UNIT 4 CASE STUDY - iOS

Structure

4.0 INTRODUCTION

iPhone operating system(iOS) is a mobile operating system developed by Apple Inc. used for apple handheld devices. It is used in devices like- iPhone, iPad and iPod. It is the second most widely used mobile operating system. It supports features like direct manipulation and can respond to various types of user gestures. It is proprietary and closed source and derived from macOS. Software development kit (SDK) is provided to create applications. It includes interfaces for developing, running and testing applications. Apps can be written using system frameworks and Objective-C programming language.

Initial three versions were introduced with the name iPhone OS. From fourth version, they renamed it to iOS. With each new version, new features and apps were added. iOS 13 is latest version which was released in 2019. Its major feature is dark mode and new Map application with street view capability. It also enhanced its previous apps and features like-Siri, Heath map and others. iOS 14 is about to release in 2020.

4.1 OBJECTIVES

After going through this unit, you should be able to:

- Understand the basic functions of iOS Operating System
- Know the history of iOS operating system
- Understand the process management in iOS and can compare with other OS
- Understand the memory management approaches in iOS
- Understand the File management in iOS
- Understand the security features in iOS

4.2 FEATURES OF IOS

iOS is the operating system for iPhone, iPad and other Apple mobile devices. Based on Mac OS, the operating system which runs Apple's line of Mac desktop and laptop computers, Apple iOS is designed for easy, seamless networking between Apple products. iOS support extremely good features and some of the common features are:

- Multitasking– it allows multipletasksto be executed concurrently. This feature allows various applications to run inbackgroundlike notifications, VoIP, audio, Bluetooth access, app updates and may more.
- SpringBoard – It is used for managing home screen.
- Gesture recognition
- Wifi, Bluetooth and support for VPN
- Access to Apple App store
- Support for integrated search – it allows files to be search simultaneously
- Safari browser
- Front and read camera with videos capturing facility
- Siri – it is an intelligent personal assistant feature which cantake voice queries and cangivevoiceresponses and recommendations, used for setting reminders etc.
- Game center – it is multiplayer gaming networkavailable online.
- Compatible with iCloud – iCloud is cloud service provided byApple.
- Push email service – Apples's email server allows mails to be delivered as theyarrive.
- Accelerometer, gyroscope, and magnetometer – these are sensor interfaces used to listen various events.
- Apple pay – it is payment technology which can store credit card details to pay for services.

- Services like Maps, contacts, web pages, messages, location
- Various security features – face ID, pass code, 2 factor authentication.
- HomePod – it can identify family members by voice and can handoff calls , music etc on other devices.
- HomeKit – it is home automation controlling system.
- CarPlay – this allows interacting with iOS during drive. Also allows access tom phone apps.

Apple keeps making iOS 14 and iPadOS 14 even better by adding valuable new capabilities and features. Most recently, Apple released iOS 14.4, adding a new workout to Fitness Plus for Apple Watch owners. It also added a new Unity watch face in February to celebrate Black History Month. The update also included a series of security fixes for vulnerabilities that were activelybeing exploited.

iOS 14.4 follows the addition of Apple ProRaw photos to the iPhone 12 Pro and 12 Pro Max. Those new features join an already impressive list of capabilities that Apple brought to its mobile devices with the release of iOS 14 in September. iOS 14.5 is currently available in beta and is shaping up to be a significant update for iPhone owners.

4.3 EVOLUTION OF IOS

iOS was first introduced with iPhone in Jan 2007 and released in June 2007. At first introduction Steve Jobs claimed it to be running OS X and running desktop class application but during release introduced it with the name "iPhone OS". Initially, there was no support for third partyapplications. In March 2008,Apple announced software development kit for iPhone. In July 2008, iOS app store opened with 500 applications which increased to 3000 in Sept 2008 and after successive growth through the years increased to 2.2 million in 2017. It is also estimated to reach 5 million by 2020. iOS has seen a lot of changes since its inception. Following are the important milestones in iOS:

- iOS was first introduced with iPhone in jan 2007 and released in june 2007.
- Apple's iOS first SDK was released on March 6, 2008.
- The initial release was named iPhone OS which was later changed to iOS on June 7, 2010.
- iPhone OS 1 was released on March 6, 2008, and is the first version of the popular operating system. The support for iPhone OS 1 ended after two years, i.e., 2010.
- iPhone OS 2, as the name suggest, is the 2nd big release for the iOS. The release was done inconjunction withiPhone 3G, and anyone with the previous version can easily upgrade to the latest version. Also, this version introduced the App store, which becomes the hub for installing new apps. New SDK was also released for developers with support ending in 2011.
- The third big release was Apple iOS 3. It came into existence in June 2009 with support ending in late 2012. New features such as copy, paste, etc. are added to the OS.

The next version is iOS 4 and is released on June 21, 2010. Clearly, this is one of the big releases for iOS as it dropped old device support instead of supporting the latest devices with multitasking features.

- iOS 5 was released on June 6, 2001. It brought support for iPad Touch (3rd generation) and iPad (1st generation).
- iOS 6 went live on September 19, 2012, for the 4thgenerationApple devices.
- iOS 7 was released for public on September 18, 2013. It supported two new phones by Apple, the Apple iPhone 5S and iPhone 5C.
- Just like the old release, iOS 8 released for public on September 9, 2014, with support for their best phone devices, the iPhone 6 and iPhone 6 Plus. They dropped support for older devices.
- iOS 9 was made public on September 16, 2015. Apple changed how they support legacyhardware and iOS 9 became the first Apple OS that supported 22 devices.
- iOS 10 was announced on June 13, 2016 at WWDC(Worldwide Developers Conference event and was released to public in September, 2016 along with iPhone 7 and iPhone 7 plus.
- iOS 11 was made public on September, 2017 along with iPhoen 8 and iPhone 8 Plus. It has dropped 32-bit applications making iOS 11 as a 64-bit OS that only runs 64-bit apps.
- iOS 12 was made public on September 2018 along with iPhone XS, iPhone XS Max, iPhone XR.
- Apple announced iOS 13 and made public on September, 2019. The principal features include Dark Mode and Memoji support. The NFC(Near Field Communication) framework supports reading several types of contactless smartcards and tags.
- Apple released iOS 14 and iPadOS 14 on 9th July 2020. All devices that support iOS 13 also support iOS 14. Some new features includes widgets that be nowbe placed directly on the home-screen, along with the App library which automatically categorizes apps into one page, Picture in Picture, Car- key technology to unlock and start a car with NFC. It also allow the user to have incomingcalls shown in banners rather than taking up the whole screen. As on date (March, 2021) 14.1 is available in beta 3 version.

4.4 ARCHITECTURE OF IOS

iOS architecture is written in Objective-C language and comprised of four layers(layered architecture). It consists of a stack of four layers – Core OS, Core services, media layer, and Cocoa Touch as shown in Figure 1. Apps installed on system communicate with the iOS which in turn communicates with the hardware. Bottom level layers in the iOS stack are responsible for basic services while the upper layers are responsible for providing interfaceand graphics designing.

Apple provides systeminterfaces called **Frameworks** which is a package that stores dynamic shared libraries. It contains various resources like – header files, supported

apps, images. Each layer of iOS contains different set of frameworks that can be used to developers to design apps.

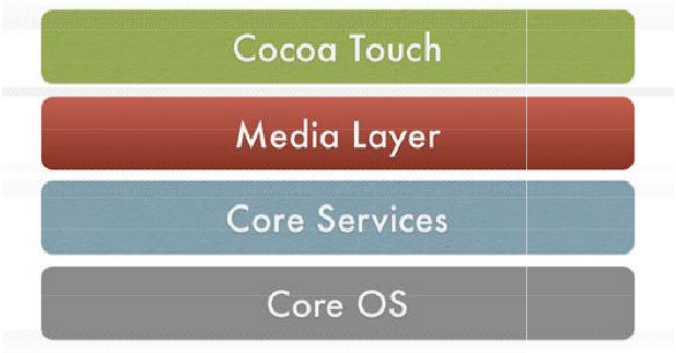

Fig 1. iOS Layered Architecture

Let us learn more about the layered architecture shown in the above figure.

4.4.1 Core OS

It is the lowest layer in the iOS architecture hence responsible for providing basic low level features. The frameworks present in this layer are:

- Securityservices framework
- Authentication framework
- Bluetooth framework
- External accessoryframework
- Accelerate framework

4.4.2 Core Services

It is the layer at the top of core Os layer. The various frameworks available at this layer are:

- Core data Framework - It is used for managing Model View Controller (MVC) app.
- Core Foundation framework - It gives interfaces used to manage data, services and features.
- Address book framework - Used for accessing u er's contacts database.
- Cloud Kit framework -Allows movement of data between apps and iCloud.
- Core Location framework - Used to give location and heading information to apps.
- Core Motion Framework - Used to access motion based data and accelerometer based information.

- Healthkit framework - Used to handle health oriented information of user
- Homekit framework - Used to connect and control user's home devices.
- Social framework - Used to handle social media accounts of user.
- StoreKit framework - Used to provides In -App Purchase.

4.4.3 Media Layer

This is the third layer from the bottom. It is used for providing audio, video and graphics features.

Audio Frameworks

- Media Player Framework - It is used to handle user's playlist.
- AVFoundation - It is used for recording and playback of audio and video.
- OpenAL- It is standard technology for providing audio. Video Frameworks
- AVKit - This interface is used for video presentation.
- AV Foundation - It gives advanced video recording and playback capability.
- Core Media - It is used to describe low level data types and interfaces for operating media.

Graphics Frameworks

- Core Graphics framework - It is used for custom 2D vector and image based rendering.
- Core Animation - It is used for adding animation in apps.
- Core Images - It is used for gives for controlling video and images
- UIKit Graphics - It is used for designing images and animating content of views.
- OpenGl ES and GLKit - It is used for managing2D and 3D rendering.

4.4.4 Cocoa touch layer

It is the top most layer in iOS architecture. It is used for providing interactive interfaces to user. The various frameworks present in this layer are:

- UIKit Framework – It is used for designing graphical, event-driven apps.
- EventKit framework - It is used for providing systeminterfaces for viewing and altering calendar related events
- GameKit Framework - It allows users to share game related information online.
- MapKit Framework - It allows maps to be included in user interface.
- PushKitFramework- It provides support for VoIP apps.
- Twitter Framework - It allows access to Twitter service.

4.5 iOS KERNEL ARCHITECTURE

iOS kernel is XNU kernel of Darwin. Initially it was used only for MacOS and later introduced as free and open source part of Darwin operating system. It is a hybrid kernel which supports both monolithic and microkernel.

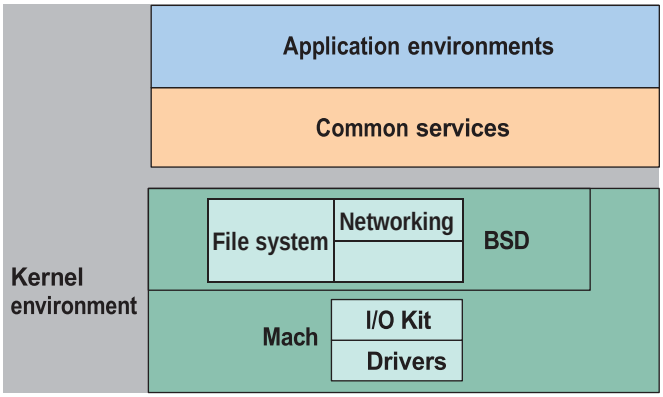

Fig 2: XNU kernel Architecture
(Source: developer.apple.com)

The foundation layer of Darwin and OS X is composed of several architectural components, as shown in Figure 2. These components form the *kernel environment*. In OS X, however, the kernel environment contains much more than the Mach kernel itself. The OS X kernel environment includes the Mach kernel, BSD, the I/O Kit, file systems, and networking components. These are often referred to collectively as the kernel. Each of these components is described briefly in the following sections. For further details, refer to the specific component chapters or to the reference material listed in the bibliography.

Because OS X contains three basic components (Mach, BSD, and the I/O Kit), there are also frequently as many as three APIs for certain key operations. In general, the API chosen should match the part of the kernel where it is being used, which in turn is dictated bywhat your code is attemptingto do. The remainder of this chapter describes Mach, BSD, and the I/O Kit and outlines the functionality that is provided by those components.

Mach

Mach manages processor resources such as CPU usage and memory, handles scheduling, provides memory protection, and provides a messaging-centered infrastructure to the rest of the operating-system layers. The Machcomponent provides untyped interprocess communication (IPC), remote procedure calls (RPC), scheduler support for symmetric multiprocessing (SMP), support for real-time services, virtual memory support, support for pagers and modular architecture.

BSD

Above the Mach layer, the BSD layer provides "OS personality" APIs and services. The BSD layer is based on the BSD kernel, primarily *FreeBSD*. The BSD component provides file systems, networking(except for the hardware device level), UNIX security model, syscall support, the BSD process model, including process IDs and signals, FreeBSD kernelAPIs, many of the *POSIX* APIs, kernel support for *pthreads* (POSIX threads).

Networking

OS X networking takes advantage of BSD's advanced networking capabilities to provide support for modern features, such as Network Address Translation (*NAT*) and *firewalls*. The networking component provides 4.4BSD TCP/IPstack and socket APIs, support for both IP and DDP (AppleTalk transport), multihoming, routing,

multicast support, server tuning, packet filtering, Mac OS Classic support (through filters).

File Systems

OS X provides support for numerous types of file systems, including *HFS, HFS+, UFS, NFS, ISO 9660,* and others. The default file-system type is HFS+; OS X boots (and "roots") from HFS+, UFS, ISO, NFS, and UDF. Advanced features of OS X file systems include an enhanced Virtual File System (*VFS*) design. VFS provides for a layered architecture (file systems are *stackable*). The file system component provides *UTF-8* (Unicode) support and increased performance over previous versions of Mac OS.

I/O Toolkit

The I/O Kit component provides true plug and play, dynamic device management, dynamic (on-demand) loading of drivers, power management for desktop systems as well as portables and multiprocessor capabilities.

4.6 PROCESSES AND THREADS MANAGEMENT

Everyprocessinanapplicationis created usingone or more threads.Athread represents single path of execution. Executionof a process starts with a single thread and canlater spawn multiple threads, where each thread performs a specific task.

All threads of a single process have same access rights and also shares same address space (virtual memory). They can also communicate with each other and other processes. Every thread has its own stack of execution and scheduled for execution by kernel separately.IOS has the capability to execute multiple programs parallel but most of the execution is done in background and does not require continuous processing. The application that runs in foreground keeps the processor and other resources busy.

4.6.1 Threading Packages

Applications may require creation of threads. IOS supports thread management and synchronization along with other new technologies. The technologies used to manage thread in iOS can be POSIX threads or Cocoa threads.

POSIX threads are based on C language. This must be used when application needs to be designed for multiple platforms. Here threads are managed using *pthread* library. Communication between threads can be done using ports, shared memory or conditions. For Cocoa applications, threads are created using NSThread class. It allows detached thread to be created. Detached thread is the one in which system automatically reclaims thread's resources when it terminates.

4.6.2 Threads Alternatives

Threadsarelowleveldoingconcurrencyand managingsynchronizationbecomesdifficult for application development. Also the use of thread adds processor and memory overheads. iOS henceoffersalternativesto thread programming. Executionofconcurrent tasks or processes on iOS is managed by asynchronous designing approach like traditional thread approach. Applications should only define specific tasks instead of creating threads as threads are low level and should leave it to system. Threads when

managed by system, gives scalability to applications. The various techniques provided by iOS to manage tasks asynchronously are:

- **Grand Central Dispatch (GCD)**

  In this approach user only defines the task to be done and add it to the appropriate dispatch queue. GCD then creates threads as needed and schedules them. Since in this approach thread management and execution is done by system, it is more efficient than traditional threads approach.

- **Operation Queues**

  They are similar to dispatch queue and are object-C objects. User defines the task to execute and add it to operation queue. Scheduling and execution is then done by operation queue.

Dispatch queues: Dispatch queue are used for execution of customized tasks. It executes the task serially or concurrently in first-in first-out order. Serial dispatcher waits for the task to complete before de-queuing, whereas concurrent dispatcher does not wait for the tasks to finish before starting newtasks.

Operation Queues: In operation queues the order of execution of task is dependent upon many factors. One such factor is the dependence of one task on the completion of another task. User when defining a task should also configure dependency.

The various advantages of using dispatch and operation queues in place of threads offer various advantages:

1. No need to store thread stack hence reduces memory requirements
2. User need not to configure and manage threads.
3. User need not to schedule threads.
4. It is easy to write code as compared to threads.

4.7 MEMORYMANAGEMENT

For effective and efficient utilization of memory, memory management is required by the iOS. Resource needs to be free up when they are not required. In iOS memory used by applications is managed by maintaining the life cycle of objects and freeing them when not required. To identify that an object is no more required; its object graph is created. The group of objects that are related to each other forms **object graph**. In this, objects can be related bythe direct reference or indirect reference.

4.7.1 Application Memory Management

Applications memory management can be done by Object - C by two methods:

i. Manual Retain Release – In this method user can themselves manage memory by keeping track of objects created by them. This is implemented using **Reference Count Model.** (for versions till iOS-5)

i. Automatic Reference Counting– In this feature, provided bycompiler, system uses Reference counting model by calling methods for user at compile time. (for new versions)

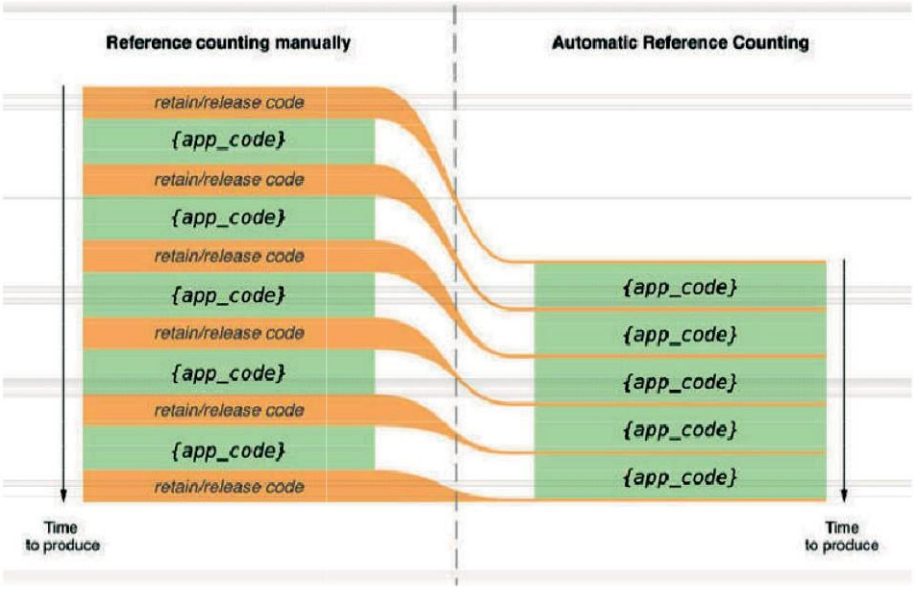

Fig3: Difference betweenAutomatic and Manual Reference Counting

(Source: developer.apple.com)

Memory management using Reference counting is based on Ownership of object. An object can be owned by one or many owners. The object continues to exist till the time it has atleast one owner. The object is destroyed automaticallyat runtime if has no owner. Hence, **ownership** is the ability of causing object to be destroyed.

a. Reference Count Model

Memory is managed by maintaining life cycle of an object. When the objects are not needed, it should be de-allocated. When an object is created or copied, its retain count is set to 1. This retaincountcanincreasewithtime ifownershipinterestisexpressed by other objects. The retain count can also be decremented if objects relinquish their interest of ownership. The object will be maintained in memory till it's retain count is not zero. The object will be deallocated once it's retain count reaches zero.

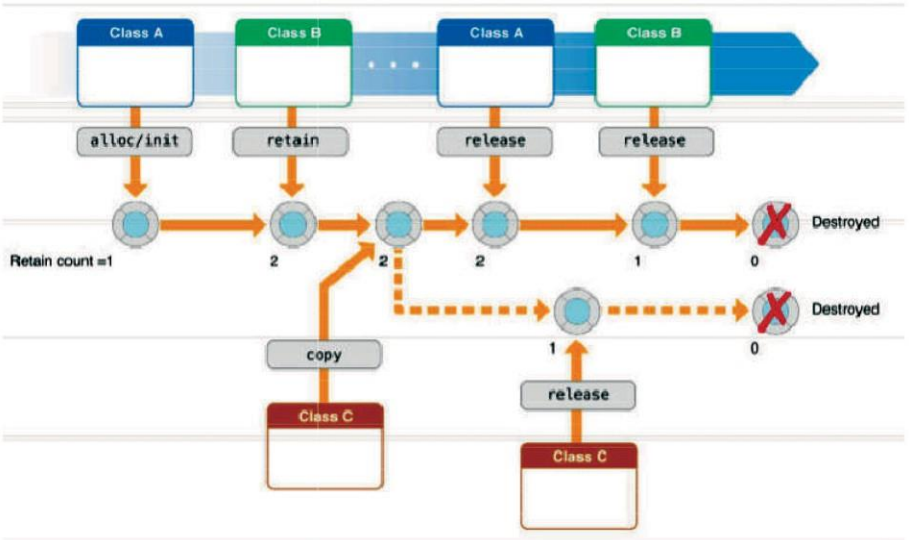

Fig 4: Reference Counting Model (Source: developer.apple.com)

b. Automatic Reference Counting

Automatic reference counting is a compiler provided feature that manages memory automatically of Object-C objects. Automatic reference counting is an ownership system which provides convention for management and transfer of ownership. It itself maintains

the object's lifetime requirements and user need not to remember about retain and releases. It allows object to be pointed by two types of references- strong and week. Anobject whichis strongly referenced will be preserved and never deleted. The objects having back reference is called weak reference and can be de-allocated (destroyed). When no strong reference to objects is left, the object automatically becomes nil. Hence weak reference does not increase life time of objects. This model also imposes some restricts to prevent memory faults or issues.

4.7.2 Virtual Memory Management

Virtual memory overcomes the limitation of physical memory. For each process a logical address space is created by dividing the memory into pages. Apage table is maintained by Memory management units to map logical address to physical address. Logical address is provided to each process but if application wants to access a page withno physicaladdress, page fault willoccur. Virtual memorysystemis then responsible for invokinga specialprogramcalled pagefault handler. The pagefault handler suspends the currently executing process, identifies free memory location and load the required page fromthe disk. The correspondingpage table entry is then updated and canbe used by returning control to the suspended process. This process of loading the page from disk is called Paging. There is no backing store in iOS hence pages cannot be moved out fromprimary memory (page write) hence, if no free space is available in primary memorythan read onlypages can be removed from memory which can be later loaded.

Page size in older versions of iOS is 4KB. While the newer version A9 supports 16KB pages.

Efficient use of memory is important for high performance of system. Reducing the amount of memoryusage decreases memoryfootprint of applications and also reduces processor consumption. Afullyintegrated virtual memory is included with IOS which is always on. For 32- bit system it can provide upto 4 GB addressable space.

Data present on the disk (code pages) which is read only, can be removed from the primary memory and can later be loaded from the disk when needed, whereas, the iOS does not remove the writable data from memory. If the free memory reduces below a certain threshold, the running applications are asked to free up memory to make roomfor new applications or data voluntarily.

4.7.3 Page List in Kernel

Kernel maintainsthree lists of physical memory pages:

i. Active list - this list contains pages which are recently accessed

ii. Inactive list - this list contains pages which are not recentlyaccessed but still present inphysical memory.

iii. Free list - this list contains pages which are not currently allocated and is available for use by processes.

The pages that are not accessed are moved from active to inactive list by kernel. Also when the free memory reduces below a threshold, the pages that at unmodified and inactive are flushed bykernel and runningapplications are asked to freeup memory by sendingthemlowmemorynotification.Applicationsuponreceivingnotificationremove strong references to as many objects as possible.

4.7.4 Page Fault

It is the process of loading the page into primary memory when needed by a process. This process is required as a result of page fault. The two types of fault that can occur are:

i. Soft fault - when the desired page is not mapped to the address space of process but it is present in physical memory.

i. Hard fault - when the desired page is not present in the physical memory.

Whenever page fault occurs, kernel finds the VM object for accessed region and checks the resident list of VM object. If the required page is found in resident list, then soft fault isgenerated otherwise hard fault is generated. For handlingsoft fault, physical memory of page is mapped to the virtual memory and the page is marked as active by kernel. For handling hard fault, pager of VM object finds the page from disk and updates its map entry. The page after loading in primary memory is marked as active.