**root权限运行adbd**

one.以 root 权限运行 adbd  
adb 的运行原理是 PC 端的 adb server 与手机端的守护进程 adbd 建立连接，然后 PC 端的 adb client 通过 adb server 转发命令，adbd 接收命令后解析运行。

所以如果 adbd 以普通权限执行，有些需要 root 权限才能执行的命令无法直接用adb xxx 执行。这时可以 adb shell 然后 su 后执行命令，也可以让 adbd 以 root 权限执行，这个就能随意执行高权限命令了。

命令：  
adb root  
正常输出：  
restarting adbd as root  
现在再运行 adb shell，看看命令行提示符是不是变成 # 了？

有些手机 root 后也无法通过 adb root 命令让 adbd 以 root 权限执行，比如三星的部分机型，会提示 adbd cannot run as root in production builds，此时可以先安装 adbd Insecure，然后 adb root 试试。

相应地，如果要恢复 adbd 为非 root 权限的话，可以使用 adb unroot 命令。

指定 adb server 的网络端口命令：  
adb -P start-server  
默认端口为 5037。  
USB 连接  
通过 USB 连接来正常使用 adb 需要保证几点：

硬件状态正常。

包括 Android 设备处于正常开机状态，USB 连接线和各种接口完好。

Android 设备的开发者选项和 USB 调试模式已开启。

可以到「设置」-「开发者选项」-「Android 调试」查看。

如果在设置里找不到开发者选项，那需要通过一个彩蛋来让它显示出来：在「设置」-「关于手机」连续点击「版本号」7 次。

设备驱动状态正常。

这一点貌似在 Linux 和 Mac OS X 下不用操心，在 Windows 下有可能遇到需要安装驱动的情况，确认这一点可以右键「计算机」-「属性」，到「设备管理器」里查看相关设备上是否有黄色感叹号或问号，如果没有就说明驱动状态已经好了。否则可以下载一个手机助手类程序来安装驱动先。

通过 USB 线连接好电脑和设备后确认状态。

adb devices  
如果能看到

xxxxxx device  
说明连接成功。

无线连接  
除了可以通过 USB 连接设备与电脑来使用 adb，也可以通过无线连接——虽然连接过程中也有需要使用 USB 的步骤，但是连接成功之后你的设备就可以在一定范围内摆脱 USB 连接线的限制啦！

操作步骤：  
1.将 Android 设备与将运行 adb 的电脑连接到同一个局域网，比如连到同一个 WiFi。  
2.将设备与电脑通过 USB 线连接。  
应确保连接成功（可运行 adb devices 看是否能列出该设备）。  
3.让设备在 5555 端口监听 TCP/IP 连接：  
adb tcpip 5555  
4.断开 USB 连接。  
5.找到设备的 IP 地址。  
一般能在「设置」-「关于手机」-「状态信息」-「IP地址」找到。  
6.通过 IP 地址连接设备。  
adb connect   
这里的 就是上一步中找到的设备 IP 地址。  
7.确认连接状态。  
adb devices  
如果能看到  
:5555 device  
说明连接成功。

如果连接不了，请确认 Android 设备与电脑是连接到了同一个 WiFi，然后再次执行 adb connect 那一步；  
如果还是不行的话，通过 adb kill-server 重新启动 adb 然后从头再来一次试试。  
断开无线连接命令：  
adb disconnect

two.重新挂载 system 分区为可写  
注：需要 root 权限。  
/system 分区默认挂载为只读，但有些操作比如给 Android 系统添加命令、删除自带应用等需要对 /system 进行写操作，所以需要重新挂载它为可读写。

步骤：  
1.进入 shell 并切换到 root 用户权限。  
命令：  
adb shell  
su  
2.查看当前分区挂载情况。  
命令：  
mount  
输出示例：  
rootfs / rootfs ro,relatime 0 0  
tmpfs /dev tmpfs rw,seclabel,nosuid,relatime,mode=755 0 0  
devpts /dev/pts devpts rw,seclabel,relatime,mode=600 0 0  
proc /proc proc rw,relatime 0 0  
sysfs /sys sysfs rw,seclabel,relatime 0 0  
找到其中我们关注的带 /system 的那一行：  
/dev/block/platform/msm\_sdcc.1/by-name/system /system ext4 ro,seclabel,relatime,data=ordered 0 0  
重新挂载。  
3.命令：  
mount -o remount,rw -t yaffs2 /dev/block/platform/msm\_sdcc.1/by-name/system /system  
这里的 /dev/block/platform/msm\_sdcc.1/by-name/system 就是我们从上一步的输出里得到的文件路径。  
如果输出没有提示错误的话，操作就成功了，可以对 /system 下的文件为所欲为了。

three.查看连接过的 WiFi 密码  
注：需要 root 权限。  
命令：  
adb shell  
su  
cat /data/misc/wifi/\*.conf  
输出示例：  
network={  
ssid="TP-LINK\_9DFC"  
scan\_ssid=1  
psk="123456789"  
key\_mgmt=WPA-PSK  
group=CCMP TKIP  
auth\_alg=OPEN  
sim\_num=1  
priority=13893  
}

network={  
ssid="TP-LINK\_F11E"  
psk="987654321"  
key\_mgmt=WPA-PSK  
sim\_num=1  
priority=17293  
}  
ssid 即为我们在 WLAN 设置里看到的名称，psk 为密码，key\_mgmt 为安全加密方式。

four.设置系统日期和时间  
注：需要 root 权限。  
命令：  
adb shell  
su  
date -s 20160823.131500  
表示将系统日期和时间更改为 2016 年 08 月 23 日 13 点 15 分 00 秒。

Five.重启手机命令：adb reboot  
six.检测设备是否已 root  
命令：  
adb shell  
su  
此时命令行提示符是 $ 则表示没有 root 权限，是 # 则表示已 root。

seven.使用 Monkey 进行压力测试  
Monkey 可以生成伪随机用户事件来模拟单击、触摸、手势等操作，可以对正在开发中的程序进行随机压力测试。  
简单用法：  
adb shell monkey -p -v 500  
adb shell monkey -p com.rct.gebao -v 500  
表示向 指定的应用程序发送 500 个伪随机事件。

eight.刷机相关命令  
重启到 Recovery 模式命令：adb reboot recovery  
重启到 Fastboot 模式命令：adb reboot bootloader  
通过 sideload 更新系统:  
如果我们下载了 Android 设备对应的系统更新包到电脑上，那么也可以通过 adb 来完成更新。  
以 Recovery 模式下更新为例：  
重启到 Recovery 模式。  
命令：  
adb reboot recovery  
在设备的 Recovery 界面上操作进入 Apply update-Apply from ADB。  
注：不同的 Recovery 菜单可能与此有差异，有的是一级菜单就有 Apply update from ADB。  
通过 adb 上传和更新系统。  
命令：  
adb sideload <path-to-update.zip>