

## MODULE 5:

### Nội dung thực hành

- Hiểu được cơ chế đồng thời (concurrency) trong Android
- Áp dụng được các kỹ thuật Asynchronous
  - Thread
  - Executor
  - HandlerThread
  - AsyncTask
  - Service & IntentService
  - AsyncQueryHandler
  - Loader

### Bài tập 1

#### Mục đích:

- Sử dụng được Thread để thực hiện cơ chế làm việc đồng thời

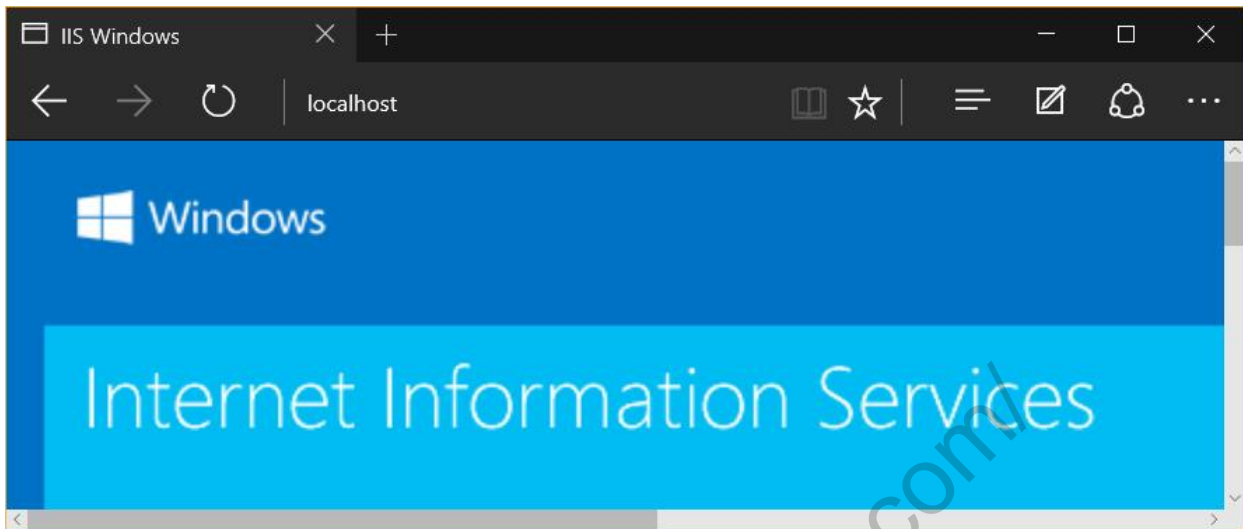
#### Yêu cầu:

Viết chương trình download tập tin trên mạng bằng Thread bên dưới và hiển thị tiến độ bằng progress bar

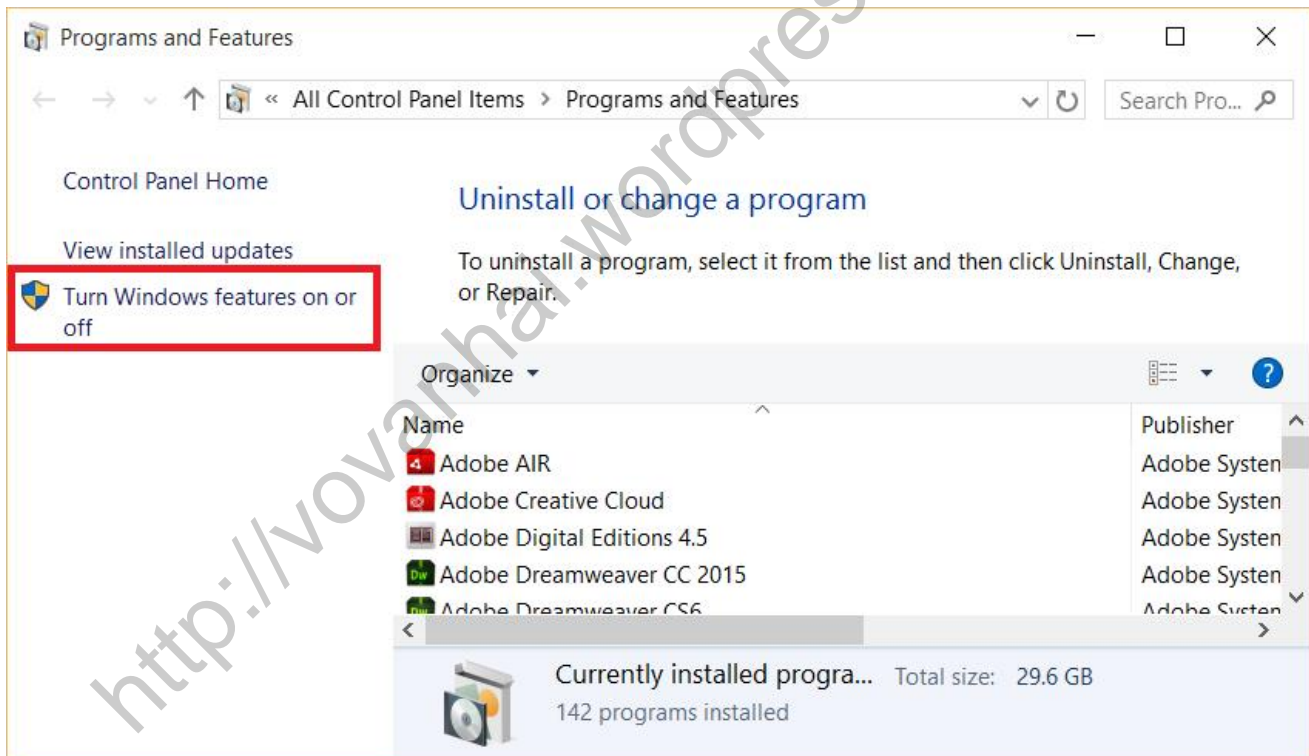
#### Hướng dẫn:

Để download một tập tin trên mạng, ta cần nối mạng. Để mô phỏng, ta cấu hình Internet Information Services (IIS) trên Windows.

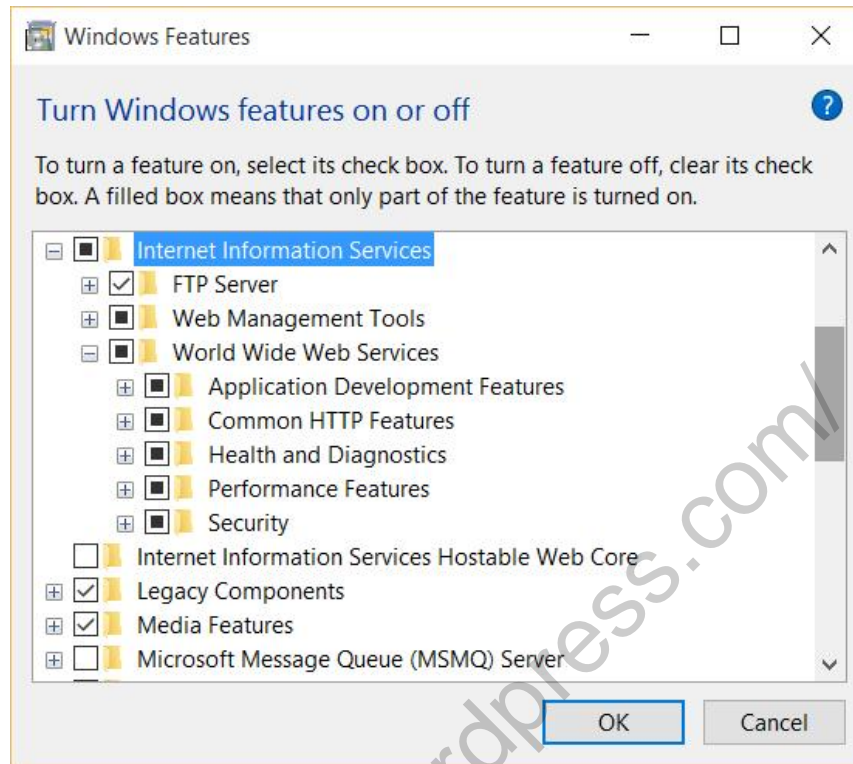
Bạn có thể thử máy mình có cài đặt IIS chưa bằng cách mở trình duyệt rồi vào <http://localhost>



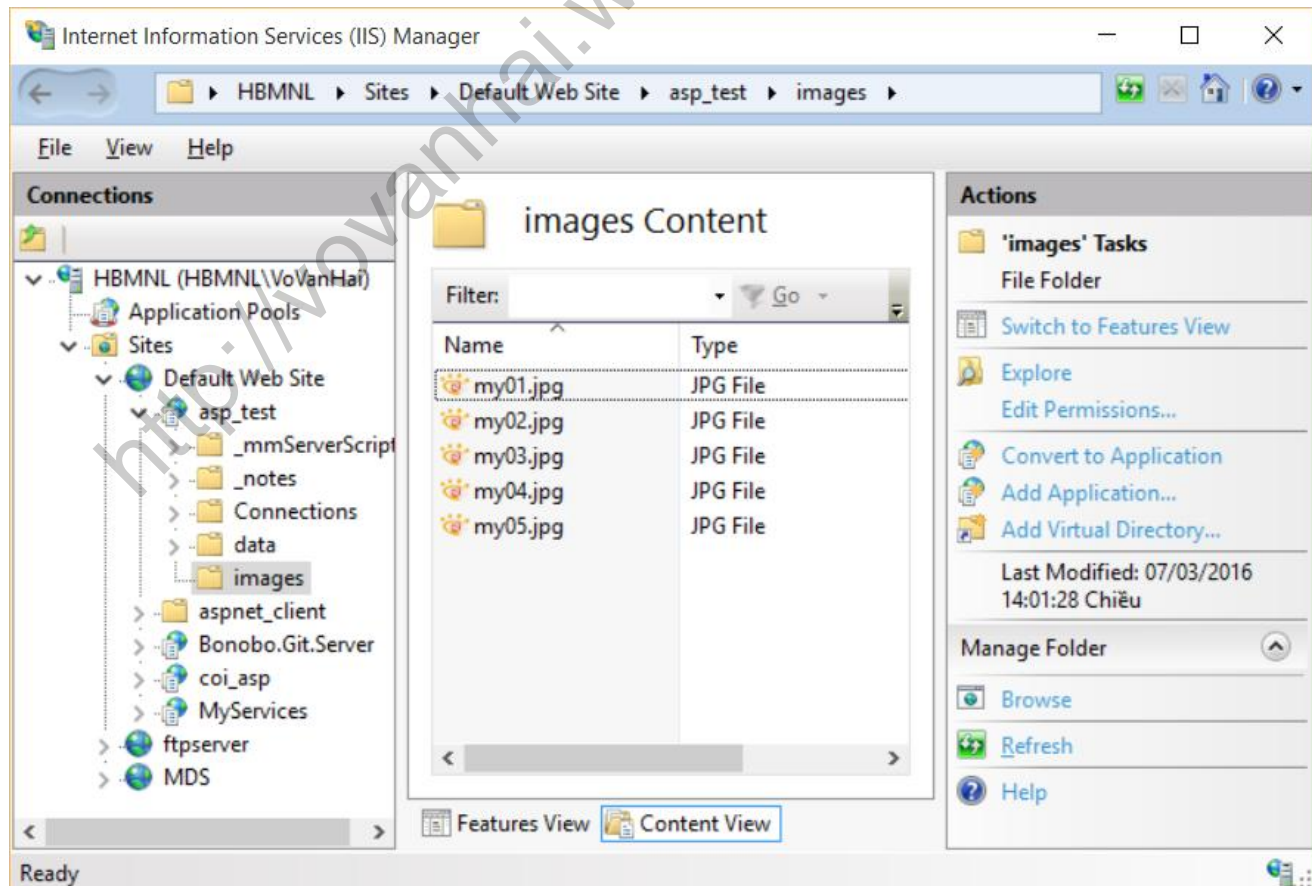
Nếu chưa có bạn có thể vào “Programs and Features” của windows và chọn như hình



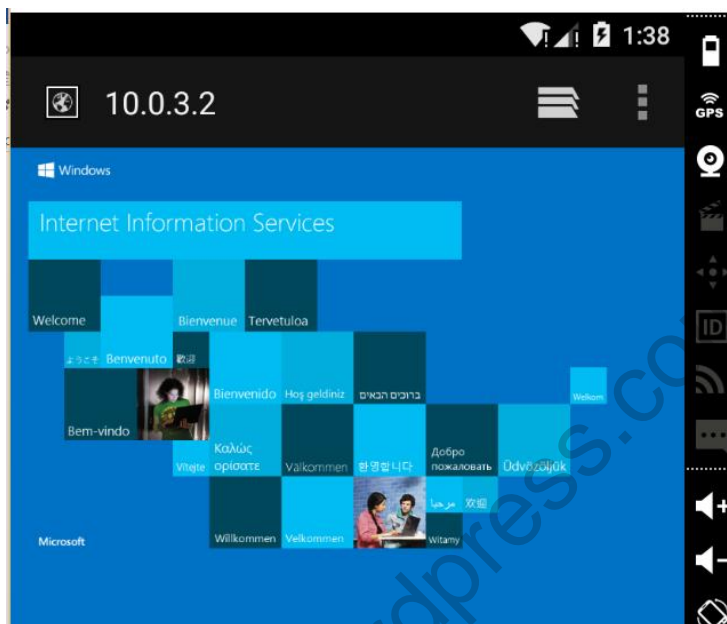
Sau đó chọn Internet Information Services như hình



Sau khi cài đặt bạn có thể mở IIS manger lên và tạo một sample application



Trong máy ảo Genymotion, bạn có thể truy xuất trang chủ của IIS trên máy host với link <http://10.0.3.2>



Lớp dùng làm tiến trình nền cho việc download

```
public class Downloader implements Runnable{
    private URL url;
    public Downloader(URL url) {
        this.url = url;
    }
    @Override
    public void run() {
        int fileSize=getFileSize(url);
        if(fileSize!=-1)
            downloadFile(fileSize);
    }
    /**
     * Download resource
     * @param fileSize là kích thước của resource
     */
    private void downloadFile(int fileSize) {
        try {
            File des=Environment.getExternalStorageDirectory();
            String x=url.getFile();
            String name=x.substring(x.lastIndexOf("/"));
            File desFile=new File(des.getAbsolutePath()+"/Download/"+name);
            FileOutputStream fis=new FileOutputStream(desFile);

            InputStream is = url.openStream();
            byte[] buffer=new byte[512];
            int len=0;
            long totalDownloaded=0;
            while((len=is.read(buffer))>0){
                totalDownloaded+=len;
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```

        Double dProgress = (totalDownloaded*100.0/ fileSize);
        int pos=dProgress.intValue();
        //cập nhật % đã download được
        pBar.setProgress(pos);
        //you can simulate for slow network
        Thread.sleep(5);
        //write to your destination
        fis.write(buffer, 0, len);
    }
    fis.flush();fis.close();is.close();
} catch (Exception e) {
    e.printStackTrace();
}
}
}
/**
 * Hàm này tính kích thước (file size) của tập tin cần
 * download để tính toán việc hiển thị progress bar
 * @param uri là resource cần download
 * @return kích thước của resource, -1 nếu resource không có
 */
private int getFileSize(URL uri) {
    HttpURLConnection conn = null;
    try {
        conn = (HttpURLConnection) uri.openConnection();
        conn.setRequestMethod("HEAD");
        conn.getInputStream();
        return conn.getContentLength();
    } catch (IOException e) {
        return -1;
    } finally {
        conn.disconnect();
    }
}
}
}

```

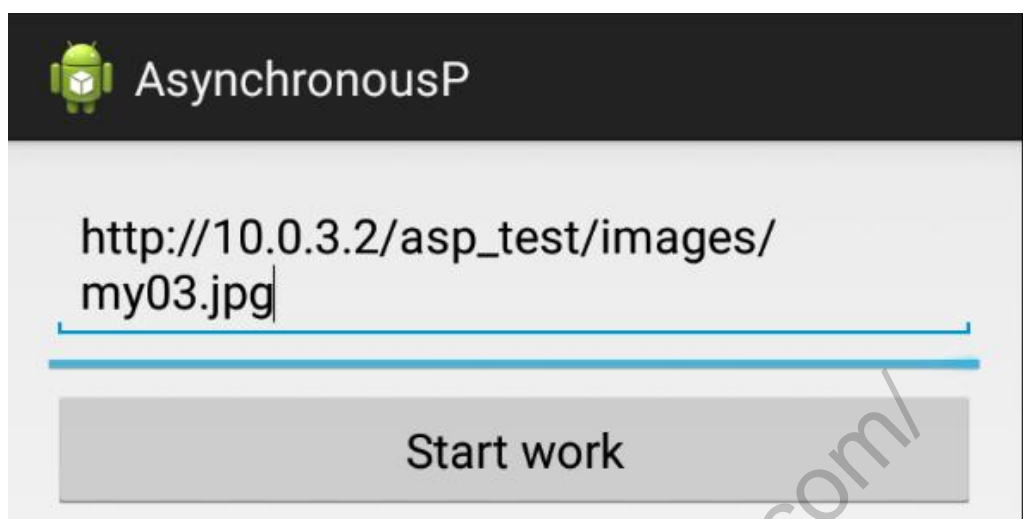
Phương thức cho nút download của ứng dụng

```

public void doWork(View v)throws Exception{
    URL url=new URL(edtURL.getText().toString());
    pBar.setProgress(0);
    new Thread(new Downloader(url)).start();
}

```

Giao diện đơn giản như sau (các bạn có thể cải tiến, thêm vào,...)



Cấp quyền truy xuất mạng và SD Card cho chương trình

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

## Bài tập 2 \*

### Mục đích

- Đồng bộ hóa tiến trình

### Yêu cầu

- Bài toán Producer/Consumer: Tiến trình sản xuất( Producer) tạo ra dòng thông tin để tiến trình tiêu thụ(Consumer) sử dụng. Phát biểu bài toán như sau:
  - o Giả sử có Hàng đợi(Queue) bao gồm một số hữu hạn các phần tử (Items) được tiến trình Producer lần lượt đưa vào các phần tử S1, S2, ...vào.
  - o Tiến trình Consumer lấy sản phẩm ra theo đúng thứ tự.
  - o Công việc của Producer phải đồng bộ với Consumer. Không đưa ra phần tử khi hết chỗ trống, không lấy được phần tử khi chưa có.
- Dùng một ListView để hiển thị các sản phẩm được đưa vào và lấy ra.

### Hướng dẫn

```
class MyQueue {
    int n;
    boolean valueSet = false;
    synchronized int get() {
        if(!valueSet)
            try { wait(); } catch (InterruptedException e) {}
        display2UI("Got: " + n);
        //assume that our work take a time to execute
        try{Thread.sleep(300);}catch(Exception x){}
        valueSet = false;
        notify();
        return n;
    }
    synchronized void put(int n) {
        if(valueSet)
            try { wait(); } catch (InterruptedException e) {}
        this.n = n;
        valueSet = true;
        display2UI("Put: " + n);
        //assume that our work take a time to execute
        try{Thread.sleep(500);}catch(Exception x){}
        notify();
    }
}
```

```
class Producer implements Runnable {
    MyQueue q;
    Producer(MyQueue q) {
        this.q = q;
        new Thread(this, "Producer").start();
    }
    public void run() {
        int i = 0;
```

```

        while(true) {
            q.put(i++);
        }
    }
}

```

```

class Consumer implements Runnable {
    MyQueue q;
    Consumer(MyQueue q) {
        this.q = q;
        new Thread(this, "Consumer").start();
    }
    public void run() {
        while(true) {
            q.get();
        }
    }
}

```

```

public void startWork(){
    ExecutorService service = Executors.newFixedThreadPool(2);
    MyQueue q = new MyQueue();

    service.execute(new Producer(q));
    service.execute(new Consumer(q));
}

```



## Bài tập 3

### Mục đích

- Hiểu cơ chế asynchronous với Handler class
  - o Sử dụng message
  - o Sử dụng post

### Yêu cầu

Viết lại chương trình download một tập tin từ internet (bài 1) dùng Handler class với hai cơ chế message và post.

### Hướng dẫn

Code sau là code cho activity download file từ mạng sử dụng Handler class với cơ chế truyền thông điệp giữa hai tiến trình

```
package vovanhai.wordpress.com.asynchronous;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.URL;

import android.app.Activity;
import android.os.Bundle;
import android.os.Environment;
import android.os.Handler;
import android.os.Message;
import android.os.StrictMode;
import android.os.SystemClock;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.ProgressBar;

public class DownloadUsingHandlerActivity extends Activity {

    private EditText edtURL;
    private ProgressBar pBar;
    private static Handler handler;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_download_using_handler);

        edtURL=(EditText) findViewById(R.id.editText1);
        pBar=(ProgressBar) findViewById(R.id.progressBar1);

        handler=new Handler(){
```

```

        public void handleMessage(android.os.Message msg) {
            super.handleMessage(msg);
            //cập nhật cho progress bar giá trị nhận được từ background
            Log.i("-----", msg.arg1+"");
            pBar.setProgress(msg.arg1);
        }
    };
}

public void downloadWithMessage(View v) throws Exception{
    //=====
    //sử dụng strictmode cấp mọi quyền cho tiến trình kết nối mạng
    //chạy tiến trình riêng thay vì cùng chung tiến trình với main
    //nếu không sẽ bị android.os.NetworkOnMainThreadException
    StrictMode.ThreadPolicy policy =
        new StrictMode.ThreadPolicy.Builder().permitAll().build();
    StrictMode.setThreadPolicy(policy);
    //=====

    final URL url=new URL(edtURL.getText().toString());
    final double fileSize=getFileSize(url);
    //do not download if filesize ==-1, ie. url not exists
    if(fileSize==1)return;

    pBar.setProgress(0);
    new Thread(new Runnable() {
        @Override
        public void run() {
            download(url, fileSize);
        }
    }).start();
}

private void download(final URL url, final double fileSize) {
    try {
        File des=Environment.getExternalStorageDirectory();
        String x=url.getFile();
        String name=x.substring(x.lastIndexOf("/"));
        File desFile=new File(des.getAbsolutePath()+"/Download/"+name);
        FileOutputStream fis=new FileOutputStream(desFile);

        InputStream is = url.openStream();
        byte[] buffer=new byte[512];
        int len=0;
        long totalDownloaded=0;
        while((len=is.read(buffer))>0){
            totalDownloaded+=len;
            Double dProgress = (totalDownloaded*100.0/ fileSize);
            int pos=dProgress.intValue();
            //gửi giá trị % download được lên cho handler cập nhật

            Message msg = handler.obtainMessage();
            msg.arg1=pos;
            handler.sendMessage(msg);
            //you can simulate for slow network
            SystemClock.sleep(10);
        }
    }
}

```

```

        //write to your destination
        fis.write(buffer, 0, len);
    }
    fis.flush();fis.close();is.close();
} catch (Exception e) {
    e.printStackTrace();
}
}
private int getFileSize(URL url) {
    HttpURLConnection conn = null;
    try {
        conn = (HttpURLConnection) url.openConnection();
        conn.setRequestMethod("HEAD");
        conn.getInputStream();
        return conn.getContentLength();
    } catch (IOException e) {
        return -1;
    } finally {
        conn.disconnect();
    }
}
}

```

### Với cơ chế post

```

public void downloadWithPost(View v)throws Exception{
    //=====
    StrictMode.ThreadPolicy policy =
        new StrictMode.ThreadPolicy.Builder().permitAll().build();
    StrictMode.setThreadPolicy(policy);
    //=====
    final URL url=new URL(edtURL.getText().toString());
    final double fileSize=getFileSize(url);
    //do not download if filesize ==-1, ie. url not exists
    if(fileSize==1)return;
    pBar.setProgress(0);
    new Thread(new Runnable() {
        @Override
        public void run() {
            try {
                File des=Environment.getExternalStorageDirectory();
                String x=url.getFile();
                String name=x.substring(x.lastIndexOf("/"));
                File desFile=new File(des.getAbsolutePath() +"/Download/"
+name);

                FileOutputStream fis=new FileOutputStream(desFile);
                InputStream is = url.openStream();
                byte[] buffer=new byte[512];
                int len=0;
                long totalDownloaded=0;
                while((len=is.read(buffer))>0){
                    totalDownloaded+=len;
                    Double dProgress = (totalDownloaded*100.0/ fileSize);
                    final int pos=dProgress.intValue();
                    //cập nhật progressbar

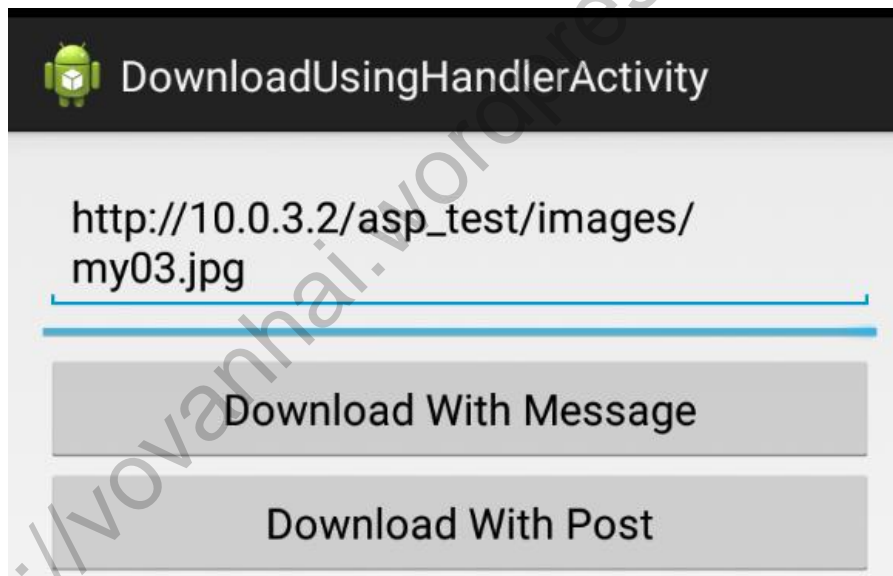
```

```

        pHandler.post(new Runnable() {
            @Override
            public void run() {
                pBar.setProgress(pos);
            }
        });
        //you can simulate for slow network
        SystemClock.sleep(10);
        //write to your destination
        fis.write(buffer, 0, len);
    }
    fis.flush();fis.close();is.close();
} catch (Exception e) {
    e.printStackTrace();
}
}
}).start();
}

```

Màn hình giao diện đơn giản



Cấp quyền truy xuất mạng và SD Card cho chương trình

```

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

```

**Bài tập 4 \*****Mục đích**

- Vận dụng được kỹ thuật asynchronous với AsyncTask class

**Yêu cầu**

Viết chương trình download các tập tin từ internet sử dụng AsyncTask class.

Chương trình sẽ download nhiều file với mỗi file có thanh tiến trình download riêng. (file nào download xong thì đóng thanh tiến trình đó).

Giao diện giả lập có một edittext cho người dùng thêm một url cần download vào một danh sách hiển thị trên listview, và một nút cho phép bắt đầu download.

**Hướng dẫn**

Trong bài hướng dẫn này, bạn sẽ thấy việc download là tuần tự cho từng file (cùng một task, vòng lặp for cho các lần download).

Yêu cầu bạn chỉnh sửa để việc download các file là đồng thời. (gợi ý: dùng một task cho một tiến trình download)

```
package vovanhai.wordpress.com.asynchronousp;

import java.net.URL;
import java.util.ArrayList;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.AlertDialog.Builder;
import android.app.Dialog;
import android.content.DialogInterface;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

public class DownloadUsingAsyncTaskActivity extends Activity {
    private EditText edtURL;
    private ListView lvURLs;
    private ArrayList<URL> lstURLs;
    private ArrayAdapter<URL> adapter;
    //tác vụ download
    private DownloadFilesTask task;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```

        setContentView(R.layout.activity_download_using_async_task);

        edtURL=(EditText) findViewById(R.id.editText1);
        lvURLs=(ListView) findViewById(R.id.ListView1);

        lstURLs=new ArrayList<URL>();
        //thêm cho việc thử nghiệm
        try {
            lstURLs.add(new URL("http://10.0.3.2/asp_test/images/my01.jpg"));
            lstURLs.add(new URL("http://10.0.3.2/asp_test/images/my02.jpg"));
            lstURLs.add(new URL("http://10.0.3.2/asp_test/images/my03.jpg"));
        } catch (Exception e) {
            e.printStackTrace();
        }

        adapter=new ArrayAdapter<URL>(this,
        android.R.layout.simple_expandable_list_item_1,lstURLs);
        lvURLs.setAdapter(adapter);
    }
    /*
    * Nút nhấnThêm một URL vào danh sách
    */
    public void addURL(View v)throws Exception{
        try {
            String urlS=edtURL.getText().toString();
            //điều này đồng thời kiểm tra một URL là hợp lệ không
            URL url=new URL(urlS);
            lstURLs.add(url);
            adapter.notifyDataSetChanged();
        } catch (Exception e) {
            Toast.makeText(this, "Invalid URL", Toast.LENGTH_SHORT).show();
        }
    }

    public void startDownloadAllItems(View v)throws Exception{
        int count=lstURLs.size();
        if(count==0) return;

        URL []params=new URL[count];
        params=lstURLs.toArray(params);

        //tạo Activity động với số progressbar
        ProgressBar[] bars=new ProgressBar[count];
        final Dialog dlg = createDialog(bars);
        dlg.show();
        //start download
        task = new DownloadFilesTask(bars){
            protected void onPostExecute(Long result) {
                dlg.dismiss();
            }
        };
        task.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR, params);
    }

    //Tạo một danh sách các progress
    private Dialog createDialog(ProgressBar[] bars) {

```

```

AlertDialog.Builder builder=new AlertDialog.Builder(this);

LinearLayout layout=new LinearLayout(this);
layout.setOrientation(LinearLayout.VERTICAL);
//sinh ra các thanh progress bar
for (int i = 0; i < bars.length; i++) {
    TextView tv = new TextView(this);tv.setText(""+(i+1));
    layout.addView(tv);
    bars[i]=new
ProgressBar(this,null,android.R.attr.progressBarStyleHorizontal);
    layout.addView(bars[i]);
}
Builder bd = builder.setView(layout);
bd.setPositiveButton("Hide", new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int which) {
        dialog.dismiss();
    }
});
bd.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int which) {
        task.cancel(true);
        dialog.cancel();//hủy bỏ tất cả các tác vụ
    }
});
return builder.create();
}
}

```

AsyncTask dùng cho việc download file

```

package vovanhai.wordpress.com.asynchronous;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.URL;

import android.os.AsyncTask;
import android.os.Environment;
import android.util.Log;
import android.widget.ProgressBar;

public class DownloadFilesTask extends AsyncTask<URL, Integer, Long> {
    private ProgressBar []pBar;

    public DownloadFilesTask(ProgressBar []pBar) {
        this.pBar = pBar;
    }
    protected Long doInBackground(URL... urls) {
        int count = urls.length;
        long totalSize = 0;
        for (int i = 0; i < count; i++) {
            totalSize += downloadFile(i, urls[i],pBar[i]);
        }
    }
}

```

```

        // Escape early if cancel() is called
        if (isCancelled()) break;
    }
    return totalSize;
}

protected void onProgressUpdate(Integer... progress) {
    //update UI progress
    int index=progress[0]; //pBar nào
    int val=progress[1]; //giá trị cho pBar đó
    pBar[index].setProgress(val);
    if(val==100) //pBar nào xong thì ẩn đi
        pBar[index].setVisibility(ProgressBar.GONE);
}

protected void onPostExecute(Long result) {
    Log.i("Downloader", "Downloaded completed. Received " + result + " bytes");
}

private long downloadFile(int index, URL url, ProgressBar pBar) {
    long downloadedSize=0;
    try {
        int fileSize=getFileSize(url);
        File des=Environment.getExternalStorageDirectory();
        String x=url.getFile();
        String name=x.substring(x.lastIndexOf("/"));
        File desFile=new File(des.getAbsolutePath()+"/Download/"+name);
        FileOutputStream fis=new FileOutputStream(desFile);

        InputStream is = url.openStream();
        int len=0;

        byte[] buffer=new byte[512];
        while((len=is.read(buffer))>0){
            downloadedSize+=len;
            //notify
            Double dProgress = (downloadedSize*100.0/ fileSize);
            int barPosition=dProgress.intValue();
            publishProgress(index,barPosition);
            //write to your destination
            fis.write(buffer, 0, len);
            //simulate slow network
            Thread.sleep(10);
        }
        fis.flush(); fis.close(); is.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return downloadedSize;
}

private int getFileSize(URL url) {
    HttpURLConnection conn = null;
    try {
        conn = (HttpURLConnection) url.openConnection();
        conn.setRequestMethod("HEAD");
        conn.getInputStream();
        return conn.getContentLength();
    }
}

```

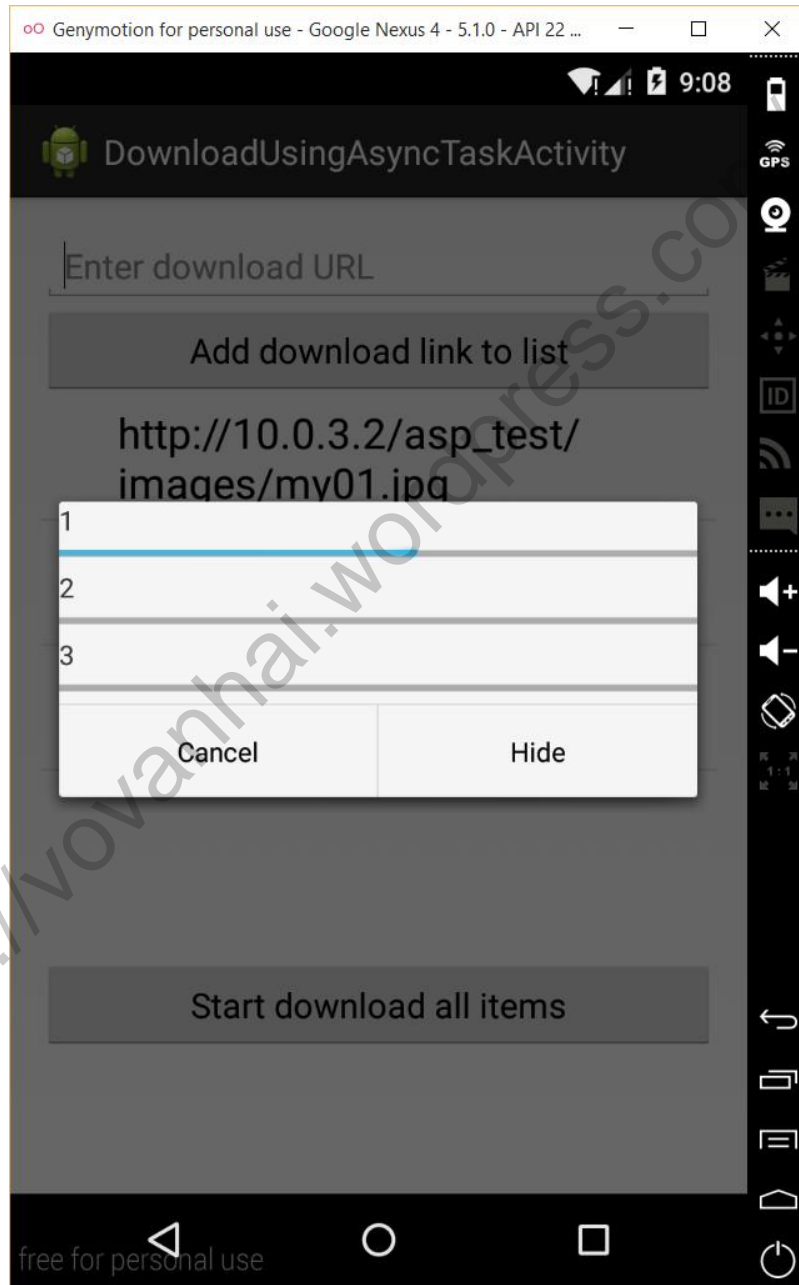


```

    } catch (IOException e) {
        return -1;
    } finally {
        conn.disconnect();
    }
}
}

```

Kết quả



**Bài tập 4 \*****Mục đích**

- Vận dụng được kỹ thuật asynchronous với Service

**Yêu cầu**

Viết chương trình download các tập tin từ internet sử dụng một service chạy nền.

**Hướng dẫn**

Trong bài hướng dẫn này tôi sẽ hướng dẫn các bạn download một resource từ internet sử dụng Service.

Đầu tiên bạn tạo một lớp kế thừa từ IntentService (nên dùng) hoặc Service

```
package vovanhai.wordpress.com.serv;  
import java.io.File;  
import java.io.FileOutputStream;  
import java.io.InputStream;  
import java.net.URL;  
import android.app.Activity;  
import android.app.IntentService;  
import android.content.Intent;  
import android.os.Environment;  
public class DownloadService extends IntentService{  
    private int result=Activity.RESULT_CANCELED;  
    public DownloadService() {  
        super("DownloadService");  
    }  
    @Override  
    protected void onHandleIntent(Intent intent) {  
        //lấy thông tin từ activity gửi đến service  
        String urlPath = intent.getStringExtra("url");  
        String fileName = intent.getStringExtra("filename");  
        File output=new File(Environment.getExternalStorageDirectory(),  
            "/Download/"+fileName);  
        if(output.exists()) output.delete();  
        try {  
            URL url=new URL(urlPath);  
            InputStream is=url.openStream();  
            FileOutputStream fos=new FileOutputStream(output);  
            byte[] buffer=new byte[512];  
            int x;  
            while((x=is.read(buffer))!=-1)  
                fos.write(buffer, 0, x);  
            fos.close();is.close();  
            result=Activity.RESULT_OK;  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
        //thông báo kết quả cho BroadcastReceiver  
        Intent it=new Intent("vovanhai.wordpress.com.download.action");  
        it.putExtra("filepath", output.getAbsolutePath());  
        it.putExtra("result", result);  
    }  
}
```

```
//Broadcasts the Intent to receivers in this app.
sendBroadcast(it);
}
}
```

Đăng ký service trong AndroidManifest.xml file

```
<service android:name="vovanhai.wordpress.com.serv.DownloadService"/>
```

Ngoài ra để truy xuất internet, SD card,... bạn phải cấp permission cho ứng dụng

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

Như vậy ta có một service và có thể triệu gọi khi cần download một resource nào đó trên internet.

Bây giờ ta thêm một activity gọi service này cho việc download. Trong activity này ta cần có một BroadcastReceiver để nhận lại kết quả từ service. Để đơn giản, ta khai báo bên trong Activity thay vì tạo một subclass của BroadcastReceiver

```
private BroadcastReceiver receiver=new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        Bundle bd=intent.getExtras();
        String filepath=bd.getString("filepath");
        int result=bd.getInt("result");
        if(result==RESULT_OK){
            Toast.makeText(DownloadUsingServiceActivity.this,
                "Download complete. Download URI: " + filepath,
                Toast.LENGTH_LONG).show();
        }
        else{
            Toast.makeText(DownloadUsingServiceActivity.this,
                "Download failed.",Toast.LENGTH_LONG).show();
        }
    }
};
```

Ta đăng ký BroadcastReceiver này trong phương thức onCreate như sau

```
registerReceiver(receiver,
    new IntentFilter("vovanhai.wordpress.com.download.action"));
```

Chú ý action uri phải giống với service gửi về

Để download một tài nguyên sử dụng service, ta tạo một intent dựa trên service class. Sau đó ta cần đưa tham số cho service(trong service này ta cần url và filename) sau đó ta startService để service bắt đầu làm việc.

```
public void startDownload(View v){
    Intent intent=new Intent(this,DownloadService.class);
    intent.putExtra("filename", "princess.jpg");
    intent.putExtra("url", "http://10.0.3.2/asp_test/images/my01.jpg");
    startService(intent);
}
```