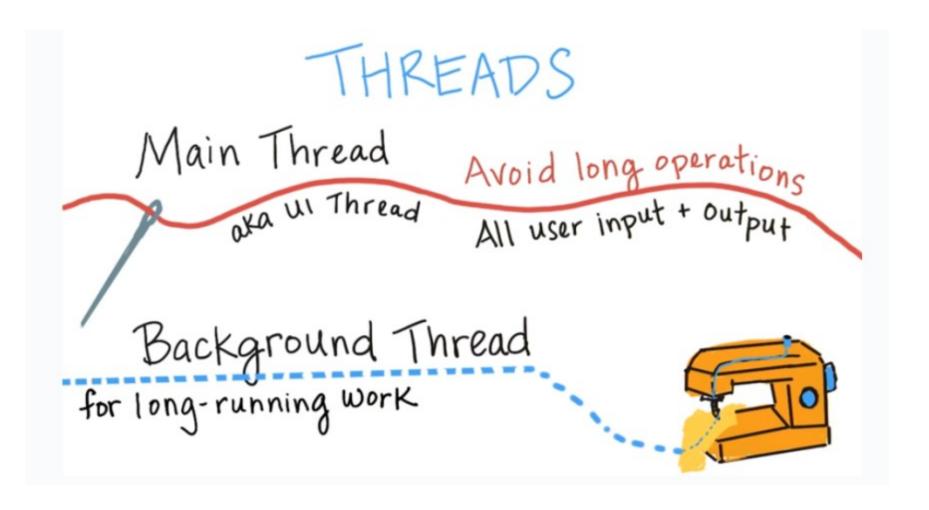
## Introduction to Service

Background processing.....

#### Services

- are used to run tasks in the background/invisible ...
- Please remember services by default run on the main thread.
- IntentService runs on background thread.

#### **Threads**



### Services (2)

- derived by extending the Service class.
- started/bound by calling:
  - startService(Intent intent)/bindService(Intent intent);

Always remember to stop/unbind a started Service

 stopService(Intent intent)/unbindService(ServiceConnection conn)

#### **Priority**

1

 Active activities or foreground services

9

 Visible activities or running services

3

Background activities

# Questions ???

### **Task** (1)

- 1. Create a service and bind it to the main activity
  - a. Create a **new Java class** (e.g. "BoundService") which extends **Service**
  - b. Implement abstract method
     public IBinder onBind(Intent intent) { ... }
  - c. Add to Android manifest <service android:name=,,package.name.ServiceName" />
  - d. Declare a **member variable** of type of your new service inside **MainActivity**
  - e. Override onStart() method inside MainActivity
    - a. Create a **new intent** with the new service class
    - **b. Start** the service
    - c. Bind the service

      Hint: You have to declare a new variable of type

      ServiceConnection!
  - f. Override onStop() method inside MainActivity (Unbind the service)

### **Task (2)**

- Create two buttons (one for printing the time stamp, one for stopping the service) and one text view (for showing the time stamp) inside the layout file for MainActivity
- 3. Create a method inside the service to return the elapsed time since starting the app Hint: Use SystemClock.elapsedRealtime()!
- 4. Create new instances of OnClickListener to the buttons
  Hint: To get the service you need to create a class (e.g. "MyBinder")
  which extends from Binder and provides a method to get the service
  (e.g. "return BoundService.this"). In the onServiceConnected
  method you can assign the member variable to the instance of the
  service. You'll also need a member variable in the service class of type
  IBinder or MyBinder.
  - See developer.android.com/guide/components/bound-services.html

## **Task** (3)

- 5. Create a Boolean member (e.g. "mServiceBound") in MainActivity to check and maintain the status of the service (only print the time stamp if the service is bound, only unbind if bound)
- Use an instance of Chronometer in the service class to keep track of the time.
  - developer.android.com/reference/android/widget/Chronometer.html

# Questions ???

#### Sources

- Android Developers developer.android.com
- Udacity Developing Android Apps classroom.udacity.com/courses/ud853
- Bound Service Example In Android truiton.com/2014/11/bound-service-example-android/