“Київський фаховий коледж зв’язку”

Циклова комісія комп’ютерної та програмної інженерії

Лекція №3

з дисципліни: «Операційні системи»

Тема: «Табличка порівняльна»

Виконала студентка

групи БІКС-13

Лось Злата Володимирівна

Перевірив викладач

Сушанова В.С.

Киів 2024

|  |  |  |
| --- | --- | --- |
| **Критерії порівняння** | **Windows 10** | **IOS 17** |
| **Архітектура ядра ОС**  **(коротка характеристика)** | **Windows NT-ядро:** Це ядро включає основні компоненти, такі як обробник завдань, менеджер пам'яті, драйвери, планувальник завдань та інші елементи, що забезпечують основні функції операційної системи.  **Драйвери:** Велика частина функціональності операційної системи, така як обробка пристроїв, мережі та інші аспекти, реалізується через драйвери, які взаємодіють з ядром.  **Win32 та інші підсистеми:** Додаткові підсистеми, такі як Win32 для сумісності з програмами, а також підсистеми для підтримки POSIX, Linux і інших середовищ, доповнюють функціонал ядра.  **Системні виклики та API:** Windows 10 надає API (інтерфейс програмування додатків) для взаємодії програм з операційною системою. Системні виклики дозволяють програмам звертатися до функцій ядра.  **Безпека:** Windows 10 включає різноманітні механізми безпеки, такі як Windows Defender, BitLocker, UAC (User Account Control) та інші, щоб забезпечити захист від загроз та зберегти конфіденційність користувачів. | **Mach Microkernel:** Ядро iOS використовує мікроядерну архітектуру, і основним компонентом є Mach microkernel. Мікроядерна архітектура розділяє ядро на мінімальний набір функцій, що дозволяє покращити надійність та підтримувати легшу розширюваність.  **XNU Kernel:** Ядро Mach об'єднане з іншими компонентами в ядро під назвою XNU (X is Not Unix). XNU включає Mach microkernel, а також елементи з монолітного ядра, такі як розроблений Apple планувальник, менеджер пам'яті та інші компоненти.  **Драйвери:** Ядро взаємодіє з різними драйверами, які забезпечують взаємодію з різноманітними пристроями, такими як камери, дисплеї, аудіо-пристрої та інші.  **Quasar:** Це компонент ядра, який відповідає за безпеку операційної системи. Включає в себе систему обробки дозволів і інші заходи безпеки.  **UIKit та інші підсистеми:** Щоб надати високорівневий інтерфейс та функціонал, ядро взаємодіє з різними підсистемами, такими як UIKit, Core Audio, Core Graphics, Core Data і т.д., які відповідають за графіку, аудіо, інтерфейс користувача та інші аспекти операційної системи. |
| **Інтерфейс користувача**  **(коротка характеристика та**  **чи можливо розширити**  **його функціонал)** | **Меню "Пуск" (Start Menu):** Воно розташоване в лівому нижньому куті екрана і містить підменю зі списком програм, доступ до файлів та параметрів системи. Крім того, у ньому можна знаходити "Живі плитки" (Live Tiles) для відображення інформації в реальному часі.  **Панель завдань (Taskbar):** Знаходиться внизу екрана і включає кнопки запущених програм, системний трей, а також панель швидкого доступу.  **Вікна програм:** Користувач може відкривати вікна програм, переміщувати їх, змінювати розмір, закривати, максимізувати та мінімізувати.  **Параметри системи:** В Windows 10 параметри системи, такі як мережеві налаштування, облікові записи користувачів, безпека і приватність, можна налаштовувати через Налаштування.  **Центр дій (Action Center):** Забезпечує швидкий доступ до повідомлень, системних подій та параметрів, таких як режим польоту, режим енергозбереження, інші налаштування.  **Файловий провідник (File Explorer):** Дозволяє користувачам керувати файлами та папками на комп'ютері, а також взаємодіяти з різними сховищами даних.  **Центр безпеки та захисту (Windows Security):** Надає інструменти для керування антивірусом, захистом від загроз та іншими аспектами безпеки.  **Вхід за допомогою PIN-коду або пароля:** Забезпечує безпеку та конфіденційність вхідних даних. | **Домашній екран (Home Screen):** Це основний екран, на якому відображаються "іконки" додатків. Користувачі можуть організовувати свої програми, створювати теки та використовувати пошук для швидкого доступу.  **Додатки та Іконки:** Додатки у iOS представлені іконками, які можна торкатися для запуску програм або взаємодії з функціями.  **Керування за допомогою жестів:** iOS використовує жести для навігації та виконання різних дій, таких як скролінг, розгортання та згортання.  **Контрольний центр (Control Center):** Забезпечує швидкий доступ до основних функцій, таких як вимкнення Wi-Fi, налаштування яскравості екрана, керування музикою тощо.  **Повідомлення (Notifications):** Відображає повідомлення в реальному часі висуваючись зверху екрана.  **Спільний центр (Today View):** Відображає важливі інформаційні блоки, такі як погода, календар, новини тощо.  **Магазин App Store:** Де користувачі можуть знаходити та завантажувати додатки, ігри та інші контенти.  **Сімейство "Живі плитки" (Live Tiles):** Деякі іконки та спеціальні віджети можуть відображати змінюючи інформацію в реальному часі.  **Анімації та Переходи:** iOS відомий своєю плавною анімацією та ефектними переходами між екранами, що створює позитивний користувацький досвід.  **Безпека та Конфіденційність:** iOS має високий рівень безпеки та конфіденційності, зокрема використання Face ID або Touch ID для розблокування пристрою та автентифікації. |
| **Системні виклики (які**  **стандарти, їх коротка**  **характеристика, які**  **бібліотеки можуть їх**  **розширити)** | **CreateFile:** Створює або відкриває файл для подальшої роботи.  **ReadFile та WriteFile:** Використовуються для читання та запису даних в файл або інший об'єкт вводу/виводу.  **CreateProcess:** Запускає новий процес.  **ExitProcess:** Завершує виконання поточного процесу.  **AllocateMemory та FreeMemory:** Використовуються для виділення та звільнення пам'яті.  **OpenProcess та CloseHandle:** Дозволяє відкривати та закривати доступ до інших процесів.  **Socket:** Для мережевого взаємодії, використовується, наприклад, при роботі з TCP/IP.  Щодо бібліотек, які можуть розширити або полегшити використання системних викликів, важливо вказати на бібліотеки, які надають високорівневий API та абстракцію над системними викликами. Наприклад, в середовищі Windows можуть використовуватися бібліотеки Microsoft WinAPI або високорівневі інтерфейси, такі як .NET Framework для роботи з багатьма системними функціями, що робить розробку програм більш зручною та продуктивною. | **Objective-C та Swift API:** Розробка під iOS часто використовує мови програмування Objective-C та Swift, і системні виклики виконуються через API, надані цими мовами.  **UIKit та Foundation Frameworks:** Ці фреймворки надають інтерфейс для взаємодії з графікою, користувацьким інтерфейсом, обробкою подій та роботою з базовими структурами даних.  **Grand Central Dispatch (GCD):** Для управління потоками та виконання асинхронних завдань.  **Core Data:** Бібліотека для роботи з базами даних та кешуванням даних.  **Core Location:** Для роботи з геолокацією та обробки географічних даних.  **Core Animation:** Використовується для створення анімацій та обробки графічних ефектів.  **NSURLSession та Alamofire для мережевого взаємодії:** Для виконання запитів до мережі та обміну даними. |