

ANDROID PROGRAMMING

LESSON 10

ANDROID THREAD

Agenda

- Giới thiệu Thread
 - Thread là gì?
- Một số kiểu Thread trong Android
 - Xử lý Thread với Handler (01)
 - Tạo Progress Bar với Handler (02)
 - Tạo Progress Bar với AsyncTask (03)
- Clip ảnh với Handler (postDelayed) (04)
- Viết ứng dụng đọc báo (RSS) sử dụng AsyncTask (05)

Thread là gì?

- Khi một ứng dụng Android được khởi chạy, thì hệ thống Android sẽ start một New Linux Process cho ứng dụng đó, gọi là UI Thread.
- Thread là một thực thể của Process, mỗi process có thể có thể có nhiều luồng.
- Đơn giản :

Process = Program + State of all excuting Threads

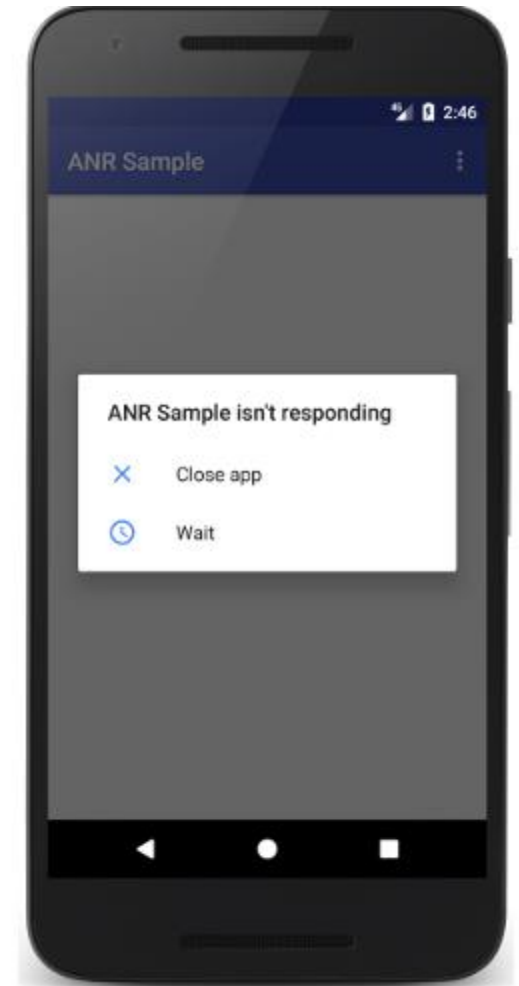
Một số kiểu Thread trong Android

- UI Thread (Main Thread) : xử lý tất cả các vấn đề liên quan đến giao diện, các widget của Android, được tạo ra ngay khi khởi chạy ứng dụng
- Worker Thread : Thread do người dùng tạo thêm để xử lý các tác vụ riêng (Background Thread)

Hiện tượng ANR(Application Not Responding) trong Android

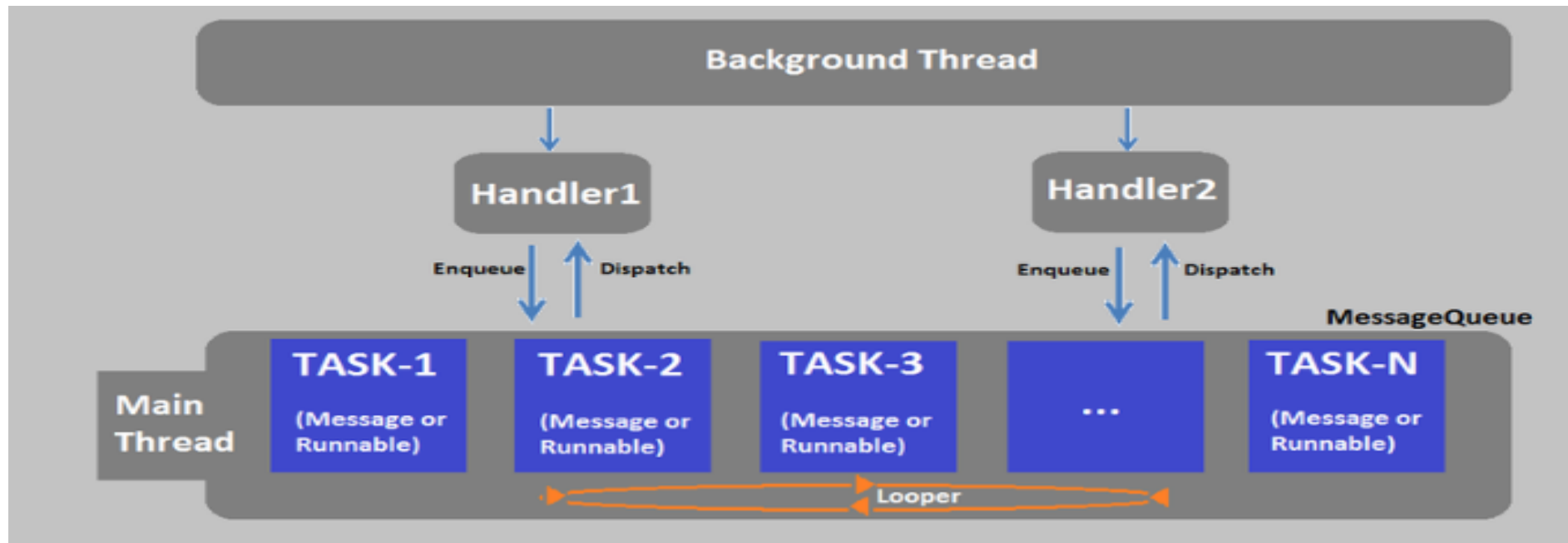
Khi có nhiều xử lý trên UI Thread, và có một xử lý phải thực hiện lâu. Ví dụ kết nối mạng hay truy vấn cơ sở dữ liệu, khi đó UI sẽ bị block. Người dùng thấy ứng dụng đang bị treo, nhưng thực ra nó đang thực thi xử lý trên UI Thread. Nếu UI bị block hơn vài giây (trung bình là 5 giây) thì hệ thống Android sẽ xuất hiện hộp thoại như bên. Để không xảy ra hiện tượng này thì Android là đề ra 2 nguyên tắc trong xử lý bất đồng bộ của Android:

1. Không được block UI Thread
2. Không được kết nối tới bộ công cụ Android UI (Android UI toolkit) từ một Thread không phải là UI Thread.

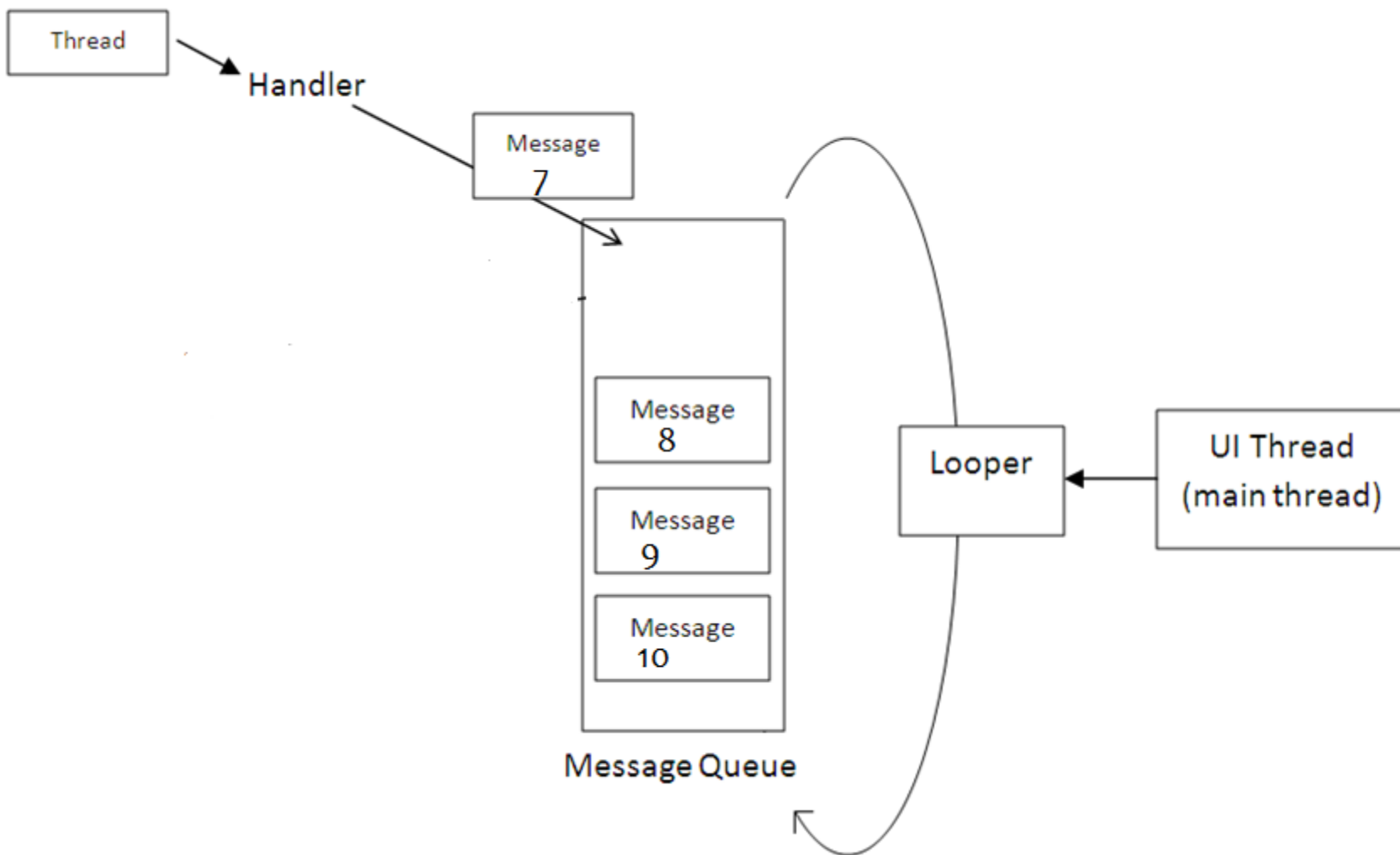


Xử lý Thread với Handler

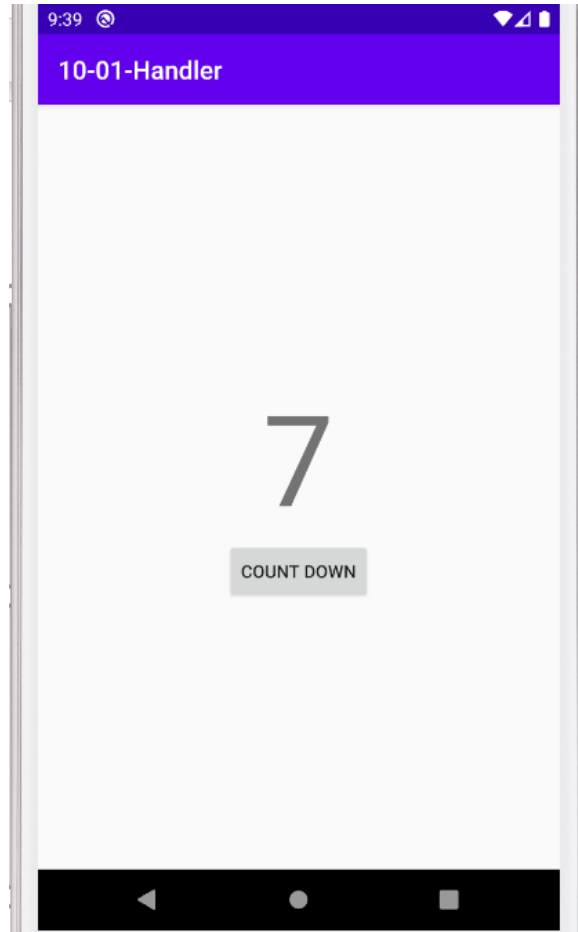
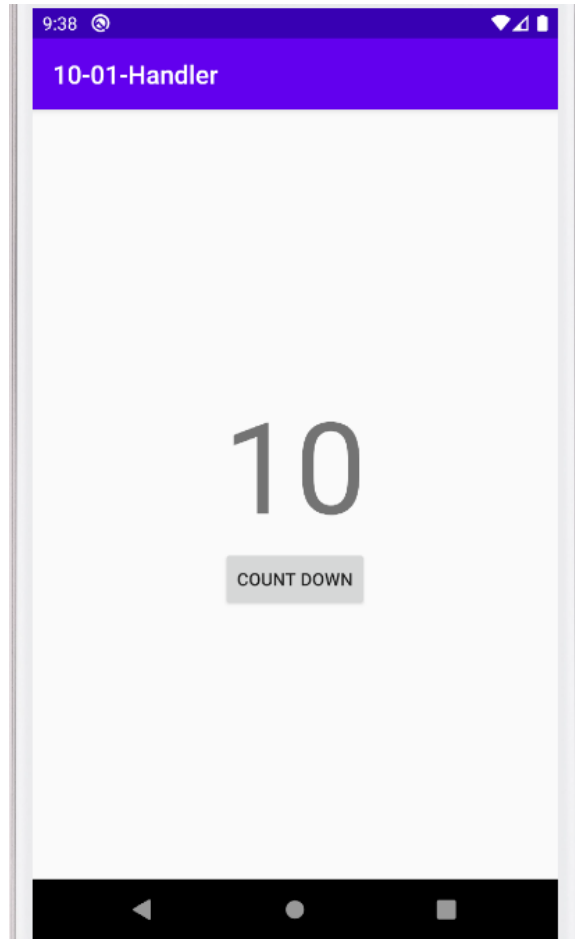
- Các Thread tương tác và trao đổi với nhau qua Handler.
- Handler có nhiệm vụ gửi và thực thi các Message hoặc Runnable tới Message Queue của Thread sinh ra nó (Handler).
- Khi dùng Handler để giao tiếp giữa Worker Thread với UI thì có một số bất tiện là phải code nhiều, dài dòng



Ví dụ bài toán COUNTDOWN (01)



Dùng Thread với Handler



File Layout

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center">
    <TextView
        android:id="@+id/tvTimer"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="100sp"
        android:text="10"/>
    <Button
        android:id="@+id/btCount"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="COUNT DOWN"/>
</LinearLayout>
```

Đối tượng handler

```
private Handler handler;
handler=new Handler(){
    @Override
    public void handleMessage(@NonNull Message msg) {
        switch (msg.what){
            case MESSAGE_COUNT_DOWN:
                tvTimer.setText(String.valueOf(msg.arg1));
                break;
            case 789:
                Toast.makeText(getApplicationContext(),
                    "finish", Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(MainActivity.this,
                    NewYearActivity.class);
                startActivity(intent);
        }
    }
};
```

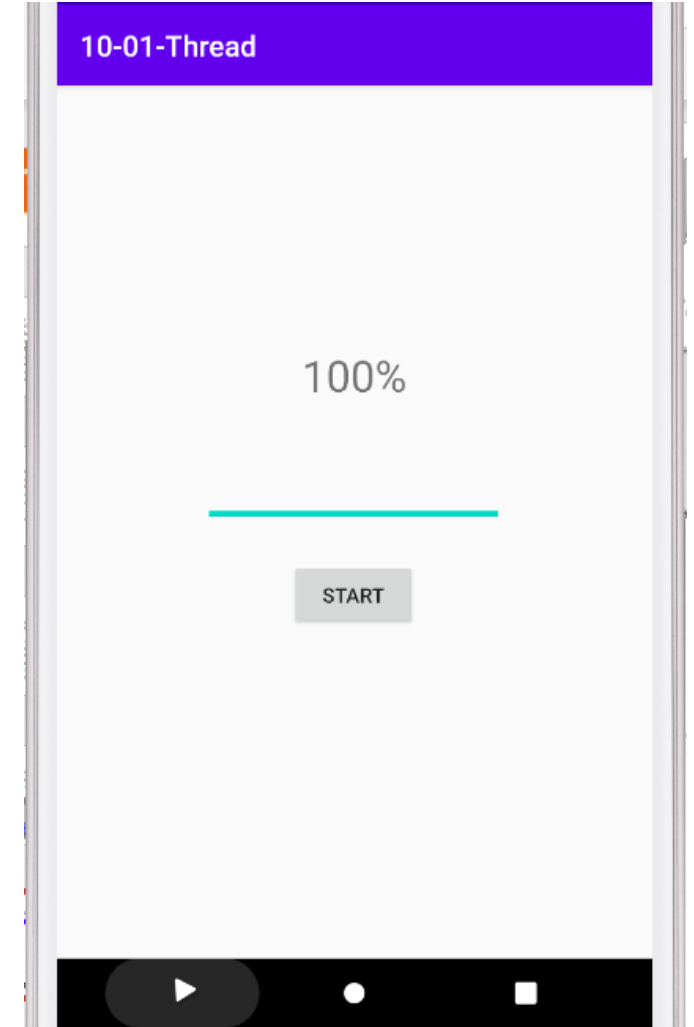
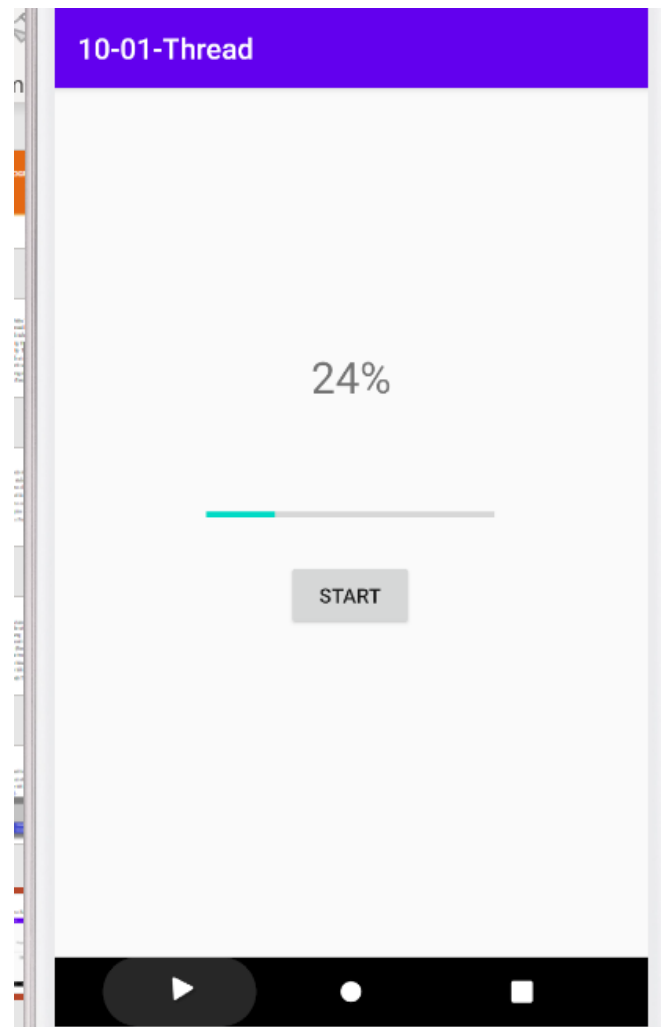
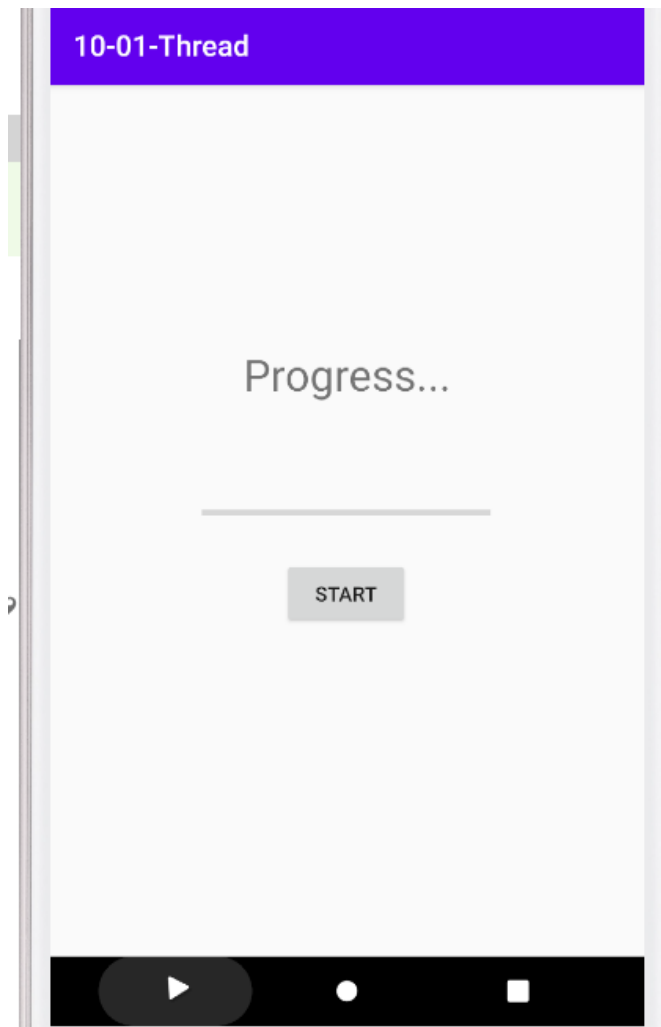
Lớp Thread (chạy ngầm)

```
class Countdown extends Thread{
    @Override
    public void run() {
        int time = 10;
        while(time>0){
            time --;
            Message msg = new Message();
            msg.what = MESSAGE_COUNT_DOWN;
            msg.arg1 = time;
            handler.sendMessage(msg);
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        handler.sendMessage(789);
    }
}
```

MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
    public static final int MESSAGE_COUNT_DOWN=1001;  
    private TextView tvTimer;  
    private Button btCount;  
    private Handler handler;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        initView();  
        btCount.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                new Countdown().start();  
            }  
        });  
    }  
}
```

Tạo ProgressBar với Handler (02)



File Layout

```
<TextView
    android:id="@+id/tv"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="180dp"
    android:textSize="30dp"
    android:text="Progress..."
/>
<ProgressBar
    android:id="@+id/progressBar"
    style="?android:progressBarStyleHorizontal"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="68dp"
/>
<Button
    android:id="@+id/btStart"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Start"/>
```

Lớp Handler

```
private TextView tv;  
private ProgressBar progressBar;  
private Handler handler;  
handler = new Handler() {  
    @Override  
    public void handleMessage(@NonNull Message msg) {  
        super.handleMessage(msg);  
        progressBar.setProgress(msg.arg1);  
        tv.setText(msg.arg1 + "%");  
    }  
};
```

Lớp Thread

```
private void doStart() {
    if (thread != null) thread.interrupt();
    progressBar.setProgress(0);
    thread=new ProgressThread();
    thread.start();
}
class ProgressThread extends Thread{
    public void run(){
        for (int i=1;i<=100;i++){
            try {
                Thread.sleep(100);
            }catch(InterruptedException e){
                System.out.println(e);
            }
            Message msg = handler.obtainMessage();
            msg.arg1 = i;
            handler.sendMessage(msg);
        }
    }
}
```


MainActivity.java

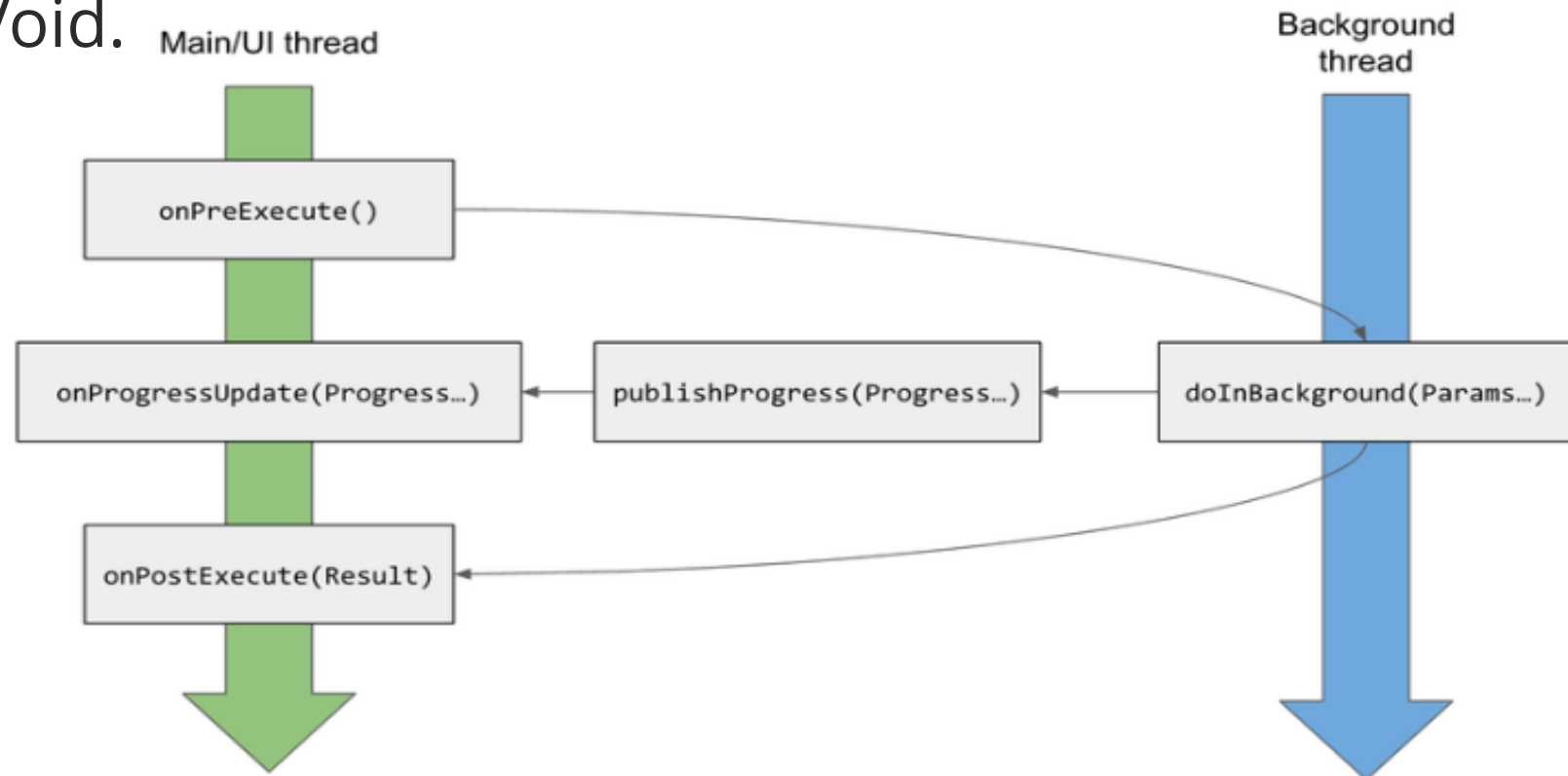
```
public class MainActivity extends AppCompatActivity {
    private TextView tv;
    private ProgressBar progressBar;
    private Button btStart;
    private Handler handler;
    private ProgressThread thread;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        initView();
    }
    private void initView() {
        tv = findViewById(R.id.tv);
        progressBar = findViewById(R.id.progressBar);
        btStart = findViewById(R.id.btStart);
        btStart.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                doStart();
            }
        });
    }
}
```

Xử lý Thread với AsyncTask

- AsyncTask là một đối tượng thích hợp dùng để giao tiếp giữa một Thread với UI Thread.
- Nó cho phép chúng ta thực hiện công việc ở background và sau đó cập nhật giao diện ở UI Thread. AsyncTask chính là sự tổ hợp của một Thread và một Handler.

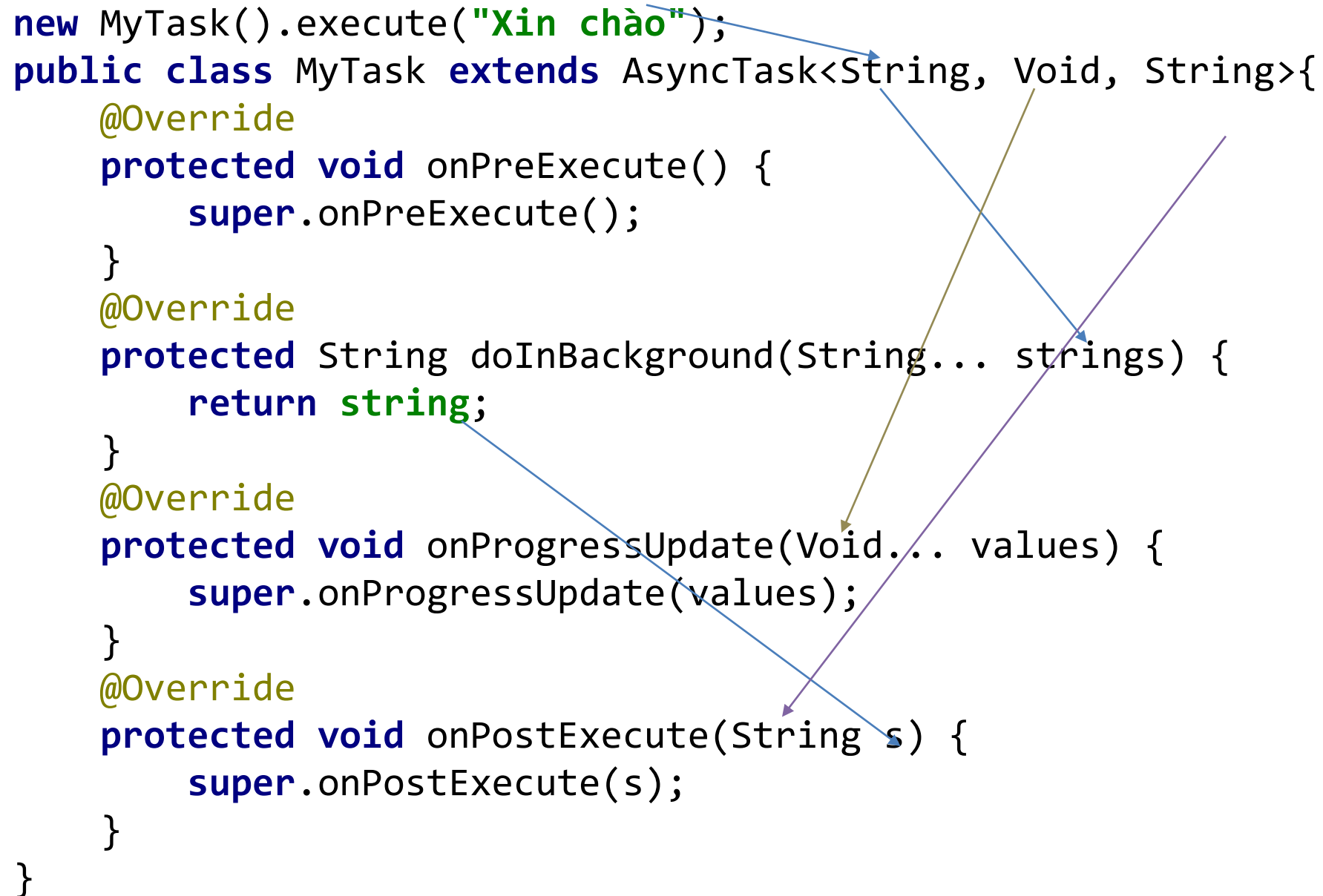
Xử lý Thread với AsyncTask

AsyncTask là một đối tượng có 3 tham số Generic Type truyền vào là *Params*, *Progress*, *Result* và 4 phương thức *onPreExecute()*, *doInBackground()*, *onProgressUpdate()* và *onPostExecute()*. 3 tham số đều phải là kiểu Object, nếu không có thì để là Void.



Lớp AsyncTask

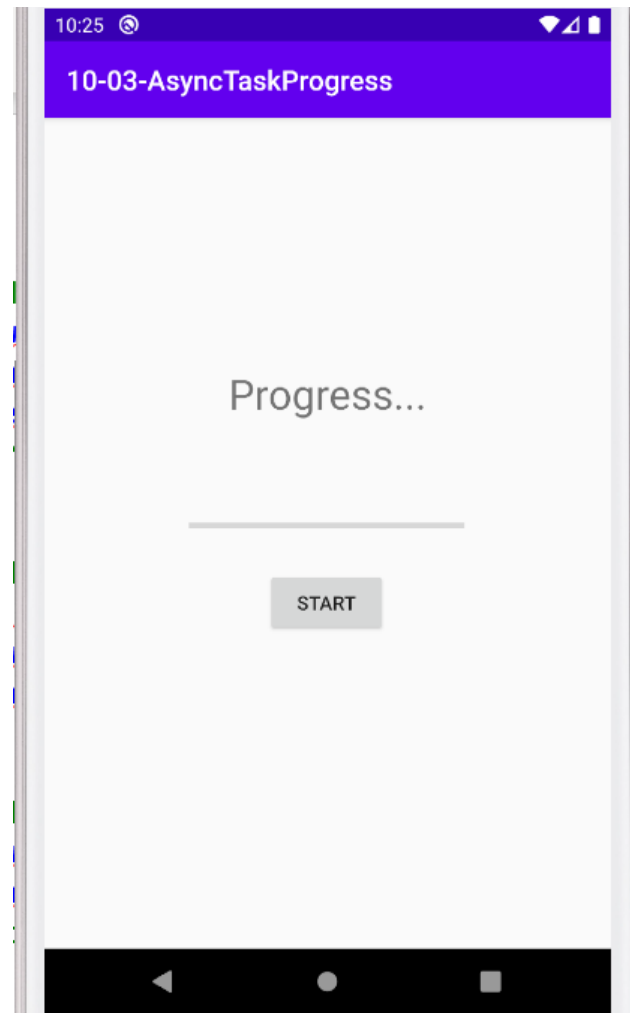
```
new MyTask().execute("Xin chào");  
public class MyTask extends AsyncTask<String, Void, String>{  
    @Override  
    protected void onPreExecute() {  
        super.onPreExecute();  
    }  
    @Override  
    protected String doInBackground(String... strings) {  
        return string;  
    }  
    @Override  
    protected void onProgressUpdate(Void... values) {  
        super.onProgressUpdate(values);  
    }  
    @Override  
    protected void onPostExecute(String s) {  
        super.onPostExecute(s);  
    }  
}
```



Tạo Progress Bar với AsyncTask (03)

```
public class MainActivity extends AppCompatActivity {
    Button btStart;
    MyTask myTask;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btStart = (Button) findViewById(R.id.btStart);
        btStart.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                myTask = new MyTask(MainActivity.this);
                myTask.execute();
                btStart.setEnabled(false);
            }
        });
    }
}
```

Layout file



Lớp AsyncTask (TaskProgress)

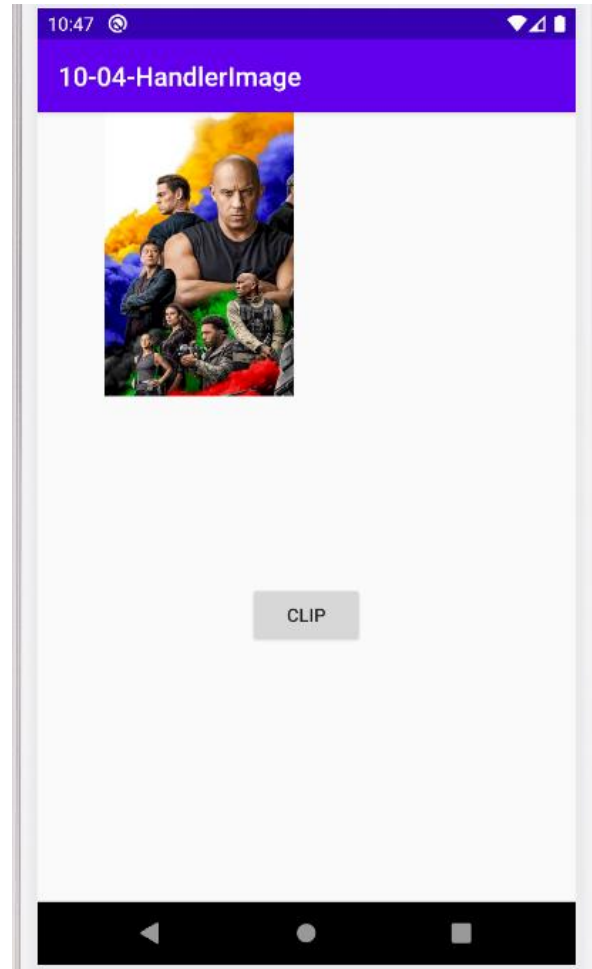
```
public class MyTask extends AsyncTask<Void,Integer,Void> {
    private Activity activity;
    public MyTask(Activity activity) {
        this.activity = activity;
    }
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        Toast.makeText(activity, "Bat dau tien trinh", Toast.LENGTH_SHORT).show();
    }
    @Override
    protected Void doInBackground(Void... paras) {
        //hàm này được chạy ngầm và thực hiện liên tiếp sau onPreExecute
        for (int i = 1; i <= 100 ; i++) {
            SystemClock.sleep(100);
            publishProgress(i);
        }
        return null;
    }
}
```

```
protected void onProgressUpdate(Integer... values) {
    super.onProgressUpdate(values);
    ProgressBar progress=activity.findViewById(R.id.progressBar);
    //vì truyền có 1 số nên mảng chỉ có 1 phần tử
    int num=values[0];
    progress.setProgress(num);
    TextView tv=activity.findViewById(R.id.tv);
    tv.setText(num+ "%");
}
@Override
protected void onPostExecute(Void unused) {
    super.onPostExecute(unused);
    Toast.makeText(activity, "Ket thuc tien trinh",
    Toast.LENGTH_SHORT).show();
    Button bt=activity.findViewById(R.id.btStart);
    bt.setEnabled(true);
}
}
```


Phương thức sau khi trì hoãn

```
final Handler handler = new Handler();  
handler.postDelayed(new Runnable() {  
    @Override  
    public void run() {  
        //Do something after 100ms  
    }  
}, 100);
```

Clip ảnh với Handler (04)



Viết ứng dụng đọc báo (RSS) sử dụng AsyncTask (05)

```
<item>
  <title>Con trai Ancelotti chỉ đạo Real cuối tuần này</title>
  <description>
    <![CDATA[ <a href="https://vnexpress.net/con-trai-ancelotti-chi-dao-real-cuoi-tuan-nay-4446025.html" >Con trai Ancelotti chỉ đạo Real cuối tuần này</a>
    1648772877-8443-1648772935.jpg?w=1200&h=0&q=100&dpr=1&fit=crop&s=N-8Mj0G2xrUC3CrtqWlEtA" ></a>
    sẽ nắm quyền chỉ đạo khi Real làm khách của Celta Vigo ở vòng 30 La Liga. ]]>
  </description>
  <pubDate>Fri, 01 Apr 2022 07:30:23 +0700</pubDate>
  <link>https://vnexpress.net/con-trai-ancelotti-chi-dao-real-cuoi-tuan-nay-4446025.html</link>
  <guid>https://vnexpress.net/con-trai-ancelotti-chi-dao-real-cuoi-tuan-nay-4446025.html</guid>
  <slash:comments>0</slash:comments>
</item>
```

Giao diện project

Demo8



Nét trong trẻo của sao nhí Bảo Ngọc Fri, 01 Apr 2022 00:00:00 +0700

[</br>Ở tuổi 14, ca sĩ Bảo Ngọc gây chú ý khán giả với nét trong trẻo và phong cách năng động.](https://vnexpress.net/net-trong-treo-cua-sao-nhi-bao-ngoc-4445672.html)

Hoa hậu Tiểu Vy đóng phim kinh dị Fri, 01 Apr 2022 08:13:50 +0700

[</br>Hoa hậu Trần Tiểu Vy lần đầu lần sân điện ảnh khi đóng chính phim kinh dị "Tử mẫu Thiên linh cái".](https://vnexpress.net/hoa-hau-tieu-vy-dong-phim-kinh-di-4445998.html)

Những phim Hàn lên sóng tháng 4 Fri, 01 Apr 2022 05:00:00 +0700

[</br>Kim Hee Sun đóng thần chết trong "Tomorrow", Kim Woo Bin trở lại màn ảnh sau chữa ung thư với phim "Our Blues".](https://vnexpress.net/nhung-phim-han-len-song-thang-4-4444162.html)

Son Ye Jin: 'Tôi chưa trưởng thành như nhân vật' Thu, 31 Mar 2022 23:21:52 +0700

Demo8



Mới nhất

Xem nhiều

Giải trí > Thời trang > Sao đẹp - Sao xấu

Thứ sáu, 1/4/2022, 00:00 (GMT+7)

Nét trong trẻo của sao nhí Bảo Ngọc

Ở tuổi 14, ca sĩ Bảo Ngọc gây chú ý khán giả với nét trong trẻo và phong cách năng động.



```
public class MyAsynsTask extends AsyncTask<Void,Void,Void> {  
    private final Context context;  
    private final String link;  
    private List<Item> list;  
    private RecyclerView recycleView;  
    RecyclerViewAdapter adapter;  
    public MyAsynsTask(Context context, String link,RecyclerView  
recycleView) {  
        this.context = context;  
        this.link = link;  
        this.recycleView=recycleView;  
    }  
}
```

```
protected Void doInBackground(Void... voids) {  
    try {  
        URL url = new URL(link);  
        URLConnection connection=url.openConnection();  
        InputStream is=connection.getInputStream();  
        list=(List<Item>)MySaxParser.xmlParser(is);  
    }catch(Exception e){  
        Log.d("Loi:",e.getMessage());  
    }  
    return null;  
}  
  
@Override  
protected void onPostExecute(Void unused) {  
    super.onPostExecute(unused);  
    adapter=new RecyclerViewAdapter(context,list);  
    LinearLayoutManager manager=new LinearLayoutManager(context,  
        RecyclerView.VERTICAL,false);  
    recycleView.setLayoutManager(manager);  
    recycleView.setAdapter(adapter);  
}
```

```
public class MainActivity extends AppCompatActivity {  
    private RecyclerView recyclerView;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        recyclerView=findViewById(R.id.recycleView);  
        String link="https://vnexpress.net/rss/giai-tri.rss";  
        MyAsyncTask asyncTask=new  
        MyAsyncTask(MainActivity.this,link,recyclerView);  
        asyncTask.execute();  
    }  
}
```

Thank you!