OPERATING SYSTEMS (JNTUK-R16)

UNIT-6; LINUX SYSTEM & ANDROID SOFTWARE PLAT FORM

SYLLABUS:

- Linux System
  - Components of Linux
  - Interprocess Communication
  - Synchronization
  - Interrupt
  - Exception
  - System Call
- Android Software Platform
  - Android Architecture
  - Operating System Services
  - Android Runtime
  - Application Development
  - Application Structure
  - Application Process Management

PART-I: LINUX SYSTEM:

Introduction to Linux Operating System:

- Linux is an open source operating system which is powerful and easy to implement.
- It can be easily installed on the system.
- It can run on various platforms and can be shared and distributed freely.
- The first version of LINUX was released in August 1991 through internet, not an official version.
- Another version 0.02 was published officially on October 5th 1991.
- This version allows GNO programs such as gcc, bash.
- Finally, the first version (0.01) was released publicly in March 1994.
- The most widely used LINUX operating systems are,
  - Arch Linux
  - CentOS
  - Debian and Raspbian
  - Fedora
  - Gentoo Linux
  - Linux Mint
  - Mageia
  - openSUSE
  - Ubuntu
  - Redhat

1. Components of Linux:

- The various components of Linux are,
  a) Process
  b) Thread
  c) Interprocess communication

www.Jntufastupdates.com

d) Synchronization
e) Bootstrap
f) Time Management
g) Kernel Module

a) Process:

- A Process is a program in execution.
- A Process contains Program code and a set of data
- The process attributes are,
  - Identifier
  - Process Sate
  - Priority
  - Program Counter
  - Memory Pointer
  - Context Data
  - I/O status Information
  - Accounting Information

Process Control Block:

- All Process information available in Process Control Block
- PCB includes
  - Process Identification
  - Process State Information
  - Process Control Information

b) Thread:

- A Thread can be thought as a basic unit of CPU utilization.
- It is also called as light-weight process.
- Multiple threads can be created by a particular process.
- Each thread shares code, data segments and open files of that process with other threads
- However, each of them has their separate register set values and stacks.

c) Inter-process communication:

Refer Topic-2

d) Synchronization:

Refer Topic-3

e) Bootstrap:

- Bootstrapping is used to initialize the operating system as required by the user.
- It is needed at startup of every system because there is no operating system when the system boots.
- Bootstrapping is carried out in four steps for X386-based systems
  - It performs POST (Power On Self Test)
  - It uses BIOS (Basic Input Output System)
  - It loads the data using boot loader
  - It finally loads the Operating Systems

f) Time Management:

- In Linux Systems, there are two different types of times.

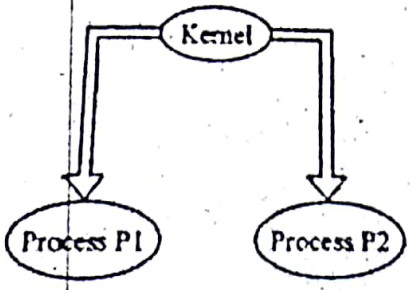
www.Jntufastupdates.com

*) Current Date and Time:
- All the systems including Linux based systems maintain the current date and time using RTC (Real Time Clock) device.
- It maintains the time even when the system is in off state.

*) Interval Timer:
- All the computer systems also maintains a PIT (Programmable Interval Timer) which issues interrupt signals based on a specific frequency set by the user.
- The default value for linux system is 1 ms.
- For every PIT execution, the kernel performs
  - Incrementing of clock
  - Incrementing of current date and time
  - Checking of failed or expired timers
  - Checking of active timer with its expiry time

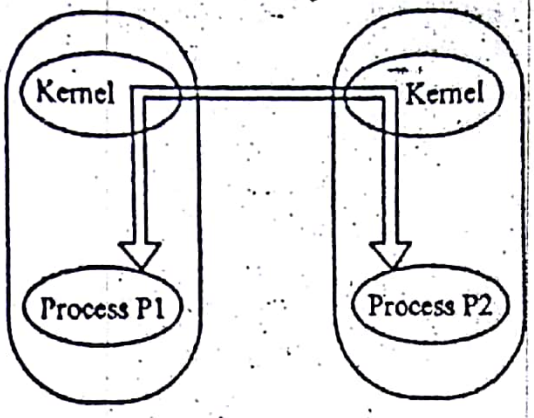
g) Kernel Module:
- Kernel Modules are used for module management in Linux systems.
- Examples of kernel modules are,
  - File systems
  - Device drivers
  - Network protocols etc
- In Linux the kernel module can be brought into action using insmod command.
- It is executed only after resolving all the pending requests present in the module program.
- However, various modules can be loaded and unloaded automatically.

2. Interprocess Communication (IPC):
- Inter Process Communication (IPC) is a mechanism that involves communication of one process with another process.
- This usually occurs only in one system.
- Communication can be of two types –
  - Between related processes initiating from only one process, such as parent and child processes.
  - Between unrelated processes, or two or more different processes.
- The design goal of IPC is to allow processes to share data among them, without interfering with one another.
- The IPC between two processes on a single computer is depicted as,



- The IPC between two processes on two different computers is depicted as,

www.Jntufastupdates.com



Methods of IPC:

- The different methods of IPC are,

a) Signals
b) Pipes
c) FIFOs
d) Semaphores
e) Message Queues
f) Shared Memory

Signals:

- Signal is a mechanism to communication between multiple processes by way of signaling.
- This means a source process will send a signal (recognized by number) and the destination process will handle it accordingly.

Pipes:

- Communication between two related processes.
- The mechanism is half duplex meaning the first process communicates with the second process.
- To achieve a full duplex i.e., for the second process to communicate with the first process another pipe is required.

FIFOs:

- Communication between two unrelated processes.
- FIFO is a full duplex, meaning the first process can communicate with the second process and vice versa at the same time.

Semaphores:

- Semaphores are meant for synchronizing access to multiple processes.
- When one process wants to access the memory (for reading or writing), it needs to be locked (or protected) and released when the access is removed.
- This needs to be repeated by all the processes to secure data.

Message Queues:

- Communication between two or more processes with full duplex capacity.
- The processes will communicate with each other by posting a message and retrieving it out of the queue.
- Once retrieved, the message is no longer available in the queue.

Shared Memory:

- Communication between two or more processes is achieved through a shared piece of memory among all processes.

www.Jntufastupdates.com

- The shared memory needs to be protected from each other by synchronizing access to all the processes.

3. Synchronization:

- Synchronization is used to ensure that the behavior of resource is in accordance with the expected behavior in a shared resource environment.
- Linux systems allow multiple kernel paths to access multiple data structures at the same time.
- The results generated from these executions are based on the way of interleaving their operations.
- To ensure appropriate behavior, these executions need to be controlled.
- The schemes used for performing synchronization in Linux systems are
    a) Interrupt Disabling
    b) Atomic Operations
    c) Kernel Semaphore
    d) Spinlock
    e) Seqlock
    f) Disabling Kernel Preemption
    g) Kernel Path Synchronization
    h) Deadlock Handling

a) Interrupt Disabling:

- This scheme is used to avoid race conditions that can be occur in a shared resource environment.
- In this scheme, the interrupt circuit is disabled when the processor is about to execute critical section.
- When the execution is completed, the circuit is enabled.
- During this, the non-maskable interrupts are ignored and are not disabled.
- Instead, they are allocated to CPU.
- In X386 systems, cli instruction is used for disabling the interrupts whereas sti instruction is used for enabling it.

b) Atomic Operations:

- In this scheme, certain atomic operations such as test-and-set, read-modify-write are used in order to avoid the race conditions.
- This typically happens when there exist trivial memory cell or integer variable in the critical section.
- In Linux based systems, atomic_t variable used and can be accessed by various operations.
- These operations are implemented with X386 systems.
- In these systems, the following instructions are considered as memory barriers.
    - I/O port instructions
    - Lock byte prefixed instructions
    - Instructions that perform write operations on registers such as debug, system and control
    - Iret instructions

c) Kernel Semaphore:

- In this scheme, the kernel paths are synchronized

www.Jntufastupdates.com

- The kernel semaphores acts as objects whose type is struct semaphore.
- Linux uses up and down atomic operations which are used before and after the execution of critical section respectively.
- The services are served in FIFO order.
- The Linux also uses read/ write semaphores which are used to improve concurrent execution.
- The atomic operations used for this purpose include down_read, down_write, up_read, up_write.

d) Spinlock:

- This scheme is preferred in multiprocessor systems where semaphore-based synchronization cannot be used.
- This is because, the semaphore-based synchronization fails when two CPUs accesses the kernel data structures concurrently where one is kept in waiting and another one has completed its critical section execution.
- In this scheme, the waiting queue is removed and the down operation iteratively checks the value of integer until a spin lock is detected.
- At this position, the preemption is disabled to avoid freezing of the entire system.
- This scheme is preferred only for multi processor systems.

e) Seqlock:

- This scheme is similar to read/ write spinlock.
- In this scheme, higher pricrity is allocated to the write operations thereby allowing them to enter the critical section even if the users are actively executing critical section.
- When writes collide with read operations, the read operations need to be performed again.

f) Disabling Kernel Preemption:

- In this scheme, the kernel disables preemption in the situations where kernel executions cannot be preempted.
- An alternative to this method is to interrupt the process which is currently executing the kernel use exceptions or interrupt.
- However this process can lead to race conditions among all the kernel paths associated with a single process.

g) Kernel Path Synchronization:

- This scheme considers kernel paths as separate agents so that they can be easily synchronized.
- Once the execution of kernel is started with a kernel path, it cannot be interrupted until the critical section is completed.
- Due to this, another execution with same critical path cannot be started.
- To allow this, local interrupts need to be disable along with a spinlock.

h) Deadlock Handling:

- Linux does not support automatic dead lock detection and prevention.
- Therefore, it is the responsibility of kernel developers to implement the synchronization process is such a way that it avoids deadlock.

www.Jntufastupdates.com

4. Interrupt:

- When the interrupt handler is in a running state, the interrupt signal corresponding to the interrupt vector gets disable for a specific period of time.
- Also there are chances that an interrupt serviced by a process is not accessed for the process.
- We observe that interrupt handler is allowed to run in arbitrary process context.
- At the same time these handlers are not entitled to carry out any sort of blocking operation or process-specific operation.
- In order to accelerate the device utilization, the speed of certain entities of interrupt service is fasten up such as critical service parts are made faster while the execution of non-critical parts is delayed.
- For instance, when the packet arrives across the Ethernet, it is received by the critical part while the deferrable part simply analyzes the packet and proceeds further.
- According to Linux, an interrupt service is divided into two parts.
    i) Top Half which is executed by the Kernel immediately
    ii) Bottom Half which is executed later

i) Top Half which is executed by the Kernel immediately:

- This part is divided into two parts
- Critical Part
- Non-critical Part

Critical Part:

- It is executed while INTR interrupts disabled.

Non-critical Part:

- It is executed while INTR interrupts enabled.

- Some of the jobs performed by this part are,
    - Acknowledging the PC
    - Updation of data structures used by interrupting device and CPU

ii) Bottom Half which is executed later:

- Here interrupts are enabled and kernel maintains interrupt service routines in a form of queue.
- The execution of bottom half is not interrupted by other bottom half.
- However, it gets interrupted by any additional top half.
- Moreover the Linux 2.6 improves concurrency in multiprocessor systems by using various types of deferrable tasks which includes,
    - Softirg
    - Tasklet
- Both are executed in a serial manner on a single CPU.

Softirq:

- There are 32 softirqs which are allocated statically.
- Ideally, only 6 are used.
- They can run concurrently on multiple CPUs.

Tasklet:

- They cannot run concurrently on multiple CPUs.

www.Jntufastupdates.com

- However, tasklets of different types can run.

5. Exception:

- The mechanism for address space switching is analogous for both interrupt as well as exception.
- The approach of kernel to both interrupt and exception is different.
- In essence, the system call in Linux is taken as exception (software interrupt)
- The kernel uses trap gate descriptors to connect exception.
- The structure of exception handlers is as follows,
  - Storage of registers on stack of kernel
  - Handling of exception in C function
  - Execution of ref_from_exception function
- On the other hand, the Linux categorizes the exceptions into the following,
  - Errors in program (division by zero)
  - Page fault, which is a kernel design choice
- Ideally, the services offered by exceptions are very fast in the category of division by zero.
- It operates by sending a POSIX signal to the process in a running state.
- While in the later case, Linux retrieves the process from the exception.

6. System Call;

- An application process attains the requested services provided by the kernel by means of invoking system calls.
- A software interrupt is a type of interrupt accountable for initiating system call in X386 microprocessor and it is caused by an exception of special instruction "int OX80" present in the instruction set.
- Each and every system call need to be transmitted or passed via an interrupt vector OX80.
- Subsequently, a process need to perform the following actions in order to make a system call.
  - Initiating registers so as to pass arguments to the system call
  - Triggering the software interrupt
- For instance, arguments like identifying the call type and system call number are passed to the system call via EAX register.
- On each occurrence of software interrupt, the processor is not only responsible for altering mode of the operation to kernel but, also responsible for reading the starting address of generic system call handler system_call that stores in trap gate descriptor.
- After reading the address, the processor jumps to the specific address and sets the segment selector in CS, DS, ES and SS registers as follows.

Code Segment (CS) Register	:	The segment selector points to kernel mode
Data Segment (DS) Register	:	The segment selector points to kernel DS
Extra Segment (ES) Register	:	The segment selector points to kernel DS
Stack Segment (SS) Register	:	The segment selector points to user segment

www.Jntufastupdates.com

PART-II: ANDROID SOFTWARE PLATFORM:

7. Android:

- Android is a complete set of software for mobile devices such as tablet computers, notebooks, smartphones, electronic book readers, set-top boxes etc.
- It contains a linux-based Operating System, middleware and key mobile applications.
- It can be thought of as a mobile operating system.
- But it is not limited to mobile only. It is currently used in various devices such as mobiles, tablets, televisions etc.
- Android is a software package and linux based operating system for mobile devices such as tablet computers and smartphones.
- It is developed by Google and later the OHA (Open Handset Alliance).
- Java language is mainly used to write the android code even though other languages can be used.
- The goal of android project is to create a successful real-world product that improves the mobile experience for end users.

Android Versions:

VERSION	CODE NAME
1.5	Cupcake
1.6	Donut
2.1	Eclair
2.2	Froyo
2.3	Gingerbread
3.1 and 3.3	Honeycomb
4.0	Ice Cream Sandwitch
4.1, 4.2 and 4.3	Jelly Bean
4.4	KitKat
5.0	Lollipop
6.0	Marshmallow
7.0	Nougat
8.0	Oreo

Features of Android:

i) Storage:

- Android uses SQLite for storing the data.
- SQLite is a light-weight relational database.

ii) Connectivity:

www.Jntufastupdates.com

- Android supports various connections such as,
  - GSM/EDGE
  - Bluetooth
  - CDMA
  - UMTS
  - EV-DO
  - WiMax
  - WiFi
  - LTE

iii) Multi-touch:

- It provides multi-touch screens.

iv) Messaging:

- Android supports SMS for text messages and MMS for video messaging

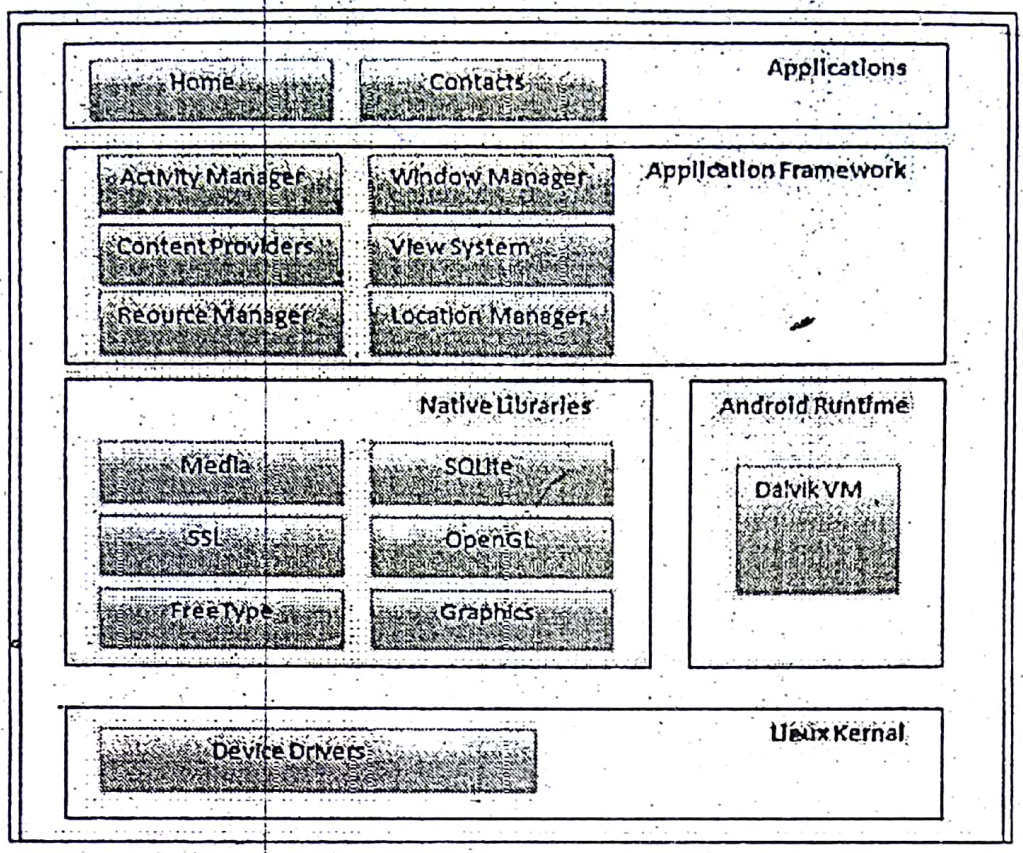
v) Flash:

- Android provides support for flash by using flash 10.1 for version 2.3
- The Other features are
  - Browser
  - Multi-tasking
  - Tethering (hotspot)
  - Camera
  - GPS
  - Digital Compass
  - Proximity Sensor
  - Accelerometer Sensor

8. Android Architecture:

- Android architecture or Android software stack is categorized into five parts:
  a) Linux kernel
  b) native libraries (middleware),
  c) Android Runtime
  d) Application Framework
  e) Applications
- Android architecture is depicted as,

www.Jntufastupdates.com



a) Linux kernel:

- It is the heart of android architecture that exists at the root of android architecture.
- Linux kernel is responsible for device drivers, power management, memory management, device management and resource access.

b) Native Libraries:

- On the top of linux kernel, their are Native libraries such as WebKit, OpenGL, FreeType, SQLite, Media, C runtime library (libc) etc.
- The WebKit library is responsible for browser support, SQLite is for database, FreeType for font support, Media for playing and recording audio and video formats.

c) Android Runtime:

- In android runtime, there are core libraries and DVM (Dalvik Virtual Machine) which is responsible to run android application.
- DVM is like JVM but it is optimized for mobile devices.
- It consumes less memory and provides fast performance.

d) Android Framework:

- On the top of Native libraries and android runtime, there is android framework.
- Android framework includes Android API's such as UI (User Interface), telephony, resources, locations, Content Providers (data) and package managers.
- It provides a lot of classes and interfaces for android application development.

e) Applications:

- On the top of android framework, there are applications.
- All applications such as home, contact, settings, games, browsers are using android framework that uses android runtime and libraries.
- Android runtime and native libraries are using linux kernal.

www.Jntufastupdates.com

8. Operating System Services:

- In Android, the operating system services are,
  - i) Memory Management
  - ii) Process/ Thread Management
  - iii) Power Management
  - iv) Network
  - v) Security
  - vi) Driver Management
  - vii) Interprocess communication

9. Android Runtime:

- The core part of android software platform is a runtime system.
- It comprises two main concepts, i.e., first is Dalvik VM and second are core libraries corresponding to the features of java programming language sustained by android.
- We observe that, every single android applications runs as a VM with in an individual local process.
- Also, an instance of Dalvik VM is present with in the process address space.
- One or many Dalvik VMs can be supported by android platform effectively just by sharing the code and data
- The java applications are then converted into Dalvik executable .exe format files, the translation of java application file into class file is done using standard java compilers.
- This java .class files are are converted into .dex files using dx tool.
- The Dalvik VM consists of platform independent byte code, where java byte code from .class files are converted into Dalvik byte code in .dex files.
- The Dalvik VM uses the Linux for implementing security and protection through uid (user identifier) and gid(group identifier)

10. Application Development:

- The android is not considered as application development platform.
- This implies that, the development of the application is done at some other place and subsequently installed in android platform for execution.
- Ideally the android has SDK, which provides tools and APIs required for creating android applications.
- A virtual mobile device called emulator is used for testing and debugging the applications.
- The guide for the development of the application can be obtained from http://developer/android.com/guide.
- This makes the development process easy and fast as large numbers of APIs are available across the site.
- Every single android application is a java application.
- They invoke android APIs to acquire services from the android platform.

www.Jntufastupdates.com

ii) Process Lifecycle:

- In android, the process lifecycle states are,

Visible:

- In this state, the activity is visible on the screen but not foreground and also not in interaction with user.

Foreground:

- In this state, the activity is in interaction with user and present on top of the screen.

Service:

- In this state, a service element is run by the process.

Background:

- In this state, the activity is paused and not made visible to the user.

Empty:

- In this state, no activity is performed by the process.

iii) IPC:

Refer Topic-2

iv) Security:

- The resources of the system cannot be used by an application without the user permission (during installation).
- For instance, if a mobile application requires vibration and the user do not permit, then the application cannot access the vibration system.

Short Answer Questions:

1. What is Inter-process Communication? List out different methods of IPC.

2. Write short notes on Spinlock and Seqlock.

3. Discuss about Operating System Services.

4. Explain about atomic operations.

5. Discuss about the classification of X386 interrupt descriptors

X386 interrupt descriptors types:

- These are 3 types
  - i) Task Gate Descriptor
  - ii) Interrupt Gate Descriptor
  - iii) Trap Gate Descriptor

i) Task Gate Descriptor:

- This type of class includes TSS selector which runs during interruptions.

ii) Interrupt Gate Descriptor:

www.Jntufastupdates.com

- This type of class includes a segment selector and offset for interrupt handler.
- Apart from this, IF bit in EFLAGS is processed by the processor for disabling INTR.

iii) Trap Gate Descriptor:

- This type of class is same as that of interrupt gate descriptor but with a difference is that instead of processing the IF bit it remains fixed.

6. Discuss about application development

7. What are the different types of elements used in application structure?

8. Discuss about operations performed by activity management

9. Explain different states of process life cycle

10. Write short notes on exception handling in Linux

Long Answer Questions:

1. Give a brief overview on Linux operating system and Discuss the different components of Linux.

2. Explain about IPC in Linux.

3. Explain about Synchronization supported by Linux

4. Explain about a) Interrupt b) Exception c) System call

5. What is android? Discuss its versions and features.

6. Discuss in detail about architecture of android software platform

7. Explain about a) Android Runtime b) Application Development c) Application Structure

8. Explain about application process management.

www.Jntufastupdates.com

11. Application Structure:

- The applications of an android platform are capable of providing its services to other applications by sharing the elements.
- This can be achieved when the system initiates the requested application and the requested element in a process.
- However the application that is intended for using the services must be permitted by the application that is providing the service.
- The application constitutes of numerous entry points and one or more elements of the following types,
  - Service Element
  - Activity Element
  - Content provider Element
  - Broadcast Receiver Element

Service Element:

- It can be defined as a task which runs in background for unlimited duration in presence of other activities.
- For example, a user can play a song in background while doing other activities.
- Using the android.app.service class, service object can be instantiated.

Activity Element:

- It is a visual user interface through a window.
- The view object specifies visual content corresponding to a window, such that every viw manages a rectangular area within the window.

Content provider Element:

- This type of element assists other applications to make use of data from their own application.
- The class of this element extends the class android.content.contentprovider.

Broadcast Receiver Element:

- This element as the name implies, receives the broadcast messages and respond to them correspondingly.
- The typical response includes notifying the user, initiating the activity.
- The object for this element is instantiated from the class android.content.BroadCastReceiver.

12. Application Process Management:

- Application process management of an android has 4 phases
  - i) Activity Management
  - ii) Process Lifecycle
  - iii) IPC
  - iv) Security

i) Activity Management:

- An application possesses in different activities where every activity is an object of the class android.app.Activity.
- This object consists of hooks for the following operations,

Create:

- This deals with the creation of an activity.

www.Jntufastupdates.com

- This is done by calling the onCreate( ) hook.

Start:

- This deals with the initialization of activity.
- It is done by calling the onStart( ) hook.

Resume:

- This deals with the activity interacting with its user.
- It is carried out by calling the onResume( ) hook.

Pause:

- This deals with stopping an activity for temporary purpose and resuming some other activity for completion.
- It is carried out by calling the onPause( ) hook

Stop:

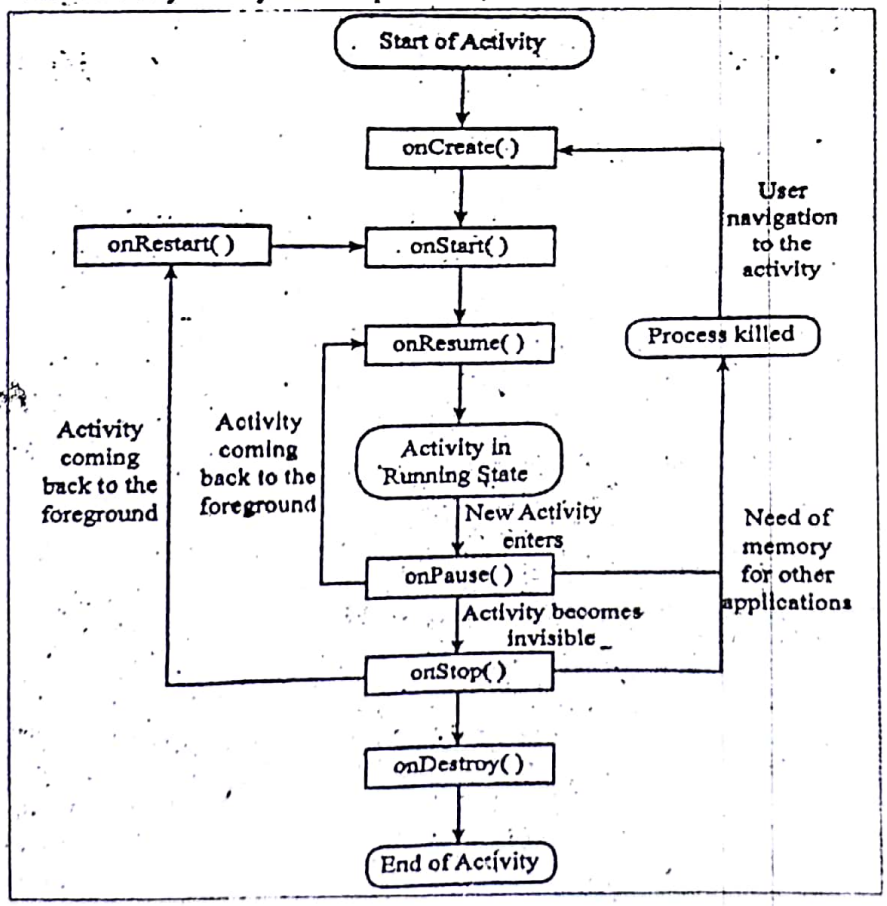
- This deals with the activity becoming invisible to the user.
- It is carried out by calling the onStop( ) hook.

Destroy:

- This deals with the activity being destroyed either manually or automatically by the system.
- It is carried out by calling the onDestroy( ) hook.

Restart:

- This deals with the activity getting restarted after it was stopped.
- It is carried out by calling the onRestart( ) hook.
- The Activity Lifecycle is depicted as,



www.Jntufastupdates.com