# Thread – Luồng, Tiểu trình

## Thread: Luồng, MultiThread: đa luồng

* Luồng: Tiến trình thực hiện tại 1 thời điểm
* Đa luồng: Thực hiện nhiều tiến trình (tác vụ, công việc) 1 thời điểm
* Android chạy ứng dụng trong 1 thread riêng
* Các loại Thread trong Java
  + Thread
  + AsyncTask

## Thread

Thread thread = new Thread(new Runnable() {

@Override

public void run() {

for (int i = 1; i <= numberButton && isRuning.get(); i++) {

// dừng luồng (milisecond)

SystemClock.sleep(100);

// Xử lý tiến trình, không xử lý được android View

}

}

});

thread.start();

Xử lý Android View trong Thread:

new Thread(new Runnable() {

public void run() {

final Bitmap bitmap = loadImageFromNetwork("http://example.com/image.png");

// Xử lý Android View trong Thread

mImageView.post(new Runnable() {

public void run() {

mImageView.setImageBitmap(bitmap);

}

});

}

}).start();

## AsyncTask (Asynchronous Task)

protected Void doInBackground(Void... params) {

// thực hiện tiến trình

total = number1 + number2;

publishProgress(i); // gọi hàm cập nhật giao diện, trả về onProgressUpdate

}

/\* Cập nhật giao diện,

\* @params Iterger thứ 2 (AsyncTask<Void, Integer, Void>)

\* "..." giống params trong C# (truyền bao nhiêu biến vào cũng được)

\*/

@Override

protected void onProgressUpdate(Integer... values) {

super.onProgressUpdate(values);

// cập nhật trạng thái khi đang thực hiện

int intValue = values[0];

}

/\* Kết thúc, lấy kết quả

\* @params: biến void thứ 3 (AsyncTask<Void, Integer, Void>)

\*/

@Override

protected void onPostExecute(Void result) {

super.onPostExecute(result);

// Trả về khi thực hiện xong

}

}

// Chạy tác vụ

protected void doProgress() {

MyAsyncTask myTask = new MyAsyncTask(this);

myTask.execute(); // kích hoạt tiến trình

}

**Chú giải:**

* **AsyncTask<Params, Progress, Result>**  có 3 đối số là các Generic Type:
  + **Params:** Là giá trị ((biến) được truyền vào khi gọi thực thi tiến trình và nó sẽ  được truyền vào **doInBackground**
  + **Progress**: Là  giá trị (biến) dùng để update giao diện diện lúc tiến trình thực thi, biến này sẽ được truyền vào hàm **onProgressUpdate**.
  + **Result:** Là biến dùng để lưu trữ kết quả trả về sau khi tiến trình thực hiện xong.
  + Những đối số nào không sử dụng trong quá trình thực thi tiến trình thì ta thay bằng **Void**.
* **Thông thường trong 1 AsyncTask sẽ chứa 4 hàm**
  + **onPreExecute() :**Tự động được gọi đầu tiên khi tiến trình được kích hoạt.
  + **doInBackground()**: Được thực thi trong quá trình tiến trình chạy nền, thông qua hàm này để ta gọi hàm onProgressUpdate để cập nhật giao diện (gọi lệnh **publishProgress**). Ta không thể cập nhật giao diện trong hàm doInBackground().
  + **onProgressUpdate ():**Dùng để cập nhật giao diện lúc runtime
  + **onPostExecute()**: Sau khi tiến trình kết thúc thì hàm này sẽ tự động sảy ra. Ta có thể lấy được kết quả trả về sau khi thực hiện tiến trình kết thúc ở đây.
  + Trong 4 hàm trên thì hàm **doInBackground()**bắt buộc phải tồn tại, còn các hàm khác có thể khuyết, nhưng theo Tui các bạn nên sử dụng đầy đủ 4 hàm đã nêu.
  + Đối với AsyncTask thì ta cần tạo một lớp kế thừa từ AsyncTask, sau đó từ MainActivity ta gọi hàm execute() của tiến trình này là OK.

## Handler

* Handler giao tiếp giữa 2 hay nhiều thread
* Handler() xử lý 1 yêu cầu nào của người dùng sau 1 khoảng thời gian nhất định

// Nhận giá trị từ Thread con

Handler.Callback myCallback=new Handler.Callback(){

@Override

public boolean handleMessage(Message msg){

// lấy giá trị trả về, xử lý UI

String value = msg.obj.toString();

return false;

}

};

Handler handler=new Handler(myCallback);

// Chạy Thread con gởi giá trị về Hander

Thread thread = new Thread(new Runnable() {

@Override

public void run(){

/\*

Xử lý gởi giá trị

\*/

Message msg=handler.obtainMessage(); //lấy Message vào

msg.obj="Giá trị: abc"; // gài hàng, gởi giá trị

handler.sendMessage(msg); // gởi lại Hander để xử lý

}

});

thread.start();

**Tham khảo:**

* <http://devpro.edu.vn/thread-asynctask-handler/>
* <https://duythanhcse.wordpress.com/2013/10/25/bai-37-xu-ly-da-tien-trinh-bang-asynctask/>