沙盒机制

# 前言

iOS系统相对于Android（或Windows）系统来说比较安全的原因有很多，其中有一点就是苹果推出的沙盒机制，每个应用都有自己对应的沙盒，每个应用程序之间不能相互访问非本程序的沙盒，所以，iOS系统相对于其他的系统来说比较安全，再是从内存上来说相对于Windows来说也比较安全，Apple的应用程序在内存消耗过高时会收到内存警告，如果不及时处理的话应用程序会自动退出，而不像Windows系统一样，中了病毒或者木马会一直消耗内存，直到内存没有了才宕机。可以说沙盒机制，让iOS系统变得更安全。

# 沙盒原理

沙盒是一种安全机制，用于防止不同应用之间互相访问。iOS系统下每个应用都有自己对应的沙盒，每个沙盒之间都是相互独立的，互不能访问（没有越狱的情况下）。

1. 每个应用程序都有自己的存储空间
2. 应用程序不能翻过自己的围墙去访问别的存储空间的内容
3. 应用程序请求的数据都要通过权限检测，假如不符合条件的话，不会被放行。

# 沙盒结构

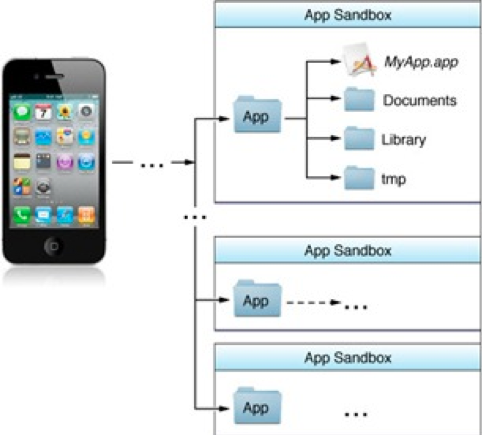
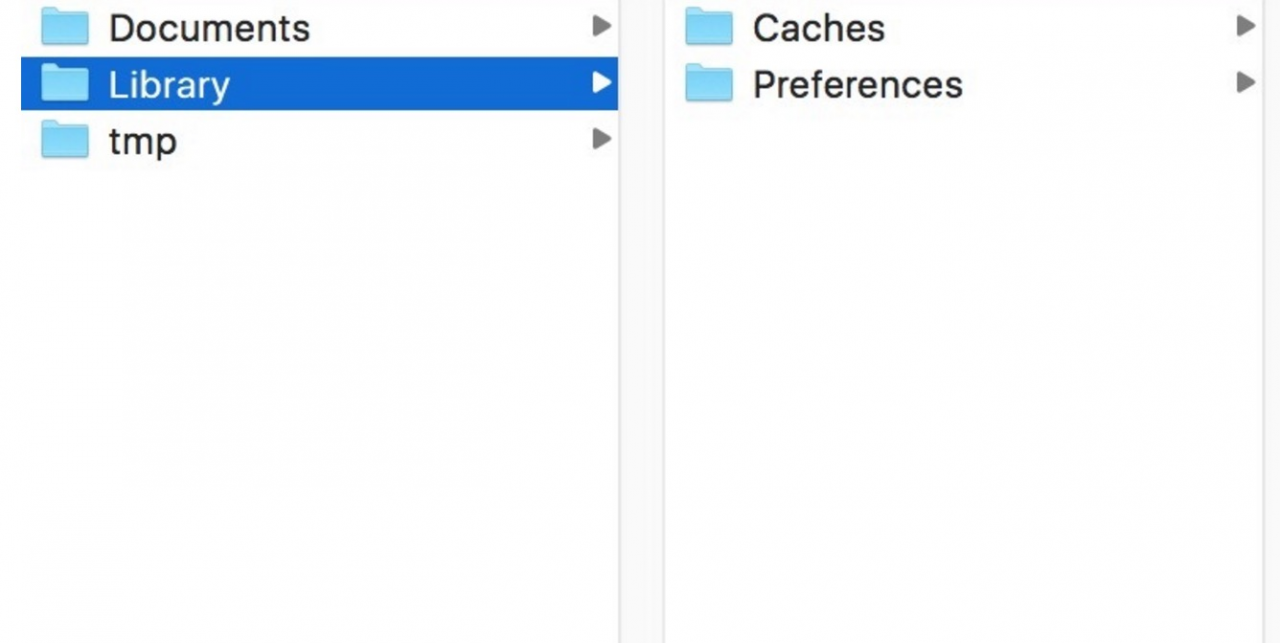


**沙盒目录中包含：Documents、Library、tmp和一个xxx.app⽂件**

1. **Documents**：用于存放程序中的文件数据，应用程序在运行时生成的一些需要长久保存的数据（比如：游戏进度存档、应用程序个人设置等等），通过 iTunes、iCloud 备份时, 会备份这个目录下的数据，此目录下保存相对重要的数据。
2. **Library** 包含Caches和Preferences⼦目录
3. **Caches**：存放缓存文件，从网络上下载的文件或者数据（如：音乐缓存、图片缓存等），此目录下的文件不会在应用退出时自动删除 ，需要程序员手动清除改目录下的数据。iTunes、iCloud 备份时不会备份此目录下的数据。主要用于保存应用在运行时生成的需要长期使用的数据，一般用于存储体积较大，不需要备份的非重要数据。
4. **Preferences**存放的是基于NSUserDefaults的设置数据，文件格式为 “plist”。设置应用的一些功能会在该目录中查找相应设置的信息，iTunes、iCloud备份时会备份此目录下的数据。该目录由系统自动管理，通常用来储存一些基本的应用配置信息。比如账号密码、自动登录等。
5. **tmp**：存放应用运行时产生的一些临时数据和文件，当应用程序退出、系统磁盘空间不足、手机重启时，都会自动清除该目录的数据。无需程序员手动清除该目录中的数据，iTunes、iCloud备份时不会备份此目录。
6. **xxx.app (应用程序包)**：包含程序中的nib⽂件、图片、音频等资源。

综述：

1. **Document** : 存储用户数据,需要备份的信息
2. **Library/Caches** : 存储缓存文件,程序专用的支持文件
3. **Library/Preferences** : 存储应用程序的偏好设置⽂件
4. **tmp** : 存储临时文件,比如:下载的zip包,解压后的再删除
5. **xxx.app** : 应用程序包
6. iTunes在与iPhone同步时，会备份 `Documents` 和 `Preferences` 目录下的文件 。

# 路径获取

## 获取程序的根目录：

NSString \*homeDirectory = NSHomeDirectory();

NSLog(@"homeDirectory = %@",homeDirectory);

## 获取document目录：

NSString \*documentPath = [NSSearchPathForDirectoriesInDomains(NSDocumentDirectory, NSUserDomainMask, YES) lastObject];

## 获取Caches目录：

NSString \*cachesPath = [NSSearchPathForDirectoriesInDomains(NSCachesDirectory, NSUserDomainMask, YES) lastObject];

## 获取Library目录：

NSString \*libraryPath = [NSSearchPathForDirectoriesInDomains(NSLibraryDirectory, NSUserDomainMask, YES) lastObject];

## 获取Tmp目录：

NSString \*tmpDirectory = NSTemporaryDirectory();

NSLog(@"tmpDirectory = %@",tmpDirectory);

## 获取xxx.app程序包：

NSBundle \*myBundle = [NSBundle mainBundle];

NSLog(@"app程序包 = %@",myBundle);