Android 开发进阶：如何读写 Android 文件

Android 主要有四大主要组件组成：Activity、ContentProvider、Service、Intent 组成。Android 文件的运行主要需要读写四大组件的文件。本文将介绍如何读写 Android 文件，希望对正在进行 Android 开发的朋友有所帮助。

文件存放位置在 Android 中文件的 I/O 是存放在/data/data/<package name>/file/filename 目录下。

提示:Android 是基于 linux 系统的，在 linux 的文件系统中不存在类似于 Windows 的磁盘分区现象，其是以一个正斜杠“/”开头。

Android 中得到输入输出流在 Android 中，对于流的操作十分简单。在 Context 类中有如下两个方法可以直接得到文件输入输出流:

```
public FileInputStream openFileInput (String name)
public FileOutputStream openFileOutput (String name, int mode)
```

顾名思义，通过如上方法就可以得到文件输入输出流。对于第二个方法中的 mode，有如下四种模式：

◆Use 0 or MODE_PRIVATE(the default operation) :用 0 表示默认值，只能够创建文件的应用程序访问该文件，每次文件写入为覆盖方式。

◆MODE_APPEND to append to an existing file: 每次文件写入为追加方式，类似于 StringBuffer 中的 append（）方法。

◆MODE_WORLD_READABLE ： 只有读权限。

◆MODE_WORLD_WRITEABLE ： 只有写权限。

提示：如果想同时得到读与写的权限，则可以在 mode 处通过如下方式创建：

```
MODE_WORLD_READABLE+ MODE_WORLD_WRITEABLE
```

对于 Java SE 部分的补充 FileOutputStream：

public void write(byte[] b) throws IOException 该方法可将指定的字节数组写入文件输出流

FileInputStream：

public int read(byte[] b) throws IOException 从此输入流中将最多 b.length 个字节的数据读入一个 byte 数组中。在某些输入可用之前，此方法将阻塞。

对于输出流直接使用 write 方法即可，可参考如下代码：

Java 代码

```
/** *  写入数据  * @param fs * @param content */
public void fileWrite(FileOutputStream fos,String content)
{
byte[] contentcontentByteArray = content.getBytes();
try
```

```
{
fos.write(contentByteArray);
}
catch (IOException e1)
{
e1.printStackTrace();
}
try
{
//关闭流
fos.close();
}
catch (IOException e)
{
e.printStackTrace();
}
}
```

对于输入流，出于性能考虑，可先使用 ByteArrayOutputStream，向内存中创建一个字符数组，当将文件读完后，在读入，参考如下代码：

Java 代码

```
/**  读数据
   * @param fis
   * @return
   */
public String fileRead(FileInputStream fis)
{
ByteArrayOutputStream baos = new ByteArrayOutputStream();
      byte[] buffer = new byte[1024];
int len = -1;
try
{
    while((len=(fis.read(buffer))) != -1)
    {
     baos.write(buffer,0,len);
        }
     }
    catch (IOException e)
     {
      e.printStackTrace();
     }
     String result = new String(baos.toByteArray());       //System.out.println(result);
      try
      {
       baos.close();
```

```
            fis.close();
        }
        catch (IOException e)
        {
          e.printStackTrace();
        }
        return result;
}
```

ByteArrayOutputStream:此类实现了一个输出流，其中的数据被写入一个 byte 数组。public void write(byte[] b,int off,int len) 将指定 byte 数组中从偏移量 off 开始的 len 个字节写入此 byte 数组输出流