RAMA UNIVERSITY

www.ramauniversity.ac.in

FACULTY OF ENGINEERING & TECHNOLOGY

BCS-503: Object Oriented Techniques

Lecture-31

Preeti Singh
Computer Science & Engineering

OBJECTIVES

In this PPT, you will learn to:

- ❖Define a thread
- ❖Define multithreading
- ❖List benefits of multithreading
- ❖Create threads
- ❖Discuss thread states

MULTITASKING VS MULTITHREADING

- **Multitasking** is the ability to run one or more programs concurrently.
- Operating system controls the way in which these programs run by scheduling them.
- Time elapsed between switching of programs is minuscule.
- **Multithreading** is the ability to execute different parts of a program, called threads, simultaneously.

THREAD

- Thread is the smallest unit of executable code that performs a particular task.
- An application can be divided into multiple tasks and each task can be assigned to a thread.
- Many threads executing simultaneously is termed as Multithreading.

BENEFITS OF MULTITHREADING

- Multithreading requires less overhead than multitasking.
- In multitasking, processes run in their own different address space.
- Tasks involved in multithreading can share the same address space.
- Inter-process calling involves more overhead than inter-thread communication.
- Multithreading allows us to write efficient programs that make maximum use of the CPU.
- Multithreading allows animation loops to sleep for a second between each frame without causing the whole system to pause.

THE ‘MAIN’ THREAD

When Java programs execute, there is always one thread running and that is the main thread.

- It is this thread from which child threads are created.
- Program is terminated when main thread stops execution.
- Main thread can be controlled through Thread objects.
- Reference of the main thread can be obtained by calling the currentThread() method of the Thread class.

CREATING THREADS

Thread objects can be created in two ways:

- **Declare a class that is a sub-class of the class Thread defined in java.lang package.**
  - class mythread extends Thread

- **Declare a class that implements the Runnable interface.**
  - **class mythread implements Runnable**

While using Applets, Thread class cannot be extended. Therefore, one has to implement the Runnable interface.

CREATING THREADS (Contd.)

- After a new thread has been initiated, we use the start() method to start the thread, otherwise it is an empty Thread object with no system resources allocated.

```
Mythread t = new Mythread();
t.start();
```

- When start() method is invoked, the system allocates resources required to run the thread and schedules the thread to run.

- It then calls the thread's run() method.

THREAD STATES

Born: A newly created thread is in a born state.

Ready: After a thread is created, it is in its ready state waiting for start() method to be called.

Running: Thread enters the running state when it starts executing.

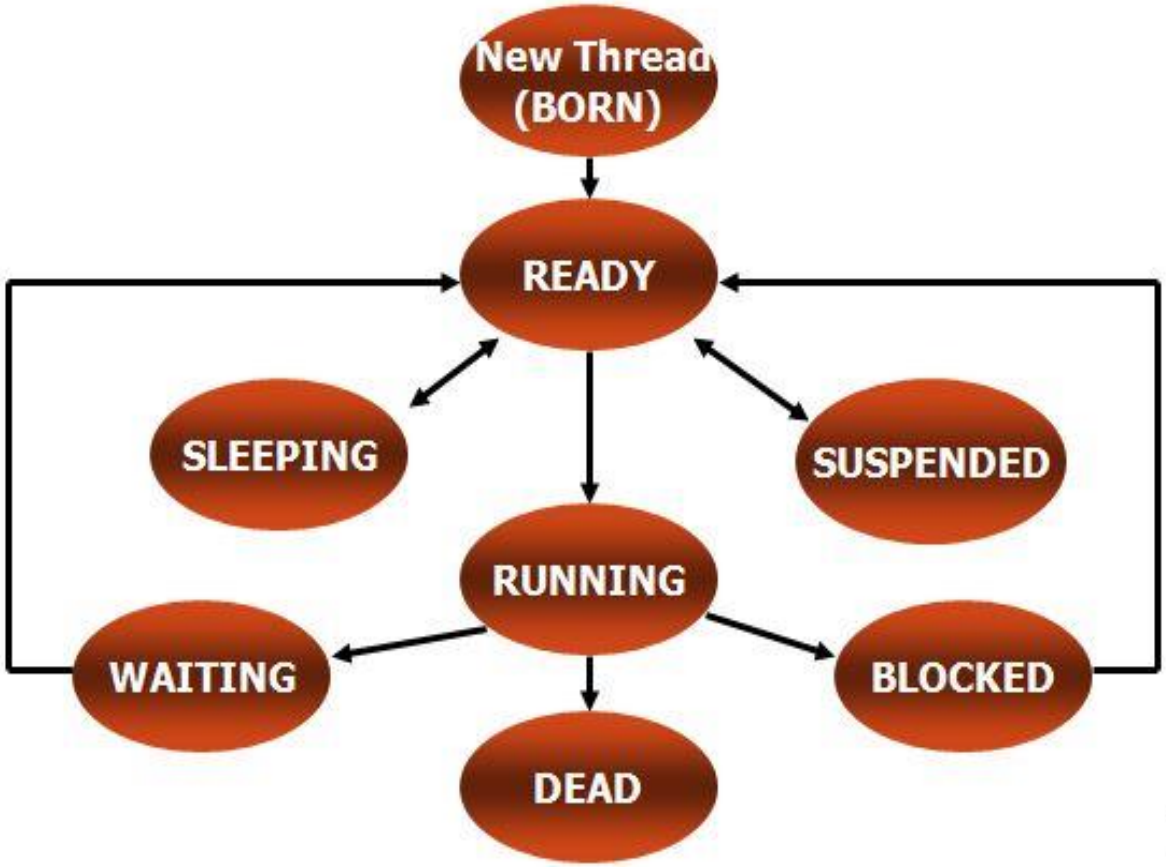
Sleeping: Execution of a thread can be halted temporarily by using sleep() method. The thread becomes ready after sleep time expires.

Waiting: Thread is in waiting state if wait() method has been invoked. Used when two or more threads run concurrently.

Blocked: The thread enters a blocked state when it waits for an event such as Input/Output operations.

Dead: The thread enters the dead state after the run() method has finished or the thread's stop() method is called.

DIFFERENT STAGES IN THE LIFE OF A THREAD



REFERENCES

1. James Rumbaughet. al, "Object Oriented Modeling and Design", PHI

2. Grady Booch, James Rumbaugh, Ivar Jacobson, "The Unified Modeling Language User Guide", Pearson Education

3. Naughton, Schildt, "The Complete Reference JAVA2", TMH

4. Mark Priestley "Practical Object-Oriented Design with UML", TMH

5. Booch, Maksimchuk, Engle, Young, Conallen and Houstan, "Object Oriented Analysis and Design with Applications", Pearson Education

6. Pandey, Tiwari, " Object Oriented Programming with JAVA" , Acme Learning

7. https://www.javatpoint.com/java-tutorial

8. https://www.tutorialspoint.com/java/index.htm

9. https://www.tutorialspoint.com/object_oriented_analysis_design/index.htm

10. https://www.slideshare.net/niitstudentcare/

Multiple Choice Question:

Q1. What is multithreaded programming?

a) It’s a process in which two different processes run simultaneously

b) It’s a process in which two or more parts of same process run simultaneously

c) It’s a process in which many different process are able to access same information

d) It’s a process in which a single process can access information from many sources

Multiple Choice Question:

Q2. Which of these are types of multitasking?

a) Process based

b) Thread based

c) Process and Thread based

d) None of the mentioned

Multiple Choice Question:

Q3. Which of these statements is incorrect?

a) By multithreading CPU idle time is minimized, and we can take maximum use of it

b) By multitasking CPU idle time is minimized, and we can take maximum use of it

c) Two thread in Java can have the same priority

d) A thread can exist only in two states, running and blocked

Multiple Choice Question:

Q4. What requires less resources?

a) Thread

b) Process

c) Thread and Process

d) Neither Thread nor Process

Multiple Choice Question:

Q5. What is true about threading?

a) run() method calls start() method and runs the code

b) run() method creates new thread

c) run() method can be called directly without start() method being called

d) start() method creates new thread and calls code written in run() method

Summary

In this PPT, you learned that:

- Multithreading allows programmers to write efficient programs that make the maximum use of the CPU.
- Java provides built-in support for multithreading in the form of classes and interfaces.
- When Java programs are executed, there is already one thread that is running and it is the main thread. This main thread is important for two reasons:
  - It is the thread from which child threads will be created.
  - Program is terminated when the main thread stops execution.
- Thread objects can be created in two ways:
  - Declare the class to be a sub-class of the Thread class where we need to override the run() method of the Thread class.
  - Declare a class that implements the Runnable interface. Then define the run() method.