CMER

Centre for Mobile Education and Research

Mobile Operating Systems

Week I

Overview

- Introduction
- Mobile Operating System Structure
- Mobile Operating System Platforms
- Java ME Platform
- Palm OS
- Symbian OS
- Linux OS
- Windows Mobile OS
- BlackBerry OS
- iPhone OS
- Google Android Platform

OS Features

- Features
  - Multitasking
  - Scheduling
  - Memory Allocation
  - File System Interface
  - Keypad Interface
  - I/O Interface
  - Protection and Security
  - Multimedia features

Introduction

- Design and capabilities of a Mobile OS (Operating System) is very different than a general purpose OS running on desktop machines:
  - mobile devices have constraints and restrictions on their physical characteristic such as screen size, memory, processing power and etc.
  - Scarce availability of battery power
  - Limited amount of computing and communication capabilities

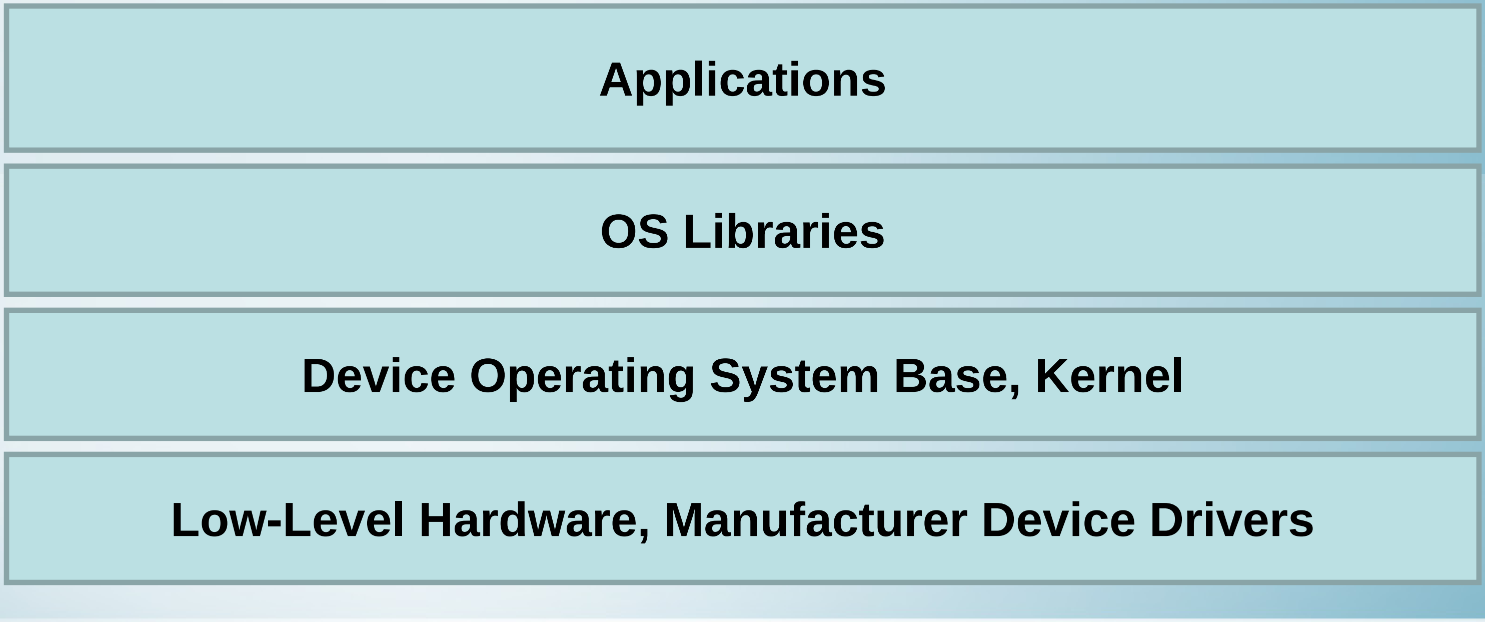
Introduction (Cont.)

- Thus, they need different types of operating systems depending on the capabilities they support. e.g. a PDA OS is different from a Smartphone OS.

- Operating System is a piece of software responsible for management of operations, control, coordinate the use of the hardware among the various application programs, and sharing the resources of a device.

Operating System Structure

- A mobile OS is a software platform on top of which other programs called application programs, can run on mobile devices such as PDA, cellular phones, smartphone and etc.



Applications

OS Libraries

Device Operating System Base, Kernel

Low-Level Hardware, Manufacturer Device Drivers

Mobile Operating System Platforms

- There are many mobile operating systems. The followings demonstrate the most important ones:
  - Java ME Platform
  - Palm OS
  - Symbian OS
  - Linux OS
  - Windows Mobile OS
  - BlackBerry OS
  - iPhone OS
  - Google Android Platform

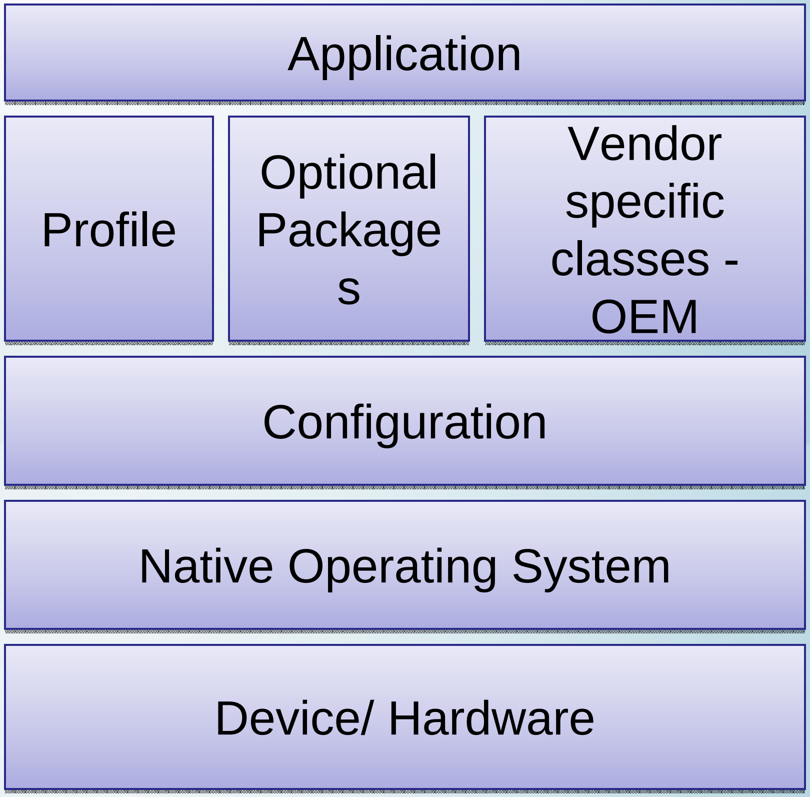
Java ME Platform

- J2ME platform is a set of technologies, specifications and libraries developed for small devices like mobile phones, pagers, and personal organizers.

- Java ME was designed by Sun Microsystems. It is licensed under GNU General Public License

Java ME Architecture

- Java ME platforms are composed of the following elements:



Application

Profile

Optional Package s

Vendor specific classes - OEM

Configuration

Native Operating System

Device/ Hardware

Java ME Main Components

- **Configuration: it defines a minimum platform including the java language, virtual machine features and minimum class libraries for a grouping of devices. E.g. CLDC**

- **Profile: it supports higher-level services common to a more specific class of devices. A profile builds on a configuration but adds more specific APIs to make a complete environment for building applications. E.g. MIDP**

Java ME

- Optional Package: it provides specialized service or functionality that may not be associated with a specific configuration or profile. The following table lists some of the available packages:

Packages	Description
JSR 75 - PIM	PDA Package
JSR 82 - BTAPI	Java APIs for Bluetooth
JSR 120 - WMA	Wireless Messaging API
JSR 172	J2ME Web Service
JSR 179	Location API for J2ME

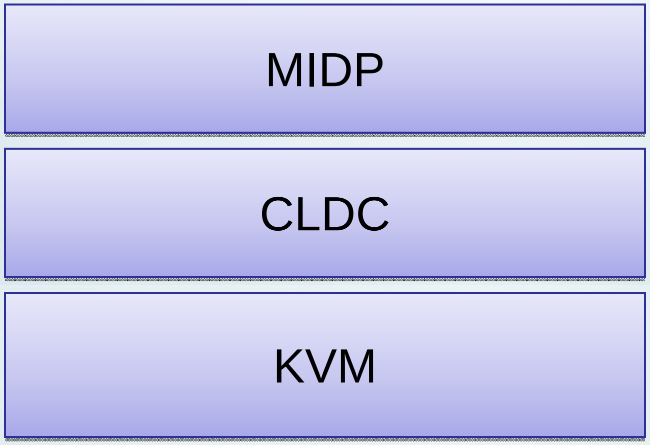
Java ME Platforms

- It includes two kinds of platforms:
  - High-end platform for high-end consumer devices. E.g. TV set-top boxes, Internet TVs, auto-mobile navigation systems
  - Low-end platform for low-end consumer devices. E.g. cell phones, and pagers

Platforms	Device Characteristics
"High-End" consumer devices	• a large range of user interface capabilities • total memory budgets starting from about two to four megabytes • persistent, high-bandwidth network connections, often using TCP/IP
"Low-end" consumer devices	•simple user interfaces •minimum memory budgets starting from about 128–256 kilobytes •low bandwidth, intermittent network connections that is often not based on the TCP/IP protocol suite. • most of these devices are battery-operated

Java ME Platforms (Cont.)

- The following figures demonstrate the elements of these two types of platforms:

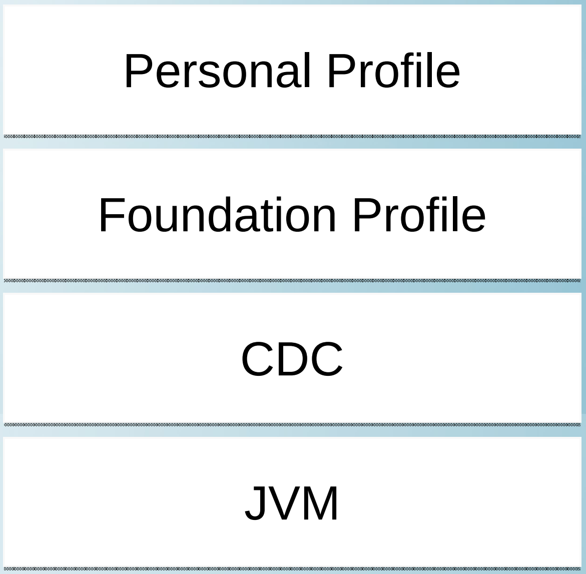


MIDP

CLDC

KVM

Fig.1- Low-end consumer device platform architecture



Personal Profile

Foundation Profile

CDC

JVM

Fig.2- High-end consumer device platform architecture

Java ME CLDC

- We focus on “Low-end” consumer devices:
  - CLDC configuration address the following areas:
    - Java language and virtual machine features
    - Core Java libraries
    - CLDC Specific Libraries (Input/output & Networking)
    - Internationalization: There is a limited support for converting Unicode characters to and from a sequence of bytes.
    - Security: CLDC addresses the following topics to security
      - At the low-level the virtual machine security is achieved by requiring downloaded Java classes to pass a class file verification step.
      - Applications are protected from each other by being run in a closed “sandbox” environment.
      - Classes in protected system packages cannot be overridden by applications.

Java ME CLDC (Cont.)

- The entire CLDC implementation (static size of the K virtual machine + class libraries) should fit in less than 128 kilobytes.
- It guarantee portability and interoperability of profile-level code between the various kinds of mobile (CLDC) devices

Palm OS

- Palm OS[1] is an embedded operating system designed for ease of use with a touchscreen-based graphical user interface.
- It has been implemented on a wide variety of mobile devices such as smart phones, barcode readers, and GPS devices.
- It is run on Arm architecture-based processors. It is designed as a 32-bit architecture.

Palm OS Features

- The key features of Palm OS [1,2,3] are:
  - A single-tasking OS:
    - Palm OS Garnet (5.x) uses a kernel developed at Palm, but it does not expose tasks or threads to user applications. In fact, it is built with a set of threads that can not be changed at runtime.
    - Palm OS Cobalt (6.0 or higher) does support multiple threads but does not support creating additional processes by user applications.
    - Palm OS has a preemptive multitasking kernel that provides basic tasks but it does not expose this feature to user applications .

Palm OS Features (Cont.)

- Memory Management:
  - The Memory, RAM and ROM, for each Palm resides on a memory module known as card. In other words, each memory card contains RAM, ROM or both. Palms can have no card, one card or multiple cards.
- Expansion support[3]:
  - This capability not only augments the memory and I/O , but also it facilitates data interchanges with other Palm devices and with other non-Palm devices such as digital cameras, and digital audio players.
- Handwriting recognition input called Graffiti 2

Palm OS Features (Cont.)

- HotSync technology for synchronization with PC computers
- Sound playback and record capabilities
- TCP/IP network access
- Support of serial port, USB, Infrared, Bluetooth and Wi-Fi connections
- Defined standard data format for PIM (Personal Information Management) applications to store calendar, address, task and note entries, accessible by third-party applications

Palm OS Features (Cont.)

– Security model:

- Device can be locked by password, arbitrary application records can be made private [2]
- Palm OS Cobalt include a certificate manager. The Certificate Manager handles X.509 certificates[3].

Symbian OS

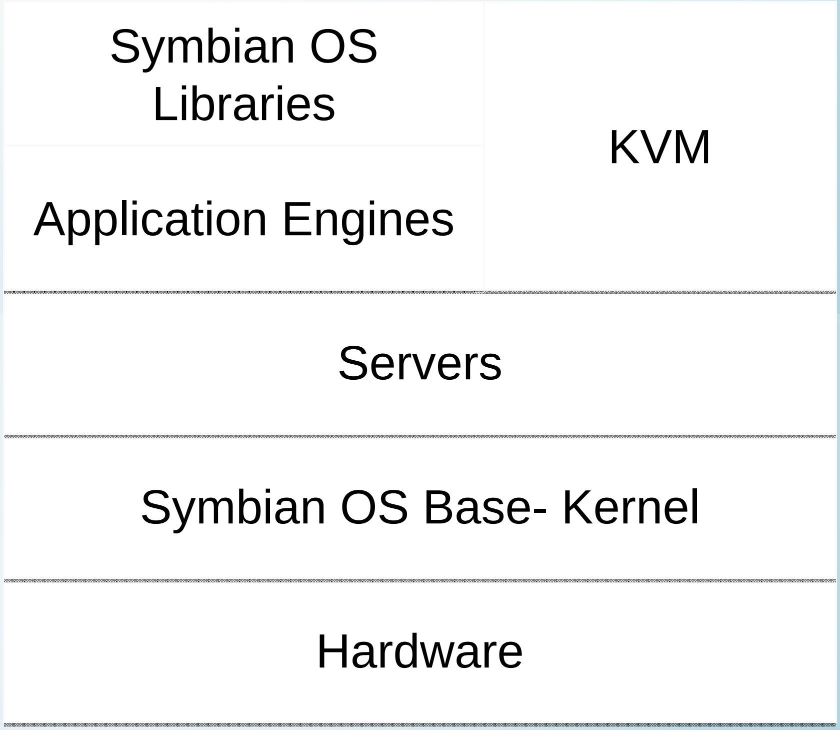
- Symbian OS is 32 bit, little-endian operating system, running on different flavors of ARM architecture[4].
- It is a multitasking operating system and very less dependence on peripherals.
- Kernel runs in the privileged mode and exports its service to user applications via user libraries.

Symbian OS Structure

- User libraries include networking, communication, I/O interfaces and etc.
- Access to these services and resources is coordinated through a client-server framework.
- Clients use the service APIs exposed by the server to communicate with the server.
- The client-server communication is conducted by the kernel.

Symbian OS Structure (Cont.)

- The following demonstrates the Symbian OS architecture[5]:



Symbian OS Libraries

Application Engines

KVM

Servers

Symbian OS Base- Kernel

Hardware

Symbian OS Features

- Real-time: it has a real-time, multithreaded kernel.
- Data Caging [6]: it allows applications to have their own private data partition. This feature allows for applications to guarantee a secure data store. It can be used for e-commerce applications, location aware applications and etc.
- Multimedia: it supports audio, video recording, playback and streaming, and Image conversion.

Symbian OS Features (Cont.)

- Platform Security [6]: Symbian provides a security mechanism against malware. It allows sensitive operations can be accessed by applications which have been certified by a signing authority. In addition, it supports full encryption and certificate management, secure protocols (HTTPS, TLS and SSL) and WIM framework.

Symbian OS Features (Cont.)

- Internationalization support: it supports Unicode standard.
- Fully object-oriented and component- based
- Optimized memory management
- Client- server architecture [6]: described in previous slides, it provides simple and high-efficient inter-process communication. This feature also eases porting of code written for other platforms to Symbian OS.

Symbian OS Features (Cont.)

- A Hardware Abstraction Layer (HAL): This layer provides a consistent interface to hardware and supports device-independency

- Kernel offers hard real-time guarantees to kernel and user mode threads.

Embedded Linux OS

- It is known as Embedded Linux which is used in embedded computer systems such as mobile phones, Personal Digital Assistants, media players and other consumer devices.

- In spite of Linux operating system designed for Servers and desktops, the Embedded Linux is designed for devices which have relatively limited resources such as small size of RAM, storage, screen, limited power and etc. Then, they should have an optimized kernel.

Embedded Linux OS

- It is a Real-Time Operating System (RTOS). It meets deadlines and switch context
- It has relatively a small footprint. Today, mobile phones can ship with a small memory. Thus, OS must not seek to occupy a large amount of available storage. It should have a small foot print. Theoretically, they deploy in a footprint of 1MB or less.
- It is open source. It has no cost for licensing.
- Examples: Motorola Mobile phones such as RAZR V8, RAZR V9, A1200 are based on MontaVista Linux.

Embedded Linux OS

- **ARM and MIPS structures [7]: Embedded CPU architectures like ARM and MIPS offer small instruction sets and special execution modes that shrinks application size and consequently generates smaller code.**

Windows Mobile OS

- Windows Mobile is a compact operating system designed for mobile devices and based on Microsoft Win32.

- It is run on Pocket PCs, Smartphones and Portable media centers.

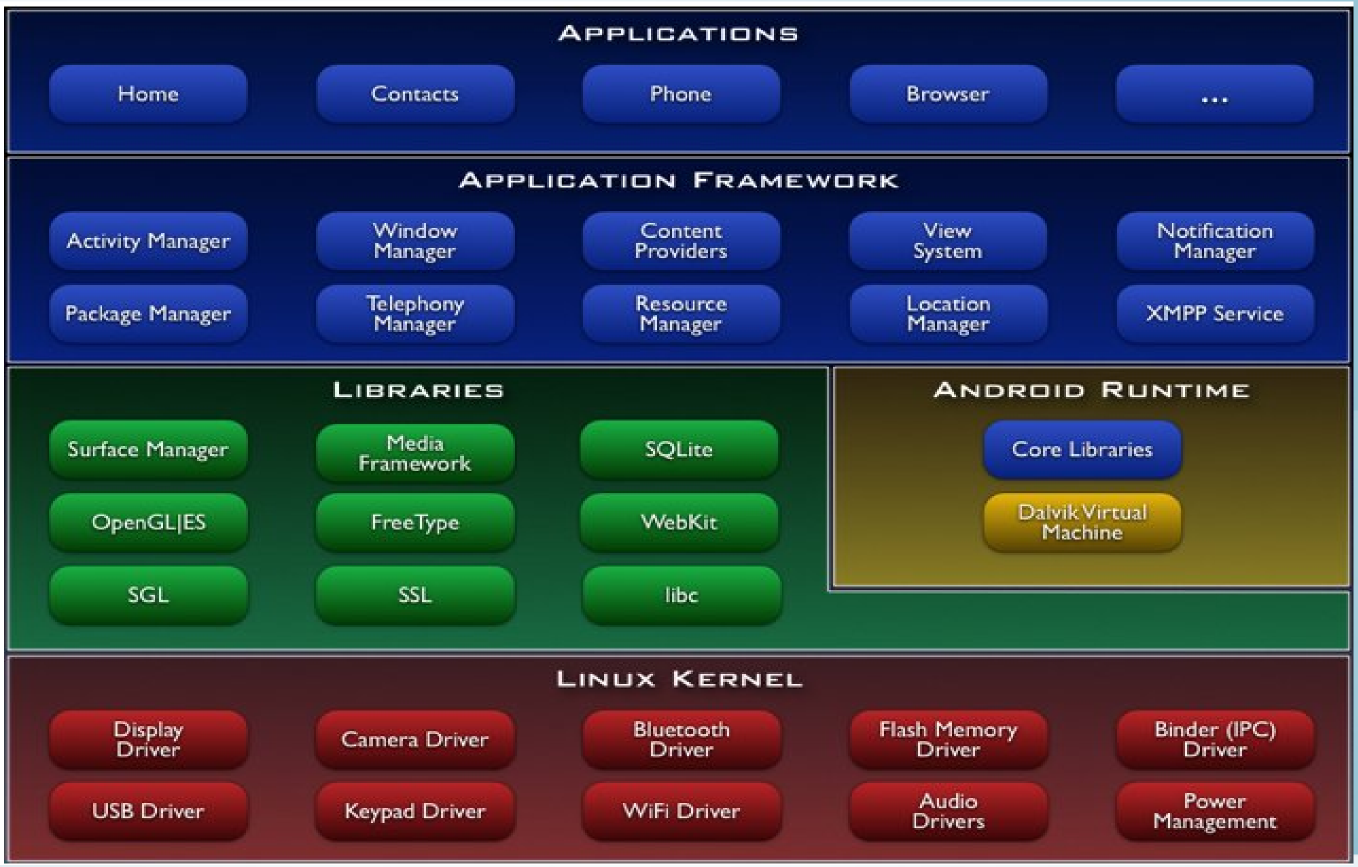
- It provides ultimate interoperability. Users with various requirements are able to manipulate their data.

Google Android Platform

- It is a platform and an operating system for mobile devices based on the Linux operating system.
- It allows developers design applications in a java-like language using Google-developed java libraries.
- It supports a wide variety of connectivity such as GSM, WiFi, 3G, …
- The Operating system has not been implemented yet (Feb, 2008). Several prototypes have been proposed.

Google Android Platform

Android architecture: http://code.google.com/android/what-is-android.html



Google Android Platform

- As demonstrated in the previous slide, the Android platform contains the following layers:
  - Linux Kernel: Android relies on Linux for core system services such as security, memory management, process management and etc.
  - Android [7] Runtime: it provides a set of core libraries which supports most of the functionality in the core libraries of Java. The Android Virtual Machine known as Dalvik VM relies on the linux kernel for some underlying functionality such as threading,…

Google Android Platform

- Libraries: Android includes a set of C/C++ libraries. These libraries are exposed to developers through the Android application framework. They include media libraries, system C libraries, surface manager, 3D libraries, SQLite and etc.

  For more details, please visit the following link:

  http://code.google.com/android/what-is-android.html

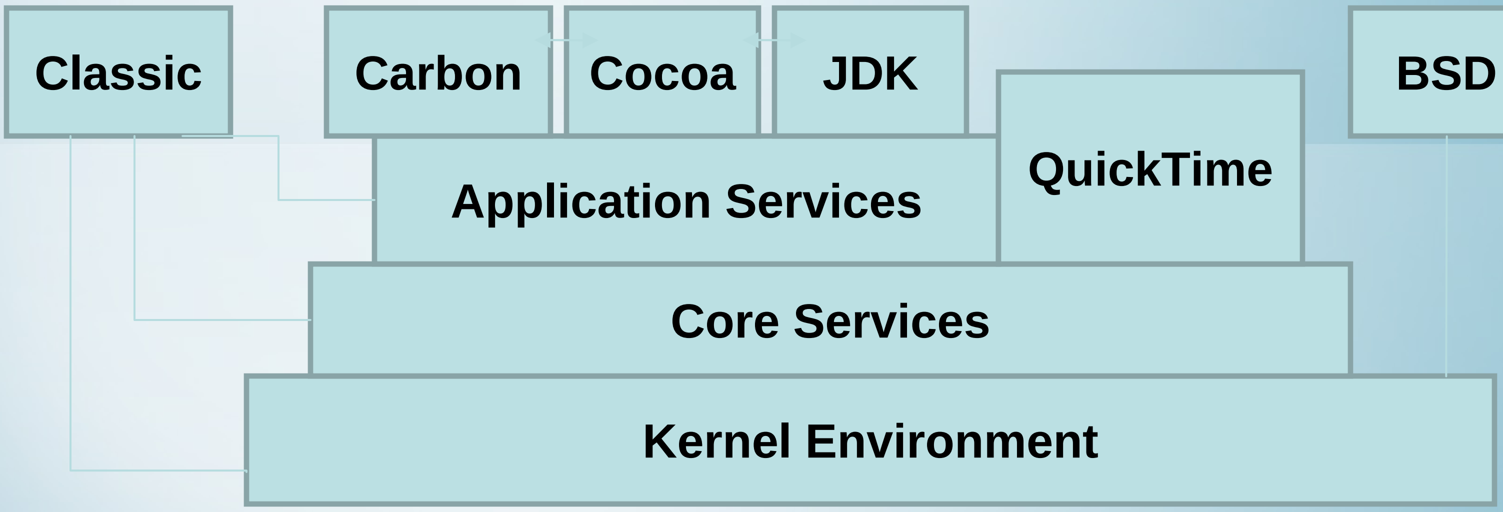
- Application Framework: it provides an access layer to the framework APIs used by the core applications. It allows components to be used by the developers.

iPhone OS

- iPhone OS is an operating system run on iPhone and iPod.
- It is based on Mach Kernel and Drawin core as Mac OS X.
- The Mac OS X kernel includes the following component:
  - Mach Kernel
  - BSD
  - I/O component
  - File Systems
  - Networking components

iPhone OS

- The following is Mac OS X Architecture [8]:



iPhone OS

- Mac OS X has a preemptive multitasking environment.
- Preempting is the act of taking the control of operating system from one task and giving it to another task.
- It supports real-time behavior.
- In Mac OS X, each application has access to its own 4 GB address space.

iPhone OS

- Not any application can directly modify the memory of the kernel. It has a strong mechanism for memory protection.
- For more details about kernel architecture, please visit the following link:

  http://developer.apple.com/documentation/Darwin/Conceptual/KernelProgramming/Architecture/Architecture.html

BlackBerry OS

- **BlackBerry OS will be discussed in Week II slides.**

References

[1] Palm OS, http://en.wikipedia.org/wiki/Palm_OS

[2] Palm OS is a multitasking operating system. How can I create a task?, http://kb.palmsource.com/cgi-bin/palmsource.cfg/php/enduser/std_adp.php?p_faqid=72

[3] Palm OS Programmer’s companion- Volume I, http://www.access-company.com/developers/documents/docs/palmos/PalmOSCompanion/CompanionTOC.html

[4] Mobile Operating Systems, http://dsonline.computer.org/portal/site/dsonline/menuitem.9ed3d9924aeb0dcd82ccc6716bbe36ec/index.jsp?&pName=dso_level1&path=dsonline/topics/os&file=MobileOS.xml&xsl=article.xsl&##archives

[5] Symbian OS, http://images.google.ca/imgres?imgurl=http://wiki.forum.nokia.com/images/7/77/Symbian_OS_Structure.png&imgrefurl=http://wiki.forum.nokia.com/index.php/Symbian_OS&h=260&w=467&sz=10&hl=en&start=45&sig2=Hx5G5bWIQo_Hc3_kni4lgw&um=1&tbnid=PeArbbX9LhuMaM:&tbnh=71&tbnw=128&ei=ZStJSIKCNaLGgwKe09nXAw&prev=/images%3Fq%3Dmobile%2Boperating%2Bsystem%2Bstructure%26start%3D36%26ndsp%3D18%26um%3D1%26hl%3Den%26sa%3DN

[6] Symbian OS, http://www.symbian.com/files/rx/file6965.pdf

[7] Mobile Phones, the Embeded Linux Challenge, http://www.linuxjournal.com/article/8762

[8] iPhone SDK, http://cssinnovations.blogspot.com/2008/05/iphone-sdk-iphone-sdk-was-officially.html