System Programming and Operating Systems Lab

ASSIGNMENT 11

Name: Shrirang Mhalgi
Roll No: 322008
Batch: B1

1 Date of Completion:

26/03/2019

2 Aim:

Study assignment on process scheduling algorithms in Android and Tizen.

3 Objectives:

- To understand Android OS
- To understand Tizen OS
- To understand Concept of process management

4 Theory:

ANDROID OS :- Android is a mobile operating system developed by Google, based on a modified version of the Linux Kernel and other open source software and designed primarily for touchscreen mobile devices such as smartphones and tablets. In addition, Google has further developed Android TV for televisions, Android Auto for cars, and Android Wear for wrist watches, each with a specialized user interface. Variants of Android are also used on game consoles, digital cameras, PCs and other electronics.

Some Android Versions :
Gingerbread (2.3)
Honeycomb (3.0)
Ice Cream Sandwich (4.0)
Jelly Bean (4.3/4.2/4.1)
KitKat (4.4)
Lollipop (5.0)

Advantages :

1. Support 2D and 3D Graphics
2. Support multiple language
3. Java support
4. Faster web browser
5. Support audio , video etc

Disadvantages :

1. Slow response
2. Heat
3. Advertisement etc

Tizen OS : - Tizen is a mobile operating system developed by Samsung that runs on a wide range of Samsung devices, including smartphones; tablets; in-vehicle infotainment (IVI) devices; smart televisions; smart cameras; smartwatches; Blu-ray players; smart home appliances (refrigerators, lighting, washing machines, air conditioners, ovens/microwaves); and robotic vacuum cleaners.

Android vs Tizen Operating system :

Easy and Convenient Navigation: Scrolling and navigation becomes smooth with Tizen
Fast and Lightweight: Tizen Operating System is easy to operate and fast as compared to Googles Android Wear
Visual Effects: Tizen extends 3D visual effects of various gaming apps installed on the device
UI: TouchWiz UI
Resizable boxes: One of the amazing features of Tizen is its ability to dynamically resize the icons on screen to display more information or less
Enhanced Processors: Tizen 3.0 will bring 64 bit processors with it, compatible with x86 processors and 64 bit RAM, which Google is also anticipating with its update.
Tizen vs. Android Gaming Platform: Tizen 3.0 will be able to make use of Vulkan APIs and will prove to be a good gaming platform unlike Android.
Supporting Devices: Tizen is being used in smart TVs, refrigerators, smart watches, smart phones, washing machines, light bulbs, vacuum cleaners while Android is visible only in smart phones, computers or smart watches.
IoT Devices: Tizen 3.0 is compatible with Artik cloud which will extend cloud services for IoT devices.
Battery Consumption: Samsungs devices with Tizen OS consume less power than Android devices according to mobile experts.
Pricing: Devices with Tizen support will be made available at various price points but focus will be on lower end markets. Unlike Android, that has its presence in both upper as well as lower end markets.

Advantages of using Tizen OS

- It is an open source Operating System
- The OS is Compatible with various mobile platform. Application built on Tizen can be launched on iOS and Android too with few changes.
- The Tizen OS is so Flexible to offer many applications and adapt too, with little changes
- Immense personalization capability supported by ARM x86 processor

PROCESS SCHEDULING ALGORITHMS IN ANDROID AND TIZEN OS :

- Normal scheduling

Android is based on Linux and uses the Linux kernels scheduling mechanisms for determining scheduling policies. This is also true for Java code and threads. The Linuxs time sliced scheduling policy combines static and dynamic priorities. Processes can be given an initial priority from 19 to -20 (very low to very high priority). This priority will assure that higher priority processes will get more CPU time when when needed. These level are however dynamic, low level priority tasks that do not consume their CPU time will fine their dynamic priority increased. This dynamic behaviour results is an overall better responsiveness. In terms of dynamic priorities it is ensured that lower priority processes will always have a lower dynamic priority than processes with real-time priorities. Android uses two different mechanisms when scheduling the Linux kernel to perform process level scheduling

- Real-time scheduling

The standard Linux kernel provides two real-time scheduling policies, SCHED FIFO and SCHED RR. The main real-time policy is SCHED FIFO. It implements a first-in, first-out scheduling algorithm. When a SCHED FIFO task starts running, it continues to run until it voluntarily yields the processor, blocks or is preempted by a higher-priority real-time task. It has no timeslices. All other tasks of lower priority will not be scheduled until it relinquishes the CPU. Two equal-priority SCHED FIFO tasks do not preempt each other. SCHED RR is similar to SCHED FIFO, except that such tasks are allotted timeslices based on their priority and run until they exhaust their timeslice. Non-real-time tasks use the SCHED NORMAL scheduling policy (older kernels had a policy named SCHED OTHER).

- Thread Scheduling

A thread scheduler decides which threads in the Android system should run, when, and for how long Androids thread scheduler uses two main factors to determine the scheduling:
- Niceness Values
- Control Groups (Cgroups) Niceness Values
- a thread with a higher niceness value will run less often than those with a lower niceness value (this sounds paradoxical)
- niceness value has the range of -20 (most prioritized) to 19 (least prioritized); default value is 0
- a new Thread inherits its priority from the thread where it is started
- it is possible to change the priority via: o thread.setPriority(int priority) - values: 0 (least prioritized) to 10 (most prioritized) o process.setThreadPriority(int priority) - values: -20 (most prioritized) to 19 (least prioritized)

- Priority Based Pre-Emptive Task Scheduling for Android Operating System

The key concept present in any operating system which allows the system to support multitasking, multiprocessing, etc. is Task Scheduling . Task Scheduling is the core which refers to the way the different processes are allowed to share the common CPU. Scheduler and dispatcher are the softwares which help to carry out this assignment . Android operating system uses O (1) scheduling algorithm as it is based on Linux Kernel 2.6. Therefore the scheduler is names as Completely Fair Scheduler as the processes can schedule within a constant amount of time, regardless of how many processes are running on the operating system . Pre-emptive task scheduling involves interrupting the low priority tasks when high priority tasks are present in the queue. This scheduling is particularly used for mobile operating system as the CPU utilization is medium, turnaround time and response time is high. Mobile phones are required to meet specific time deadlines for the tasks to occur.

- Fixed-priority pre-emptive scheduling

Fixed-priority preemptive scheduling is a scheduling system commonly used in real-time systems. With fixed priority preemptive scheduling, the scheduler ensures that at any given time, the processor executes the highest priority task of all those tasks that are currently ready to execute. The preemptive scheduler has a clock interrupt task that can provide the scheduler with options to switch after the task has had a given period to executethe time slice. This scheduling system has the advantage of making sure no task hogs the processor for any time longer than the time slice. However, this scheduling scheme is vulnerable to process or thread lockout: since priority is given to higher-priority tasks, the lower-priority tasks could wait an indefinite amount of time. One common method of arbitrating this situation is aging, which gradually increments the priority of waiting processes and threads, ensuring that they will all eventually execute. Most Real-time operating systems (RTOSs) have preemptive schedulers. Also turning off time slicing effectively gives you the non-preemptive RTOS.

Preemptive scheduling is often differentiated with cooperative scheduling, in which a task can run continuously from start to end without being preempted by other tasks. To have a task switch, the task must explicitly call the scheduler. Cooperative scheduling is used in a few RTOS such as Salvo or TinyOS

- Dynamic priority pre-emptive scheduling

earliest-deadline first scheduling:
a job's priority is inversely proportional to its absolute deadline. The difference between deadline monotonic scheduling and earliest-deadline first scheduling is that DM is a static priority algorithm, EDF is a dynamic priority algorithm. [3]EDF can guarantee that all deadlines are met provided that the total CPU utilization is less than 1.

5 Conclusion:

In this assignment we learn in detail the concept of process scheduling of Android and Tizen OS.