"Kyiv Professional College of Communication"

Cycle commission of computer and software engineering

Lecture 3

in the discipline: "Operating systems"

Topic: "Comparative table"

Performed by a student

of the group BICS-13

Los Zlata Volodymyrivna

Checked by the teacher

Sushanova V.S.

Kyiv 2024

|  |  |  |
| --- | --- | --- |
| **Comparison criteria** | **Windows 10** | **IOS 17** |
| **OS kernel architecture**  **(brief description)** | **Windows NT kernel:** This kernel includes the main components such as the task handler, memory manager, drivers, task scheduler, and other elements that provide the basic functions of the operating system.  **Drivers:** Most of the operating system's functionality, such as device handling, networking, and other aspects, is implemented through drivers that interact with the kernel.  **Win32 and other subsystems:** Additional subsystems, such as Win32 for application compatibility, as well as subsystems to support POSIX, Linux, and other environments, complement the functionality of the kernel.  **System calls and APIs:** Windows 10 provides APIs (application programming interfaces) for programs to interact with the operating system. System calls allow programs to access kernel functions.  **Security:** Windows 10 includes a variety of security mechanisms, such as Windows Defender, BitLocker, UAC (User Account Control), and others, to provide protection against threats and maintain user privacy. | **Mach Microkernel:** The iOS kernel uses a microkernel architecture, and the main component is the Mach microkernel. The microkernel architecture divides the kernel into a minimal set of functions, which allows for improved reliability and supports easier extensibility.  **XNU Kernel:** The Mach kernel is combined with other components into a kernel called XNU (X is Not Unix). XNU includes the Mach microkernel as well as elements from the monolithic kernel, such as the Apple-designed scheduler, memory manager, and other components.  **Drivers:** The kernel interacts with various drivers that enable interaction with a variety of devices such as cameras, displays, audio devices, and others.  **Quasar:** This is the kernel component that is responsible for the security of the operating system. It includes a permission processing system and other security measures.  **UIKit and other subsystems:** To provide a high-level interface and functionality, the kernel interacts with various subsystems such as UIKit, Core Audio, Core Graphics, Core Data, etc., which are responsible for graphics, audio, user interface, and other aspects of the operating system. |
| **The user interface**  **(brief description and**  **whether it is possible to expand**  **its functionality)** | **The Start Menu:** Located in the lower left corner of the screen, this menu contains submenus that list programs, access to files, and system settings. You can also find Live Tiles to display real-time information.  **Taskbar:** Located at the bottom of the screen, it includes buttons for running programs, the system tray, and a shortcut bar.  **Program windows:** The user can open program windows, move them, resize them, close them, maximize them, and minimize them.  **System settings:** In Windows 10, system settings such as network settings, user accounts, security, and privacy can be customized through Settings.  **Action Center:** Provides quick access to notifications, system events, and settings such as airplane mode, power saving mode, and other settings.  **File Explorer:** Allows users to manage files and folders on their computer and interact with various storage devices.  **Windows Security Center:** Provides tools for managing antivirus, threat protection, and other security aspects.  **Log in with a PIN or password:** Ensures that your sign-in information is secure and confidential. | **Home Screen:** This is the main screen that displays application "icons". Users can organize their apps, create folders, and use search for quick access.  **Apps and Icons:** Applications in iOS are represented by icons that you can tap to launch programs or interact with features.  **Gesture Control:** iOS uses gestures to navigate and perform various actions, such as scrolling, expanding, and collapsing.  **Control Center:** Provides quick access to basic functions, such as turning off Wi-Fi, adjusting screen brightness, controlling music, and more.  **Notifications:** Displays real-time notifications in a sliding bar at the top of the screen.  **Today View:** Displays important information blocks such as weather, calendar, news, and more.  **App Store:** Where users can find and download apps, games, and other content.  **Live Tiles family:** Some icons and special widgets can display changing information in real time.  **Animations and Transitions:** iOS is known for its smooth animations and spectacular screen transitions, which creates a positive user experience.  **Security and Privacy:** iOS has a high level of security and privacy, including the use of Face ID or Touch ID to unlock your device and authenticate. |
| **System calls (which**  **standards, their brief**  **description, which**  **libraries can extend them**  **can extend them)** | **CreateFile:** Creates or opens a file for further work.  **ReadFile and WriteFile:** Used to read and write data to a file or other I/O object.  **CreateProcess:** Starts a new process.  **ExitProcess:** Exits the current process.  **AllocateMemory and FreeMemory:** Used to allocate and free memory.  **OpenProcess and CloseHandle:** Allows you to open and close access to other processes.  **Socket:** For network communication, used, for example, when working with TCP/IP.  Regarding libraries that can extend or facilitate the use of system calls, it is important to point out libraries that provide a high-level API and abstraction over system calls. For example, in the Windows environment, Microsoft WinAPI libraries or high-level interfaces such as the .NET Framework can be used to work with many system functions, making program development more convenient and productive. | **Objective-C and Swift APIs:** iOS development often uses the Objective-C and Swift programming languages, and system calls are made through APIs provided by these languages.  **UIKit and Foundation Frameworks:** These frameworks provide an interface for interacting with graphics, user interface, event handling, and basic data structures.  **Grand Central Dispatch (GCD):** For managing threads and performing asynchronous tasks.  **Core Data:** A library for working with databases and data caching.  **Core Location:** For working with geolocation and processing geographic data.  **Core Animation:** Used for creating animations and processing graphic effects.  **NSURLSession and Alamofire for networking:** For making network requests and exchanging data. |