ADB (Android Debug Bridge)

说明：下面一些命令需要有root权限才能执行成功

快速启动dos窗口执行adb：  
1. adb.exe所在路径添加到系统环境变量中  
2. 配置快捷键启动dos  
进入C:\WINDOWS\system32目录下，找到cmd.exe.  
右击菜单 "发送到" -> 桌面快捷方式。  
在桌面上右击"快捷方式 到 cmd.exe" -> "属性" -> "快捷方式"页   
-> 光标高亮"快捷键" -> 按下自定义快捷键 (如：Ctrl + Alt + Z)  
  
任何情况下，按下Ctrl + Alt + Z启动dos窗口就可以执行adb命令了  
  
-----------查看设备连接状态 系列-----------  
adb get-serialno 获取设备的ID和序列号serialNumber  
adb devices 查询当前计算机上连接那些设备（包括模拟器和手机），输出格式: [serialNumber] [state]  
adb get-state 查看模拟器/设施的当前状态.  
  
说明：  
序列号[serialNumber]——由adb创建的一个字符串，这个字符串通过自己的控制端口<type>-<consolePort>   
唯一地识别一个模拟器/设备实例。一个序列号的例子： emulator-5554  
  
-----------发送命令到设备 系列-----------  
adb [-d|-e|-s <serialNumber>] <command>  
-d 发送命令给usb连接的设备  
-e 发送命令到模拟器设备  
-s <serialNumber> 发送命令到指定设备  
  
如启动手机设备shell: adb -d shell   
  
adb forward <local> <remote>发布端口,可以设置任意的端口号，  
做为主机向模拟器或设备的请求端口。如：adb forward tcp:5555 tcp:8000  
  
adb reboot 重启手机  
adb remount 将system分区重新挂载为可读写分区  
adb kill-server 终止adb服务进程  
adb start-server 重启adb服务进程  
adb root 已root权限重启adb服务  
adb wait-for-device 在模拟器/设备连接之前把命令转载在adb的命令器中  
adb jdwp 查看指定的设施的可用的JDWP信息.   
可以用 forward jdwp:<pid> 端口映射信息来连接指定的JDWP进程.例如：   
adb forward tcp:8000 jdwp:472   
jdb -attach localhost:8000   
  
adb shell am 命令可以启动应用程序  
  
adb shell input text <string> 向设备输入文本（光标所在的文本框）  
adb shell input keyevent <event\_code> 向设备发送按键事件  
如：   
在编辑短信时，往文本框输入文本：adb shell input text "hello"   
向手机发送键值回Home：adb shell input keyevent 3  
event\_code 参考view/KeyEvent.java中的 KEYCODE\_\*   
 public static final int KEYCODE\_SOFT\_LEFT = 1;  
 public static final int KEYCODE\_SOFT\_RIGHT = 2;  
 public static final int KEYCODE\_HOME = 3;  
 public static final int KEYCODE\_BACK = 4;  
 public static final int KEYCODE\_CALL = 5;  
 public static final int KEYCODE\_ENDCALL = 6;  
  
-----------安装卸载 系列-----------   
adb install [-l] [-r] <file> - push this package file to the device and install it  
 ('-l' means forward-lock the app)  
 ('-r' means reinstall the app, keeping its data)  
adb uninstall [-k] <package> - remove this app package from the device  
 ('-k' means keep the data and cache directories)  
如：  
adb install d:\hello.apk  
adb unstall com.huawei.hello  
说明：如果带-r选项重新安装apk时，安装在 /data/local/tmp/目录下，手机重启后还是使用原来的apk.  
  
-----------文件操作 系列-----------   
adb push <local> <remote> - copy file/dir to device  
adb pull <remote> <local> - copy file/dir from device  
  
-----------基本linux shell命令 系列-----------   
adb shell [command]   
ls 列出目录下的文件和文件夹  
cd 切换目录  
rm 删除目录和文件  
cat 查看文件内容  
ps 可以看那个进程再跑  
ps -x [PID] 查看单个进程的状态  
top 可以看那个进程的占用率最高  
su 切换到root用户  
kill [pid] 杀死一个进程  
chmod 777 <file> 修改该文件为可执行权限  
  
详细使用情况可以登录一台Linux服务器在shell下查看帮助手册, man <command>  
  
-----------查看系统状态和信息 系列-----------  
adb shell procrank 查询各进程内存使用情况  
adb shell service list 查看services信息  
adb shell cat /proc/meminfo 查看当前的内存情况  
adb shell cat /proc/cpuinfo 查看CPU信息（硬件）  
adb shell cat /proc/iomem 查看IO内存分区  
  
adb shell getprop 列出系统所有属性  
adb shell getprop | findstr "gsm" 列出包含gsm的属性  
adb shell setprop <key> <value> 修改系统属性  
  
adb shell sqlite3 可以执行sql语句查看数据库信息， 具体使用情况待调查  
  
-----------Log 系列-----------   
adb logcat [ <filter-spec> ] - View device log  
  
1~~~~~~~~~~~查看可用日志缓冲区:  
adb logcat -b radio — 查看缓冲区的相关的信息.   
adb logcat -b events — 查看和事件相关的的缓冲区.   
adb logcat -b main — 查看主要的日志缓冲区   
  
2~~~~~~~~~~~过滤日志输出:  
过滤器语句按照下面的格式描tag:priority ... , tag 表示是标签, priority 是表示标签的报告的最低等级  
adb logcat \*:W 显示优先级为warning或更高的日志信息  
adb logcat ActivityManager:I MyApp:D \*:S  
  
日志的标签是系统部件原始信息的一个简要的标志。（比如：“View”就是查看系统的标签）.  
优先级有下列集中，是按照从低到高顺利排列的:   
V — Verbose (lowest priority)  
D — Debug  
I — Info  
W — Warning  
E — Error  
F — Fatal  
S — Silent (highest priority, on which nothing is ever printed)   
  
如果你电脑上运行logcat ，相比在远程adbshell端，你还可以为环境变量ANDROID\_LOG\_TAGS :输入一个参数来设置默认的过滤   
export ANDROID\_LOG\_TAGS="ActivityManager:I MyApp:D \*:S"  
需要注意的是ANDROID\_LOG\_TAGS 过滤器如果通过远程shell运行logcat 或用adb shell logcat 来运行模拟器/设备不能输出日志.  
  
3~~~~~~~~~~~控制日志输出格式:  
日志信息包括了许多元数据域包括标签和优先级。可以修改日志的输出格式，所以可以显示出特定的元数据域。可以通过 -v 选项得到格式化输出日志的相关信息.   
  
brief — Display priority/tag and PID of originating process (the default format).  
process — Display PID only.  
tag — Display the priority/tag only.  
thread — Display process:thread and priority/tag only.  
raw — Display the raw log message, with no other metadata fields.  
time — Display the date, invocation time, priority/tag, and PID of the originating process.  
long — Display all metadata fields and separate messages with a blank lines.   
当启动了logcat ，你可以通过-v 选项来指定输出格式:  
  
[adb] logcat [-v <format>]  
下面是用 thread 来产生的日志格式:   
  
adb logcat -v thread  
需要注意的是你只能-v 选项来规定输出格式 option.  
  
4~~~~~~~~~~~Logcat命令列表  
-b <buffer> 加载一个可使用的日志缓冲区供查看，比如event 和radio . 默认值是main 。具体查看Viewing Alternative Log Buffers.   
-c 清楚屏幕上的日志.   
-d 输出日志到屏幕上.   
-f <filename> 指定输出日志信息的<filename> ，默认是stdout .   
-g 输出指定的日志缓冲区，输出后退出.   
-n <count> 设置日志的最大数目<count> .，默认值是4，需要和 -r 选项一起使用。   
-r <kbytes> 每<kbytes> 时输出日志，默认值为16，需要和-f 选项一起使用.   
-s 设置默认的过滤级别为silent.   
-v <format> 设置日志输入格式，默认的是brief 格式，要知道更多的支持的格式，参看Controlling Log Output Format   
  
adb bugreport - return all information from the device  
 that should be included in a bug report.  
  
adb shell dmesg 查询内核缓冲区信息  
adb shell dumpstate 各类信息，比如进程信息，内存信息，进程是否异常，kernnel的log等  
adb shell dumpcrash  
adb shell dumpsys 查询所有service的状态  
  
-----------其他 -----------   
  
模拟器使用镜像sdcard  
用SDK里的mksdcard工具来创建FAT32磁盘镜像并在模拟器启动时加载它。这样创建镜像：? mksdcard <size> <file>,  
比如我要创建一个64M的SD卡模拟文件，文件路径是在D:\workspace\sdcard.img  
 mksdcard 64000000 D:\workspace\sdcard.img  
  
Emulator –sdcard D:\workspace\sdcard.img  
或者在eclipse的run菜单的open run dialog对话框中配置启动参数。  
  
#top   
Usage: top [ -m max\_procs ] [ -n iterations ] [ -d delay ] [ -s sort\_column ] [ -t ] [ -h ]  
 -m num Maximum number of processes to display.  
 -n num Updates to show before exiting.  
 -d num Seconds to wait between updates.  
 -s col Column to sort by (cpu,vss,rss,thr).  
 -t Show threads instead of processes.  
 -h Display this help screen.  
  
 \*\*\*\*\*\*\*\*\* simple selection \*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\* selection by list \*\*\*\*\*\*\*\*\*  
-A all processes -C by command name  
-N negate selection -G by real group ID (supports names)  
-a all w/ tty except session leaders -U by real user ID (supports names)  
-d all except session leaders -g by session OR by effective group name  
-e all processes -p by process ID  
T all processes on this terminal -s processes in the sessions given  
a all w/ tty, including other users -t by tty  
g OBSOLETE -- DO NOT USE -u by effective user ID (supports names)  
r only running processes U processes for specified users  
x processes w/o controlling ttys t by tty  
\*\*\*\*\*\*\*\*\*\*\* output format \*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\* long options \*\*\*\*\*\*\*\*\*\*\*  
-o,o user-defined -f full --Group --User --pid --cols --ppid  
-j,j job control s signal --group --user --sid --rows --info  
-O,O preloaded -o v virtual memory --cumulative --format --deselect  
-l,l long u user-oriented --sort --tty --forest --version  
-F extra full X registers --heading --no-heading --context  
 \*\*\*\*\*\*\*\*\* misc options \*\*\*\*\*\*\*\*\*  
-V,V show version L list format codes f ASCII art forest  
-m,m,-L,-T,H threads S children in sum -y change -l format  
-M,Z security data c true command name -c scheduling class  
-w,w wide output n numeric WCHAN,UID -H process hierarchy  
  
netstat -ano 查看网络连状态  
 显示协议统计信息和当前 TCP/IP 网络连接。  
  
NETSTAT [-a] [-b] [-e] [-n] [-o] [-p proto] [-r] [-s] [-v] [interval]  
  
 -a 显示所有连接和监听端口。  
 -b 显示包含于创建每个连接或监听端口的  
 可执行组件。在某些情况下已知可执行组件  
 拥有多个独立组件，并且在这些情况下  
 包含于创建连接或监听端口的组件序列  
 被显示。这种情况下，可执行组件名  
 在底部的 [] 中，顶部是其调用的组件，  
 等等，直到 TCP/IP 部分。注意此选项  
 可能需要很长时间，如果没有足够权限  
 可能失败。  
 -e 显示以太网统计信息。此选项可以与 -s  
 选项组合使用。  
 -n 以数字形式显示地址和端口号。  
 -o 显示与每个连接相关的所属进程 ID。  
 -p proto 显示 proto 指定的协议的连接；proto 可以是  
 下列协议之一: TCP、UDP、TCPv6 或 UDPv6。  
 如果与 -s 选项一起使用以显示按协议统计信息，proto 可以是下列协议之一:  
 IP、IPv6、ICMP、ICMPv6、TCP、TCPv6、UDP 或 UDPv6。  
 -r 显示路由表。  
 -s 显示按协议统计信息。默认地，显示 IP、  
 IPv6、ICMP、ICMPv6、TCP、TCPv6、UDP 和 UDPv6 的统计信息；  
 -p 选项用于指定默认情况的子集。  
 -v 与 -b 选项一起使用时将显示包含于  
 为所有可执行组件创建连接或监听端口的  
 组件。  
 interval 重新显示选定统计信息，每次显示之间  
 暂停时间间隔(以秒计)。按 CTRL+C 停止重新  
 显示统计信息。如果省略，netstat 显示当前  
 配置信息(只显示一次)  
  
pm  
usage: pm [list|path|install|uninstall]  
 pm list packages [-f]  
 pm list permission-groups  
 pm list permissions [-g] [-f] [-d] [-u] [GROUP]  
 pm list instrumentation [-f] [TARGET-PACKAGE]  
 pm list features  
 pm path PACKAGE  
 pm install [-l] [-r] [-t] [-i INSTALLER\_PACKAGE\_NAME] PATH  
 pm uninstall [-k] PACKAGE  
 pm enable PACKAGE\_OR\_COMPONENT  
 pm disable PACKAGE\_OR\_COMPONENT  
  
The list packages command prints all packages. Options:  
 -f: see their associated file.  
  
The list permission-groups command prints all known  
permission groups.  
  
The list permissions command prints all known  
permissions, optionally only those in GROUP. Options:  
 -g: organize by group.  
 -f: print all information.  
 -s: short summary.  
 -d: only list dangerous permissions.  
 -u: list only the permissions users will see.  
  
The list instrumentation command prints all instrumentations,  
or only those that target a specified package. Options:  
 -f: see their associated file.  
  
The list features command prints all features of the system.  
  
The path command prints the path to the .apk of a package.  
  
The install command installs a package to the system. Options:  
 -l: install the package with FORWARD\_LOCK.  
 -r: reinstall an exisiting app, keeping its data.  
 -t: allow test .apks to be installed.  
 -i: specify the installer package name.  
  
The uninstall command removes a package from the system. Options:  
 -k: keep the data and cache directories around.  
after the package removal.  
  
The enable and disable commands change the enabled state of  
a given package or component (written as "package/class").  
  
查看stdout 和stderr  
在默认状态下，Android系统有stdout 和 stderr (System.out和System.err )输出到/dev/null ，  
在运行Dalvik VM的进程中，有一个系统可以备份日志文件。在这种情况下，系统会用stdout 和stderr 和优先级 I.来记录日志信息  
  
通过这种方法指定输出的路径，停止运行的模拟器/设备，然后通过用setprop 命令远程输入日志  
  
$ adb shell stop  
$ adb shell setprop log.redirect-stdio true  
$ adb shell start系统直到你关闭模拟器/设备前设置会一直保留，可以通过添加/data/local.prop 可以使用模拟器/设备上的默认设置  
  
UI/软件 试验程序 Monkey  
当Monkey程序在模拟器或设备运行的时候，如果用户出发了比如点击，触摸，手势或一些系统级别的事件的时候，  
它就会产生随机脉冲，所以可以用Monkey用随机重复的方法去负荷测试你开发的软件.  
最简单的方法就是用用下面的命令来使用Monkey，这个命令将会启动你的软件并且触发500个事件.  
  
$ adb shell monkey -v -p your.package.name 500  
更多的关于命令Monkey的命令的信息，可以查看UI/Application Exerciser Monkey documentation pag