

将数据保存到文件

1.1 知识解析

Android 为数据存储提供了多种方式，分别有如下几种：

- 文件
- SharedPreferences
- SQLite 数据库
- 内容提供者 (Content Provider)
- 网络

Android 系统中和数据相关的几个目录

每个 Android 应用均有一个存放数据的目录：/data/data/[app_pkg]，在这个目录下可以有：

- files：内部存储的数据
- databases：数据库
- shared_prefs：使用 SharedPreferences 存储的数据
- cache：临时文件
- 内部存储 (internal storage)：手机的内存。始终可用。默认情况下只有本应用可以访问
- 外部存储 (external storage)：手机的“SD 卡”，并非总是可用。它是全局可读的。

使用内部存储

Android 提供的几个和内部存储文件相关的方法：

- File Context.getDir(String name, int mode)：获取指定的目录，用于保存应用的数据。如果目录不存在，将创建目录

- `File Context.GetFilesDir()`：将返回使用 `openFileOutput(String name, int mode)` 所存储的文件所在的路径，即应用内部的目录

mode 取值：

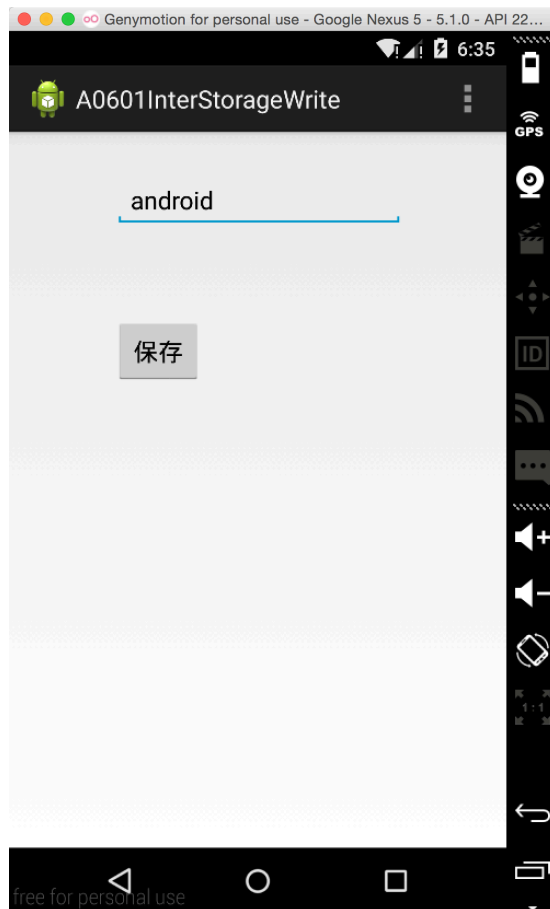
- `MODE_PRIVATE`
- `MODE_APPEND`
- `MODE_WORLD_READABLE` (deprecated from api 17)
- `MODE_WORLD_WRITABLE` (deprecated from api 17)

`File getCacheDir()`：返回应用所需要保存临时文件的目录，在系统空间不足的时候，将会无警告的情况下删除这些文件

可以使用 `FileOutputStream Context.openFileOutput(String filename, int mode)` 来打开一个文件并往里写数据。

可以使用 `FileInputStream Context.openFileInput(String filename)` 来打开一个文件流并从中读取数据。

1.2 功能演示



1.3 实战操作

```
byte[] buffer = str.getBytes();

try {
    FileOutputStream fos =
MainActivity.this.openFileOutput("abc.txt", Context.MODE_APPEND);

    fos.write(buffer);
} catch (FileNotFoundException e) {
    // TODO Auto-generated catch block
```

```
        e.printStackTrace();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }
```

1.4 职业素质

android 手机内存本来就不大，要是老把数据放在手机里，很明显会让手机的使用者体验到什么是“卡”，所以，我们把数据要放到 SD 卡中，以减少手机内存的使用。