Difference between Android Service,Thread,IntentService and AsyncTask

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Service** | **Thread** | **IntentService** | **AsyncTask** |
| **When to use ?** | Task with no UI, but shouldn't be too long. Use threads within service for long tasks.  eg.many download at a time in parallel. Makes process faster but slower bcoz of nw speed. | - Long task in general.  - For tasks in **parallel** use Multiple threads (traditional mechanisms) | - Long task **usually** with no communication to main thread.  **sequential** **(Update)**- If communication is required, can use main thread handler or broadcast intents  - When callbacks are needed (Intent triggered tasks)  eg. Image download  one download at a time and sending just done broadcast | - **Small task having to communicate with main thread**.  **Progress update**  - For tasks in parallel use multiple instances OR Executor  Eg. Web page download…progress I showned |
| **Trigger** | Call to method onStartService() | Thread start()  method | Intent | Call to method execute() |
| **Triggered From (thread)** | Any thread | Any Thread | Main Thread (Intent is received on main thread and then worker thread is spawed) | Main Thread |
| **Runs On (thread)** | Main Thread | Its own thread | Separate worker thread | Worker thread. However, Main thread methods may be invoked in between to publish progress. |
| **Limitations /** **Drawbacks** | May block main thread | - Manual thread management  - Code may become difficult to read | - Cannot run tasks in parallel.  - Multiple intents are queued on the same worker thread. | - **one instance can only be executed once (hence cannot run in a loop)**   - Must be created and executed from the Main thread |

**Toast**

Toast.*makeText*(getActivity(),**"inside serverfrag"**,Toast.***LENGTH\_SHORT***).show();

//**Read** Read a short text file to the internal storage

**private void** populateTask() {  
  
 **try** {  
 Scanner scan = **new** Scanner(openFileInput(**"TaskList.txt"**));  
 **while** (scan.hasNextLine())  
 {  
 String line = scan.nextLine();  
 String[] pieces = line.split(**"\t"**);  
 **arrayListTask**.add(**new** Task(pieces[0], pieces[1]));  
 **taskvwadapter**.notifyDataSetChanged();  
 **etTitle**.setText(**""**);  
 **etDesc**.setText(**""**);  
  
  
 }  
 Toast.*makeText*(MainActivity.**this**,**"Read from file done"**,Toast.***LENGTH\_SHORT***).show();  
  
  
 } **catch** (FileNotFoundException e) {  
 e.printStackTrace();  
 }  
  
 }

**//Write file**  
 PrintStream output = **null**;  
 **try** {  
 output = **new** PrintStream(openFileOutput(**"TaskList.txt"**, ***MODE\_APPEND***));  
 output.println(**titletobeadded** + **"\t"** + **desctobeadded**);  
 Toast.*makeText*(MainActivity.**this**, **"write done"**, Toast.***LENGTH\_SHORT***).show();  
  
 } **catch** (FileNotFoundException e) {  
 e.printStackTrace();  
 }

//**checkbox**

**private void** checkButtonClick() {  
 **lstvw**.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(AdapterView<?> parent, View view, **int** position, **long** id) {  
 Task delTask = **arrayListTask**.get(position);  
 **if** (delTask.isChk() == **true**) {  
 **arrayListTask**.remove(delTask);  
 **taskvwadapter**.notifyDataSetChanged();  
 }  
 }  
 });  
  
 }

//**Restoreinstance**  
  
 @Override  
 **protected void** onRestoreInstanceState(Bundle savedInstanceState) {  
  
 Log.*d*(**"restore"**,**"inside restore"**);  
  
 **if**(savedInstanceState == **null** || !savedInstanceState.containsKey(**"tasklist"**)){  
  
 **arrayListTask**=**new** ArrayList<>();  
 }  
 **else** {  
 **arrayListTask** = savedInstanceState.getParcelableArrayList(**"tasklist"**);  
  
 }  
 **taskvwadapter**=**new** CustomTaskAdapter(**this**,**arrayListTask**);  
 **lstvw**.setAdapter(**taskvwadapter**);  
 **super**.onRestoreInstanceState(savedInstanceState);  
 }

//**saveinstance**  
  
 @Override  
 **public void** onSaveInstanceState(Bundle outState) {  
 outState.putParcelableArrayList(**"tasklist"**,**arrayListTask**);  
 **super**.onSaveInstanceState(outState);  
 Log.*d*(**"save"**,**"inside save"**);  
  
  
 }  
  
}

//**Listview**

arrayListTask=**new** ArrayList<>();  
**taskvwadapter**=**new** CustomTaskAdapter(**this**,arrayListTask);  
**lstvw**.setAdapter(**taskvwadapter**);

//**Listview after update**

**public void** onItemClick(AdapterView<?> parent, View view, **int** position, **long** id) {  
 Task delTask = **arrayListTask**.get(position);  
 **if** (delTask.isChk() == **true**) {  
 **arrayListTask**.remove(delTask);  
 **taskvwadapter**.notifyDataSetChanged();  
 }  
}

EditText **etTitle**,**etDesc**;  
Button **btnBdd**;  
ListView **lstvw**;  
ArrayList<Task> **arrayListTask**;  
ArrayAdapter<Task> **taskvwadapter**;  
String **titletobeadded**,**desctobeadded**;

etTitle=(EditText) findViewById(R.id.***titl***);  
**etDesc**=(EditText) findViewById(R.id.***desc***);  
**lstvw**=(ListView) findViewById(R.id.***lstvwtask***);  
**btnBdd**=(Button) findViewById(R.id.***addbtn***);

**public class** MainActivity **extends** AppCompatActivity {  
 ListView **lstvwid**;  
 **private** String[] **names** = {**"a"**, **"b"**, **"c"**, **"d"**, **"e"**};  
 ArrayAdapter<String> **arrAdapter**;  
 ArrayList<String> **arrNames**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 **lstvwid** = (ListView) findViewById(R.id.***lstvw***);  
 *// arrNames = new ArrayList<>();* **arrAdapter** = **new** ArrayAdapter<String>(**this**, android.R.layout.***simple\_list\_item\_1***, **new** ArrayList<String>());  
 **lstvwid**.setAdapter(**arrAdapter**);

**new** myclass().execute();  
 }

//**//Fragment(land)Activity(Port) in java in main activity**

public void profileClick(View view) {

String name = view.getTag().toString();

Intent intent = new Intent(this,ProfileActivity.class);

intent.putExtra("name",name);

if(getResources().getConfiguration().orientation == Configuration.ORIENTATION\_PORTRAIT) {

startActivity(intent);

}

DetailsFragment frag = (DetailsFragment)this.getFragmentManager().findFragmentById(R.id.detailsFragment);

frag.updateFragment(name);

}

**//Getintent on second activity**

Intent intent = getIntent();

String extra = intent.getExtra("name");

**//wait for result on main**

public class FirstActivity extends Activity {

private static final int **REQ\_CODE** = 123; // MUST be 0-65535

public void myOnClick(View view) {

Intent intent = getIntent(this, SecondActivity.class);

startActivityForResult(intent, REQ\_CODE);

}

protected void onActivityResult(int requestCode,

int resultCode, Intent intent) {

super.onActivityResult(requestCode, resultCode, intent);

if (requestCode == REQ\_CODE) {

// came back from SecondActivity

String data = intent.getStringExtra("name");

Toast.makeText(this, "Got back: " + data,

Toast.LENGTH\_SHORT).show();

}

}

}

**//pass data to main back**

Intent intent = new Intent();

intent.putExtra("name", value);

setResult(RESULT\_OK, intent);

finish(); // calls onDestroy

**//Fragment in java**

**if**(getResources().getConfiguration().**orientation**== Configuration.***ORIENTATION\_PORTRAIT***)  
{  
  
 android.support.v4.app.Fragment fragmentById = getSupportFragmentManager().findFragmentById(R.id.***fragment\_container***);  
  
 **if** (fragmentById != **null**) {  
 **if** (fragmentById.isAdded()) {  
  
 getSupportFragmentManager().beginTransaction()  
 .remove(fragmentById).commit();  
  
 }  
 }  
 getSupportFragmentManager().beginTransaction().addToBackStack(**null**)  
 .add(R.id.***fragment\_container***, **locFrag**).commit();  
}

**//Fragmnt in landscape xml**

<**fragment  
 android:layout\_weight="1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/localFrag"  
 android:name="com.mobileappclass.assignment3.localFragment"  
 tools:layout="@layout/fragment\_local"** />  
<**fragment  
 android:layout\_weight="1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/serverFrag"  
 android:name="com.mobileappclass.assignment3.serverFragment"  
 tools:layout="@layout/fragment\_server"  
 tools:ignore="DuplicateIds"** />

// **Fragmnt in portrait xml**

<**FrameLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/fragment\_container"  
 android:layout\_marginTop="10dp"**></**FrameLayout**>

**//Strat a service**

Intent intent = new Intent(this,DownloadService.class);

intent.putExtra("link",link);

startService(intent);

**// Service using thread**

public class DownloadService extends Service {

@Override

public int onStartCommand(Intent intent, int flags, int startId) {

final String link = intent.getStringExtra("link");

Toast.makeText(DownloadService.this, "Test", Toast.LENGTH\_SHORT).show();

Thread thread = new Thread(new Runnable() {

@Override

public void run() {

DownloadHelper.downloadFile(link,0);

Intent broadcast = new Intent();

broadcast.setAction("FINISHED");

sendBroadcast(broadcast);

}

});

Toast.makeText(DownloadService.this, "Test 1 2 3", Toast.LENGTH\_SHORT).show();

thread.start();

return super.onStartCommand(intent, flags, startId);

}

@Override

public IBinder onBind(Intent intent) {

return null;

}

}

**//inside myservice extends service**

**public int** onStartCommand(**final** Intent intent, **int** flags, **int** startId) {  
  
  
 **final** Handler mHandler = **new** Handler();  
 **mRunnable**=**new** Runnable() {  
 @Override  
 **public void** run () {  
  
  
  
 **helper**.addEntry(**mLastLocation**.getLatitude(), **mLastLocation**.getLongitude());  
  
  
 *// ArrayList<String> words = helper.getEntireColumn();* Intent brod = **new** Intent();  
 brod.putExtra(**"result"**, **mLastLocation**.getLatitude() + **" "** + **mLastLocation**.getLongitude());  
 brod.setAction(localFragment.***broadcastString***);  
  
 *//intent.setAction("finish");* sendBroadcast(brod);  
 Toast.*makeText*(getApplicationContext(), **"Brod"** + **mLastLocation**.getLatitude() + **" "** + **mLastLocation**.getLongitude(), Toast.***LENGTH\_LONG***).show();  
  
 *// If desired, stop the service  
 //stopSelf();* Log.*d*(**"Service"**, **mLastLocation**.getLatitude() + **" "** + **mLastLocation**.getLongitude());  
  
  
  
 mHandler.postDelayed(**mRunnable**, 10 \* 1000);  
 }  
 };  
  
 mHandler.postDelayed(**mRunnable**,10\*1000);  
  
 **return** Service.***START\_STICKY***;  
  
  
 *//return super.onStartCommand(intent, flags, startId);*}

**//Broadcast receiver in fragment**

**private** BroadcastReceiver **testReceiver** = **new** BroadcastReceiver() {  
 @Override  
 **public void** onReceive(Context context, Intent intent) {  
  
 populateList();  
  
 String result = intent.getStringExtra(**"result"**);  
 **words**.add(result);  
 **adapter**.notifyDataSetChanged();  
 **for** (String i : **words**)  
 {  
  
 *//adapter.notifyDataSetChanged();  
 // Toast.makeText(getApplicationContext(),"inside add database:2010-05-28T15:36:56.200" , Toast.LENGTH\_LONG).show();* }  
  
  
 Log.*d*(**"receiver"**, result);  
 Toast.*makeText*(getActivity(), **"inside receiver"**, Toast.***LENGTH\_SHORT***).show();  
 *// Toast.makeText(MainActivity.this, result, Toast.LENGTH\_SHORT).show();* Intent stopIntent = **new** Intent(getActivity(), GPStracker.**class**);  
 getActivity().stopService(stopIntent);\*/  
  
 }  
};

**//fragment with receiver**

@Override  
**public void** onPause() {  
 LocalBroadcastManager.*getInstance*(getActivity()).unregisterReceiver(**testReceiver**);  
  
 **super**.onPause();  
}  
  
@Override  
**public void** onResume() {  
 getActivity().registerReceiver(**testReceiver**, **filter**);  
  
 **super**.onResume();  
}  
  
@Override  
**public void** onDestroy() {  
 **if**(**testReceiver** != **null**){  
 getActivity().unregisterReceiver(**testReceiver**);  
 }  
 **super**.onDestroy();  
}  
  
@Override  
**public void** onActivityCreated(@Nullable Bundle savedInstanceState) {  
 **filter** = **new** IntentFilter();  
 **filter**.addAction(***broadcastString***);  
  
  
  
 populateList();  
  
 **super**.onActivityCreated(savedInstanceState);  
}

**//Asynch task**

**private class** myclass **extends** AsyncTask<String ,String,String>{  
 ArrayAdapter<String> **asynchAdapter**;  
 ProgressBar **pgbar**;  
 **int cnt**;  
  
 **public** myclass() {  
 **super**();  
 }  
  
 @Override  
 **protected void** onPreExecute() {  
 **asynchAdapter**= (ArrayAdapter<String>) **lstvwid**.getAdapter();  
  
 **pgbar**=(ProgressBar) findViewById(R.id.***pbar***);  
 **pgbar**.setMax(5);*//since max is 5 string in array* **pgbar**.setProgress(0);*//current is on 0* **pgbar**.setVisibility(View.***VISIBLE***);  
  
  
 **super**.onPreExecute();  
 }  
 @Override  
 **protected** String doInBackground(String ... params) {  
 **for**(String str:**names**)  
 {  
 publishProgress(str);  
 Log.*d*(**"in bg"**,str);  
 **try** {  
 Thread.*sleep*(1000);  
 } **catch** (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 **return "Done"**;  
 }  
 @Override  
 **protected void** onProgressUpdate(String... values) {  
 **asynchAdapter**.add(values[0]);  
 **cnt**++;  
 **pgbar**.setProgress(**cnt**);  
 *// Toast.makeText(getApplicationContext(),,Toast.LENGTH\_SHORT).show();* }  
  
  
 @Override  
 **protected void** onPostExecute(String res) {  
 Toast.*makeText*(getApplicationContext(),res,Toast.***LENGTH\_SHORT***).show();  
 **pgbar**.setVisibility(View.***GONE***);*//pgbar gone* }  
  
  
 }  
  
  
  
  
}

**//sql data listview**

**public void** populateList()  
 {  
 SqlHelper helper = **new** SqlHelper(getActivity());  
  
 *// This is the only time we will use SQLiteDatabase in an Activity  
// SQLiteDatabase db = openOrCreateDatabase("Locationdb", null);  
  
 // helper.onCreate(db);  
 // helper.addEntry(44.67,43.97);  
 // helper.addEntry(32.67,64.97);* **if**(helper.getEntireColumn()!=**null**){  
 **words** = helper.getEntireColumn();  
 **adapter** = **new** ArrayAdapter(getActivity(), android.R.layout.***simple\_list\_item\_1***,**words**);  
 **locationList** = (ListView) getActivity().findViewById(R.id.***LocLIST***);  
 **locationList**.setAdapter(**adapter**);  
 }  
 *// ArrayList<String> words = helper.getEntireColumn();  
 /\* for (String i : words)  
 {  
 // Toast.makeText(getApplicationContext(),"inside add database:2010-05-28T15:36:56.200" , Toast.LENGTH\_LONG).show();  
 Log.d("Activity", i);  
 } \*/* }

//

SharedPreferences pref = getApplicationContext().getSharedPreferences("MyPref", 0); // 0 - for private mode

Editor editor = pref.edit();

editor.putBoolean("key\_name", true); // Storing boolean - true/false

editor.putString("key\_name", "string value"); // Storing string

editor.putInt("key\_name", "int value"); // Storing integer

editor.putFloat("key\_name", "float value"); // Storing float

editor.putLong("key\_name", "long value"); // Storing long

editor.commit(); // commit changes

pref.getString("key\_name", null); // getting String

pref.getInt("key\_name", null); // getting Integer

pref.getFloat("key\_name", null); // getting Float

pref.getLong("key\_name", null); // getting Long

pref.getBoolean("key\_name", null); // getting boolean