Telerik Academy

Android Threads

Dimitar Ivanov

Mobile Applications with Android

Telerik Software Academy
http://academy.telerik.com

Telerik Academy

Today's Topics

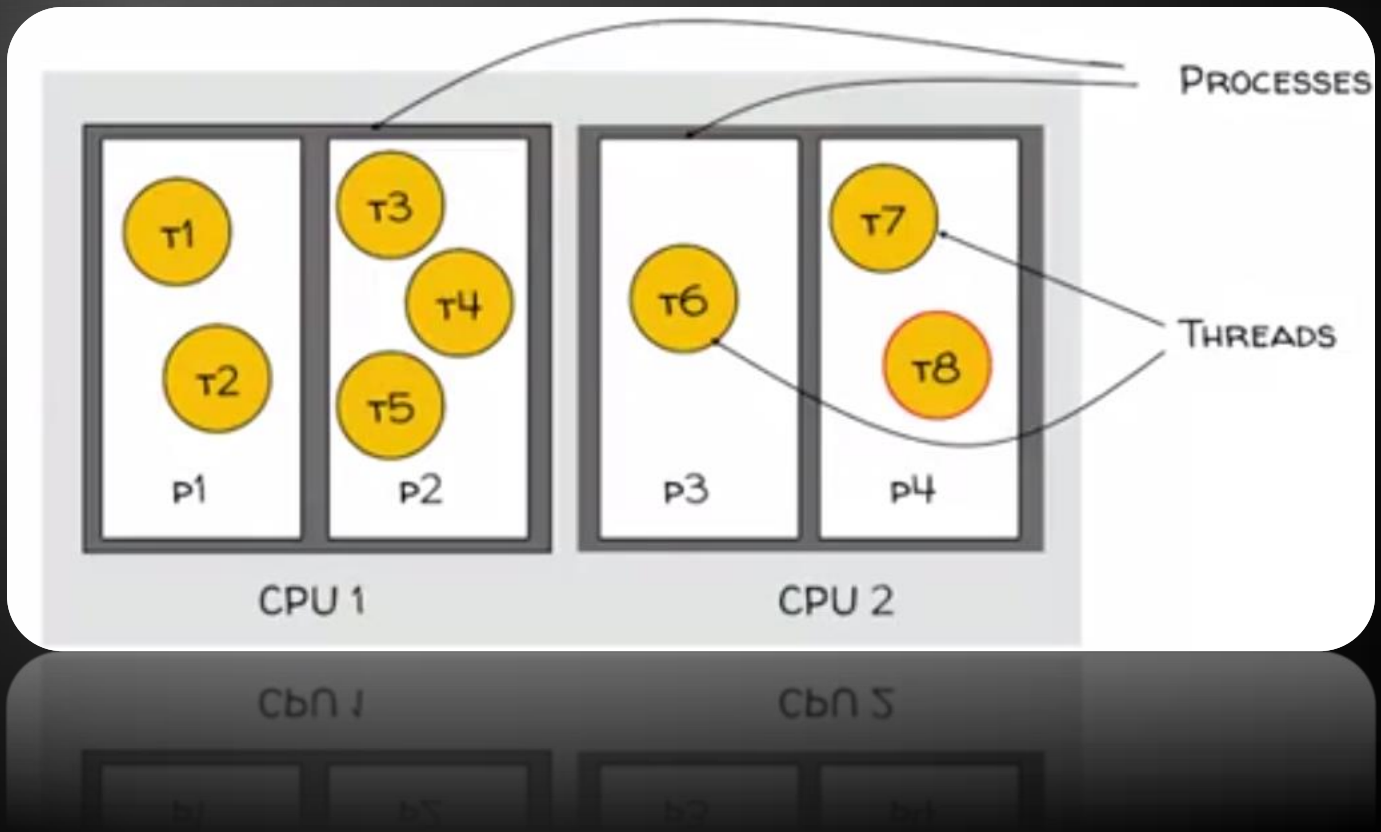
- Threading Overview
- Android's UI Thread
- The AsyncTask Class
- The Handler Class

What is a Thread ?

- Conceptual View
  - Parallel computation running in a process
- Implementation View
  - A program counter and a stack
  - With heap and static areas that are shared with other threads.

Telerik Academy

What is a Thread ?



Java Threads

- Represented by an object of type Java.Lang.Thread.
  - Implements the runnable interface
    - void run()
- Some thread methods
  - void start()
    - Starts the Thread
  - void sleep(long time)
    - Sleeps for the given period

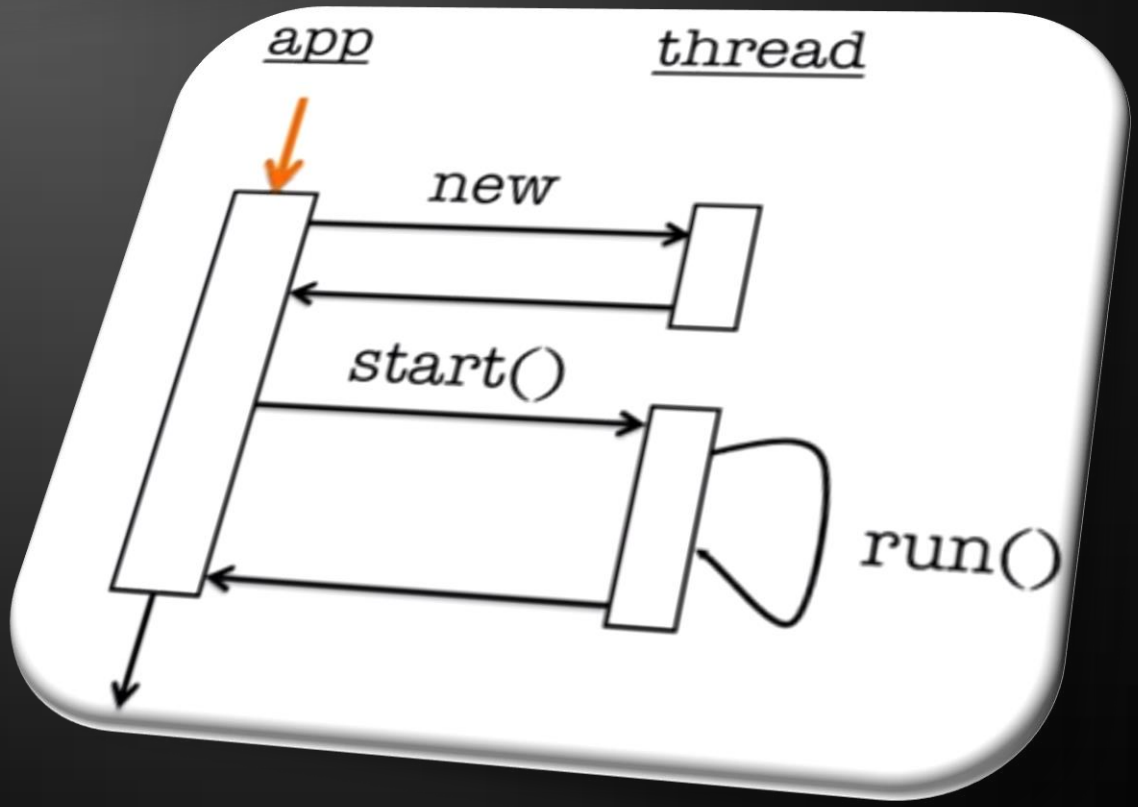
Java Threads

- Void wait()
  - Current thread waits until another thread invokes notify() on this object.
- Void notify()
  - Wakes up a single thread that is waiting on this object.

Basic Thread Use Case

- Instantiate a Thread object
- Invoke the Thread's start() method
  - Thread's run() method get called
- Thread terminates when run() returns

Basic Thread Use Case



Telerik Academy

ThreadingNoThreading Demo

- Application displays two buttons
- LoadIcon
  - Load a bitmap from a resource file & display
- Show loaded bitmap
- Other Button
  - Display some Toast text

AsyncTask

- Provides a structured way to manage work involving background & UI Threads
- Background Thread
  - Perform work
  - Indicate progress
- UI Thread
  - Does setup
  - Publishes intermediate progress
  - User results

AsyncTask

- Generic Class

```
Class AsyncTask<Params,Progress,Result> {
…
}
```

- Generic Type Parameters
  - Params – Type used in background work

AsyncTask

- void onPreExecuted()
  - Runs in UI Thread before doInBackground()
- Result
  - doInBackground(Params...params)
    - Performs work in background Thread
    - May call
      - void publishProgress(Progress... values)

AsyncTask

- void onProgressUpdate(Progress... values)
  - Invoked in response to publishProgress()
- void onPostExecute(Result result)
  - Runs after doInBackground()

Async Demo

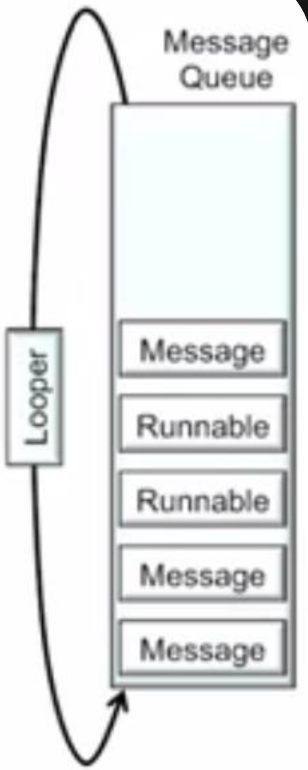
Handler

- Each handler is associated with a Thread
- One thread can hand off work to another Thread by sending
- Messages & Posting Runnables to a handler associated with other Thread.
- Runnable
  - Contains an instance of the Runnable interface
  - Sender implements response
- Message
  - Can contain a message code, an object & integer arguments

Handler Architecture

Each Android Thread is associated with a MessageQueue & a Looper

A MessageQueue holds Messages and Runnables to be dispatched by the Looper



Runnables & Handlers

- boolean post(Runnable r)
  - Add Runnable to the MessageQueue
- Boolean postAtTime(Runnable r,long uptimeMillis)
  - Add Runnable to the MessageQueue.Run at a specific time(based on SysteClock.upTimeMillis())
- Boolean postDelayed(Runnable r,long delayMillis)
  - Add Runnable to the message queue.Run after the specific amount of time elapsed.

Messages & Handlers

- Create Message & set Message content
  - Handler.obtainMessage()
  - Message.obtain()
- Message parameters include
  - Int arg1,arg2, what
  - Object obj
  - Bundle data
- Many variants.See documentation

Messages & Handlers

- sendMessage()
  - Queue Message now
- sendMessageAtFrontOfQueue()
  - Insert Message now at front of queue
- sendMessageAtTime()
  - Queue message at the stated time

Handler

Live Demo

Telerik Academy

Android Threads

http://academy.telerik.com

Homework

1. Asynchronously download an image from a random web resource and indicate the progress of the download
   - i.e. http://images.google.com