

## Android数据的存储

### 内部存储空间:

1. RAM内存: 运行内存, 相当于电脑的内存
2. ROM内存: 存储内存, 相当于电脑的硬盘

### 外部存储空间:

1. SD卡: 相当于电脑的移动硬盘

案例实践: 通过对ROM的读写完成登录保存功能(截图如下):



相应的UI代码如下:

```
1. <EditText
2.     android:id="@+id/txt_password"
3.     android:layout_width="match_parent"
4.     android:layout_height="wrap_content"
5.     android:hint="请输入密码" />
6.
7. <RelativeLayout
8.     android:layout_width="match_parent"
9.     android:layout_height="wrap_content">
10.    <CheckBox android:id="@+id/chk_save"
11.        android:layout_width="wrap_content"
12.        android:layout_height="wrap_content"
13.        android:layout_centerVertical="true"
14.        android:hint="保存密码" />
15.    <Button
16.        android:onClick="btn_login"
17.        android:layout_width="wrap_content"
18.        android:layout_height="wrap_content"
19.        android:layout_alignParentRight="true"
20.        android:text="确认" />
21. </RelativeLayout>
```

把用户名与密码, 保存到ROM的代码如下:

```
1. // 1: 获取用户名密码
2. EditText txt_username = (EditText) findViewById(R.id.txt_username);
3. EditText txt_password = (EditText) findViewById(R.id.txt_password);
4. CheckBox chk_save = (CheckBox) findViewById(R.id.chk_save);
5. String name = txt_username.getText().toString();
6. String pass = txt_password.getText().toString();
```

```
7.  if (chk_save.isChecked()) {
8.      // 2: 保存密码
9.      File file = new File(getFilesDir(), "user.txt");
10.     OutputStream outputStream = null;
11.     try {
12.         outputStream = new FileOutputStream(file);
13.         outputStream.write((name + "," + pass).getBytes());
14.     } catch (Exception e) {
15.         throw new RuntimeException();
16.     } finally {
17.         try {
18.             outputStream.close();
19.         } catch (Exception e) {
20.             throw new RuntimeException();
21.         }
22.     }
23. }
```

从文件中读取相应的数据代码如下：

```
1.  File file = new File(getFilesDir(), "user.txt");
2.  if (file.exists()) {
3.      FileInputStream inputStream = null;
4.      try {
5.          inputStream = new FileInputStream(file);
6.          // 返回下一次对此输入流调用的方法可以不受阻塞地从此输入流读取（或跳过）的估计
           剩余字节数
7.          byte[] b = new byte[inputStream.available()];
8.          int len=inputStream.read(b);
9.          String data = new String(b,0,len);
10.         EditText txt_username = (EditText) findViewById(R.id.txt_username);
11.         txt_username.setText(data);
12.     } catch (Exception e) {
13.         throw new RuntimeException();
14.     } finally {
15.         try {
16.             inputStream.close();
17.         } catch (Exception e) {
18.             throw new RuntimeException();
19.         }
20.     }
21. }
```