



FPT POLYTECHNIC

LẬP TRÌNH ANDROID CƠ BẢN

Bài 6: Lưu trữ dữ liệu

www.poly.edu.vn

Nội dung bài học

- Lưu trữ thông tin trên thiết bị
- Lưu trữ thông tin trên thẻ nhớ
- Shared Preferences



Lưu trữ dữ liệu

- Android cung cấp một số lựa chọn để lưu trữ dữ liệu
- Lựa chọn giải pháp nào tùy thuộc vào nhu cầu bảo mật của ứng dụng và kích thước của file cần lưu trữ
- Các tùy chọn lưu trữ bao gồm:
 - Shared Preferences
 - Internal Storage
 - External Storage
 - SQLite Database
 - Network Connection

SharedPreferences

- Lớp SharedPreferences cung cấp một framework chung để lưu trữ dữ liệu dưới dạng key-value của dữ liệu kiểu nguyên thủy
- Có thể sử dụng để lưu trữ dữ liệu kiểu boolean, float, int, long, string
- Dữ liệu này được lưu trữ trong các phiên làm việc của người dùng (ngay cả khi ứng dụng bị kill)



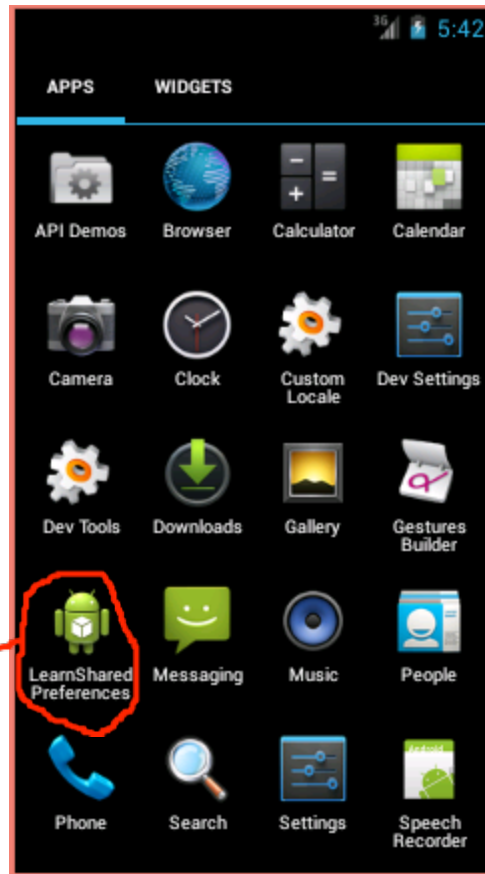
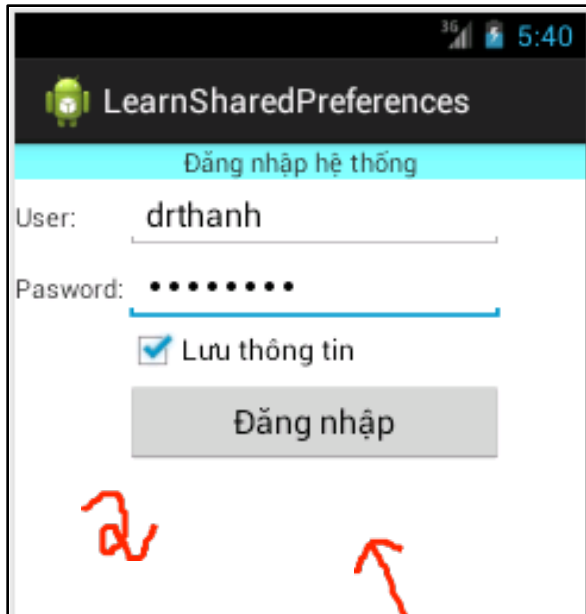
SharedPreferences

SharedPreferences

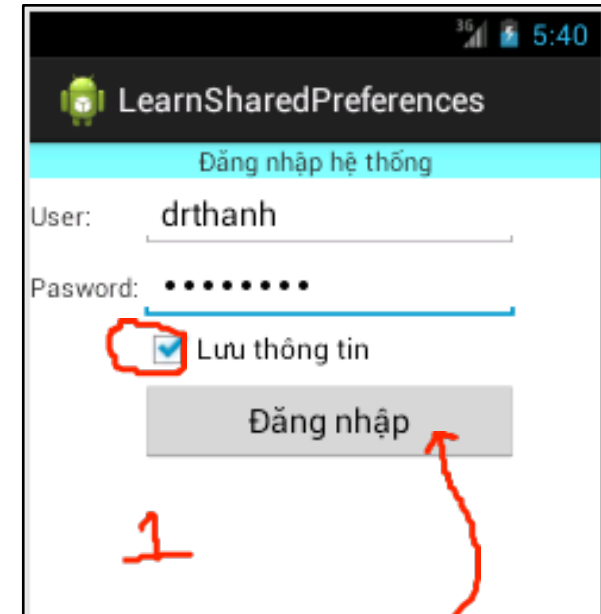
- Gọi `getSharedPreferences()`
- Ghi thông tin:
 - Gọi `edit()` để nhận `SharedPreferences.Editor`
 - Bổ sung giá trị sử dụng `putBoolean`, `putString`,...
 - Commit giá trị mới sử dụng `commit()`
- Đọc thông tin: sử dụng phương thức như `putBoolean`, `putString`,...



SharedPreferences



Mở lại ứng dụng →
thông tin được khôi
phục lại



Check và click vào nút Login

SharedPreferences

```
private void savingPreferences() {  
    //tao doi tuong getSharedPreferences  
    SharedPreferences sharepreference = getSharedPreferences(filename, MODE_PRIVATE);  
    //tao doi tuong Editor de luu thay doi  
    SharedPreferences.Editor editor = sharepreference.edit();  
    String user = txtUsername.getText().toString();  
    String password = txtPassword.getText().toString();  
    boolean chk = chkStatus.isChecked();  
    if (!chk)  
    {  
        //Xoa moi luu tru truoc do  
        editor.clear();  
    }  
    else {  
        //luu vao editor  
        editor.putString("username", user);  
        editor.putString("password", password);  
        editor.putBoolean("savestatus", chk);  
    }  
    //chấp nhận lưu xuống file  
    editor.commit();  
}
```

SharedPreferences

```
private void restoringPreferences() {  
    SharedPreferences pref = getSharedPreferences(filename, MODE_PRIVATE);  
    boolean chk = pref.getBoolean("savestatus", false);  
    if (chk)  
    {  
        //lấy user, pwd, nếu không thấy giá trị mặc định là rỗng  
        String user = pref.getString("username", "");  
        String password = pref.getString("password", "");  
        txtUsername.setText(user);  
        txtPassword.setText(password);  
    }  
    chkStatus.setChecked(chk);  
}
```

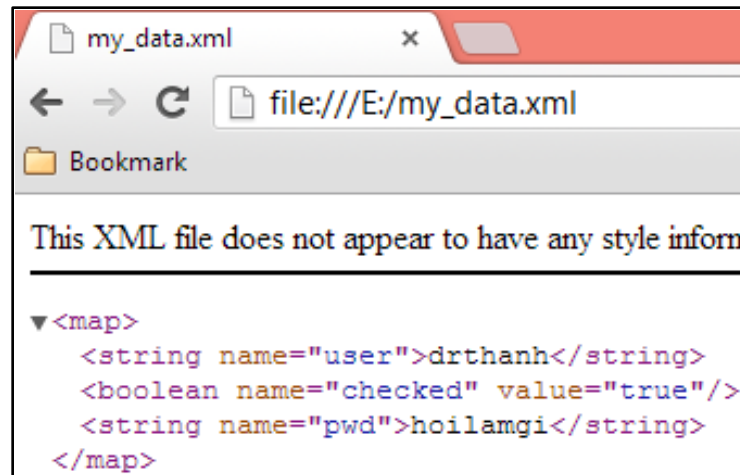

SharedPreferences

Threads Heap Allocation Tracker Network Statistics File Explorer

Name	Size	Date	Time	Perm
▼ fpoly.android.com		2014-04-03	03:13	drwxr
▶ cache		2014-04-01	17:07	drwxr
lib		2014-04-03	03:13	lrwxr
▼ shared_prefs		2014-04-02	17:36	drwxr
mydata.xml	203	2014-04-02	17:36	-rw-r



Vị trí lưu
SharedPreferences
(DDMS/File
Explorer/data/data/
package
name/shared_prefs



Định dạng
XML

Internal Storage

- Bạn có thể lưu file trực tiếp vào bộ nhớ trong của thiết bị
- Mặc định, các file được lưu ở bộ nhớ trong là riêng tư và các ứng dụng khác không có quyền truy cập đến
- Khi người dùng xóa ứng dụng, các file này sẽ tự động bị xóa theo



Internal Storage

- Tạo và ghi private file tới internal storage
 - Gọi `openFileOutput()` với tên file và tham số chế độ hoạt động. Trả về một `FileOutputStream`
 - Ghi tới file sử dụng `write()`
 - Đóng stream sử dụng `close()`

```
FileOutputStream fos = openFileOutput(FILENAME, Context.MODE_PRIVATE);  
fos.write(string.getBytes());  
fos.close();
```

Internal Storage

- MODE_PRIVATE sẽ tạo file (hoặc thay thế file nếu trùng tên) và thiết lập file này là riêng tư đối với các ứng dụng khác
- Các mode khác có thể là:
MODE_APPEND, MODE_WORLD_READABLE, MODE_WORLD_WRITEABLE



Internal Storage

- Đọc file từ Internal Storage
 - Gọi `openFileInput()` và truyền tên file muốn đọc. Trả về `FileInputStream`
 - Đọc sử dụng `read()`
 - Sau đó đóng stream sử dụng `close()`



Lưu trữ Cache files

- Nếu bạn muốn lưu cache file, sử dụng `getCacheDir()` để mở File đại diện cho thư mục nội tại là nơi mà ứng dụng lưu thông tin cache tạm thời
- Khi thiết bị ở trạng thái thiếu dung lượng, Android có thể xóa một số file cache để giải phóng dung lượng.
- Tuy nhiên, không nên dựa vào Android sẽ dọn dẹp cho bạn
- Cỡ của file cache tối đa 1 MB
- Khi bạn xóa bỏ ứng dụng, file cache sẽ bị xóa theo

Tạo cache file

```
public void createCache()  
{  
    try {  
        File pathCacheDir = getCacheDir();  
        String strCacheFileName = "myCacheFile.cache";  
        String strFileContents = editdata.getText()+"";  
        File newCacheFile = new  
            File(pathCacheDir, strCacheFileName);  
        newCacheFile.createNewFile();  
        FileOutputStream foCache =  
            new FileOutputStream(  
                newCacheFile.getAbsolutePath());  
        foCache.write(strFileContents.getBytes());  
        foCache.close();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

Đọc cache file

```
public void readCache() {  
    try {  
        File pathCacheDir = getCacheDir();  
        String strCacheFileName = "myCacheFile.cache";  
        File newCacheFile = new  
            File(pathCacheDir, strCacheFileName);  
        Scanner sc=new Scanner(newCacheFile);  
        String data="";  
        while(sc.hasNext())  
        {  
            data+=sc.nextLine()+"\n";  
        }  
        editdata.setText(data);  
        sc.close();  
    } catch (FileNotFoundException e) {  
        e.printStackTrace();  
    }  
}
```


Lấy thông tin tất cả cache files

```
public void loadAllCache()  
{  
    File pathCacheDir = getCacheDir();  
    File []listCache= pathCacheDir.listFiles();  
    for(File f :listCache)  
    {  
        //process f here  
        f.delete();  
    }  
}
```

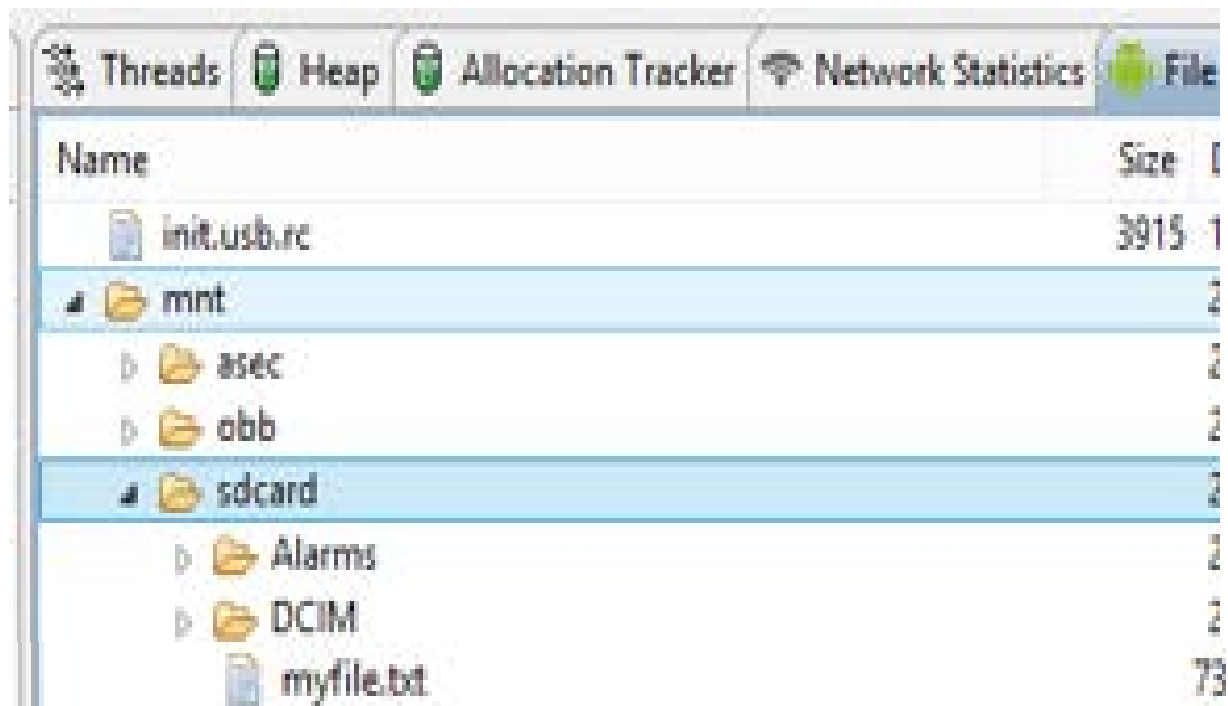
Lưu trữ thông tin trên thẻ nhớ

- Lưu trữ thông tin trên thẻ nhớ SDCard
- Ưu điểm là dung lượng lưu trữ lớn hơn



External Storage

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



The screenshot shows the File Explorer tab in Android Studio. The file system structure is as follows:

Name	Size
init.usb.rc	3915
mnt	
asec	
obb	
sdcard	
Alarms	
DCIM	
myfile.txt	73

Đọc từ thẻ nhớ

```
public void readData()  
{  
    String sdcard=Environment  
        .getExternalStorageDirectory()  
        .getAbsolutePath()+"/myfile.txt";  
    try {  
        Scanner scan=new Scanner(new File(sdcard));  
        String data="";  
        while(scan.hasNext())  
        {  
            data+=scan.nextLine()+"\n";  
        }  
        scan.close();  
        editdata.setText(data+"");  
    } catch (FileNotFoundException e) {  
        e.printStackTrace();  
    }  
}
```

Ghi vào thẻ nhớ

```
public void writeData()  
{  
    String sdcard=Environment  
        .getExternalStorageDirectory()  
        .getAbsolutePath()+"/myfile.txt";  
    try {  
        OutputStreamWriter writer=  
            new OutputStreamWriter(  
                new FileOutputStream(sdcard));  
        writer.write(editdata.getText()+"");  
        writer.close();  
    } catch (FileNotFoundException e) {  
        e.printStackTrace();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

Tổng kết nội dung bài học

- Lưu trữ thông tin trên thiết bị
- Lưu trữ thông tin trên thẻ nhớ
- Shared Preferences



Kết thúc!

