# Unit 3: iOS Operating System

3

## Unit Structure

## 3.1 LEARNING OBJECTIVE

After studying this unit, the student should be able to Understand:

- iOS architecture, subcomponents, and its working mechanism.
- Comparing the iOS with Android architecture and various layers.
- Jailbreaking and related issues.
- Comparison between Android and iOS.

## 3.2 INTRODUCTION

iOS is a mobile OS created and developed by Apple Inc. It is the second most popular mobile OS after Android. Xcode is an integrated development environment (IDE) for macOS containing a suite of software development tools for developing software for macOS, iOS, watchOS, and tvOS. iOS is based on the Mach microkernel and Darwin system. programmers use Objective-C and Swift to create software for iOS platforms.

### 1.2.1 IMPORTANT FEATURES OF IOS

Following are the important key points that make iOS as a unique choice for certain users.

- Consistency user interface that facilitates the sense of control, familiarity, and reliability. The consistency in iOS is governed by the human interface design guidelines, similar to the material design in Android OS. The consistency intern increases the usability and minimizes the confusion.
- The consistency prioritizes the content , make it navigational, logical structured, and evokes the positive emotional response.
- ios is providing stability due to the use of only signed apps. Untrusted binaries are not allowed, and hence no worry about the malware. As iOS is intended only for the devices by Apple, hence they do not deal with the multiple compatibility issues.
- It follows the strict guidelines of user interface design.
- Control central in iOS provides quick access to the most frequently useful functionalities. iOS 11 gives customization to the control center.

- 3D touch (Force Touch) functionality supports the pressure sensitive screens, and introduces two new standard gestures - Peek and Pop, in addition to the swipe, pinch and tap. Press the screen with medium strength and you perform a Peek; push it harder, and you perform a Pop. For example, Peek allows previewing the contents of an email without opening the app involved. The Pop opens the item in its relevant app.
- There are few apps come first on the iOS, e.g. Instagram, snapchat, Augmented Reality mobile game Pokémon Go, etc.
- As soon as the apple produces the updates the users receive it.
- iTunes media player developed by Apple Inc. for play, download, and organize digital multimedia files, including music and video.
- Apple provides a Cloud-based storage service, called iCloud, that makes it possible for us to sync our documents, photos, videos, files across all of our iPhone, iPad, and Mac devices. iCloud dependent on the Amazon cloud services and also Google cloud, and Microsoft Azure.
- With the advantages, the new release of the iPhones, the old phones also get good value.
- iOS battery optimization is better. To save the battery power, it is required to Turn off Background App Refresh, and the low signal should be turned into airplane mode. In a low power, mode Mail will not download content in the background, AirDrop, iCloud sync, and Continuity will be disabled. Wi-fi access uses less battery power than the cellular network. The permissible temperature for battery life is 0$^{o}$C to 35 $^{o}$C. The battery works better with the latest updates of the software, the settings for the auto-brightness, and low power mode.
- Security and multi-tasking are excellent in Ios.

### 3.2.2 DISADVANTAGES

- The hardware limitation in one of the constraints in the case of iOS. The hardware manufacturer and the OS developer is same and hence less flexibility to use hardware with varied features.

- To maintain stability and consistency, the internal quality assurance cell spends time and efforts, which increases the cost of the product. The budget constraint is one of the limitations.
- The kernel of the iOS is having microkernel, which is a contrast to the monolithic kernel of the Android OS.
- High amount of restrictions on the customization, and no app installation other than the apple app store keeps the user is very much in control.
- iOS is not open source, apps are costly and no widgets.
- No hybrid SIM card support.
- Battery performance is reduced when used with 4G.
- No Near Field Communication or Radio support.

## 3.3 IOS ARCHITECTURE

### 3.3.1 INTRODUCTION TO IOS LAYERS & FRAMEWORKS

As shown in Figure-68, the iOS architecture is having five layers, including Coca touch, media layer, core services layer, core operating system layer & kernel, and Device Driver layer.

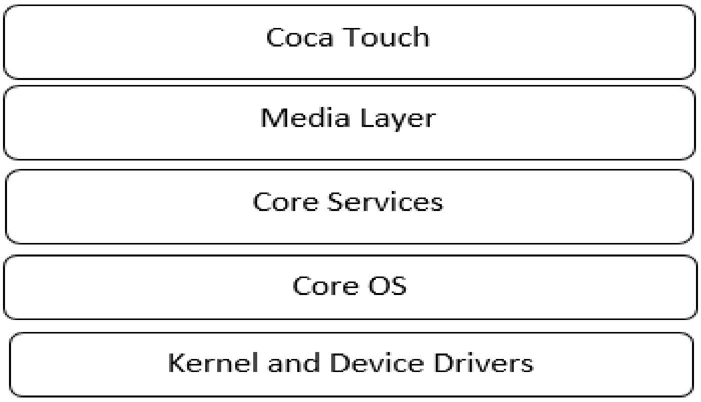

**Figure-68 Layers in the iOS Stack (Architecture)**

### 3.3.2 COCOA TOUCH LAYER

Cocoa is a collection of tools, libraries, frameworks, and APIs used to build applications for the Mac OS. It provides the rich quality functionalities for drawing to display, working with text, saving and opening data files, interacting with the operating system, and network. The Cocoa frameworks include classes for user interface and classes for memory management, networking, file system operations, and time management.

Cocoa Touch is a modified version of Cocoa with device-specific libraries for the iPhone and iPod Touch. Cocoa Touch works in conjunction with other layers in the iPhone and iPod Touch OS. Cocoa Touch provides an abstraction layer of iOS, the OS for the iPhone, iPod Touch, and iPad. It is based on the macOS Coca API toolset. Cocoa Touch allows the use of hardware and features. It follows the Model-View-Controller (MVC) architecture. Cocoa Touch contains a different set of graphical control elements to Cocoa. The iOS SDK consists of the Tools for developing applications based on Cocoa Touch.

The features and technologies of the cocoa touch and framework are App Extension, Handoff, Document Picker, AirDrop (File Transfer using Bluetooth and Wi-Fi), TextKit (classes to control the layout of text), UIKit Dynamics, Multitasking, Auto Layout, Storyboards, UI State Preservation, Apple Push Notification Service, Local Notifications, Gesture Recognisers, and Standard System View Controllers. The cocoa touch framework Foundation Kit Framework, UIKit Framework (based on Application Kit), GameKit Framework, MapKit Framework, Address Book UI Framework, EventKit UI Framework, Message UI Framework, Notification Centre Framework, PushKit Framework, and Twitter Framework.

Gamekit is set of classes to develop the games for more than one player. It allows to play games, share scores, dashboard through the network sharing. The player can allow other player to join the game, challenge the player, and show the accomplishment of the players. The PushKit is useful for the notifications based on VoIP invitations.

### 3.3.3 COCOA TOUCH FRAMEWORKS

The cocoa application layer is responsible for the responsiveness and appearance of the user interface of the app. Cocoa layer implemented the Notification Center, full-screen mode, and Auto Save options. Cocoa is the application layer and programming interface for app development. Auqa is the term used for the appearance of the iOS app development. The layout, color, and texture, along with the human-computer interface guidelines, make the user interface beautiful and compelling.

Appkit is useful for the various operations at this layer, including menu management, window management, document management, open and saves dialogue, pasteboard, and clipboard behavior. There is a font, color, image graphics, gesture, typography, and font operations.

Most Cocoa classes in Cocoa are subclasses of NSObject. In UIKit, all classes that respond to user input inherit from UIResponder. There are UI kit classes having base class as UIResponder and other child classes UIApplication, UIView, UIWindow, UILabel, UIpickerView, and other UI classes. Moreover, there are controller, value objects, and device classes.

The Foundation layer of Cocoa Touch provides the abstraction for the core elements of the OS. It handles the object behavior, memory management, inter-object notification mechanism, event dispatching, accessing file bundles, internationalization, and localization of resources including text, string, and images. It also supports file handling and SQLite data management tools. The object wrappers are similar to the wrapper classes in Java. It also has value objects, XML, Strings, URL handling, inter-processing communication, locking, and thread management.

## 3.3.4 MEDIA LAYER

The beautiful graphics and high-quality multimedia is the point of attraction for Ios. Media layer provides the 2D & 3D graphics, animation, image effects, and professional grade audio and video functionality.

The media layer supports web streaming protocols like HTTP, RTP, and RTSP. It also supports the iTunes .m4v format, and other standard formats such as AVI, AVR, DV, M-JPEG, MPEG-1, MPEG-2 other many formats. It also supports the various image, audio, and data file format. Figure-69 shows the media layer subcomponents and libraries.

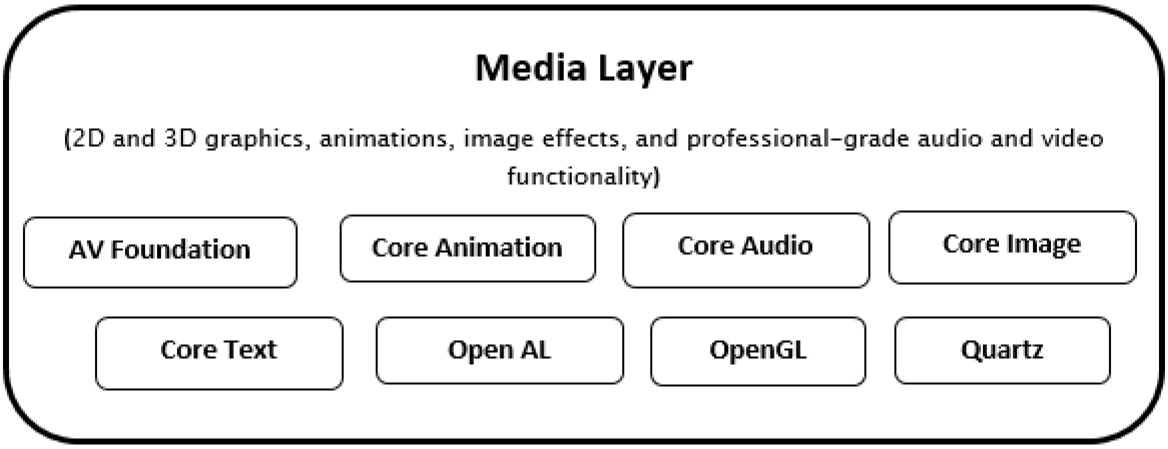
Media Layer
(2D and 3D graphics, animations, image effects, and professional-grade audio and video functionality)
AV Foundation
Core Animation
Core Audio
Core Image
Core Text
Open AL
OpenGL
Quartz

**Figure-69 Components of the Media layer**

## 3.3.5 MEDIA LAYER FRAMEWORKS

The AppKit framework provides the Quartz 2D and drawing primitive shapes such as lines, rectangles, ovals, arcs, and Bezier paths. Metal programming provides the maximization of the graphics. There are other core graphics like Quartz-2D, 2D vector, image rendering, and device independent capabilities. There is also a core animation Quarts framework. The Spriekit, Scene kit, Metal kit, OpenGL, GL kit are the other libraries.

The APIs are also available for the color, Text, Typography, and fonts operations. The image operation frameworks are also available. There are printing facility APIs are available. The audio technologies support for high-quality audio recording, synthesis, manipulation, and playback. The AV Foundation framework, OpenAL, Core Audio frameworks are available. There is video technologies AVKit, AV Foundation, Core media framework, Core video framework.

## 3.3.6 CORE SERVICE LAYER

The core services provide accounts and social media services such as Facebook and Twitter. It gives the iCloud storage service API where the user can store the documents, key-value pair, and core data storage. Sometimes the user can store the state of the complex game.

The Cloudkit is another important control mechanism to Save, search, and fetch data for specific to an individual user, and in public area shared by all users. CloudKit has minimal caching and relies on a network connection.

The file coordination service manages the file access mechanism when more than one users try to access the same file. If both users access the same file in reading mode, then there is no issue. Whereas, if the first user is trying to save the changes and another user is trying to read the same document then the write operation should be performed to maintain the consistency. If the two users are trying to write the same file, then the locking mechanism is performed, and file coordination plays an essential role. The NSDocument and NSFilePresentor are the two important classes for the programmer.

Bundles and packages are essential mechanisms for software distribution. The Bundle is the single encapsulated entity mechanism for software distribution, whereas there are hierarchical files located inside. The package is also called as the installation package. The package consists of the files and directories inside. User can double click the package and install the app.

There are internationalization and localization support available for the app such as the text, images, and another look of the app is available in the local languages then it will be more attractive. To support several languages, the Unicode is popular nowadays due to which the text can be converted into the local languages. The important factors such as format numbers, currencies, dates, and times need in the local languages. There is some strings resources provision available in this layer for such support.

Similar to the lambda expression in the java, and serverless computing in the Cloud Computing concepts, the closure or block or lambda expression is also available in the C, C++, and Objective C language.

The multiple threading mechanism tries to achieve the fake parallelism on the single CPU capability. However, if the CPUs are with multiple cores, then it is the skill of the programmer. The Grand Central Dispatch (GCD) is a more efficient mechanism similar to the Threads creation, where the dispatch queues are used instead of the Threads. The task can be executed serially or concurrently. The NSOperationQueue class is useful for achieving the multiple queue usage for the task accomplishment. The queue can be synchronous or asynchronous queue. The queues are useful to assign on the particular core of the CPU.

The Bonjour is the implementation of the zero-configuration networking architecture, i.e., a robust system for publishing and discovering services, devices, and computers over an Internet Protocol (IP) network. It assigns the IP without the Dynamic Host Configuration Network Protocols (DHCP). Network-based devices such as printers, scanners, Redundant Array of Independent Disks (RAID) servers, and wireless routers can be configured without using the Domain Name System (DNS) server. Bonjour works on multiple subnets.

The security services provide user authentication, keys, Kerberos, certificates, trust services, user authorization, secure transport, and Smart cards with the CryptoTokenKit framework. This security framework provides all the dimensions of the security, including the privacy, authentication, authorization, integrity, non-repudiation, encryption, decryptions, message digest, and digital signature.

The mapkit framework provides the Display map or satellite imagery directly from the app's interface. The location and map-based services can be efficiently provided based on the longitude and latitude of the location. Also the iBeacon and Bluetooth (BLE) low energy services can contribute to the various applications.

The address book is an essential element of any mobile device to access contacts, emails, street address, etc. quickly. The developer has to integrate and access the address book information. There is a CardDAV protocol for functionality purpose and synchronization.

The critical part is the text to speech, speech to text and speech synthesizer mechanism useful for normal as well as to the person with certain disabilities such as vision, and attention.

There are identity services management and Access Control List (ACL) gives fine-grained access to file-system objects. The time machine support provides the automatically backing up data to a different hard drive, in an efficient way.

Keychain Services provides a secure way to store passwords, keys, certificates, and other sensitive information associated with a user. Users often have to manage multiple user IDs and passwords to access various login accounts, servers, secure websites, instant messaging services, and so on. A keychain is an encrypted container that holds passwords for multiple apps and secure services.

Access to the keychain is provided through a single master password. Once the keychain is unlocked, Keychain Services–aware apps can access authorized information without bothering the user.

The SQLite database which is also used by the other mobile application development platforms such as Android can be used as like a local database in iOS. You can create local database files and manage the tables and records in those files. The library is designed for general purpose use but is still optimized to provide fast access to database records. XMLParser and Distributed notification services are important services in this layer.

## 3.3.7 CORE SERVICE FRAMEWORKS

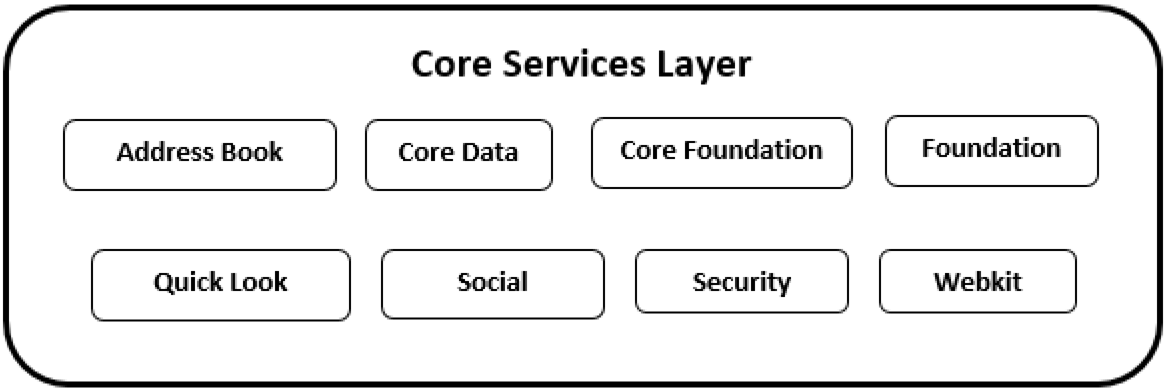

**Figure-70 Components of the Core Services layer**

Launch Services gives a programmatic way to open apps, documents, URLs, or files with a given MIME type in a way similar to the Finder or the Dock. The Launch Services framework also provides interfaces for programmatically registering the document types your app supports. Launch Services is in the Core Services umbrella framework. Figure-70 shows the components of the Core services layer.

The Metadata framework helps you to create Spotlight importer plug-ins. It also provides a query API that you can use in your app to search for files based on metadata values and then sort the results based on certain criteria. (The Foundation framework offers an Objective-C interface to the query API.)

Search Kit provides the fast search, summarize, and retrieve documents, and contents written in most human languages in an app. This framework is part of the Core Services umbrella framework.

Web Services Core provides support for the invocation of web services using CFNetwork. The available services cover a wide range of information and include things such as financial data and movie listings. Web Services Core is part of the Core Services umbrella framework.

Dictionary Services lets you create custom dictionaries that users can access through the Dictionary app. Through these services, your app can also access dictionaries programmatically and can support user access to dictionary look-up through a contextual menu.

The Accounts framework (Accounts.framework) provides a single sign-on model for supported account types such as Twitter and Facebook. Single sign-on improves the user experience because it prevents your app from having to prompt a user separately for login information related to an account. It also simplifies the development model for you by managing the account authorization process for your app.

There is an address book, Automator, core data, Event Kit, Foundation and core foundation, quick look, social framework, storekit, and WebKit are few more to mention in this list.  The Other Frameworks in the Core Services Layer includes Collaboration, Input Method Kit, and Latent Semantic Mapping

## 3.3.8 CORE OS LAYER

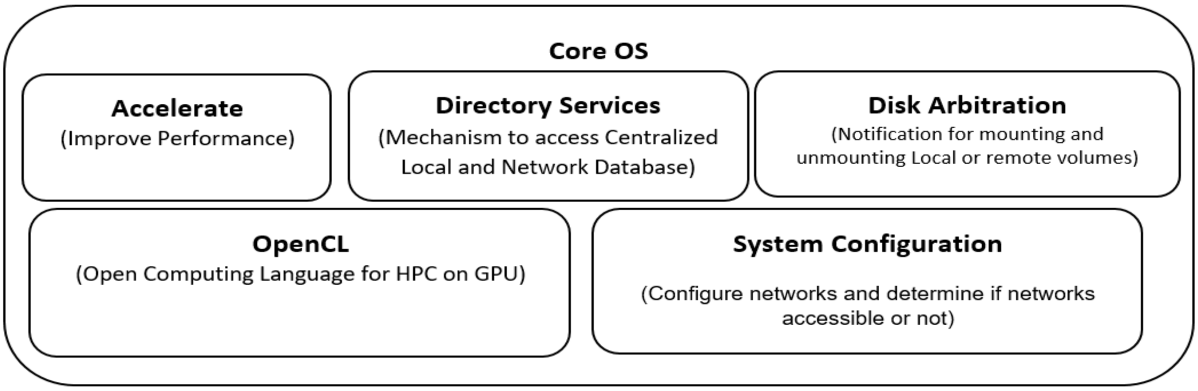
Core OS
Accelerate
(Improve Performance)
Directory Services
(Mechanism to access Centralized Local and Network Database)
Disk Arbitration
(Notification for mounting and unmounting Local or remote volumes)
OpenCL
(Open Computing Language for HPC on GPU)
System Configuration
(Configure networks and determine if networks accessible or not)

**Figure-71 Components of the Core OS**

The core OS Layer is low-level services related to hardware and networks. These services are based on facilities in the Kernel and Device Drivers layer. There are high-level app security features such as gatekeeper, app sandbox and code signing. The gatekeeper keeps the app safe by avoiding the installations of apps other than the Mac app store and unknown developers. The app must have the Developer ID certificate and code sign your app. Whenever someone developing a new app the necessary Developer ID and code sign must be done from the Mac Store. Figure-71 shows the components of the core OS.

The app sandbox provides the security to the app from stolen, corrupted, or deleted user data if malicious code exploits your app. The code signing is useful to avoid the accidental code change of the app, and it protects from the malicious code change. The X-code IDE facilitates the necessary security through the various options.

The Accelerate framework contains APIs that helps to accelerate complex operations. Hardware-based vector units boost the performance of any app that exploits data parallelism, such as those that perform 3D graphics imaging, image processing including the digital signal processing, video processing, audio compression, and software-based cell telephony. The linear algebra and other computationally expensive mathematical operations are performed using the vecLib and vImage libraries.

Disk arbitration framework is useful for the disk mounting and unmounting operations.

The Open Computing Language (OpenCL) makes the high-performance parallel processing power of Graphics Processing Units (GPUs) available for general-purpose computing. The OpenCL can also be useful for the CPU and GPU combination utilizing the vector and scalar operations. The parallel operations using data parallelization for the various applications such as signal processing, image manipulations, finite element analysis can be successfully performed using OpenCL.

Open Directory is a directory service about users, groups, computers, printers, and other information that exists in a networked environment in local or network database. Open Directory provides some APIs to interact with the Light-weight Directory Access Protocols (LDAP), NetInfo, AppleTalk, SLP, Server Message Block (SMB), Domain Name System (DNS), Microsoft Active Directory, Bonjour protocols and few more.

The system configuration is the framework useful in case if the app is trying to connect the particular network. The present situation of the network to be compared as well as the dynamic changing parameters of the network are identified and used using the system configuration framework. It allows the necessary configuration of the network before the connection to the network.

## 3.3.9 KERNEL AND DEVICE DRIVERS

The inter-process communication (XPC) is integrated with Grand Central Dispatch (GCD).  The Cache APIs are important for the performance of the app. The cache APIs are useful to manage the available memory for all kind of operations. If the demand exceeds the available memory, then free the memory efficiently is possible through the libcache. There is kernel level video capture APIs where the important class IOVideoDevice is useful written in C++.

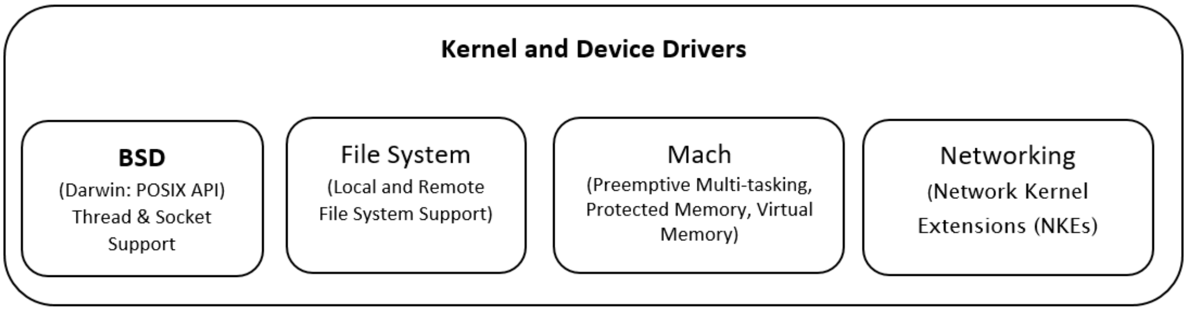
Kernel and Device Drivers
BSD
(Darwin: POSIX API) Thread & Socket Support
File System
(Local and Remote File System Support)
Mach
(Preemptive Multi-tasking, Protected Memory, Virtual Memory)
Networking
(Network Kernel Extensions (NKEs)

**Figure-72 Components of the Kernel and Device Driver Layer**

The basic OS operations such as provided through the Mach such as pre-emptive multitasking, protected memory, virtual memory operations, and low latency real-time operations are supported. Figure-72 shows the components of the kernel and device drivers.

There is device driver support in the form of the C++ APIs for the input and output operations such as plug and play, dynamic device management, power management features. The Network Kernel Extensions (NKE) provides control over the data-link and networking layer of the ISO-OSI model for the monitoring and modifying the traffic. The NKE provides the notifications for the changed conditions of the network.

There is Berkeley Software Distribution (BSD) OS which provides the process model such as process IDs, signals, and so on. Necessary security policies such as file permissions, user and group IDs, threading support (POSIX threads) and networking support (BSD sockets).

The IPC and notification related services such as file system events, kernel queues and kernel events, BSD notifications, Sockets and Ports, Streams, Pipes, shared memory, etc. are also part of the Kernel.

There is kernel support for protocols and services such as Dynamic Host Configuration Protocol (DHCP), Transport Layer Security (TLS), Hyper-text Transfer Protocol (HTTP), HTTP with Security (HTTPS), Point to Point Protocol (PPP), SLP, SSH, TCP/IP, UDP/IP, and Remote Procedure Call (RPC). The support for the network technologies such as Ethernet with all versions, routing, network diagnostics, a file system with various formats, and file sharing protocols.

There is scripting language support like through the Darwin for the Perl, Python, Ruby, Ruby on Rails and others. The cocoa classes can be accessed through the scripting languages.

## 3.4 APPLICATIONS AND TASK MANAGEMENT IN IOS

The task management in iOS is Manage your app's work and how it interacts with system services like Handoff. The operations such as undo and redo can be performed using UndoManager. To perform this undo and redo operation, it has to be registered first for the Undo operations. The to revert to the state of the last undo operation, the undo () method is called. UndoManager saves the operations you reverted to so that you can call redo() automatically.

The is the possibility that the task can progress slowly. The progress of the task can be reported to the user using the class ProgressReporting. There is the Operation, OperationQueue, and BlockOperation class which represented the code and data of the task. It also shows the state of the operations such as started, finished, executing, or waiting. The Timer is the class to fire after a certain time interval has elapsed, sending a specified message to a target object. The timers work with loops for the scheduling purpose and can be validated or invalidated.

The activity sharing called as the hand-off can be performed for task management. The hand-off can create, send, and receive user activities directly. Generally, the hand-off terminology is used in the mobile communication when the handset owner (mobile station) can travel and relocate from one hexagonal cell to the another then the hand-off can happen between the two base stations. Here in case of the Apple iOS it is hand-off of the app's operations to continue between the devices. There is system interaction, and the user notification classes are part of the task management in iOS.

## 3.5 JAILBREAKING

There are many happy and satisfied users of iPhone. They happily use the available features but there are many other users, those who wish to use the other restricted apps on their iPhone. They try to take control on the phone using the process of jailbreaking. The restrictions usually emphasize to install apps only from the app store. There are several reasons to jailbreak such as, if you are owner of the phone you want to customize it, install the apps other than default apps, want to change the look and feel of the phone.

There are certain restricts and rules defined by the Apple for app installation. These are provided in the form of the level of the user escalations. However, users may use the kernel patches to access the root user to install the apps that are not available on the Apple App Store. Formally, this kind of practice is not supported by Apple, by stating that the device may go to the unstable state. The jailbreaking can be tethered, un-tethered, semi-teetered, or semi-untethered. The activity of jailbreaking is illegal in few countries. Jailbreaking modify a smartphone or other electronic device to remove restrictions imposed by the manufacturer or operator, e.g. to allow the installation of unauthorized software, and to get more control on the device. However, Jailbreaking finishes the contract with the vendor, and invalidate your warranties. [5]

## 3.6 DIFFERENCE BETWEEN ANDROID OS AND IOS

| Features | Android OS | iOS |
|---|---|---|
| Developer Website | https://developer.android.com | https://developer.apple.com |
| IDE | Android Studio | Xcode |
| Programming Language | Java, Kotlin, C++ | Swift Compatible with Objective-C |
| Current Version | Android 9 Pie,<br>Android Q (Beta Version April 2019) | iOS 12.2 (March 2019) |
| Consistency and Stability | Consistent and Stable | ➢ Better Consistency and stability.<br>➢ Phone hanging or the crashing is very less in iOS.<br>➢ Refined experience |
| Notification | Better notification | - |
| Phone price range | Available in all price range | Premium expensive phones |
| Customization | Better Customization | Low to Moderate Level |
| Power Management | Recent version having better battery optimizations | Better battery optimization |
| Updates | Frequent updates but available in some intervals | Not frequent but available to the user on an immediate basis |
| Touch Functionality | 2D Touch Functionality | 3D Touch Functionality |
| Security | Linux kernel security | No worry about malware. |
| Notable features | ➢ Better AI Assistant<br>➢ More functionality<br>➢ Many choices | |

**Table-10 Difference between Android and iOS**

## 3.7 SECUIRTY FEATURES IN IOS

The Ios, all apps share the same sandbox. It has the code signing technology, and hardware encryption. It do not have external storage and it is difficult for unwanted code to access built in storage. iOS do not require antivirus since there is checking been done in the apps store [1] [2] [3] [4].

## 3.8 LET US SUM UP

- The iOS compared with the Android in general and the architecture.
- The architecture of the Android and iOS are more or less similar however the kernel development approaches are opposite i.e. Android having monolithic modular and iOS micro-kernel.
- The quality in the iOS app development is high priority whereas the recent android versions releases have given interesting features.
- The iOS architecture is having five layers including Coca touch, media layer, core services layer, core operating system layer and kernel and Device Driver layer. The detailed discussion on the sub-components and libraries of the iOS architecture has been done.
- Concept of Jailbreaking is discussed.

## 3.9 CHECK YOUR PROGRESS

- **Answer the following in True or False.**

1. The closure or block in Objective C is similar to the anonymous function in Java.
2. Network Kernel Extensions can take control on the Data link and Network layer.
3. Grand Central Dispatch is not performing well over the threading system.
4. CryptoTokenKit is useful for the geographical location monitoring.
5. OpenCL can extend the CPU processing functionality up-to GPU.
6. Address Book, Webkit, Social are the components of the core services layer.

- **Answer the following Questions in short.**

1. What are the dis-advantages of jailbreaking iPhone?
2. What is the difference in architecture of the Android and iOS?
3. What are the libraries available in Media layer?
4. Explain the different classes for the app and task management?
5. What is the code signing, how it affects the security of the app?
6. What is difference between multi-threading and GCD?

## 3.10 CHECK YOUR PROGRESS: POSSIBLE ANSWERS

- **True/False Answers**

1. True
2. True
3. False
4. False
5. True
6. True

## 3.11 FURTHER READING

1. Toby Boudreaux, Programming the iPhone User Experience, O'Reilly Safari, available at https://www.oreilly.com/library/view/programming-the-iphone/9780596805760/ch01.html [Accessed on 30th April 2019].
2. https://developer.apple.com/library/archive/documentation/MacOSX/Conceptual/OSX_Technology_Overview/MediaLayer/MediaLayer.html [Accessed on 30th April 2019].
3. https://developer.apple.com/library/archive/documentation/MacOSX/Conceptual/OSX_Technology_Overview/SystemTechnology/SystemTechnology.html#//apple_ref/doc/uid/TP40001067-CH207-BCICAIFJ [Accessed on 30th April 2019].
4. https://www.appcoda.com/grand-central-dispatch/ [Accessed on 30th April 2019].

## 3.12 REFERENCES

1. Mohd Shahdi Ahmad, Nur Emyra Musa, Rathidevi Nadarajah, Rosilah Hassan and Nor Effendy Othman, (2013). Comparison Between Android and iOS Operating System in terms of Security, 8th International Conference on Information Technology in Asia (CITA), IEEE.
2. Fattoh AI-Qershi, Muhammad AI-Qurishi, Sk Md Mizanur Rahman, and Atif AI-Amri, .(2014). Android vs. iOS: The Security Battle, IEEE.
3. Muneer Ahmad Dar,Syed Nisar Bukhari,Ummer Iqbal Khan,(2018), Evaluation of Security and Privacy of Smartphone Users, 4th International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB-18), IEEE.
4. Tae Oh, Bill Stackpole, Emily Cummins, Carlos Gonzalez, Rahul Ramachandran, Best Security Practices for Android, BlackBerry, and iOS, 2012 The First IEEE Workshop on Enabling Technologies for Smartphone and Internet of Things (ETSIoT),pp. 42-47.
5. Feng Liu,Ke-sheng Liu, Chao Chang, Yan Wang, (2016). Research on the technology of iOS jailbreak, Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control, IEEE, PP. 644-647.
6. Ovidiu Constantin Novac, Mihaela Novac, Cornelia Gordan, Tamas Berczes, Gyöngyi Bujdosó (2017). Comparative Study of Google Android, Apple Ios and Microsoft Windows Phone Mobile Operating Systems, 14th International Conference on Engineering of Modern Electric Systems (EMES), pp. 154-159.