

## Lab 1: THREAD



### MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Tìm hiểu Thread trong Android
- ✓ Sử dụng AsyncTask
- ✓ Tìm hiểu Class Handler
- ✓ Sử dụng ProgressBar

### BÀI 1 (2 ĐIỂM)

Viết chương trình cho phép tải và hiển thị hình ảnh từ phía server về. Sử dụng class Thread trong android để xử lý.

### HƯỚNG DẪN:

- ✓ Chạy Eclipse
- ✓ Tạo một dự án android
- ✓ Xây dựng giao diện cơ bản như bên dưới



- ✓ Viết mã cho MainActivity: Khai báo, ánh xạ các view vừa tạo

implements hàm OnClickListener và bắt sự kiện click cho button

```
public class MainActivity extends Activity implements OnClickListener {

    private Button btnLoad;
    private ImageView imgAndroid;
    private TextView tvMessage;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnLoad = (Button) findViewById(R.id.btnLoad);
        imgAndroid = (ImageView) findViewById(R.id.imgAndroid);
        tvMessage = (TextView) findViewById(R.id.tvMessage);
        btnLoad.setOnClickListener(this);
    }
}
```

✓ Tạo hàm load hình ảnh từ server

```
private Bitmap loadImageFromNetWork(String link) {
    URL url;
    Bitmap bmp = null;
    try {
        url = new URL(link);
        bmp = BitmapFactory.decodeStream(url.openConnection().getInputStream());
    } catch (IOException e) {
        e.printStackTrace();
    }

    return bmp;
}
```

✓ Xử lý sự kiện click button

```

@Override
public void onClick(View v) {
    final Thread myThread = new Thread(new Runnable() {

        @Override
        public void run() {
            // TODO Auto-generated method stub
            final Bitmap bitmap = loadImageFromNetWork("http://i64.tinypic.com/28vaq8k.png");
            imgAndroid.post(new Runnable() {

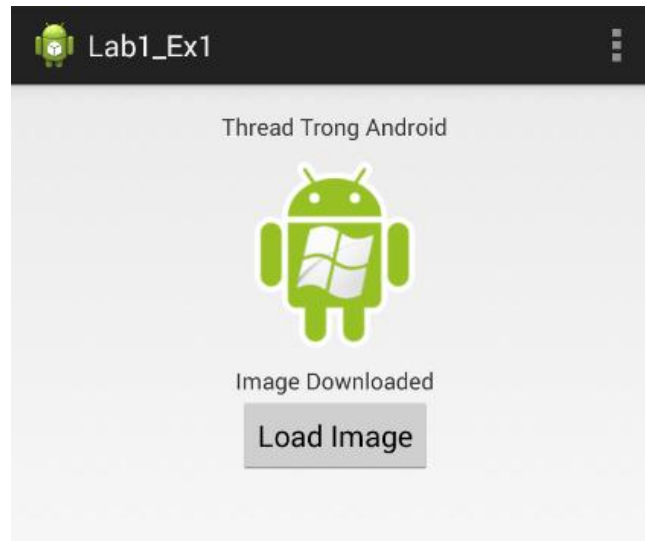
                @Override
                public void run() {
                    // TODO Auto-generated method stub
                    tvMessage.setText("Image Downloaded");
                    imgAndroid.setImageBitmap(bitmap);
                }
            });
        }
    });
    myThread.start();
}

```

- ✓ Thêm permission INTERNET cho ứng dụng

```
<uses-permission android:name="android.permission.INTERNET"/>
```

- ✓ Chạy ứng dụng và xem kết quả



## BÀI 2 (2 ĐIỂM)

Viết chương trình có hiển thị Splashscreen (Màn hình chào) trong android và load hình ảnh từ phía server về. Sử dụng class Handler

### HƯỚNG DẪN:

- ✓ Tạo mới activity đặt tên SplashScreen. Xây dựng giao diện cơ bản như bên dưới



- ✓ Chuyển qua file activity Khai báo thời gian chạy Splashscreen

```
public class SplashScreen extends Activity {  
  
    // Splash Screen Timer  
    private static int SPLASH_TIME_OUT = 3000;
```

- ✓ Khởi tạo lớp Handler và xử lý chuyển trang sau thời gian đã khai báo

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_splash_screen);
    new Handler().postDelayed(new Runnable() {

        @Override
        public void run() {

            Intent intent = new Intent(SplashScreen.this, MainActivity.class);
            startActivity(intent);

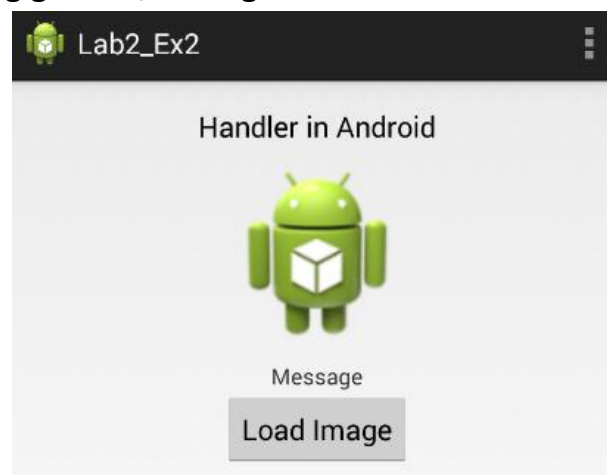
            finish();
        }
    }, SPLASH_TIME_OUT);
}
```

- ✓ Vào file AndroidManifest.xml sửa splashscreen thành activity launcher

```
<activity
    android:name=".SplashScreen"
    android:label="@string/title_activity_splash_screen" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

- ✓ Chạy ứng dụng và xem kết quả
- ✓ Tiếp tục xây dựng giao diện trang chính như sau



- ✓ Khai báo view và ánh xạ, khai báo đường dẫn hình (Có thể chọn hình khác), implements OnClickListener, bắt sự kiện click cho button

```
public class MainActivity extends Activity implements OnClickListener {

    private ImageView imgLoad;
    private Button btnLoad;
    private ProgressDialog progressDialog;
    private String url = "http://i64.tinypic.com/28vaq8k.png";
    private Bitmap bitmap = null;
    private TextView tvMessage;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        imgLoad = (ImageView) findViewById(R.id.imgLoad);
        btnLoad = (Button) findViewById(R.id.btnLoad);
        tvMessage = (TextView) findViewById(R.id.tvMessage);
        btnLoad.setOnClickListener(this);
    }
}
```

- ✓ Tạo Handler xử lý kết quả

```
private Handler messageHandler = new Handler() {
    public void handleMessage(Message msg) {
        super.handleMessage(msg);
        Bundle bundle = msg.getData();
        String message = bundle.getString("message");
        tvMessage.setText(message);
        imgLoad.setImageBitmap(bitmap);
        progressDialog.dismiss();
    }
};
```

- ✓ Xử lý sự kiện click của button. Show dialog, tạo biến runnable, xử lý nền việc tải hình, gửi tin nhắn kết quả trong hàm run. Cuối cùng tạo Thread mới và start Thread.

```

@Override
public void onClick(View v) {
    // TODO Auto-generated method stub
    progressDialog = ProgressDialog.show(MainActivity.this, "", "Downloading...");
    Runnable aRunnable = new Runnable() {

        @Override
        public void run() {
            // TODO Auto-generated method stub
            bitmap = downloadBitmap(url);
            Message msg = mHandler.obtainMessage();
            Bundle bundle = new Bundle();
            String threadMessage = "Image downloaded";
            bundle.putString("message", threadMessage);
            msg.setData(bundle);
            mHandler.sendMessage(msg);
        }
    };
    Thread aThread = new Thread(aRunnable);
    aThread.start();
}

```

- ✓ Tạo hàm downloadBitmap kết nối server cho việc tải hình

```

private Bitmap downloadBitmap(String link) {
    try {
        URL url = new URL(link);
        HttpURLConnection connection = (HttpURLConnection) url.openConnection();
        connection.connect();
        InputStream inputStream = connection.getInputStream();
        Bitmap bitmap = BitmapFactory.decodeStream(inputStream);
        return bitmap;
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

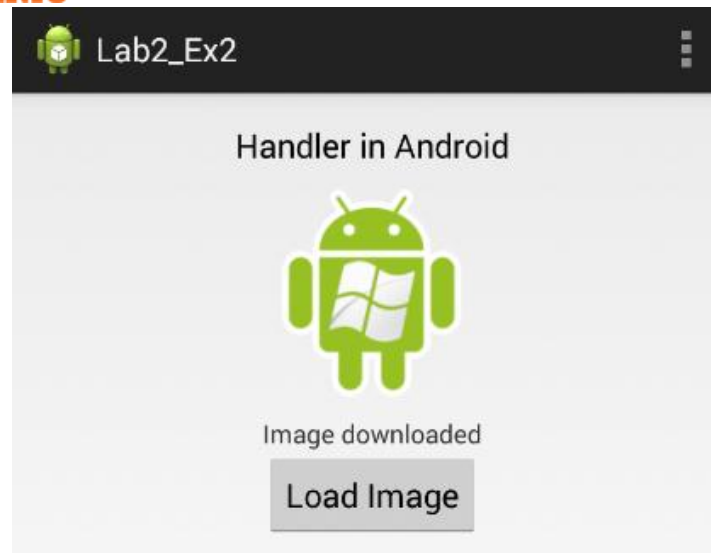
    return null;
}

```

- ✓ Thêm permission INTERNET cho ứng dụng

```
<uses-permission android:name="android.permission.INTERNET"/>
```

- ✓ Chạy ứng dụng và xem kết quả



### BÀI 3 (2 ĐIỂM)

Viết chương trình load hình ảnh từ server về. Sử dụng class AsyncTask trong android.

#### HƯỚNG DẪN:

- ✓ Thiết kế giao diện chính như hình sau



- ✓ Tạo một interface listener khai báo 2 phương thức như bên dưới

```
public interface Listener{  
    void onImageLoaded(Bitmap bitmap);  
    void onError();  
}
```

- ✓ Tạo class mới đặt tên LoadImageTask kế thừa lớp AsyncTask



```
public class LoadImageTask extends AsyncTask<String, Void, Bitmap>{

    private Listener mListener;
    private ProgressDialog progressDialog;
    public LoadImageTask(Listener listener, Context context) {
        // TODO Auto-generated constructor stub
        mListener = listener;
        progressDialog = new ProgressDialog(context);
    }
}
```

- ✓ Nạp chồng hàm onPreExecute và xử lý việc hiển thị dialog

```
@Override
protected void onPreExecute() {
    // TODO Auto-generated method stub
    super.onPreExecute();
    progressDialog.setMessage("Downloading image..");
    progressDialog.show();
}
```

- ✓ Tiến hành xử lý load hình từ server trong hàm doInBackground

```
@Override
protected Bitmap doInBackground(String... params) {

    try {
        return BitmapFactory.decodeStream((InputStream)new URL(params[0]).getContent());
    } catch (IOException e) {

        e.printStackTrace();
    }
    return null;
}
```

- ✓ Đóng dialog và set kết quả về cho hàm onImageLoaded ở interface listener

```
@Override
protected void onPostExecute(Bitmap result) {
    // TODO Auto-generated method stub
    super.onPostExecute(result);
    if(progressDialog.isShowing()){
        progressDialog.dismiss();
    }
    if(result != null){
        mListener.onImageLoaded(result);
    }else{
        mListener.onError();
    }
}
```

- ✓ Qua file activity Khai báo, ánh xạ cho view đã tạo, implements OnClickListener và Listener ở lớp LoadImageTask, Tạo biến kiểu String chứa đường dẫn hình, và bắt sự kiện click cho button

```
public class MainActivity extends Activity implements OnClickListener, Listener {

    private TextView tvMessage;
    private Button btnLoad;
    private ImageView imgLoad;

    public static final String IMAGE_URL = "http://i64.tinypic.com/28vaq8k.png";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        tvMessage = (TextView) findViewById(R.id.tvMessage);
        btnLoad = (Button) findViewById(R.id.btnLoad);
        imgLoad = (ImageView) findViewById(R.id.imgLoad);
        btnLoad.setOnClickListener(this);
    }
}
```

✓ Xử lý sự kiện click cho button

```
@Override
public void onClick(View arg0) {
    // TODO Auto-generated method stub
    switch (arg0.getId()) {
        case R.id.btnLoad:
            new LoadImageTask(this, this).execute(IMAGE_URL);
            break;
    }
}
```

✓ Gán hình cho ImageView, thông báo kết quả

```
@Override
public void onImageLoaded(Bitmap bitmap) {
    // TODO Auto-generated method stub
    imgLoad.setImageBitmap(bitmap);
    tvMessage.setText("Image Downloaded");
}

@Override
public void onError() {
    // TODO Auto-generated method stub
    tvMessage.setText("Error download image");
}
```

✓ Thêm permission INTERNET cho ứng dụng

```
<uses-permission android:name="android.permission.INTERNET"/>
```

- ✓ Chạy ứng dụng và xem kết quả

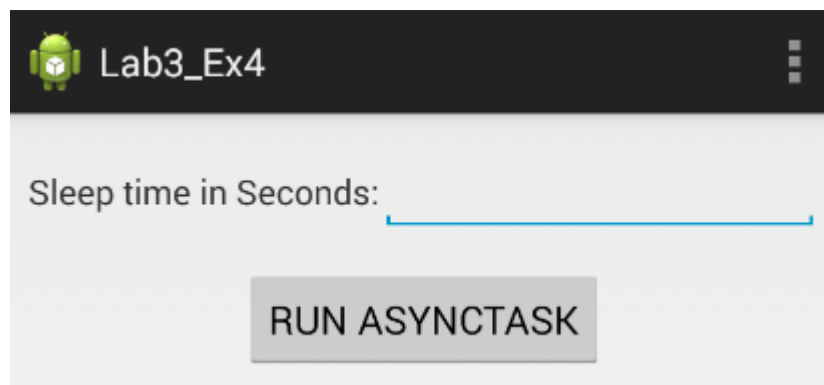


#### BÀI 4 (2 ĐIỂM)

Viết chương trình đơn giản cho phép nhập số giây để chạy nền một công việc nào đó. Sử dụng class AsyncTask

##### HƯỚNG DẪN:

- ✓ Xây dựng giao diện như hình bên dưới



- ✓ Khai báo, ánh xạ các view vừa tạo, implements OnClickListener và bắt sự kiện click cho button

```
public class MainActivity extends Activity implements OnClickListener {
    private EditText edtTime;
    private Button btnRun;
    private TextView tvResult;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        edtTime = (EditText) findViewById(R.id.edtTime);
        btnRun = (Button) findViewById(R.id.btnRun);
        tvResult = (TextView) findViewById(R.id.tvResult);
        btnRun.setOnClickListener(this);
    }
}
```

- ✓ Tạo class AsyncTaskRunner kế thừa AsyncTask, khai báo các biến để thao tác với view, tạo Constructor gán đối số tương ứng với view đã khai báo

```
public class AsyncTaskRunner extends AsyncTask<String, String, String> {

    private String resp;
    ProgressDialog dialog;
    TextView tvResult;
    EditText time;
    Context context;

    public AsyncTaskRunner(Context context, TextView tvResult, EditText time) {
        this.tvResult = tvResult;
        this.context = context;
        this.time = time;
    }
}
```

- ✓ Tiếp tục code xử lý các hàm của AsyncTask như bên dưới

```

@Override
protected void onPreExecute() {
    super.onPreExecute();
    dialog = ProgressDialog.show(context, "ProgressDialog", "Wait for "
+ time.getText().toString() + " seconds");
}

@Override
protected String doInBackground(String... params) {
    publishProgress("Sleeping...");
    try {
        int time = Integer.parseInt(params[0]) * 1000;
        Thread.sleep(time);
        resp = "Slept for " + params[0] + " seconds";
    } catch (Exception e) {
        e.printStackTrace();
        resp = e.getMessage();
    }
    return resp;
}

```

- ✓ Kết quả gán lại cho view sau khi hoàn tất công việc

```

@Override
protected void onPostExecute(String result) {
    super.onPostExecute(result);
    if (dialog.isShowing()) {
        dialog.dismiss();
    }
    tvResult.setText(result);
}

```

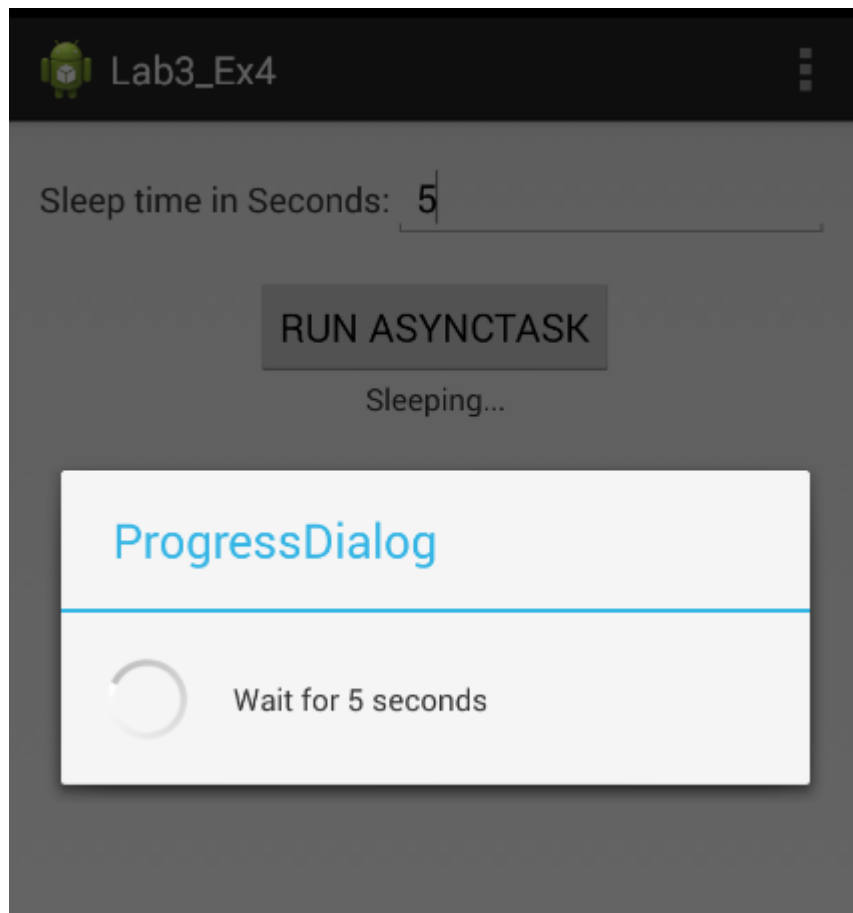
- ✓ Xử lý sự kiện cho button, khởi tạo class AsyncTaskRunner vừa tạo, truyền tham số

```

@Override
public void onClick(View v) {
    // TODO Auto-generated method stub
    switch (v.getId()) {
        case R.id.btnRun:
            AsyncTaskRunner asyncTaskRunner = new AsyncTaskRunner(this, tvResult, edtTime);
            String sleepTime = edtTime.getText().toString();
            asyncTaskRunner.execute(sleepTime);
            break;
    }
}

```

- ✓ Chạy ứng dụng và xem kết quả



### **BÀI 5 (2 ĐIỂM):**

Giảng viên cho thêm