Lab3 Flow

Slides page 1-17

Fast Talk

Slides page 18

thread demo:

In new project,

Create a service called MyService.

In MainActivity onCreate,

Add code:

startService(new Intent(MainActivity.this, MyService.class));

In MyService

Add code:

@Override

public void onCreate() {

ComputationRun computation=new ComputationRun();

Thread computationThread=new Thread(computation);

computationThread.start();

//to close compuation thread

//computationThread.interrupt();

}

class ComputationRun implements Runnable

{

// volatile makes sure reading it from memory, not cache,

volatile boolean isRunning = false;

@Override

public void run() {

isRunning = true;

//

int usedTime = 0;

while (isRunning)

{

usedTime = usedTime + 1;

Log.d("haha", "run once");

if (usedTime>12)

Thread.currentThread().interrupt();

if(Thread.currentThread().isInterrupted()) //Thread refers to current thread

{

// when not blocked, interrupt() will go to this part

// release resources

Log.d("haha", "quit when not blocked");

isRunning = false;

// quit the service if u want

//MyService.this.stopSelf();

return;

}

try {

Thread.sleep(1000);

} catch (InterruptedException e) {

// when blocked, interrupt() will go to this part

// e.g., when the service calls interrupt()

Log.d("haha", "quit when blocked");

e.printStackTrace();

// release resources

isRunning = false;

// quit the service if u want

//MyService.this.stopSelf();

}

}

}

}

Slides page 19-34

Fast Talk

Slides page 35

sensor demo:

In new project,

Create a service called MyService.

In MainActivity onCreate,

Add code:

startService(new Intent(MainActivity.this, MyService.class));

In MyService

Change the MyService class to

public class MyService extends Service implements SensorEventListener {

private static float accelerometer\_x = 0;

private static float accelerometer\_y = 0;

private static float accelerometer\_z = 0;

SensorManager mSensorMgr;

Handler handler;

HandlerThread mHandlerThread;

int usedTime = 0;

public MyService() {

}

@Override

public void onDestroy() {

Log.d("haha", " service stopped...");

}

@Override

public void onCreate() {

//here is still main thread!

//dont do anything heavy here

mSensorMgr = (SensorManager) getSystemService(Context.SENSOR\_SERVICE);

mHandlerThread = new HandlerThread("sensorThread");

mHandlerThread.start();

//to stop

//mHandlerThread.quitSafely();

handler = new Handler(mHandlerThread.getLooper());

mSensorMgr.registerListener(this, mSensorMgr.getDefaultSensor(Sensor.TYPE\_ACCELEROMETER),

SensorManager.SENSOR\_DELAY\_FASTEST, handler);

//handle it in a new thread

//if service paused, even if you kill activity, thread will run unless you close it

//close sensor

//mSensorMgr.unregisterListener(this);

//write a new thread for compuation

ComputationRun computation=new ComputationRun();

Thread computationThread=new Thread(computation);

computationThread.start();

//to close compuation thread

//computationThread.interrupt();

}

class ComputationRun implements Runnable

{

// volatile makes sure reading it from memory, not cache,

volatile boolean isRunning = false;

@Override

public void run() {

isRunning = true;

//

while (isRunning)

{

usedTime = usedTime + 1;

//Log.d("SENSORS", "accelerometer\_x= " + accelerometer\_x);

Log.d("SENSORS", "accelerometer\_x= " + accelerometer\_x);

Log.d("SENSORS", "accelerometer\_y= " + accelerometer\_y);

Log.d("SENSORS", "accelerometer\_z= " + accelerometer\_z);

if (usedTime>12)

Thread.currentThread().interrupt();

if(Thread.currentThread().isInterrupted()) //Thread refers to current thread

{

// release resources

mSensorMgr.unregisterListener(MyService.this);

mHandlerThread.quitSafely();

isRunning = false;

// quit the service if u want

//MyService.this.stopSelf();

return;

}

try {

Thread.sleep(1000);

} catch (InterruptedException e) {

e.printStackTrace();

// release resources

mSensorMgr.unregisterListener(MyService.this);

mHandlerThread.quitSafely();

isRunning = false;

// quit the service if u want

//MyService.this.stopSelf();

}

}

}

}

@Override

public IBinder onBind(Intent intent) {

// TODO: Return the communication channel to the service.

throw new UnsupportedOperationException("Not yet implemented");

}

public void onAccuracyChanged(Sensor sensor, int accuracy) {

}

public void onSensorChanged(SensorEvent event) {

Sensor sensor = event.sensor;

if (sensor.getType() == Sensor.TYPE\_ACCELEROMETER) {

// accelerometer data

accelerometer\_x = event.values[0];

accelerometer\_y = event.values[1];

accelerometer\_z = event.values[2];

//Log.d("SENSORS", "accelerometer\_x= " + accelerometer\_x);

//Log.d("SENSORS", "accelerometer\_y= " + accelerometer\_y);

//Log.d("SENSORS", "accelerometer\_z= " + accelerometer\_z);

}

}

}

Slides page 36-37

Fast Talk

Slides page 39

localisation demo:

Download zip and open it in Android Studio

Explain In AndroidManifest.xml

<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION"/>

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"/>

<uses-feature android:name="android.hardware.location.network" />

<uses-feature android:name="android.hardware.location.gps" />

Explain MainActivity class code

Slides page 40-

Fast Talk