MOBILE COMPUTING
ASSIGNMENT 1

Done By

Harikrishnan V

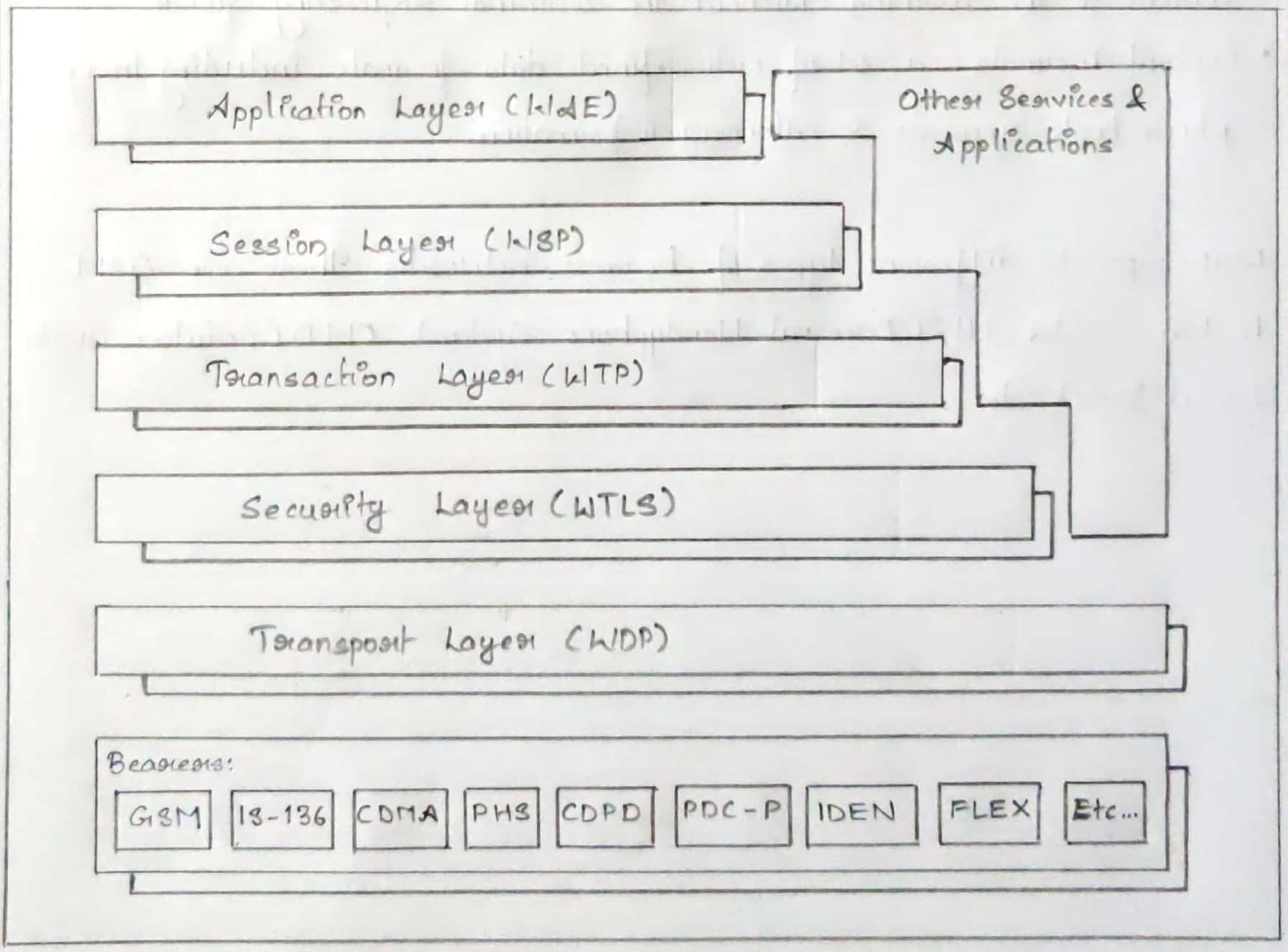
CS6A

Roll No : 27

WAP

WAP forum develops standards for application deployment over wireless devices like PDAs & mobile phones. WAP is based on layered architecture. The WAP Protocol Stack is similar to the OSI network model. These layers consist (from top to bottom) of:

- Wireless Application Environment (WAE)
- Wireless Session Protocol (WSP)
- Wireless Transaction Protocol (WTP)
- Wireless Transport Layer Security (WTLS)
- Wireless Datagram Protocol (WDP)



WAP layered architecture and protocol stack

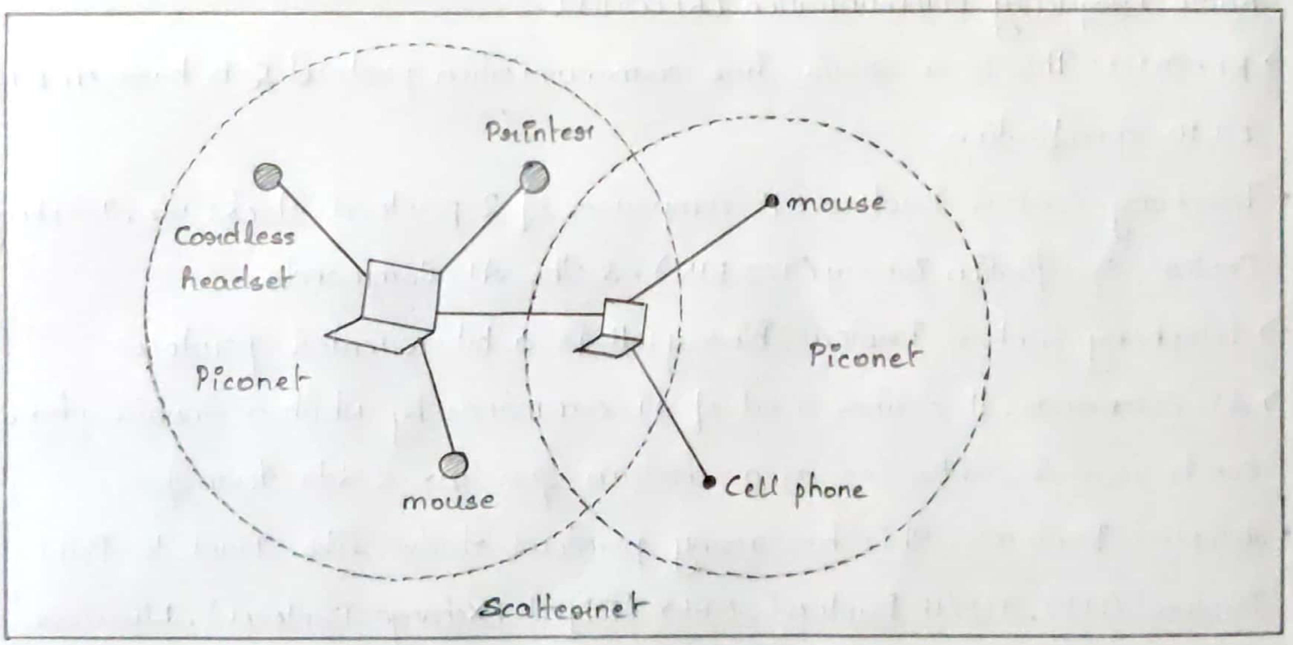
The application environment of WAE comprises multiple components to provide facilities like:

- **User agent**: the browser or a client program.
- Wireless Markup Language (WML): a lightweight markup language, similar to HTML, but optimized for use in wireless devices.
- WMLScript: a lightweight client side scripting language, similar to JavaScript in Web.
- Wireless Telephony Application: telephony services & programming interfaces.
- WAP Push Architecture: mechanisms to allow origin servers to deliver content to the terminal without the terminal requesting for it.
- Content Formats: a set of well-defined data formats, including images phone book records & calendar information.

WAP supports different types of bearer networks. These are GSM, IS-136, CDMA, PHS (Personal Handyphone System), CDPD (Cellular Digital Packet Data), etc.

BLUETOOTH

Bluetooth was the nickname of a Danish king Harald Blåtand, who unified Denmark & Norway in the 10th century. The concept behind Bluetooth wireless technology was unifying the telecom & computing industries. Bluetooth technology allows users to make ad hoc wireless connections between devices like mobile phones, desktop or notebook computers without any cable. Devices carrying Bluetooth-enabled chips can easily transfer data at a speed of about 720 Kbps within 50 meters (150 feet) of range or beyond through walls, clothing & even luggage bags.



Bluetooth scatternet as a combination of Piconets

Bluetooth Protocol Stack

Bluetooth protocol stack can be divided into 4 basic layers according to their functions. These are:

- Bluetooth Core Protocols: this comprises of Baseband, Link Manager Protocol (LMP), Logical Link Control & Adaptation Protocol (L2CAP), & Service Discovery

Protocol (SDP).

→ Baseband: The Baseband & Link Control layer enables the physical RF link between Bluetooth units forming a piconet.

→ Link Manager Protocol (LMP): When 2 Bluetooth devices come within each other's radio range, link managers of either device discover each other.

→ Logical Link Control & Adaptation Protocol (L2CAP): This layer is responsible for segmentation of large packets & reassembly of fragmented packets.

→ Service Discovery Protocol (SDP): It enables a Bluetooth device to join a piconet.

- Cable Replacement Protocol: This protocol stack has only 1 member viz., Radio Frequency Communication (RFCOMM).

→ RFCOMM: This is a serial line communication protocol & is based on ETSI 07.10 specification.

- Telephony Control Protocol: It comprises of 2 protocol stacks viz., Telephony Control Specification Binary (TCS BIN), & the AT-Commands.

→ Telephony Control Protocol Binary: It is a bit-oriented protocol.

→ AT-Commands: It defines a set of AT-commands by which a mobile phone can be used & controlled as a modem for fax & data transfers.

- Adopted Protocols: This has many protocol stacks like Point-to-Point Protocol (PPP), TCP/IP Protocol, OBEX (OBject EXchange Protocol), Wireless Application Protocol (WAP), vCard, vCalendar, Infrared Mobile Communication (IrMC), etc.

→ PPP: Bluetooth offers PPP over RFCOMM to accomplish point-to-point connections.

→ TCP/IP: It is used for communication across the Internet.

→ OBEX Protocol: OBEX is a session protocol developed by the Infrared

Data Association (IrDA) to exchange objects.

→ Content Formats : vCard & vCalendar specifications define the format of an electronic business card & personal calendar entries developed by the Versit consortium.

XML

In mobile computing through telephone, the IVR is connected to the server through client/server architecture. It is also possible to host the IVR & the application on the same system. Today Internet (http) is used in addition to client/server interface between the IVR & the server. This increases the flexibility in the whole mobile-computing architecture. Http is used for voice portals as well. In the case of a voice portal, a user uses an Internet site through voice interface. For all these advanced features, VoiceXML has been introduced. Recent IVRs are equipped with DSP (Digital Signal Processing) & are capable of recognizing voice. The output is synthesized voice through TTS (Text to Speech).

The Voice eXtensible Markup Language (VoiceXML) is an XML-based markup language for creating distributed voice applications. Voice XML is designed for creating audio dialogs that feature synthesized speech, digitized audio, recognition of spoken voice & DTMF key input. Using VoiceXML, we can create Web-based voice applications that users can access through telephone.

VoiceXML supports dialogs that feature:

- Spoken input
- DTMF (telephone key) input
- Recording of spoken input
- Synthesized speech output ('text-to-speech')
- Recorded audio output
- Dialog flow control
- Scoping of input

J2ME

The coffee cup for the small devices is christened Java 2 Micro Edition (J2ME). J2ME was conceived from the need to define a computing platform that could accommodate consumer electronics & embedded devices. The low-end PDAs may offer only offline data storage with a serial cable to sync with the PC while the high-end communicators would be microcomputers. Mobile phones are likely to have low bandwidth intermittent connectivity while the set top boxes would have uninterrupted connectivity. It was not practical to attempt to define a single J2ME platform for all of these. The biggest challenge for J2ME was to specify a platform that could support a consistent set of services across a broad spectrum of devices with a large multitude of capabilities. To be able to support the large brood of devices, a modular structure was essential. The designers of J2ME camp up with a concept of configurations & profiles towards achieving this goal.

A Configuration defines the lowest common denominator or the minimum capabilities that will be available across a given range of devices. It is a complete Java runtime environment, consisting of:

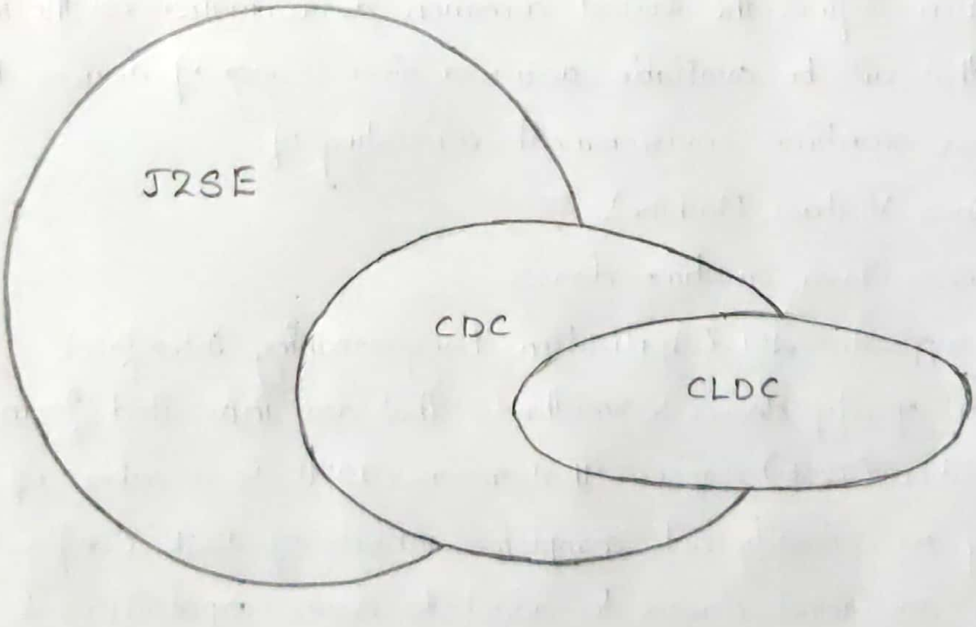
- A JVM (Java Virtual Machine).
- A set of Core Java runtime classes.
- A set of supported API (Application Programming Interface)

Configuration specify classes & methods that are inherited from Java 2 Standard Edition (J2SE) classes. That means, J2ME is a subset of J2SE. However they are generally not complete subsets of J2SE. Configurations also include additional classes to adapt to device capabilities & constraints.

To avoid fragmentation & a deluge of incompatible platforms, J2ME defines only 2 configurations. They represent the 2 distinct categories of devices.

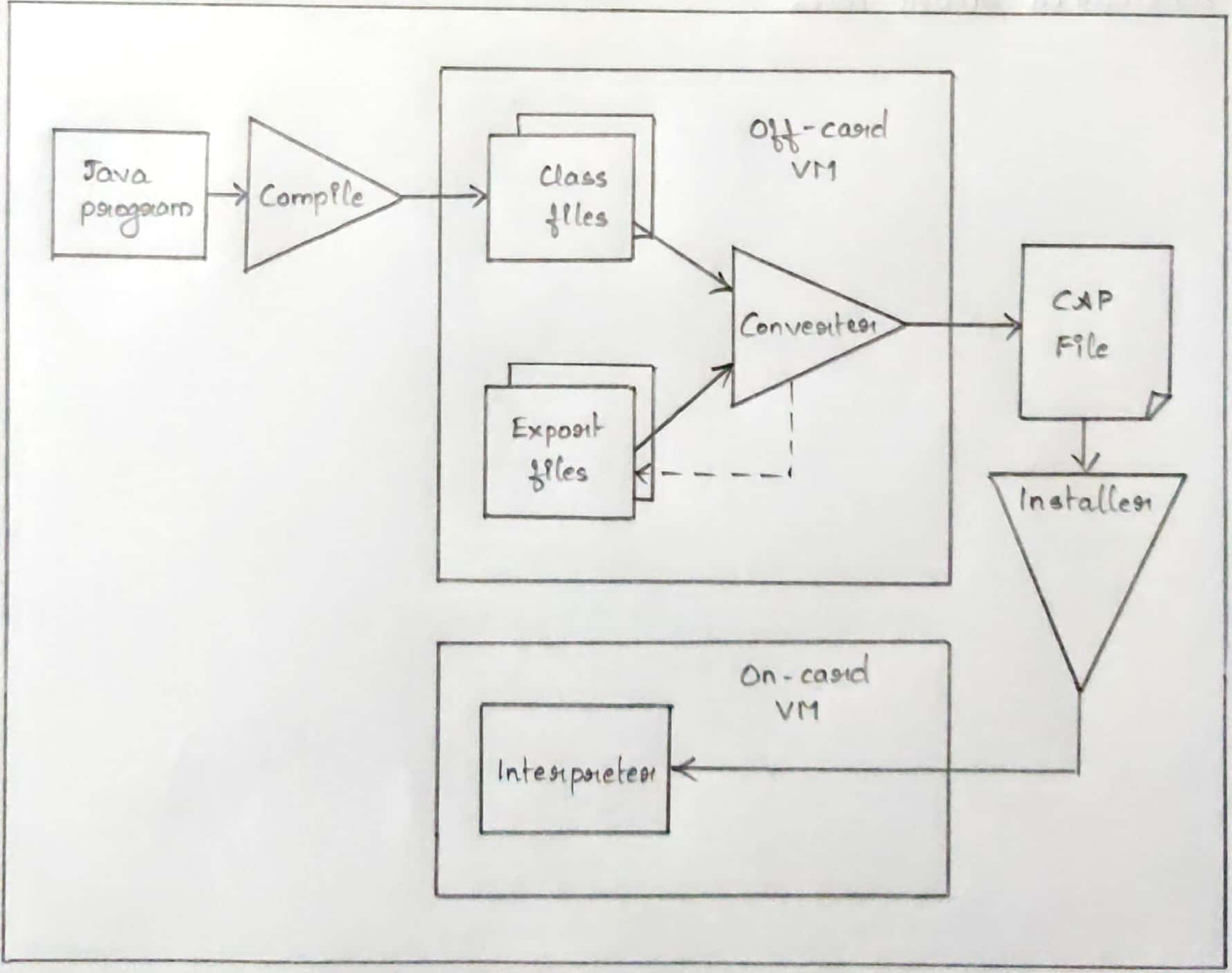
The 1st category is devices that have superior UI facilities, higher computing power & are constantly connected. These implement the Connected Device Configuration (CDC). Eg. set-top boxes, Internet TVs, Internet-enabled screen phones, high-end communicators, & car entertainment / navigation systems.

The 2nd being personal, mobile information devices that are capable of intermittent communications. These implement the Connected, Limited Device Configuration (CLDC). Eg. mobile phones, 2-way pagers, personal digital assistants (PDAs), & organizers.



JAVA CARD

Java Card is a smart card with Java framework. Smart card is a plastic card with intelligence & memory. A smart card is embedded with either (i) a microprocessor & a memory chip or (ii) only a memory chip with non-programmable logic. A microprocessor card can have an intelligent program resident within the card which can add, delete, & otherwise manipulate information on the card. Smart cards are capable of carrying data, functions, & information on the card. Therefore, unlike memory strip cards, they do not require access to remote databases at the time of the transaction.



Architecture of Java card applications development process

Smart cards have now emerged as multi-function cards. To allow interoperability, Java was chosen as the vehicle for interoperability. All the microprocessor based smart cards now offer Java API framework on the smart card. This is why smart cards with Java framework are also called Java Cards. Java Card technology preserves many of the benefits of the Java programming languages such as: productivity, security, robustness, tools, & portability. For Java card, the Java Virtual Machine (JVM), the language definition, & the core packages have been made more compact to bring Java technology to the resource constrained smart cards.

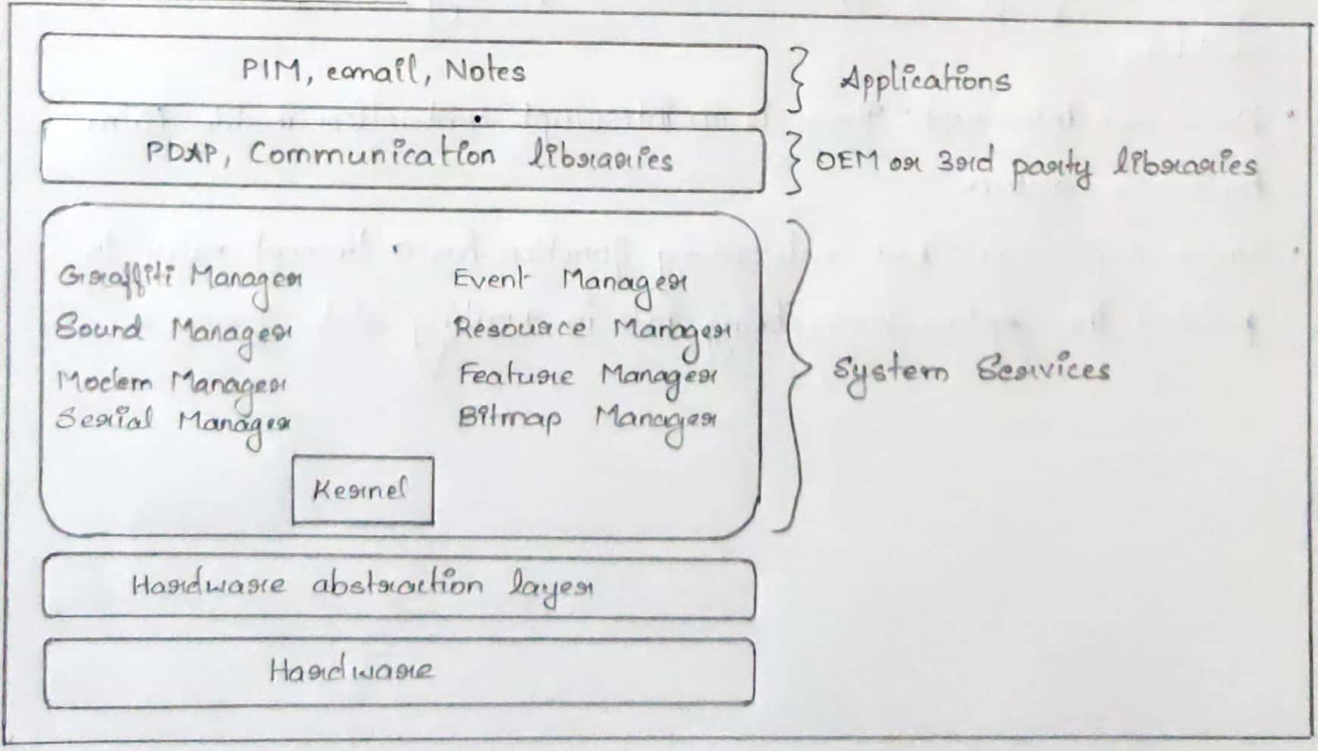
PALM OS

The travails of Palm OS closely follow the fortunes of Palm Computing, the company under whose aegis Palm OS was developed. The story of Palm's conception is legendary. Right from when its founding father Jeff Hawkins carried a block of wood to every meeting to taking a practical approach that the user should learn the hieroglyphics of graffiti rather than putting together software that understands all nuances of human handwriting. These were based on learnings from costly mistakes made earlier & feedback from customers of an earlier product Zoomer.

The Palm was designed with 3 commandments

- Handwriting recognition to be limited to simplified hieroglyphics
- Size should be small enough to fit into the pocket.
- A cradle to synchronize data with a PC.

Palm OS Architecture



The kernel used in Palm OS is the AMX real-time, multitasking kernel, a product from KADAK Products Ltd. The kernel itself supports a lot of features but not all of these are available to the applications. Some important features supported by the kernel are listed below.

Kernel Features

Multitasking: The kernel itself supports pretty advanced multitasking, including semaphores.

Interrupts: The kernel supports both maskable & non-maskable interrupts in normal & nested modes.

Time slicing & Scheduling: This essentially allows the execution of several tasks according to their priority thereby supporting timers & time procedure. There are 3 types of triggers for task switching:

- Context Switching: An application task requesting an implicit context switching.
- Hardware Interrupt: There is an interrupt controller inside Palm hardware system.
- Timer Expiration: Each networking function has a timeout value to prevent the system from being idle in waiting state forever.

LINUX FOR MOBILE DEVICES

Linux kernel-based operating systems are ubiquitously found on diverse hardware platforms.

This is a list of many Linux kernel-based operating systems used on mobile devices. They differ from 1 another in parts of the middleware or the entire middleware, & in that they employ individual UIs.

- Android (operating system)
- Replicant (operating system)
- AsteroidOS
- Plasma Mobile
- postmarketOS
- Sailfish OS
- SHR (operating system)
- Tizen
- webOS
- LuneOS (based on HP webOS)
- PureOS

Middlewares:

- BusyBox
- Toybox
- mer
- Smart Common Input Method
- Maliit
- Intelligent Input Bus
- Uim

- Fcitx
- Halium

UI:

- KDE Plasma Workspaces
- Unity
- GPE Palmtop Environment
- OPIE user interface
- Phosh

ANDROID

Android is a mobile operating system based on a modified version of the Linux kernel & other open source software, designed primarily for touchscreen mobile devices such as smartphones & tablets. Android is developed by a consortium of developers known as the Open Handset Alliance, with the main contributor & commercial marketer being Google.

Features

Interface

Android's default user interface is mainly based on direct manipulation, using touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, & reverse pinching to manipulate on-screen objects, along with a virtual keyboard.

Applications

Many, to almost all, Android devices come with preinstalled Google apps including Gmail, Google Maps, Google Chrome, YouTube, Google Play Music, Google Play Movies & TV, & many more.

Memory Management

Since Android devices are usually battery-powered, Android is designed to manage processes to keep power consumption at a minimum.

Hardware

The main hardware platform for Android is ARM (the ARMv7 & ARMv8-A architectures), with x86 & x86-64 architectures also officially supported in later versions of Android.