

BÀI THỰC HÀNH BUỔI 5

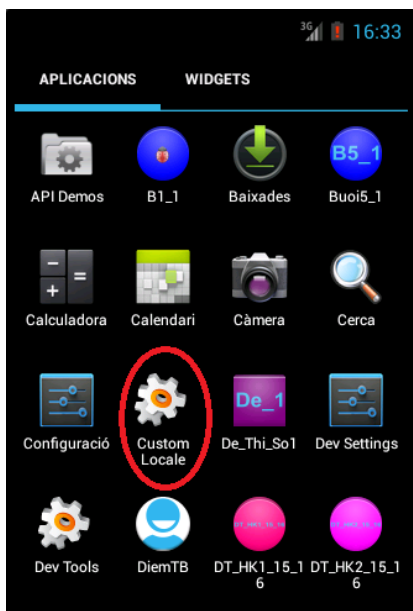
I. LÝ THUYẾT CẦN ÔN TẬP

- Lưu trữ dữ liệu bằng **Hệ thống tập tin** truyền thống (xem giáo trình/Slide bài giảng).
- Đa ngôn ngữ trong Android

Đa ngôn ngữ trong android giúp cho ứng dụng android có thể hiển thị giao diện với nhiều ngôn ngữ khác nhau tùy thuộc vào việc người dùng lựa chọn ngôn ngữ hiển thị trên thiết bị của họ.

Android mặc định tiếng Anh là ngôn ngữ chính, tạo sẵn và nạp chuỗi từ tập tin strings trong thư mục values (đường dẫn: **res/values/strings.xml**). Khi ta muốn bổ sung thêm những ngôn ngữ khác, ta cần phải tạo thư mục mới với tên được đặt theo quy tắc là **value-mã ngôn ngữ quốc gia**. Ví dụ nếu bạn muốn thêm tiếng Việt, ta phải tạo một thư mục với tên **values-vi**, sao chép tập tin **strings.xml** vào trong thư mục này và dịch tất cả các chuỗi từ tiếng Anh sang ngôn ngữ tiếng Việt.

Sau đây là vài mã ngôn ngữ quốc gia thông dụng: Pháp - fr, Đức - de, TQ - zh, Nhật - ja, Hàn quốc - ko,



Hình 5.1

Việc thay đổi chuỗi sang ngôn ngữ khác nhau được gọi là bản địa hóa (localization). Với ứng dụng được hỗ trợ nhiều ngôn ngữ, người dùng có thể chọn ngôn ngữ như sau:

1. Trên điện thoại (kể cả máy ảo, hình 5.1), chọn Settings hoặc Custom Locale ⇒ check chọn ngôn ngữ mà ta muốn.
2. Nếu ứng dụng hỗ trợ lựa chọn ngôn ngữ, android tìm kiếm các nguồn tài nguyên phù hợp với ngôn ngữ được chọn từ **values-(mã ngôn ngữ)**
3. Nếu các chuỗi nội địa hoá bị thiếu, android sẽ nạp các chuỗi đó từ tập tin **strings.xml** (**res** ⇒ **values** ⇒ **strings.xml**)

Một số lưu ý:

Nên làm

1. Luôn luôn khai báo các chuỗi trong tập tin strings.xml và sử dụng nó trong tập tin Layout.xml, thay vì khai báo trực tiếp trong tập tin Layout.

Ví dụ: Ta muốn hiển thị 1 chuỗi như “Hãy nhập vào một message”

Ta khai báo chuỗi này trong tập tin strings.xml như sau”

```
<string name="txt1">Hãy nhập vào một message</string>
```

Và sử dụng trong tập tin layout:

```
android:text="@string/txt1"
```

Không nên khai báo trực tiếp chuỗi trong Layout.xml hay trong mã java

```
android:text="Hãy nhập vào một message"
```

2. Dùng hàm setText() trong mã java khi cần

```
txtName.setText("Enter your address");
```

Trong đó chuỗi “Enter your address” là nội dung mới cần thay thế cho nội dung cũ.

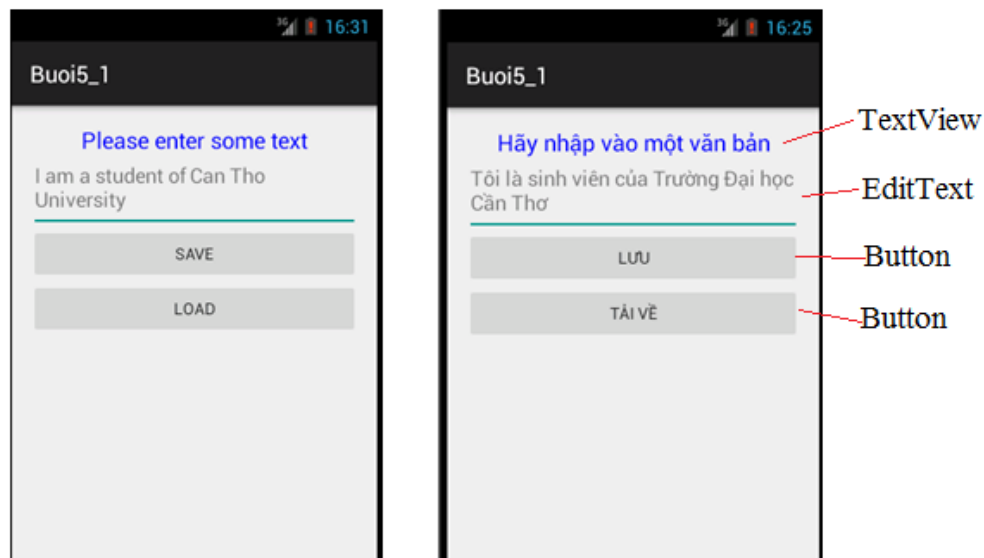
Áp dụng vào các bài sau đây để xây dựng giao diện tiếng Anh và Việt.

II. THỰC HÀNH

BÀI 10: LƯU TRỮ DỮ LIỆU TRÊN BỘ NHỚ TRONG

Trước tiên ta sẽ thực hành cách lưu dữ liệu bằng tập tin vào bộ nhớ trong của thiết bị. Các bước thực hiện như sau:

- 1) Mở Android project, trong ví dụ này project được đặt tên là **B5_1**.
- 2) Sửa lại tập tin Layout.xml để có giao diện tiếng Anh và tiếng Việt như hình 5.2.a và 5.2.b:



Hình 5.2.a: Giao diện tiếng Anh Hình 5.2.b: Giao diện tiếng Việt

Cách thực hiện:

- Lập trình Layout với các thuộc tính text dùng định danh `android:text="@string/<tên text>"` và text được khai báo trong tập tin `string.xml` (Ngôn ngữ ban đầu là Tiếng Anh)

Mã nguồn như sau:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.doanhoaminh.b5_1.MainActivity">

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textColor="#0000ff"
        android:textSize="18sp"
        android:gravity="center"
        android:background="#ffffaa"
        android:text="@string/txt1" />
    <EditText
        android:id="@+id/txtText1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="@string/edittext1"/>
    <Button
        android:id="@+id/btnSave"
        android:text="@string/btn1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:onClick="onClickSave" />
    <Button
        android:id="@+id/btnLoad"
        android:text="@string/btn2"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:onClick="onClickLoad" />

</LinearLayout>
```

Trập tin `strings.xml`

```
<resources>
    <string name="app_name">B5_1</string>
    <string name="txt1">Please enter some text</string>
    <string name="edittext1">I am a student of CanTho University</string>
    <string name="btn1">SAVE</string>
    <string name="btn2">LOAD</string>
</resources>
```

- Nhấp chuột phải vào thư mục **src** → Chọn **New** → Chọn **Android Resource Directory** → Cửa sổ **New Resource Directory** hiện ra → Đặt tên Directory là **values-vi** → Nhấp OK.

- Copy tập tin **strings.xml** trong thư mục **values** dán vào thư mục **values-vi**
→ Đổi các text tiếng Anh thành tiếng Việt.

Tập tin string.xml trong thư mục values-vi như sau:

```
<resources>
  <string name="app_name">B5_1</string>
  <string name="txt1">Hãy nhập vào một văn bản</string>
  <string name="edittext1">Tôi là sinh viên trường đại học Cần Thơ</string>
  <string name="btn1">LƯU</string>
  <string name="btn2">TẮT</string>
</resources>
```

- Trên máy ảo Android, kích hoạt ứng dụng Custom Locale và chọn vi-Tiếng Việt.

3) Sửa lại tập tin MainActivity.java như sau:

```
package com.example.b5_1;

import android.support.v7.app.ActionBarActivity;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends ActionBarActivity {
    EditText textBox;
    static final int READ_BLOCK_SIZE = 100;
    /** Called when the activity is first created. */

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        textBox = (EditText) findViewById(R.id.txtText1);
    }
    public void onClickSave(View view) {
        String str = textBox.getText().toString();
        try
        {

            FileOutputStream fOut = openFileOutput("textfile.txt",
                                                    MODE_WORLD_READABLE);

            OutputStreamWriter osw = new
                OutputStreamWriter(fOut);
            //---write the string to the file---
            osw.write(str);
        }
    }
}
```

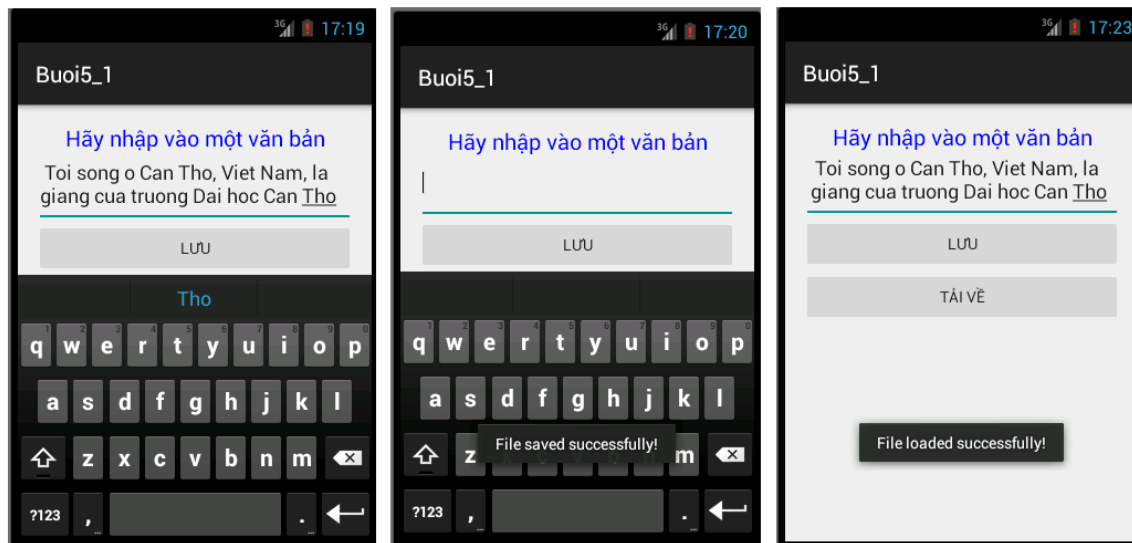
```

        osw.flush();
        osw.close();
        //---display file saved message---
        Toast.makeText(getBaseContext(), "File saved successfully!",
            Toast.LENGTH_SHORT).show();
        //---clears the EditText---
        textBox.setText(" ");
    }
    catch (IOException ioe)
    {
        ioe.printStackTrace();
    }
}

public void onClickLoad(View view) {
    try
    {
        //.....
        FileInputStream fIn = openFileInput("textfile.txt");
        InputStreamReader isr = new InputStreamReader(fIn);
        char[] inputBuffer = new char[READ_BLOCK_SIZE];
        String s = "";
        int charRead;
        while ((charRead = isr.read(inputBuffer)) > 0)
        {
            //---convert the chars to a String---
            String readString = String.valueOf(inputBuffer, 0, charRead);
            s += readString;
            inputBuffer = new char[READ_BLOCK_SIZE];
        }
        //---set the EditText to the text that has been
        // read---
        textBox.setText(s);
        Toast.makeText(getBaseContext(), "File loaded successfully!",
            Toast.LENGTH_SHORT).show();
    }
    catch (IOException ioe) {
        ioe.printStackTrace();
    }
}
}

```

- 4) Chạy thử ứng dụng với thiết bị giả lập. (Giao diện như hình 5.13).
- 5) Nhập văn bản vào EditText và nhấn Button Save.
- 6) Nếu tập tin được lưu thành công thì sẽ xuất hiện một Toast là “File save successfully”. Văn bản trong EditText sẽ biến mất.
- 7) Nhấn Button Load ta sẽ thấy văn bản xuất hiện lại trên EditText. Việc này cho thấy văn bản đã được lưu. (Hình 5.3)



Hình 5.3

8) Kiểm tra

Trên Eclipse: Mở DDMS → Chọn mục File → Tìm file theo đường dẫn: data/data/Thư mục có tên miền của ứng dụng (ví dụ: [com.example.buoil5_1/files/textfile.txt](#)) → Nhấp chọn tập tin này → Chọn mục “Full a file from the device” (📁) ở góc trên phải của cửa sổ File và xuất tập tin này ra 1 thư mục nào đó trên PC (ví dụ Desktop) → Mở file để xem nội dung.

Trên Android studio: Chọn View (trên Navigation bar) → Chọn Tools Windows → Chọn Device File Explorer (có thể chọn phương tiện này trực tiếp tại góc phải bên dưới của màn hình) → vào thư mục data/data/<tên miền của ứng dụng>/files/textfile.txt, nhấn chuột phải, chọn open để mở tập tin. (nếu không thấy cây thư mục của máy ảo thì vào Tools, Chọn Android, Chọn Enable ADB)

Lưu ý: Vì emulator không có phần mềm gõ tiếng Việt, nên trong hình 5.3, các chữ nhập vào EditText không bỏ dấu được.

Giải thích:

Để lưu văn bản vào tập tin ta đã dùng lớp `FileOutputStream`, dùng phương thức `openFileOutput()` mở tập tin để nhập văn bản, với chế độ xác định. Trong ví dụ này, ta đã dùng hằng `MODE_WORLD_READABLE` để chỉ định rằng tập tin có thể đọc bởi ứng dụng khác. Nếu ta dùng hằng `MODE_PRIVATE` thì tập tin chỉ có thể đọc bởi ứng dụng đã tạo ra nó, và nếu dùng `MODE_WORLD_WRITEABLE` thì tập tin có thể được truy xuất để đọc và chỉnh sửa với mọi ứng dụng.

Để chuyển đổi một dòng ký tự thành một dòng byte, ta sử dụng một thể hiện của lớp `OutputStreamWriter`, bằng cách đi qua nó một thể hiện của đối tượng `FileOutputStream`:

```
OutputStreamWriter osw = new
```

```
OutputStreamWriter(fOut);
```

Sau đó, ta sử dụng phương thức **write ()** để viết các chuỗi ký tự vào tập tin. Để đảm bảo rằng tất cả các byte được ghi vào tập tin, ta dùng phương thức **flush()**. Cuối cùng, dùng phương thức **close()** để đóng tập tin.

Để đọc nội dung của tập tin ta dùng lớp **FileInputStream** kết hợp với lớp **InputStreamReader**. Vì không biết trước kích thước của tập tin sẽ đọc, nội dung được đọc từng khối 100 ký tự đưa vào bộ đệm (character array). Sau đó các ký tự đã đọc được chép vào một đối tượng String.

Phương thức **read()** của lớp **InputStreamReader** kiểm tra số ký tự đã đọc và trả về giá trị -1 khi kết thúc tập tin.

Khi thử ứng dụng với thiết bị giả lập, ta có thể sử dụng các DDMS để xem và kiểm chứng rằng ứng dụng đã thực sự lưu tập tin vào thư mục của nó.

BÀI 11: LƯU TRỮ DỮ LIỆU TRÊN BỘ NHỚ NGOÀI (SD Card)

Phần trên đã trình bày cách lưu dữ liệu vào bộ nhớ trong của thiết bị Android. Tuy nhiên, người dùng thường muốn lưu dữ liệu vào bộ nhớ ngoài (chẳng hạn như SD card), bởi vì dung lượng dữ liệu cần lưu lớn và cũng để dễ dàng chia sẻ với người dùng khác (chuyên SD card qua thiết bị khác).

Mở project mới, đặt tên B5_2, Chúng ta sẽ sử dụng project trong Bài 5_1 và chỉnh sửa lại để có thể lưu văn bản do người dùng nhập vào trên SD Card. Các bước thực hiện như sau:

- 1) Trên tập tin mã nguồn Java, ta chỉnh sửa phương thức **onClick()** của Button Save như sau (phần in đậm):

```
package com.example.file;

import android.support.v7.app.AppCompatActivity;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStreamWriter;

import android.os.Bundle;
import android.os.Environment;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    EditText textBox;
    static final int READ_BLOCK_SIZE = 100;
    /** Called when the activity is first created. */

    @Override
```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    textBox = (EditText) findViewById(R.id.txtText1);
}
public void onClickSave(View view) {
    String str = textBox.getText().toString();//.....
    try
    {
        //---Luu tren SD Card---
        File sdCard = Environment.getExternalStorageDirectory();
        File directory = new File (sdCard.getAbsolutePath() + "/MyFiles");
        directory.mkdirs();
        File file = new File(directory, "textfile.txt");
        FileOutputStream fOut = new FileOutputStream(file);
        OutputStreamWriter osw = new OutputStreamWriter(fOut);

        /*FileOutputStream fOut =
            openFileOutput("textfile.txt",
                MODE_WORLD_READABLE);
        OutputStreamWriter osw = new
            OutputStreamWriter(fOut); */

        //---write the string to the file---
        osw.write(str);
        osw.flush();
        osw.close();
        //---display file saved message---
        Toast.makeText(getApplicationContext(),
            "File saved successfully!",
            Toast.LENGTH_SHORT).show();
        //---clears the EditText---
        textBox.setText(" ");
    }
    catch (IOException ioe)
    {
        ioe.printStackTrace();
    }
}
public void onClickLoad(View view) {
    try
    {
        FileInputStream fIn = openFileInput("textfile.txt");
        InputStreamReader isr = new InputStreamReader(fIn);
        char[] inputBuffer = new char[READ_BLOCK_SIZE];
        String s = "";
        int charRead;
        while ((charRead = isr.read(inputBuffer))>0)
        {
            //---convert the chars to a String---
            String readString =
                String.copyValueOf(inputBuffer, 0,
                    charRead);
            s += readString;
            inputBuffer = new char[READ_BLOCK_SIZE];
        }
    }
}

```



```

    }
    //---set the EditText to the text that has been
    // read---
    textBox.setText(s);
    Toast.makeText(getBaseContext(), "File loaded successfully!",
        Toast.LENGTH_SHORT).show();
    }
    catch (IOException ioe) {
        ioe.printStackTrace();
    }
}
}

```

Ta đã dùng phương thức **getExternalStorageDirectory()** để trả về đường dẫn đầy đủ tới bộ nhớ ngoài. Cụ thể là đường dẫn **“/sdcard”** cho thiết bị thật và **“/mnt/sdcard”** cho thiết bị giả lập. Tuy nhiên, ta không nên “mã cứng” đường dẫn đến SD card, vì có thể nhà sản xuất thiết bị chỉ định tên đường dẫn khác tới SD card. Vì vậy, cần phải dùng phương thức **getExternalStorageDirectory()** để trả về đường dẫn đầy đủ đến SD card.

Cuối cùng, ta lưu tập tin vào thư mục này.

Để tải tập tin từ bộ nhớ ngoài, ta sửa phương thức **onClickLoad()** của nút Load như sau (phần in đậm và highlight):

```

public void onClickLoad(View view) {
    try
    {
        //--- Đọc file lưu trên SD---
        File sdCard = Environment.getExternalStorageDirectory();
        File directory = new File (sdCard.getAbsolutePath() +
            "/MyFiles");
        File file = new File(directory, "textfile.txt");
        FileInputStream fIn = new FileInputStream(file);
        InputStreamReader isr = new InputStreamReader(fIn);

        /*
        // Đọc file lưu trên bộ nhớ trong
        FileInputStream fIn = openFileInput("textfile.txt");
        InputStreamReader isr = new InputStreamReader(fIn);
        */

        char[] inputBuffer = new char[READ_BLOCK_SIZE];
        String s = "";
        int charRead;
        while ((charRead = isr.read(inputBuffer))>0)
        {
            //---convert the chars to a String---
            String readString =String.valueOf(inputBuffer, 0,
                charRead);
            s += readString;
            inputBuffer = new char[READ_BLOCK_SIZE];

```

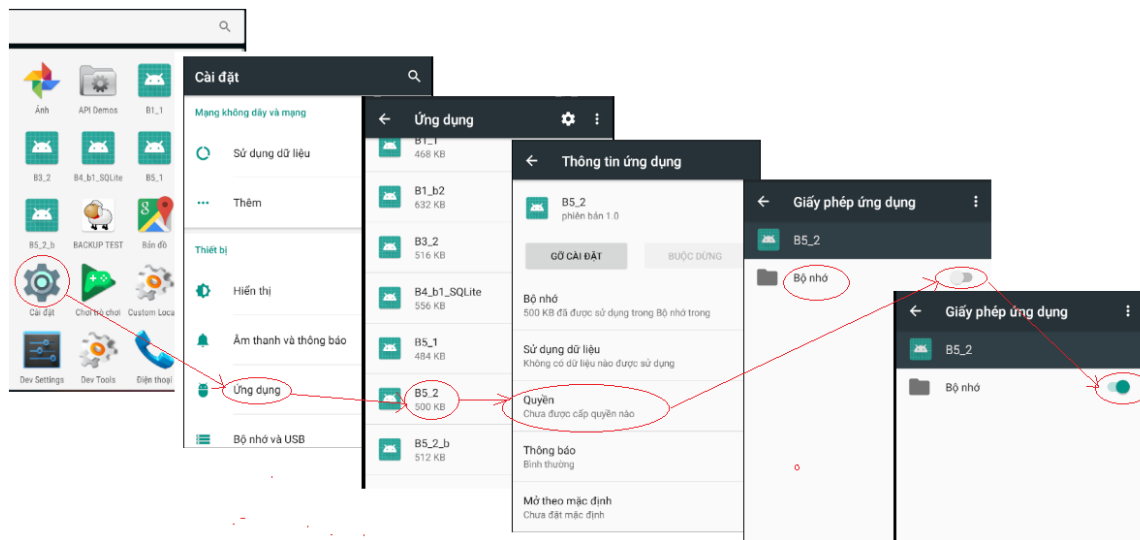
```
    }  
    // Đặt lại EditText với text đọc thành công  
    textBox.setText(s);  
    Toast.makeText(getBaseContext(),  
        "File loaded successfully!",  
        Toast.LENGTH_SHORT).show();  
}  
catch (IOException ioe) {  
    ioe.printStackTrace();  
}  
}
```

*Lưu ý, để có thể ghi lên bộ nhớ ngoài, ta cần thêm permission **WRITE_EXTERNAL_STORAGE** vào tập tin **AndroidManifest.xml**, như sau (phần in đậm và highlight):*

```
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.example.file"  
    android:versionCode="1"  
    android:versionName="1.0" >  
  
    <uses-sdk  
        android:minSdkVersion="11"  
        android:targetSdkVersion="19" />  
  
    <uses-permission  
        android:name="android.permission.WRITE_EXTERNAL_STORAGE" />  
    <uses-permission  
        android:name="android.permission.READ_EXTERNAL_STORAGE" />  
  
    <application  
        android:allowBackup="true"  
        android:icon="@drawable/ic_launcher"  
        android:label="@string/app_name"  
        android:theme="@style/AppTheme" >  
        <activity  
            android:name=".MainActivity"  
            android:label="@string/app_name" >  
            <intent-filter>  
                <action android:name="android.intent.action.MAIN" />  
  
                <category android:name="android.intent.category.LAUNCHER" />  
            </intent-filter>  
        </activity>  
    </application>  
</manifest>
```

(2) Chạy thử ứng dụng với thiết bị giả lập và kiểm tra tập tin đã được lưu vào thư mục theo đường dẫn `/mnt/sdcard/MyFiles/`, mở và kiểm tra nội dung.

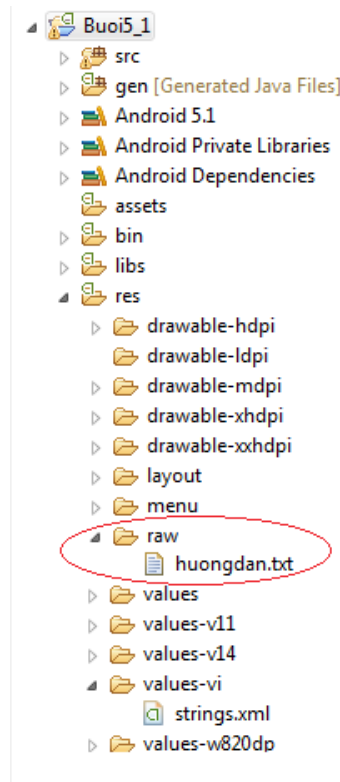
Lưu ý: Nếu không lưu được tập tin vào thẻ nhớ (sdCard) trên Android emulator, ta kiểm tra lại lệnh permission trong AndroidManifest.xml, nếu đã có lệnh này rồi thì ta mở ứng dụng nhấn chọn ứng dụng “Cài đặt” của thiết bị → Chọn “Ứng dụng” → Chọn ứng dụng muốn thử → Chọn “Quyền” → Nhấn bật thẻ nhớ (Hình 5.4).



Hình 5.4

BÀI 12: SỬ DỤNG TÀI NGUYÊN TĨNH

Bên cạnh việc sinh ra tập tin trong thời gian chạy, ta cũng cần tạo ra các tập tin trong thời gian phát triển ứng dụng, sau đó sẽ được dùng trong thời gian chạy. Ví dụ, ta muốn tạo ra tập tin trợ giúp (Help) cho ứng dụng. Trong trường hợp này, ta thêm tập tin help vào thư mục res/raw. Hình 5.4 trình bày thư mục res/raw chứa tập tin có tên là helpfile.txt đã được tạo ra khi lập trình.



Hình 5.4: Thư mục res/raw chứa tập tin textfile.

Sử dụng project Bài 1 hoặc Bài 2 để xây dựng thêm một tài nguyên tĩnh. Các bước thực hiện:

- (1) Trong thư mục project của Bài 2 (nếu chọn project Bài 2) ta tạo 1 thư mục mới tên là **raw** trong thư mục **res**.
- (2) Mở Notepad để tạo ra 1 tập tin văn bản và lưu vào thư mục res/raw với tên là **huongdan** (không có đuôi).
- (3) Trong Layout của Bài 2 thêm 1 Button có tên là HELP (tiếng Anh) và HƯỚNG DẪN (tiếng Việt). Chú ý rằng, các Button đều có thuộc tính onClick để nhận tương tác của người dùng trong Java code. Ví dụ:
`android:onClick="onClickSave"` ; cho Button Save, tương ứng với hàm onClickSave trong Java code.
`android:onClick="onClickLoad"` ; cho button Load, tương ứng với hàm onClickLoad trong Java code.
`android:onClick="onClickHelp"` ; cho Button Help, tương ứng với hàm onClickHelp trong Java code.
- (4) Xây dựng một Activity mới có tên là **StaticfileActivity**. Trong activity này, ta sử dụng phương thức getResources() để trả về một đối tượng resources InputSteam và dùng phương thức openRawResource() để mở tập tin chứa trong thư mục res/raw, dùng lớp InputStreamReader để tạo ra một luồng dữ liệu đọc, dùng lớp BufferedReader để tạo ra bộ đệm chứa dữ liệu đọc, dùng thành phần Toast để hiển thị văn bản. Mã nguồn của lớp này như sau:

```
import android.support.v7.app.ActionBarActivity;
import android.widget.EditText;
import android.widget.Toast;
import android.os.Bundle;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;

public class StaticfileActivity extends ActionBarActivity {
    EditText textBox;
    static final int READ_BLOCK_SIZE = 100;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_staticfile);
        textBox = (EditText) findViewById(R.id.txtText1);

        InputStream is = this.getResources().openRawResource(R.raw.huongdan);
        InputStreamReader ir=new InputStreamReader(is);
        BufferedReader br = new BufferedReader(ir);
        String str = null;

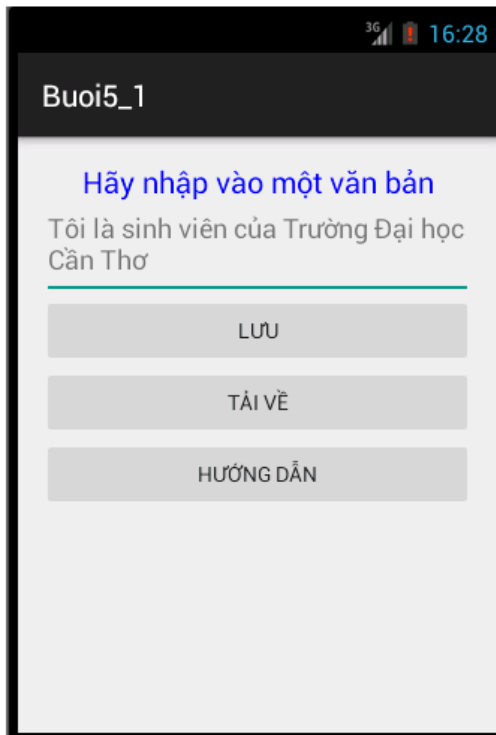
        try {
            while ((str = br.readLine()) != null) {
                Toast.makeText(getApplicationContext(),
                    str, Toast.LENGTH_SHORT).show();
            }
            is.close();
            br.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Lưu ý, ID của tài nguyên được lưu trong thư mục res/raw được đặt tên không có phần mở rộng sau tên tập tin. Ví dụ, với tên tập tin là *huongdan.txt*, thì resource ID là *R.raw.huongdan*.

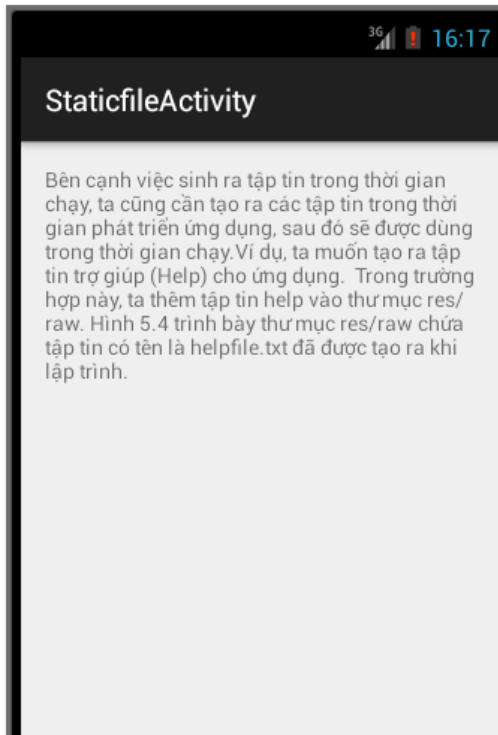
- (5) Thêm hàm **onClickHelp** vào activity chính **B5_2_Activity**, để tạo tương tác khi người dùng nhấn nút HELP. Code như sau:

```
public void onClickHelp(View view) {
    // TODO Auto-generated method stub
    Intent hd = new Intent(B5_2_MainActivity.this,
        StaticfileActivity.class);
    startActivity(hd);
}
```

- (6) Chạy thử, ta thấy tập tin hướng dẫn hiển ra rồi lặn nhanh, không đọc kịp, bởi vì trong trong mã nguồn đã dùng thành phần Toast để hiển thị.
- (7) Sinh viên hãy tham khảo lý thuyết và **cải tiến lại mã nguồn Java** trong lớp **StaticfileActivity** để hiển thị tập tin *huongdan* lâu dài có thể đọc được.



Hình 5.5.a: Giao diện được thêm vào nút HƯỚNG DẪN



Hình 5.5.b: Tập tin huongdan được hiển thị khi người dùng chọn nút HƯỚNG DẪN