Mobile OSs, Development Environments, iOS and Android

Lesson 01

Mobile Operating System

Mobiles

- Specialized hardware
- Number of Apps
- GUIs, VUIs, communication interfaces for service provider and Internet
- constraints of connectivity, CPU speed, memory, battery life, display, and size of input devices

Operating system (OS)

- The master control program
- Manages all software and hardware resources
- Controls, allocates, frees, and modifies the memory by increasing or decreasing it
- Enables the assignment of priorities for requests to the system

Operating system (OS)

- Controls IO devices and network
- Manages files, disks, device drivers and GUIs, apps, APIs and Hardware abstraction layer
- Manages security

Operating system (OS)

- Includes utility programs, for example, file manager, configuration of OS (memory and resource allocation and enabling and disabling the use of specific resources and functions)
- Can be accompanied by a specific suite of apps, for example, Internet Explorer and MS Office

Mobile OS

- An OS specialised for which enables running of tasks and threads of the apps, taking into account mobile devices constraints of hardware and network
- Enables a programmer to develop app without considering the specifications, drivers, and functionalities of the hardware

Two special features of the mobile OSs

1. OS packs a number of apps
2. OS has a hardware abstraction layer. The layer enables running of the apps on changes in mobile hardware at later hardware versions of mobiles.

NUMBER OF APP SOFTWARE COMPONENTS

- Micro secure digital (SD) card, touch screen, camera, video camera, speech recognition, video calls, voice recorder, music player, and communication APIs for micro Universal Serial Bus (USB), IR, Bluetooth, Wi-Fi, global positioning system (GPS) mobile navigation, NFC, enterprise, and Cloud.

Mobile OS

- Enables an app to run by simply abstracting the mobile system hardware
- Enables the programmer to abstract the devices such that the app need not know full details of the font and font size on mobile device display
- app need not know how the message will be displayed by the screen display hardware

Mobile OS Examples

- Apple iOS 11
- Android– released in 2008 by **Google** [open-source code which powers mobile operating system]
- Latest Andoid 8.0 Oreo

Example of hardware abstraction by the OS

- Assume that keypad, screen display, serial input, and serial output devices are abstracted by an app as the input and output devices with device numbers 1, 2, 3, and 4, respectively

Example of hardware abstraction by the OS

- write (1, 'Welcome to ABC Telecom') when a message Welcome to ABC Telecom is sent in the output for display
- Program line code "*write* ('Welcome to ABC Telecom')" when display device is taken as default output device

Mobile OS

- Facilitates execution of software components on diversified mobile device hardware
- app need not be aware of the details of the screen driver and memory at which the CPU will send the message for display

Driver

- Software component which enables the use of a device, port, or network by configuring (for *open*, *close*, *connect*, or specifying a buffer size, mode, or control word) and sends output or receives input

Mobile OS

- Provides interfaces for communication between processes, threads, and ISRs at the app and middleware layers
- Provides middleware for the system hardware
- Provides management functions (such as creation, activation, deletion, suspension, and delay) for tasks

Mobile OS

- Provides memory management
- Enables running of processes
- Helps the processes in obtaining access to system resources

Application

- Application tasks
- The OS provides the functions used for scheduling the multiple tasks in a system
- Synchronization of the tasks by using semaphores (tokens)
- A task may have multiple threads

Mobile OS

- Provides for synchronization of the threads and their priority allocation
- Accomplishes real-time execution of the tasks and threads of an app

App

- Uses the system resources, for example, CPU, memory keypad, display unit, modem interface, USB or serial port, and battery
- Resources shared concurrently by the apps running on the system

User app

- GUIs (graphic user interfaces)
- VUI (voice user interface) components
- Phone API
- Mobile OS provides configurable libraries for the GUI in the device
- Provides for multi-channel and multi-modal user interfaces

Process

- A program unit which runs when scheduled to do so by OS and each state of which is controlled by OS
- Can call a function (method) but cannot call another process directly

States of a process

- Can be in any of the states—

1. created
2. active
3. running
4. suspended
5. pending for a specified time interval

Pending state of a process for a specific communication from other process

- Signal
- Semaphore
- Mailbox-message
- Queue-message
- Socket

Task

- A process of an app that runs according to its schedule set by the OS
- Each state of which is controlled by OS
- Can be a real-time task which has time constraints or maximum defined latency within which it must run or finish

Thread

- An process or a process sub-unit (when a process or task has multiple threads)
- Runs as scheduled by the OS
- Each state controlled by OS
- Runs as a light-weight process

Light-weight

- Does not depend on certain system resources, for example, memory management unit (MMU), GUI functions provided by the OS, or the functions which need running of other processes or threads for their implementation

Interrupt service routine (ISR)

- A program unit (function, method, or subroutine) which runs when a hardware or software event occurs
- Running of which can be masked and can be prioritized by assigning a priority
- Higher priority than any other process or task or thread

Hardware events for interrupts

- Time-out of a timer (clock tick)
- Division by zero
- Overflow
- Underflow detection by hardware during computation

Hardware events for interrupts

- Finishing of DMA (direct memory access by a peripheral) transfer
- Data abort
- External FIQ (fast interrupt request through a pin input)
- External IRQ (interrupt request through a pin input)
- A memory buffer becoming full

Hardware events for interrupts

- Port, transmitter, receiver, or device buffer— becoming half filled, buffer with at least one memory address filled, and buffer becoming empty
- Buffer— associated with the memory addresses for the LCD, printer, serial or USB port, keypad, or modem

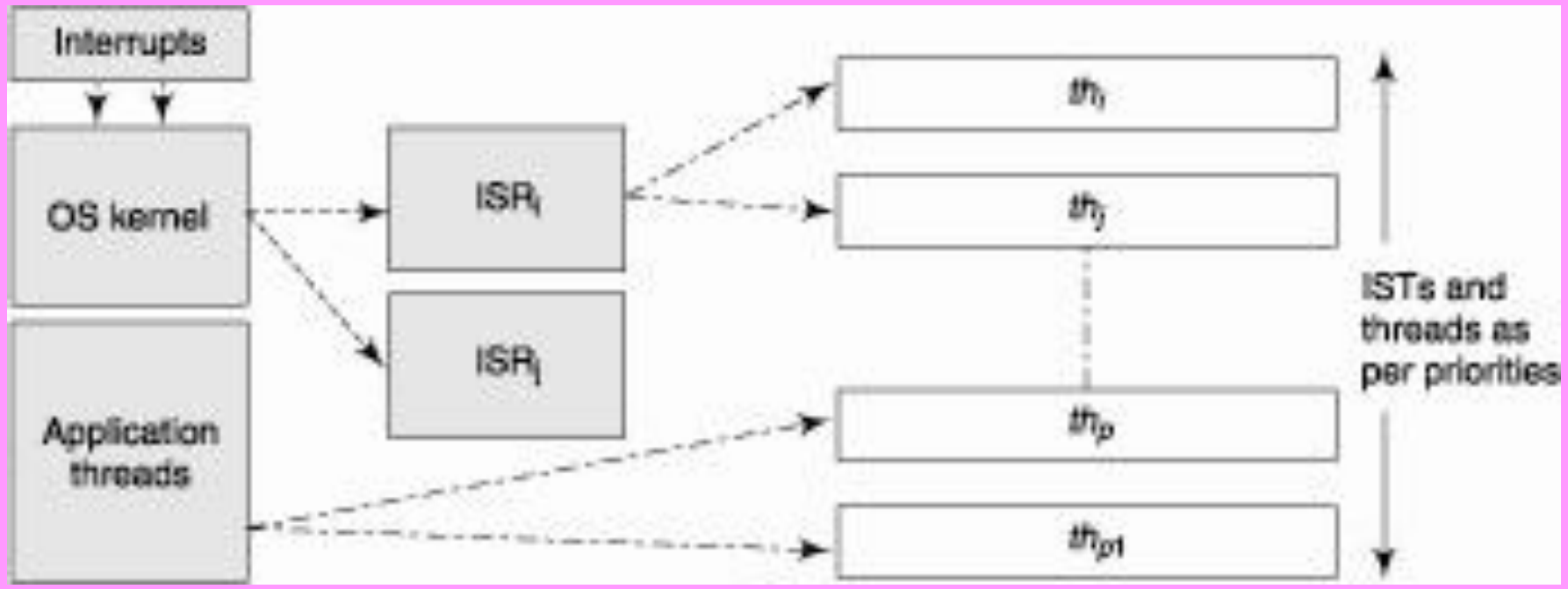
Software related events for interrupts

- Exception– software instruction for interrupt on detection of a certain condition during computations or error while logging in
- Illegal operation code provided to CPU

Interrupt service thread (IST)

- A special type of ISR or ISR unit (function, method, or subroutine) which initiates and runs on an event or message from an high priority ISR
- ISTs can be prioritized by assigning a priority
- The type of IST depends on the specific OS

App as event-driven ISTs (i, j, ...) and threads (p, q, ...)



Windows IST

- One which is placed in a priority queue so that the ISTs execute turn by turn—FIFO (first-in first-out)
- An IST is initiated and put in the FIFO according to ISR priority after the execution of an ISR starts on an event and the ISR puts a message or semaphore or signal for the IST

ISR and IST

- The ISR, therefore, has an event-initiated short piece of code, which runs only the critical part of the code and rest of the code runs at the IST initiated by it
- The IST priority– lower than the ISR but higher than the processes or tasks or threads

Page

- A unit of memory which can load from a program stored in a hard drive or from any other storage device to the program memory, RAM, before the execution of a program
- A contiguous memory address block of 4 kB (in x86 processors), 2 kB, or 1 kB

Page table

- For address mapping
- Provides the mapping of fragmented physical memory pages with the pages of the virtual addresses which are the memory addresses

Page table

- Pages of memory are spread over the memory-address space leading to fragmentation of codes and data in physical memory space

MMU

- Creates and maintains the page table and hence performs address mapping and translation

MMU

- Program during execution first translates the accessed address (virtual address) into a physical address using the page table at the MMU and then accesses the physical address and fetches the code or data

Priority inversion

- Takes place when a process or thread which is to provide a waiting object to a higher priority process or thread gets preempted by a middle priority process or thread and thus the middle priority process or thread starts running on obtaining the object for which it was waiting

Priority inheritence

- Process or thread which is to provide a waiting object to a higher priority process or thread, gets the priority of that process or thread
- Waiting object– signal, semaphore, queue, mailbox message, or bytes from a pipe

Pipe

- A virtual device which sends the bytes from a thread to another thread

Example of Linux OS – Mobile

- An open source OS
- Enables the user to customize their device to suit their specific needs
- Provides ease to suit different sorts of hardware and software applications

Linux

- Embedded Linux Consortium (ELC) standards for Linux for designing user interfaces, managing power consumption in devices, and real-time operation
- Also considered to be more secure than most other operating systems
- Several international mobile phone manufacturers use Linux for their OS requirements

Summary

- OS– a master control program
- Manages all software and hardware resources
- Controls, allocates, frees, and modifies the memory by increasing or decreasing it
- Controls processes, tasks, threads, ISRs and ISTs

Summary

- Mobile OS– An OS which enables running of application tasks taking into account mobile system constraints of hardware and network
- Enables a programmer to develop app without considering the specifications, drivers, and functionalities of the hardware of the system

… Summary

- Many Apps
- Device drivers
- GUIs
- VUIs
- Phone APIs

… Summary

- Manages files, disks, and security, provides device drivers and GUIs for desktop or mobile computer, other functions, and apps
- MMU
- Priority inversion and inheritence
- Linux– An open source OS
- Enables the user to customize their device to suit their specific needs

End of Lesson 01

Mobile Operating System