

前提

- 配置环境变量

`ANDROID_HOME=E:\AndroidSDK`

- 在path中加入

`%ANDROID_HOME%\platform-tools`

`%ANDROID_HOME%\tools`

ADB 命令

本章大纲

- ADB介绍
- ADB原理
- 常见ADB命令

命令行调试adb

- adb (Android Debug Bridge) 是Android的主要调试工具，它可以通过网络或者USB连接真实设备，也可以连接虚拟设备。
adb就是连接android手机与pc的**桥梁**，可以让用户在电脑上对手机进行全面的操作。
- 借助adb工具，可以管理设备或手机模拟器的状态。还可以进行很多手机操作，如安装软件、系统升级、运行shell命令等待。

注意：adb.exe位于sdk开发platform-tools目录下

命令行调试adb

- 它是一个**客户端/服务器架构**的命令行工具，其中客户端是用来操作的电脑，服务器端是android设备。由三部分组成
 - adb客户端：运行于PC的客户端，通过命令行的方式启动
 - adb服务器：用于开发的机器上作为后台进程运行的服务器，该服务器负责管理**客户端**与运行于模拟器或设备上的**守护进程**之间的通信
 - adb daemon（守护进程）：一个以后台进程的形式运行于模拟器或手机设备上的守护程序。



本章大纲

- ADB介绍
- ADB原理
- 常见ADB命令

ADB工作原理

- 当用户启动一个adb客户端，客户端首先确认是否已有一个adb服务进程在运行。如果没有，则启动服务进程。当服务器运行，adb服务器就会绑定本地的TCP端口5037并监听adb客户端发来的命令，所有的adb客户端都是用端口5037与adb服务器进行对话的。
- 接着服务器将所有运行中的模拟器或设备实例建立连接。它通过扫描所有5555到5585范围内的奇数端口来定位所有的模拟器或设备。一旦服务器找到了adb守护程序，它将建立一个到该端口的连接。请注意任何设备和模拟器实例会取得两个连接的端口，一个偶数端口用来控制与控制台的连接，和一个奇数端口用来控制与adb连接。

可能存在的报错

- ADB server didn't ACK
- cannot bind to 127.0.0.1:5037

```
C:\Users\think>adb devices
List of devices attached
adb server version (32) doesn't match this client (36); killing...
error: could not install *smartsocket* listener: cannot bind to 127.0.0.1:5037:
通常每个套接字地址<协议/网络地址/端口>只允许使用一次。 (10048)
could not read ok from ADB Server
* failed to start daemon *
error: cannot connect to daemon
```

- 解决办法：端口被占用
输入netstat -ano | findstr "5037" |more, 查看那些程序占用了5037端口

本章大纲

- ADB介绍
- ADB原理
- 常见ADB命令

常见adb命令

- 显示当前运行的虚拟设备: adb devices
- 查看adb所支持的所有命令: adb -help
- 查看adb的版本序列号: adb version
- 在模拟器安装软件: adb install -r *.apk
- 在模拟器卸载软件: adb uninstall 包名
- 向模拟器中传输文件: adb push
- 从模拟器中获取文件: adb pull
- 进入模拟器的shell模式: adb shell
- 重启server: adb start-server
- 关闭server: adb kill-server

```
C:\Users\think>adb install E:\ContactManager.apk
[100%] /data/local/tmp/ContactManager.apk
      pkg: /data/local/tmp/ContactManager.apk
Success
C:\Users\think>_
```

常见adb命令

- adb devices 可能包含三种状态
 - device: 表示设备或模拟器已经连接到服务器上
 - offline: 表示设备没有连接到服务器上或者没有响应
 - no device: 表示没有设备连接。

常见adb命令

`adb install -r *.apk`

- 只连接一个设备

– `adb install` 安装包路径

- 连接多个设备，需要指定一个设备

– `adb -s 设备序列号 install` 安装包路径

- 覆盖安装，保留数据

– `adb -s 设备序列号 install -r` 安装包路径

例：`adb -s 192.168.56.101:5555 install -r E:/test.apk`

常见adb命令

- adb -s 设备名称 uninstall 应用的包名
- 保留配置和缓存文件

adb -s 设备名称 shell pm uninstall -k 应用的包名

adb -s 192.168.56.101:5555 uninstall

com.example.think.today

或者使用两条命令完成:

adb shell

pm uninstall -k 应用的包名

adb -s 192.168.56.101:5555 uninstall

com.example.think.today

常见adb命令

- adb push PC源文件 设备的路径
- adb pull 设备文件 PC的路径

例:

```
adb push d:/demo/a.bat /data/
```

```
adb -s 192.168.56.101:5555 pull /data/a.bat d:/demo
```

常见adb命令

- `adb start-server` 启动adb服务进程
- `adb kill-server` 终止adb服务进程

常见adb命令

- 使用adb启动停止一个app

- -W 等待启动完成

- -S强制停止再次启动

- adb shell am start -W -S package/activity

- adb shell am force-stop package

查看包名以及启动程序的类名

```
aapt dump badging d:\group-612_0-meituan.apk
```

```
adb shell am start -W -S com.sankuai.meituan/.acvity.Welcome
```


常见adb命令

- `adb logcat` 将日志数据输出到屏幕上
- `adb logcat > E:/demo/test.log` (查找包名)

优先级按照从低到高顺利排列

V Verbose

D Debug

I Info

W Warning

E Error

F Fatal

S Silent

下面的过滤语句指显示优先级为warning或更高的日志信息

```
adb logcat *:W
```

adb shell

- 安卓系统支持常见的Linux命令, /system/bin
- `adb shell`
- `adb shell ls`
- `adb shell pm list packages` 打印所有包信息
- `adb shell pm list packages -f keyword` (所有包含关键字)
- `adb shell pm list packages -3 keyword` (第三方安装的)
- `adb shell pm list packages -i keyword` (已经安装的包)

adb shell

- 获取CPU、内存状态
 - adb shell dumpsys meminfo package
 - adb shell dumpsys cpuinfo | findstr package

截屏、录屏

- `adb shell screencap /data/local/test1.jsp`
- `adb shell screenrecord /data/local/test2.mp4`
(模拟器不支持)

adb shell

- adb shell 通过远程shell命令来控制模拟器/设备实例
- 常用的shell 命令，如下
 - ✓ cd——改变当前目录
 - ✓ pwd——查看当前所在目录完整路径
 - ✓ ls——查看目录或者文件的属*，列举出任一目录下面的文件
 - ✓ mkdir——建立目录
 - ✓ cp——拷贝文件

adb shell

- 常用的shell 命令，如下
 - ✓ rm——删除文件和目录
 - ✓ mv——移走目录或者改文件名
 - ✓ chmod/chown——权限修改
 - ✓ df——命令用来检查文件系统的磁盘空间占用情况
 - ✓ top——查看系统资源
 - ✓ touch——创建文件

详细请参考： /system/bin