UNIX- LIKE OS – iOS

S.MOHAN
ASSISTANT PROFESSOR OF COMPUTER SCIENCE AND ENGINEERING
V.S.B.COLLEGE OF ENGINEERING TECHNICAL CAMPUS
KINATHUKADAVU-COIMBAYORE-642100
TAMIL NADU- INDIA.

ABSTRACT:

In this paper we will discuss about iOS because it is UNIX- LIKE OS with its mobile operating system developed by apple Inc.… derived from macOS (now macOS) and built up on the Darwin foundation . it shared same Unix heritage utilize component of the Mach micro kernel and free BSD. Berkeley software distribution) making its Unix-like system. It is handled the graphics, audio and video competitive. it provide the frame work for creating the user-interface (UI)and user interaction. whole derived from open-source component (Darwin/BSD). iOS closed source proprietary operating system developed and controlled by apple iOS is designed for touch-based mobile service such as iPhone, iPad< and iPod touch emphasizing a service seamless user experience. iOS is designed with a strong focus on security , featuring server coding practices and encryption . it introduced in later version , supporting and ability to run multiple application simultaneously . iOS developed in primarily consider "front-end" because it is focus on user interface (UI) user experience(UX) and client-side instruction on the apple devices. However it is often in valued "back-end "talks like local data persistence (core data) . API integration and business logic modern OS devs often acts as full-stack with in the app.

KEYWORDS:BSD, iOS API, UI,UX,GPS.

INTRODUCTION:

The iOS is front-end focus designing and interface with swift UI/UI kits ensuring smooth user interaction and handling the animation, but it is "back-end" manage the data , API calls and authentication using networking tools or using BaaS(backend-as-a- services)like firebase . the iOS client-side. but does not use HTML/CSS.it focus on native app performs which serve consider mobile a distinct field ("device-end")if you must categories it is the front-end of the client service system .

The iOS development is primarily consider front-end development because it focus on the user-facing interface and experience , however modern iOS app often request a "back-end" to handle data storage user authentication and complex business logic. The most iOS developed spent there time building what user see and touch. The user- experience (UX) animation and device specification features like GPS ,camera, and local storage . it acting as the client .that request data from the remote server in iOS back-end option ,if you app needed to sync data between devices or support multi user . you will need a back-end (BaaS) backend-as-a-service. Tool likes firebase or AWS amplify allows you to manage a back-end without writing complex server – side code.

The Unix is the back-end for iOS. Because this iOS is built on a foundation descendent from Unix specifically iOS is derived from Darwin . which is an open-source operating system based on the BSD branch(free BSD).

UNIX is back-end for iOS:

The iOS uses the Darwin operating system , which serve as the core language of apple devices including iPhone, iPad, and apple TV . the center of iOS is the (XNU) kernel . which is hybrid kernel (combining mach and free BSD component). Making it as a UNIX -LIKE system.

Because it is built on BSD UNIX iOS mainframe capabilities with many POSIX standard allowing it to leverage UNIX principles from file management , processing handling and security.

Where it is UNIX based iOS is a closed system that lack direct terminal access or standard user-facing UNIX common like tools, unlike its course mac OS .this iOS is a "mobile version "of the UNIX -BASED macOS with specialized user interface (Cocoa touch) built on the top of that UNIX back-end .

UNIX FOUNDATION:

The iOS fundamentally built on a UNIX foundation while you don't typically see and command-line interface an a iPhone, the "back-end")of the operating system is derivative of UNIX , the iOS based on Darwin , an open-source operating system created by apple . Darwin itself derived from BSD(Berkeley software distribution). While it is direct descendent of the original reach UNIX.

The heart of the iOS is the XNU kernel that combine the Mach micro kernel with component from free BSD and an object-oriented driver APUI called I/O kits. Because iOS and macOS share this Darwin foundation , they use same low-level process for file management , networking (BSD socket) and permission .

The apple hides the UNIX language to ensure security and a single user experience unlike macOS ,iOS does not include a nature terminal app. The excel app is strictly isolated (sandbox) providing the kind of deep system access typically available in a standard UNIX environment . the iOS is consider "UNIX-LIKE". Because it lack of certain standard utility and command required for official certificate .

Even though the interface is modern and touch-based .the UNIX back-end matter for development . the iOS uses APFS(apple file system) which utilize the UNIX-TIME.

FRONT-END OF iOS:

The iOS front-end technology for native iOS developed is SWIFT consider with the SWIFT-UI frame work which is apple modern development Ui frame work for OLDER or Legacy project , UI Kit with swift or objective-c is used to create user-facing interactive interface for iPhone, and iPad and other apple devices.

Because it is popular cross- platforms frame work that allow develop to use java script(react native)or dart(flutter) to built native facing front-end for both iOS and android simultaneously . in the contest of iOS development the "FRONT-END "refer to the user friendly part of the an application . the visual animation and interaction element that use see and touch. Unlike web development while user HTML/CSS iOS front-end are primarily built using native apple frame work as cross- platform tools.

PRIMARY NATIVE:

Swift UI: apple modern declarative frame work for building the user interface across all it is platforms (iOS ,macOS . watch OS etc.…) it uses a core single syntax when you describe what the UI should do and the frame work handle the reading

UI Kits: the traditional , imperative framework for building iOS apps, while older it is still widely used in many existing app and for complex customs interfaces that required deep system-level control.

CROSS-PLATFORMS alternatives;

If you want to use single base for both iOS and android these front-end technology and popular

FLUTTER: a UI frame work by google that uses the dart-language to create high-performs , virtually constant app.

REACT NATIVE :a frame work by meta that uses that allow you to built mobile app using java script and react.

THE SWIFT: is the primarily modern language for all native iOS development.

THE objective-C is the older language for apple development , now mostly such in legacy project .

While an iOS app itself is often consider a " front-end" or (client-side) application .it may connect to a (back-end like firebase) to store data or handled complex server-side logic.

Back end-as-a-service: using fire base or supabase which provide SDK(software development kit)that handle data persistent and authentication , eliminating the need to built a customs back end API.

The CONNECT an iOS front-end (built with swift /objective-C) to a back end you primarily uses API (application programming interfaces) specially RESTful API to exchange the data over the internet using HTTP/HTTPs

The front-end and back-end act as two separate entities that communicate via structure loop .

Request the iOS app send and HTTP request (eg GET,POST,PUT, DELETE) to specific URL (end point On the server .

Process in; the back end receives the request , interact with a database (like Post gre SQL, mango DB,or my SQL) and performs business logic.

Response: the back-end send back data , almost always JSON format while the iOS app then decoded and display in the UI.

Common way to built/ connect : depending on your project need you can choose between managed service and customers service .

Back end-as-a-service(BaaS):I deal for most app as the provided ready-made data bases, authentication and hosting .

Firebase :highly popular , often real-time data syncing and easy swift integration .

Supabase : a powerful open-source alteration for relational database.

Cloud kit: apples native solution great for app exclusive to the app ecosystem.

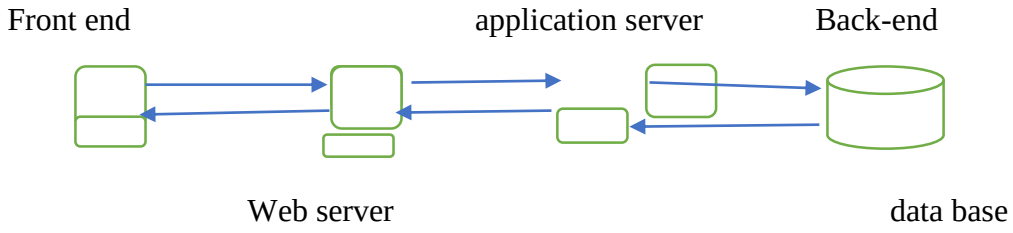
Custom back end API :if you needful control over complex logic you built your own services using frame work like .

Node.js;(express) which used and easy to setup.

Swift(vapor): allow you to use the same language for both front-end and back-end

Python(Django/flash): excellent for data heavy application.

ARCHITECTURE:



Unix-based OS: one family of multi-tasking . multi user system derived from or inspired by the original 1970 AT&T bell labs Unix they are know for possibility modularity and security with major branches including certified UNIX(mac OS , Solaris, A IX) and open-source UNIX-like system (Linux, free BSD)and mobile system like android and iOS.

Unix-like system(GNU/LINUX): not based an original code but adhere to UNIX design principles POSIX standard.

LINUX distribution: ubuntu , federo , debin cent OS, red hat enterprises Linux , arch Linux, kali Linux.

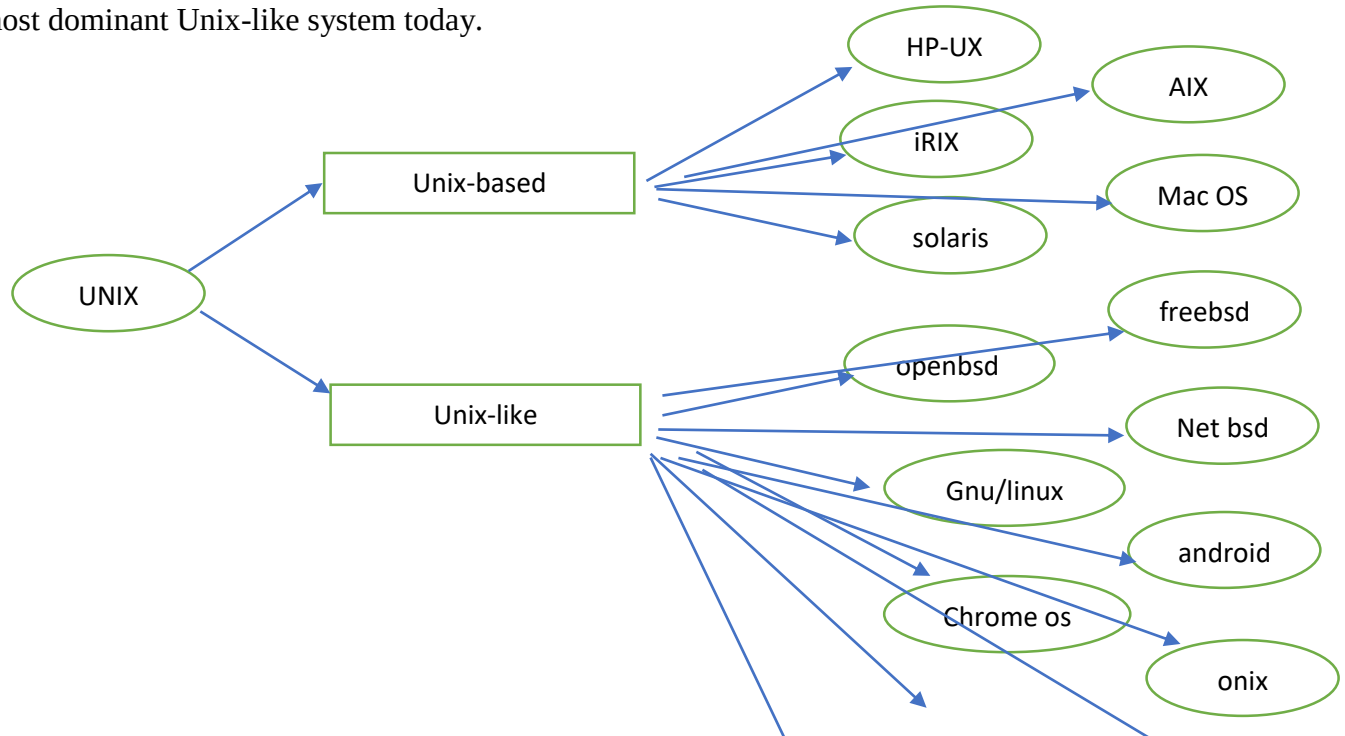
BSD (Berkeley software distribution) derived from the original UI Berkeley research , free BSD, net BSD, open BSD, known for stability and net working capability .

Mobile embedded system:

Android uses a modified UNIX kernel .

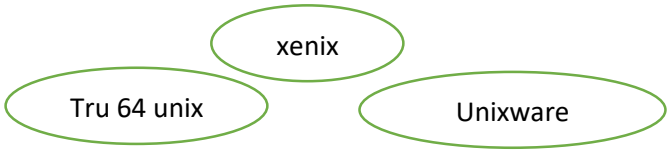
iOS/iPad OS : shares win /BSD under pinning with macOS

UNIX vs LINUX: while often used interchangeably original UNIX was properties and modular where as Linux is open-source and based on a monolithic kernel , however Linux implement the UNIX philosophy .making it the most dominant Unix-like system today.



TYPES OF UNIX OS:

1. UNIX-BASED
2. UNIX-LIKE



UNIX-BASED: OS derived from the original UNIX OS share same feature and follow the principle of the original UNIX based OS have become popular because they often many benefit of the original UNIX OS as well as some modern feature such as a graphical user interface (GNU) support for multi-media application , support VPN and fire wall etc.

UNIX-LIKE: not necessary derived from the original UNIX OS but mimic it behavior . the UNIX-like OS include some improvement and that reliable in the open-source software (OSS). If share many design principle and features of the original UNIX OS . but not it is property code OR specific implementation completely.

COCLUSION:

Here the conclusion regarding iOS an a UNIX-LIKE *OS* based on the provided search result iOS is family rooted in UNIX heritage deriving as a property mobile optimization derived of mac OS . while itself is built as a Darwin .a system incorporating BSD (Berkeley software distribution) components which iOS is not directly certified UNIX version . it adheres to many of the source design principle and functions structure making it a many modern example of UNIX-LIKE operating system . unlike LINUX which is open-source iOS is a properties system heavily restricted and customized for a user friendly experience an apple devices iOS is a modern, mobile implementation of the UNIX philosophy a serve , stable , multi-tasking system that powers million of devices while leveraging the fundamental work of the BSD branch of UNIX.

REFERENCES:

1.Android Developers, *Machine Learning*, https://developer:android:com/ml, [Accessed: 23 - 02 - 2021] (2020).

2.S. Brahler, Analysis of the Android Architecture, *Karlsruhe Institute for Technology* 7 (8) (2010).

3.O. C. Novac, M. Novac, C. Gordan, T. Berczes, G. Bujdos´o, ComparativeStudy of Google Android, Apple iOS and Microsoft Windows PhoneMobile Operating Systems, in: *2017 14th International Conference onEngineering of Modern Electric Systems (EMES)*, 2017, pp. 154–159. doi: 10:1109/EMES:2017:7980403.

4.*Monolithic Hybrid Micro Nano Kernel OS and examples*, https://dwaves:de/2021/02/13/monolithic-hybrid-micro-nano-kernel-osand-examples/, [Accessed: 27 - 02 - 2021] (2021).

5.G. Vashisht, R. Vashisht, A Study on the Tizen Operating System, *International Journal of Computer Trends and Technology* 12 (1) (2014) 14–15. doi: 10:14445/22312803/ijctt-v12p104. URL http://dx:doi:org/10:14445/22312803/IJCTT-V12P104.

6. Mathieu, P. Oechslin, *Security analysis of the KaiOS feature phoneplatform for DFS applications*, https://www:itu:int/dmspub/itu-t/opb/tut/T-TUT-DFS-2020-3-PDF-E:pdf, [Accessed: 27 - 02 - 2021] (2020).

7..O. Okediran, O. Arulogun, R. Ganiyu, C. Oyeleye, Mobile operatingsystems and application development platforms: A survey, *InternationalJournal of Advanced Networking and Applications* 6 (1) (2014) 2195.

8.T. Grønli, J. Hansen, G. Ghinea, M. Younas, Mobile Application Platform Heterogeneity: Android vs Windows Phone vs iOS vs Firefox OS, in: *2014 IEEE 28th International Conference on Advanced InformationNetworking and Applications*, 2014, pp. 635–641. doi: 10:1109/AINA:2014:78.