models. Not all devices can be emulated accurately. Driver conflicts may arise, and virtual machines demand significant resources, run slower, and have certain limitations, making them unsuitable for performance testing as results can be skewed.

In this laboratory work, we will explore the most commonly used virtual machines and emulators useful for testing and discuss their advantages and disadvantages.

VMWare

VMware Workstation is an efficient and convenient virtual machine for professional use.

Advantages:

- VMware Workstation Player is available for free for non-commercial and educational use.
- Simple and user-friendly interface.
- Installing a new operating system is significantly easier compared to a standard installation on a computer.
- It allows you to take snapshots of the operating system, making it easy to restore previous states.
- High stability and reliability.
- Efficient and fast performance.
- Ability to protect virtual machines with a password.
- Quality support for 3D graphics.

Disadvantages:

- VMware Workstation Player is paid for commercial use.
- VMware Workstation Pro is only available in the paid version.
- Some applications are required to work with different operating systems.

VirtualBox

VirtualBox is one of the most popular and accessible virtual machines.

Advantages:

- VirtualBox supports a wide variety of operating systems, both for installing VirtualBox itself and for the guest operating systems it can run.
- It creates snapshots of the operating system (Snapshots), which allow easy restoration of previous states.
- It is available for free with open-source code, under the GPLv2 license.

Disadvantages:

- VirtualBox is less performant compared to some other virtual machines, especially paid ones.
- Bugs, various issues, freezes, and crashes are relatively common.
- Limited support for 3D graphics.
- The interface is relatively complex, at least compared to paid virtual machines.

Hyper-V

Hyper-V was created as a replacement for Microsoft Virtual PC.

Advantages:

- It comes bundled with many versions of Windows 10 (Pro, Enterprise, and Education).
- It provides good support for installing guest operating systems, such as various older versions of Windows.
- It also supports the installation of guest operating systems like Linux and FreeBSD.

Disadvantages:

- Not designed to run on earlier versions of Windows.
- macOS installation is not supported.
- The interface is less user-friendly compared to VMware and VirtualBox.

Boot Camp

Boot Camp is a utility for Mac computers that allows you to install Windows.

Advantages:

- It is provided with Mac computers.

Disadvantages:

- It is specifically designed to run Windows as a guest operating system.
- It does not work with some devices, such as motion sensors.
- Issues may occur with resizing partitions.

Parallels Desktop

Parallels Desktop is a virtual machine used to run Windows on computers with the Mac operating system.

Advantages:

- Can use existing data from Boot Camp.
- Supports various guest operating systems, including Windows, Linux, different versions of macOS, and others.

Disadvantages:

- Works only on macOS.
- The program is paid, though there is a free 14-day trial version available.

Procedure for completing the task:

6.1.Download VirtualBox from https://www.virtualbox.org/wiki/Downloads. Select the version for your Operating System.

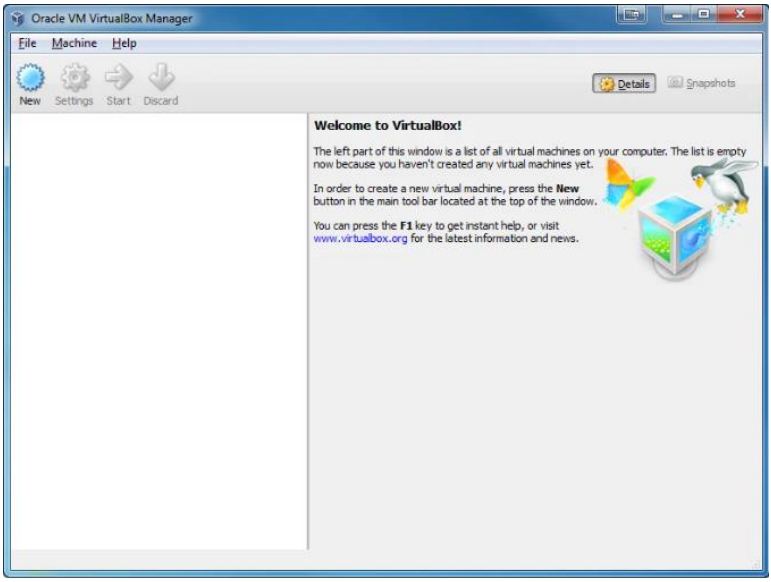


Figure 6.1 – Download and Install VirtualBox

6.2. Create a Virtual Machine. Start VirtualBox and Click on 'New' in the menu. Enter the Name of your VM. This is how you will identify it in VirtualBox so name it something meaningful to you. Select Type and Version. This depends on what OS you are installing.

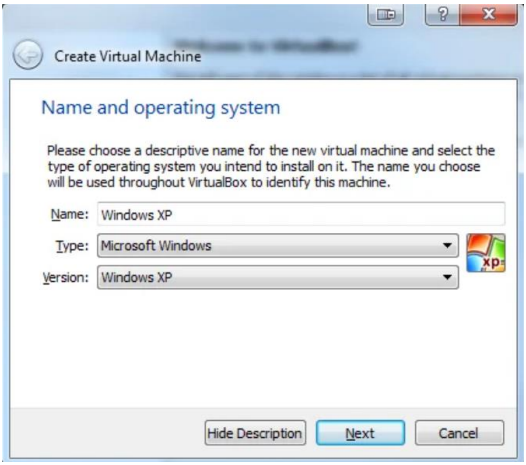


Figure 6.2 – Create a Virtual Machine

6.3. This depends on how much memory you have on your host computer. Never allocate more than half of your available RAM. If you are creating a Windows VM I recommend at least (1-2 GB) If you are creating a Linux VM I recommend at least (512 MB)

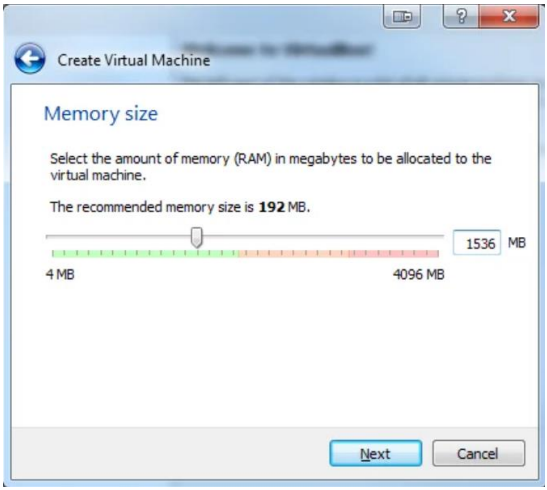


Figure 6.3 – Allocate Memory

6.4. If you already have an existing VM that you want to add select "Use an existing Virtual hard drive file." Otherwise select "Create a virtual hard drive now."

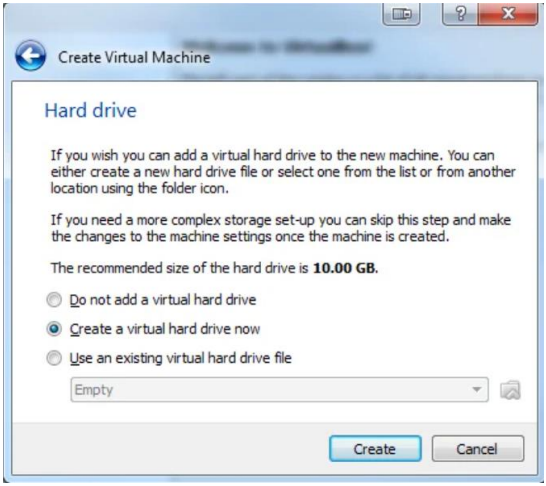


Figure 6.4 – Setup the Hard Drive

6.5. Select 'VDI.' This is usually the best option. The VM will be stored in a single file on your computer with the .vdi extension.

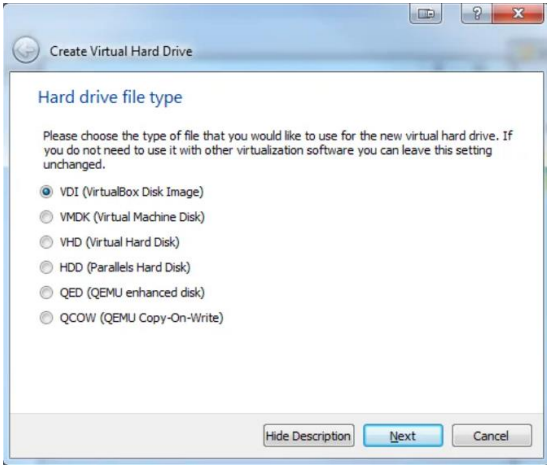


Figure 6.5 – Select Hard Drive File Type

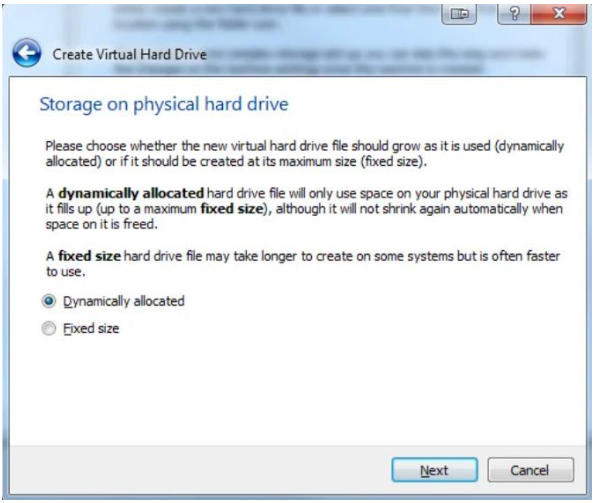


Figure 6.6 – Select Storage on Physical Hard Drive

6.6. By default, Virtualbox selects the minimum size you should choose. Depending on what you want to do with the VM you may want to select a bigger size.

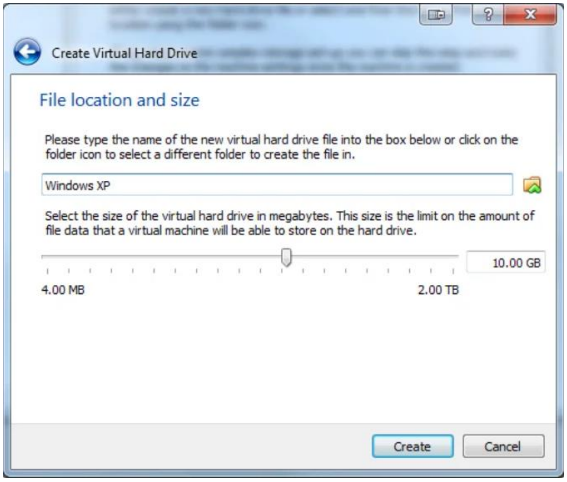


Figure 6.7 – Setup File Location and Size

6.7. Double click on your newly created VM (It will be on the left hand side and will have the name you gave it in Step 2). Browse to your installation media or .iso file. Finish installation. The actual installation process will depend on which OS you are installing.

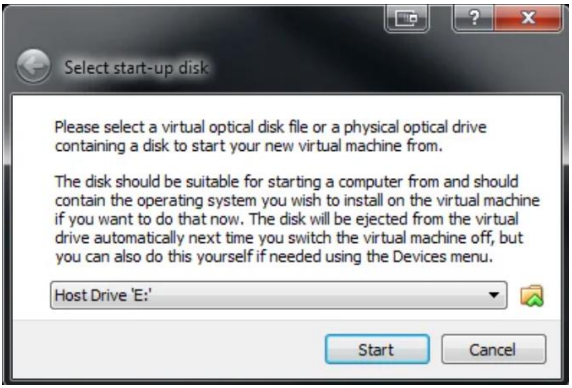


Figure 6.8 – Install the Operating System

6.8. Guest additions add more functionality to your VM, including the option to make the VM fullscreen.

Figure 6.9 – Install Guest Additions

6.9. Boot in Safe Mode. Instructions for Windows XP. When you have booted in Safe Mode, click Devices --> Install Guest Additions. Follow the Prompts and Install.

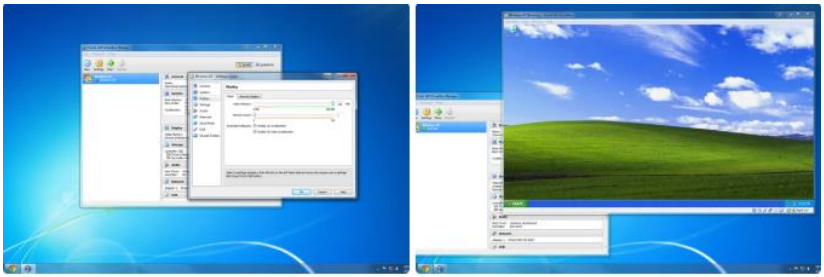


Figure 6.10 – Update Settings for 2D and 3D Acceleration

6.10. Go to the Display settings for your VM. Check both 2D and 3D acceleration. Increase Video Memory (I recommend 128 MB). Start your VM.

Questions for independent training

1. What is a virtual computer?
2. What is the purpose of a virtual computer?
3. Explain the concept of the HOST (primary) operating system.
4. Explain the concept of the GUEST (secondary) operating system.
5. Why is a virtual computer practically indistinguishable from a real one?
6. What is meant by an image of a CD, floppy disk, or hard drive?

Laboratory work 7

Study of the file system and system resources through the command line

Goals

Getting acquainted with the principles of virtualization, exploring the key functions of a virtual machine, and gaining hands-on experience in configuring and installing it on various operating systems to optimize resource usage and ensure environment isolation.

Procedure:

1. Familiarize yourself with the theoretical information.
2. Complete the practical tasks.
3. Prepare a report on the laboratory work.
4. Answer the control questions.

Theoretical part

1. **File System.** A file system is a method employed by an operating system to manage and organize data on storage devices such as hard drives, SSDs, and USB drives. It determines how files are stored, identified, accessed, and arranged into a hierarchical structure of directories (folders).

Common file systems include:

- FAT (File Allocation Table): An older file system used in previous Windows versions and on USB drives.

- NTFS (New Technology File System): Used in modern Windows versions, it supports large data volumes, provides greater flexibility, access control, and journaling.

- EXT (Extended File System): Predominantly used in Linux, with modern versions like EXT4 offering improved performance and support for large file systems.

- APFS (Apple File System): Used on Apple devices such as Macs, iPhones, and iPads.

2. **Command Line.** The command line is a text-based interface that allows users to interact with the operating system by inputting commands. It serves as an alternative to the graphical user interface (GUI), enabling users to manage files, run programs, execute system commands, and configure the system.

Advantages of using the command line:

- Speed: It allows multiple tasks to be executed simultaneously.

- Flexibility: Through commands, users can adjust the system and work with all its components.

- Automation: The command line enables the automation of tasks using scripts.

The two primary command line types are:

- Windows Command Prompt (CMD): A command-line interface built into Windows operating systems.

- Unix/Linux Shell: In Unix-based systems like Linux, the shell is used, with bash being the most widely used.

3. Key Commands for File System Operations.

- dir (Windows) / ls (Linux/Mac): Display the contents of the current directory.

- cd: Change the current directory. For example, `cd /home/user` (Linux) or `cd C:\Program Files` (Windows).

- mkdir: Create a new directory.
- rmdir: Delete an empty directory.
- del (Windows) / rm (Linux/Mac): Delete files.
- copy (Windows) / cp (Linux/Mac): Copy files and directories.
- move (Windows/Linux/Mac): Move files and directories.
- rename (Windows) / mv (Linux/Mac): Rename a file or directory.
- attrib (Windows) / chmod (Linux/Mac): Change file permissions.

4. System Resource Monitoring via Command Line.

System resources encompass memory, CPU time, disk space, and other components necessary for executing tasks. Monitoring these resources is crucial for ensuring the smooth operation of the system.

Important commands for monitoring system resources:

- tasklist (Windows) / top (Linux/Mac): Display active processes and their resource usage (CPU, memory).
- taskkill (Windows) / kill (Linux/Mac): Terminate processes.
- free (Linux/Mac) / systeminfo (Windows): Check memory usage.
- df (Linux/Mac) / chkdsk (Windows): Check disk space usage.

- vmstat (Linux): Show system resource statistics (memory, CPU, disk usage).

- uptime (Linux/Mac) / systeminfo (Windows): Check system uptime and load.

5. Event Logs and System Notifications.

Operating systems maintain logs that track critical events such as errors, warnings, and informational messages. These logs enable the monitoring of system health and provide insights into potential issues.

- eventvwr (Windows): View event logs on Windows, including errors, warnings, and informational messages.

- dmesg (Linux): Display kernel messages regarding the current system status and events.

- journalctl (Linux): Access system logs for various Linux components.

Procedure for completing the task:

7.1. Part 1: Working with the File System via Command Line

7.1.1. Viewing the contents of a directory

• On Windows, use the dir command to view the contents of the current directory.

• On Linux/Mac, use the ls command.

7.1.2. Changing the current directory

• Use the cd command to move between directories. Example:

• *cd /home/user* (Linux)

• *cd C:\Program Files* (Windows)

7.1.3. Creating a new directory

• On Windows: *mkdir NewFolder*

• On Linux/Mac: *mkdir NewFolder*

7.1.4. Deleting an empty directory

• On Windows: *rmdir EmptyFolder*

• On Linux/Mac: *rmdir EmptyFolder*

7.1.5. Creating, deleting, and moving files

- On Windows:

• Create a file: *echo Hello World > example.txt*

• Delete a file: *del example.txt*

• Move a file: *move example.txt C:\NewFolder*

- On Linux/Mac:

• Create a file: *echo Hello World > example.txt*

• Delete a file: *rm example.txt*

• Move a file: *mv example.txt /home/user/NewFolder*

7.1.6. Changing file permissions

• On Windows: *attrib +r example.txt*

• On Linux/Mac: *chmod 755 example.txt*

7.2. Part 2: Monitoring System Resources via Command Line

7.2.1. Viewing CPU and memory usage

• On Windows: *tasklist*

• On Linux/Mac: *top*

7.2.2. Viewing memory usage

• On Windows: *systeminfo*

• On Linux/Mac: *free -h*

7.2.3. Checking disk space usage

• On Windows: *chkdsk*

• On Linux/Mac: *df -h*

7.2.4. Viewing system resource usage statistics

• On Linux: *vmstat*

7.2.5. Viewing system uptime

• On Windows: *systeminfo*

• On Linux/Mac: *uptime*

7.3. Part 3: Event Logs and System Messages

7.3.1. Viewing event logs in Windows

• Enter the *eventvwr* command to view event logs (errors, warnings, informational messages).

7.3.2. Viewing system messages in Linux

• Enter the *dmesg* command to view system messages.

7.4. Part 4: Conclusion

7.4.1. After executing all commands, draw conclusions regarding the file system and system resource monitoring.

7.4.2. Evaluate the effectiveness of the command line for system administrator tasks.

Questions for independent training

1. What is a file system and why is it essential for data storage on a computer?

2. How does the structure of the file system affect the organization and management of files?

3. What are the differences between FAT, NTFS, EXT, and APFS file systems? When would you use each one?

4. What are the commands used to move, rename, and delete files in Windows and Linux/Mac? Provide examples for each.

5. What is the purpose of the attrib (Windows) or chmod (Linux/Mac) command, and how is it used to modify file permissions?

6. How can you check the available disk space in both Windows and Linux/Mac? What are the commands for this?

7. How do you check the system's uptime and its load on Linux and Windows? What is the importance of this information for system management?

Laboratory work 8

Memory Interfaces and Direct Memory Mapping

Goals

Learn to solve problems related to memory interfaces, learn to solve problems related to direct memory mapping. Master the methodology for determining key values.

Procedure:

1. Familiarize yourself with the theoretical information.
2. Complete the practical tasks.
3. Prepare a report on the laboratory work.
4. Answer the control questions.

Theoretical part

Computer memory is classified into two types: internal and external. Internal memory is made up of tiny cells, each with a unique address or identifier. Information is stored in memory by assigning it a specific address. To retrieve this information, the computer "reads" the cell and transfers its contents to the processor. The size of a single memory cell is known as a word. For personal computers, a word is usually 16 bits (or 2 bytes). A byte is equal to 8 bits. Large computers typically use words ranging from 32 to 128 bits (4 to 16 bytes), while minicomputers use words of 16 to 64 bits (2 to 8 bytes). Microcomputers often use words of 8, 16, or 32 bits (1, 2, or 4 bytes).

There are two primary types of internal memory: Random Access Memory (RAM) and Read-Only Memory (ROM). RAM is fast: the microprocessor can access it within 10-20 nanoseconds. Commercial RAM modules can store up to 256 MB (1 MB equals 1,048,576 bytes). RAM is reliable and can perform billions of operations over many years. However, RAM only retains data temporarily; once the power is turned off, it loses all information. RAM stores programs and data while the computer is operating. It is often considered temporary storage since it retains data only while the computer is powered on or until the reset button is pressed. Before shutting down or resetting, any changed data must be saved to a permanent storage device (like a hard drive). Upon reboot, this information can be reloaded into RAM.

The term "RAM" refers not only to the physical chips that make up the memory device but also to concepts such as logical mapping and placement. Logical mapping describes how memory addresses are represented on the actual chips, while placement refers to how data and instructions are organized at specific memory addresses.

The central processor (CPU) is connected to RAM. The primary memory component is essential for holding the data and programs that run in the CPU. In modern computers, RAM, like

solid-state memory, is directly connected to the CPU through the memory bus, also known as the address bus. In addition to RAM, there is cache memory, a small portion of memory used by the CPU to reduce access times and enhance performance. Cache memory helps improve CPU performance, which in turn enhances the overall computer performance. RAM is generally the most crucial part of a computer's memory and is made from integrated semiconductor chips.

The more RAM a computer has, the greater its ability to store and process programs and data. To expand RAM, additional memory (Expanded Memory) on extra boards or extended memory (Extended Memory) placed directly on the motherboard is used. When working with expanded memory, the processor treats data as if it were in regular RAM (up to 1 MB), but the data is redirected to expanded memory on an additional board, which can have several megabytes of capacity. To use extended memory, the processor switches from real mode to protected mode.

ROM, on the other hand, stores data permanently. ROM is ideal for tasks that involve repeatedly executing the same instructions. Although ROM is generally slower than RAM, it is non-volatile, meaning it retains its contents even without power. Not all ROMs are fully permanent; some have semi-permanent

memory that keeps data until it is erased and rewritten. Erasure can occur through methods such as exposure to high-intensity ultraviolet light, or other techniques used in modern memory chips that support erasure and rewriting.

External memory is usually located outside the main computer system. Since it works more slowly than internal memory, it is primarily used for storing information that is not urgently needed. To access external memory, the computer typically transfers the necessary data from external storage to internal memory. Because internal memory is limited, computer designers aim to maximize the use of external memory for storing large amounts of data.

Procedure for completing the task:

1. According to the list number (subgroup), select M and N from Table 8.1. Solve the problem.

2. According to the list number (subgroup), select the values from Table 8.2. Solve Task 8.2.

Task to complete:

Task 8.1: The access time to the cache memory is M ns, and to the main memory, it is N ns. What is the average access time of the processor, assuming a hit rate of 80%? Solve for 2 types of architecture. Compare the results.

Table 8.1

Input Data Variants

List number	M, ns	N, ns
1	20	140
2	25	135
3	15	130
4	18	128
5	20	150
6	17	185
7	19	150
8	30	140
9	28	130
10	35	155
11	15	110
12	18	120

Task 8.2. Given the main memory size, cache size, block size, and word size, find:

1) How many bits will be used to represent the physical address (P.A.), how many for the row number, and how many for the block offset?

2) The size of the tag directory (P.tag.dir)?

Table 8.2

Input data options.

List number	
1	RAM 2 GB (Main memory) Cache size 1 MB (Cache size) Block size 2 KB (Block size) Word size 1 byte (Word size)
2	
3	
4	
5	
6	
7	RAM 16 GB (Main memory) Cache size 4 MB (Cache size) Block size 16 KB (Block size) Word size 4 bytes (Word size)
8	
9	
10	
11	
12	

Questions for independent training

1. What are memory interfaces and what are their main types?
2. How does direct memory mapping work, and what are its advantages?
3. What are the main principles of cache memory operation?
4. Explain what a physical memory address is and how it is determined in systems with direct memory mapping.
5. How can the number of bits required to represent the physical address, row number, and block offset be determined in cache memory systems?
6. How does the memory access speed change when using cache memory compared to direct memory mapping?
7. What are the main differences between direct-mapped memory and associative memory?
8. How does the selection of cache parameters (size, number of blocks) affect memory utilization efficiency?
9. What are tag directories in the context of cache memory, and how are they used to search for data?
10. How is the average memory access time in a system with cache memory calculated, and what factors affect this time?
11. What optimization methods for memory access are used in modern computer systems?

Laboratory work 9

Memory Interfaces. Problem Solving of PYQs

Goals

Learn to determine associative organizations of direct-mapped sets. Learn to determine decimal values for an n-bit register for single and double precision in IEEE 754 format.

Procedure:

1. Familiarize yourself with the theoretical information.
2. Complete the practical tasks.
3. Prepare a report on the laboratory work.
4. Answer the control questions.

Theoretical part

Two cache memory organizations. The first is a 32 KB, two-way associative set with a 32-byte block size. The second has the same size (32 KB) but is direct-mapped (the mapping procedure will be different). The address size is 32 bits in both cases. This means the physical address is 32 bits in both cases. A 2-to-1 multiplexer has a delay of 0.6 ns, while a K-bit comparator has a delay of K/10 ns. The delay for accessing the associative set organization is considered h1, while the delay for direct mapping is h2. You need to determine the values of h1 and h2. Determine the correct values from the table below.

Table 9.1.

Values of h1 and h2

Value for h1			
A: 2,4 ns	B: 2,3 ns	C: 1,8 ns	D: 1,7 ns
Value for h2			
A: 2,4 ns	B: 2,3 ns	C: 1,8 ns	D: 1,7 ns

Solution: 1) First, we need to calculate h1, considering that the cache size is 32 kilobytes (Cache size = 32 KB), which equals 2^{15} bits. Since it is a two-way associative cache, the size of the set will be 2 lines, meaning that each set will have 2 lines, each located in a block. The size of a line is given as 32 bytes, which can be represented as 2^5 bits.

Cache size = 32 KB = 2^{15} B
Set size = 2 lines
Block size = 32 B = 2^5 B

This gives us a block offset of 5 bits.

Block offset = 5

Now, it is necessary to determine the number of cache lines

№ of lines = 2^{15} / 2^5 = 2^{10},

This means that there are 2^{10} lines inside the cache. Next, we need to determine the number of sets inside the cache, where the size of each set is 2 lines, or 2^1 lines. Therefore, by dividing

the number of lines by the established size, we can calculate the number of sets in the cache:

№ of Sets = $2^{10} / 2^{1} = 2^{9}$

Now, the physical address is presented as 32 bits, which means that out of the 32 bits, 5 bits are used for the block offset, 9 bits are used for the set index, and 32-9-5=18 bits are used for the tags.

32 bits		
18 bits	9 bits	5 bits
Tags	№ of Sets	Block offset

Next, we need to understand the cache organization. We know that there are sets inside the cache, ranging from 2 to 9. To access any of them, we will need a multiplexer.

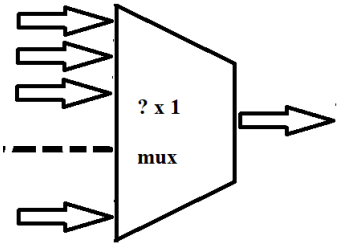


Figure 9.1. Multiplexer

Now the question remains, what should the configuration of the multiplexers be, as we are using 9 bits that will be used as selection lines.

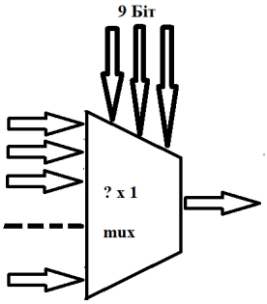


Figure 9.2. Multiplexer with 9-bit input.

In this case, the configuration will be 2^9 x 1, meaning the number of sets = 2^{10} / 2^1 = 2^9, which indicates that the cache contains 2^9 sets, as required. Now the question remains: how many of these multiplexers do we actually need? Since the size of a set consists of two lines (i.e., each set will have 2 lines).

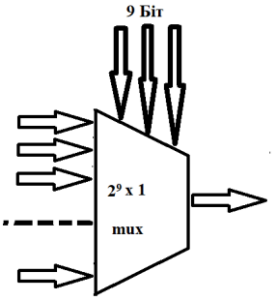


Figure 9.3. Multiplexer 2^9

If one multiplexer reads 1 bit, and we have 18 bits for tags, then we will need 18 such multiplexers. Additionally, since this is a two-way set, we need to address all the tags for all rows, so 18 x 2 = 36 (2^9 multiplexers). Therefore, in total, for associative

cache configurations with K setting bits and t tag bits, we will need (t + K) * 2^5-to-1 muxes.

Since we use 18 bits for tags and this is a two-way associative cache organization, we will have two 18-bit comparators, whose outputs are combined into one.

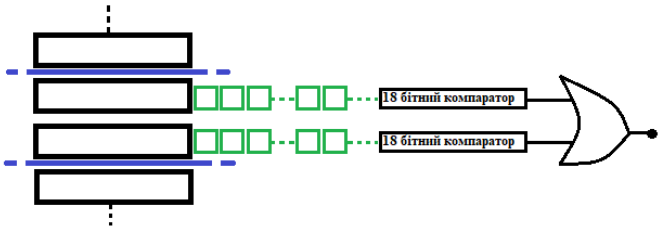
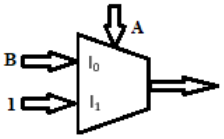


Figure 9.4. The diagram of a two-way cache organization with two comparators.

Now let's select the inputs.

A	B	O/P
0	0	0
0	1	1
1	0	1
1	1	1

$$\bar{A}B + A\bar{B} + AB = \bar{A}B + A(\bar{B} + B) = \bar{A}(B) + A(1)$$



Using the multiplexer, we implement any logical function, in this case B_0 and A_1, and store A for the selection row. Further, the task states that a K-bit comparator has a delay of K/10 ns. In this case, K = 18, which means the comparator delay is 18/10 = 1.8 ns. Therefore, the access delay h1 = 1.8 ns. However, the multiplexer also has a delay of 0.6 ns, so:

h1 = 1.8 + 0.6 = 2.4 ns – Option A.

2. Now, let's determine the value of h2. We know that...

Cache size = 32 KB = 2^{15} B

Block size = 32 B = 2^5 B : Block offset = 5

№ of lines = 2^{15} / 2^5 = 2^{10}

We know that the physical address size is 32 bits in both cases, so we have the following scheme:

32 bits		
17 bits	10 bits	5 bits

Let's determine the number of tag bits and enter it into the table: 32 - 10 - 5 = 17 bits. Similarly to the previous task, we need to determine the number of multiplexers. Here, the configuration will be 2^{10} x 1.

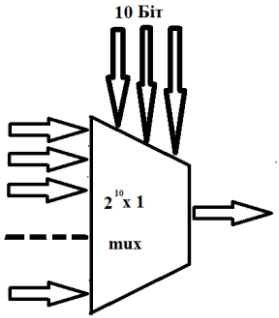


Figure 9.3. Multiplexer 2^{10}

Since we have 17-bit tags, we will need one 17-bit comparator. Therefore, the comparator will have a delay of: **h2 = K/10 ns = 17/10 = 1.7 ns – Variant D**.

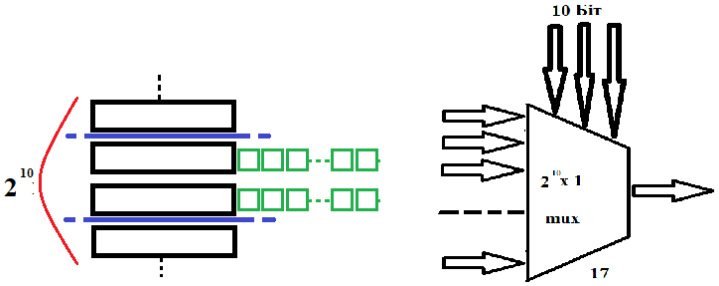


Figure 9.3. Diagram of a multiplexer with a single-side comparator.

Answer: h1 = 2.4 ns, h2 = 1.7 ns.

Value for h1			
A: 2,4 ns	B: 2,3 ns	C: 1,8 ns	D: 1,7 ns
Value for h2			
A: 2,4 ns	B: 2,3 ns	C: 1,8 ns	D: 1,7 ns

Task to perform:

Task 1. Two cache memory organizations. The first one is an N KB, two-way associative set with an N-byte block size. The second one has the same size (N KB) but is direct-mapped (the mapping procedure will be different). The address size is M bits in both cases. This means that the physical address is M bits in both cases. The 2-to-1 multiplexer has a delay of L ns, while the K-bit comparator has a delay of K/10 ns. The access delay for

the associative set organization is considered h1, while the access delay for the direct-mapped organization is h2. You need to determine the values of h1 and h2.

Table 9.2.

Task Variants

Variant	N	M	L
1	64	64	0,91
2	128	128	0,73
3	254	254	0,81
4	64	64	0,32
5	128	128	0,46
6	254	254	0,64
7	64	64	0,65
8	128	128	0,78
9	254	254	0,28
10	64	64	0,15

Task 2. Consider a 32-bit register that stores floating-point numbers in IEEE single-precision format. Find the decimal value of the following 32 bits.

Table 9.3.

Task Variants

Variant	Task
1	0 1000 0011 110010…..0
2	1 1010 0011 1100110…..0
3	0 1011 0011 110110…..0
4	1 1001 0011 110010…..01
5	0 1110 0011 110010110…..0

Variant	Task
6	0 1001 1011 111010…..0
7	1 1110 0011 110010…..010
8	0 1001 0011 1000110…..0
9	1 1111 0011 110010…..01010
10	0 1100 1011 010110…..0

Task 3: Consider a 64-bit register that stores floating-point numbers in IEEE double precision format. Find the decimal number for the following 64 bits.

Table 9.4.

Task variants

Variant	Task
1	0 000 1000 0011 010110…..0
2	1 101 1010 0011 110010…..0
3	0 110 1011 0011 1100110…..0
4	1 001 1001 0011 110110…..0
5	0 000 1110 0011 110010…..01
6	0 111 1001 1011 110010110…..0
7	1 101 1110 0011 111010…..0
8	0 000 1001 0011 110010…..010
9	1 010 1111 0011 1000110…..0
10	0 001 1100 1011 110010…..01010

Questions for independent training

1. What are associative set organizations for direct-mapped memory, and how can they be correctly determined?
2. How do you calculate the number of bits required to represent the physical address, line number, and block offset in memory systems with cache memory?
3. What is the IEEE 754 floating-point format, and how do you determine the decimal value for an n-bit register when a number is given in this format?
4. How do you determine the number of bits needed to represent a number in single-precision and double-precision IEEE 754 formats?
5. What is the difference between associative and direct-mapped memory organizations? How does this affect cache memory efficiency?
6. How do you build a table for determining addresses in systems with associative set organizations and direct-mapped memory?
7. What are the main steps for converting a number to IEEE 754 format (single and double precision)?
8. How do you calculate the average memory access time in different cache organizations: associative and direct-mapped?
9. Explain how a comparator works in associative caches and how to determine its delay when the number of bits is given.
10. How does the cache memory structure influence the overall memory access speed in computer systems?

Bibliography

1. Hwu W.-M.W., Kirk D.B., Hajj I.E. Programming Massively Parallel Processors: A Hands-on Approach. 4th ed. Cambridge : Morgan Kaufmann, 2022. 580 p.

2. Deakin T., Mattson T. G. Programming Your GPU with OpenMP: Performance Portability for GPUs. Cambridge : Morgan Kaufmann, 2023. 336 p.

3. Barlas G. Multicore and GPU Programming: An Integrated Approach. 2nd ed. Burlington : Morgan Kaufmann, 2022. 1024 p.

4. Hennessy J. L., Patterson D. A. Computer Architecture: A Quantitative Approach. 6th ed. Cambridge : Elsevier, 2017. 992 p.

5. Tanenbaum A. S. Structured Computer Organization. 6th ed. Upper Saddle River : Pearson Custom Publishing, 2013. 544 p.

6. Bishop R. L. The Architecture of Computer Hardware and Systems Software. 3rd ed. Hoboken : Wiley, 2005. 456 p.

7. Morse L. Computer Organization and Design: The Hardware/Software Interface. 5th ed. Berkeley : University of California, 2014. 720 p.

8. Darringer D. J. Advanced Computer Architecture. El Kuwait : Kuwait University, 2016. 352 p.

9. Petrosyan O. O. Computer Architecture. Kyiv : Kondor, 2010. 368 p.

10. Melnyk M.I. Architecture of Computers and Systems. Lviv : Lviv Polytechnic, 2018. 450 p.