Multi-threading in Java

Prepared by Harish Patnaik

School of Computer Engineering, KIIT Deemed to be University

Content

1. Intro to Multi-threading
2. Thread creation
3. Multiple threads
4. Synchronization
5. Thread priority
6. Inter-thread communication

Multithreading

- Different parts of the same program run concurrently
- Each part of such program is called Thread and each thread defines a separate path of execution.
- Multithreading enables us to write efficient program that make maximum use of the CPU because idle time can be reduced.

- Thread class -
  - String getName()
  - setName(String)
  - int getPriority()
  - setPriority(int)
  - void start()
  - void sleep(int)

Thread creation

- A thread can be created on any object that implements Runnable interface
  - It has to implement run() method

Example - testt.java

- A thread can be created on any object that extends Thread class

Example - threadcls.java

Multiple thread

- Multiple threads can be created out of a single object

  Example - mthread.java

- Synchronization among the threads

  Example - mthsync.java

Thread priority

- Threads can be assigned some Priority value between 1 to 10. The default priority value (NORM_PRIORITY) is 5.
- Thread with highest priority will enter the synchronized method first.
- void setPriority(int)

  int getPriority()
- Sleep method stops execution of the thread for some miliseconds.

  void sleep(int)

  Example - mthreadsl.java

Inter-thread communication

- Communication is based on following final methods defined in Object class.
  - wait() - tells the calling thread to giveup the monitor and go to sleep mode until some other thread enters the same monitor and call notify()
  - notify() - wakes up the first thread that called wait() on the same object
  - notifyAll() - wakes up all the thread that called wait() on the same object. The highest priority thread will run first.

- All three methods can be called only within synchronized method.

Example - mthnotify.java

Thank you