Android

Motorola Milestone (EU) bzw. Droid (US)

Vorbereitung(en)

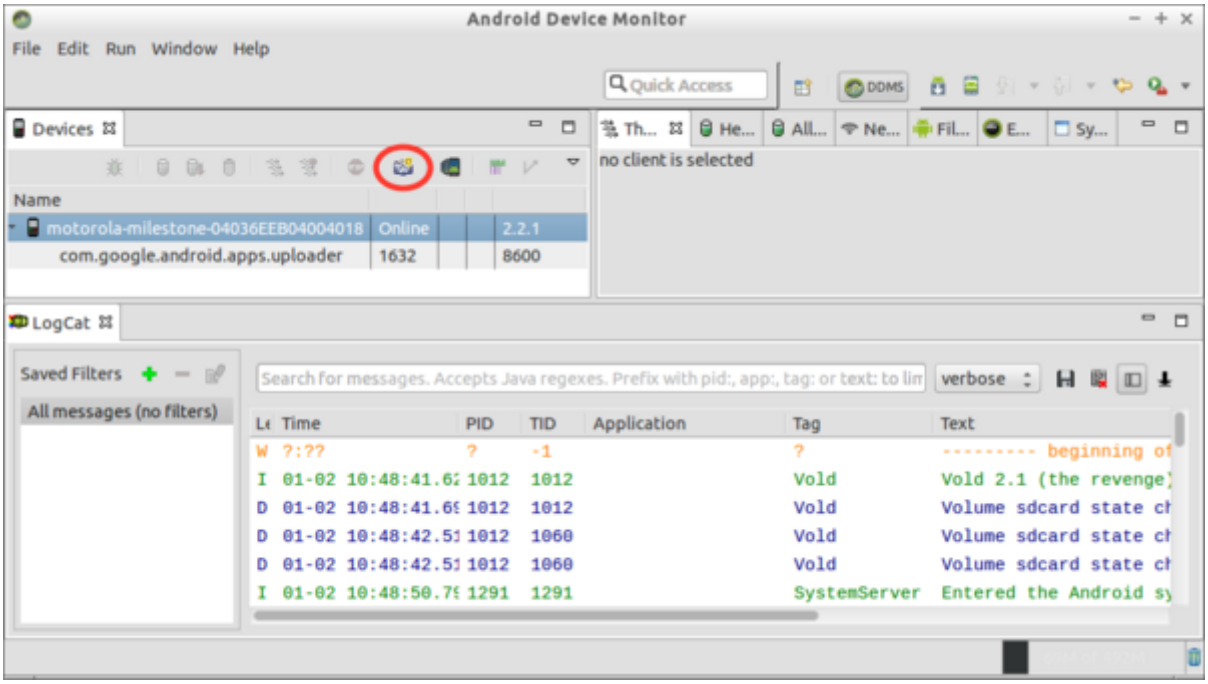
- das ausgediente Motorola Milestone vom Schwiegervater ausleihen *(erledigt)*
- „USB-Debugging“ + „Aktiv lassen“ aktivieren! *(erledigt)*
- Santoku Linux (VirtualBox VM) installieren *(erledigt)*

(nachträgliche) Dokumentation

Santoku Linux enthält von Haus aus sowohl den „Dalvic Debug Monitor“ (deprecated) als auch den „Android Device Monitor“.

Zuletzt genannter kann z. B. aus einem Terminal wie folgt gestartet werden:

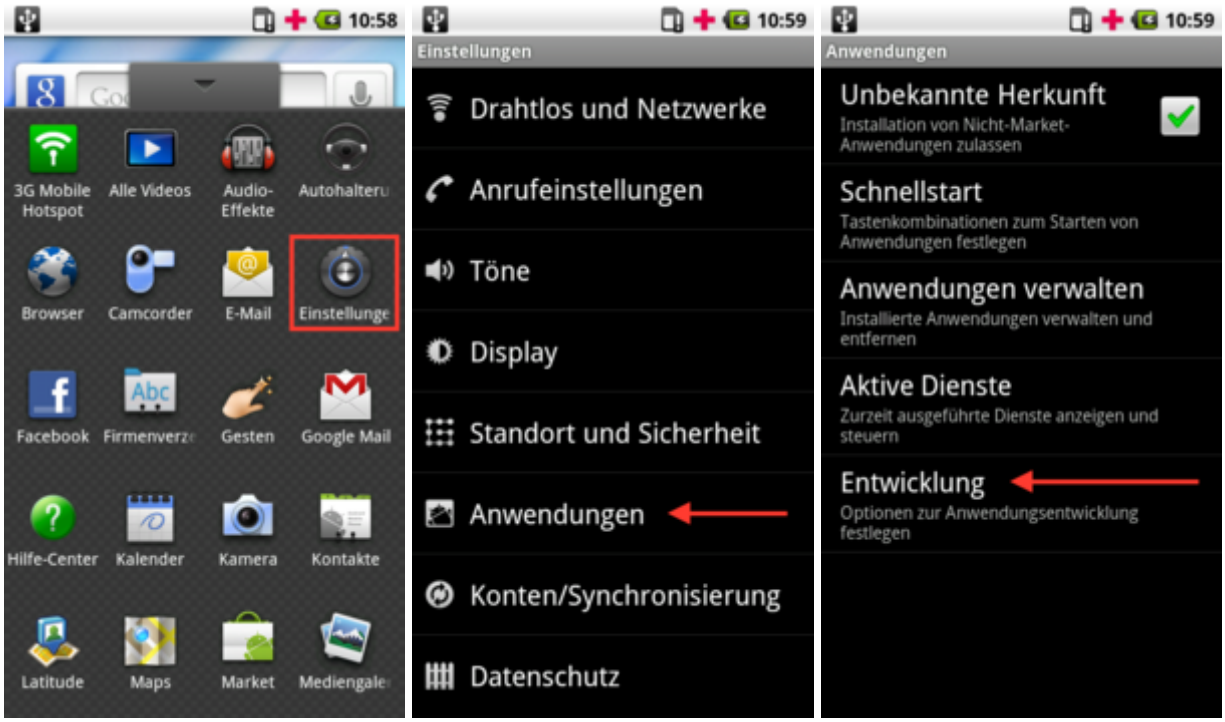
```
user@Santoku:~$ monitor
```



Das „Device Screen Capture“-Fenster kann über die Fotoapperat-Schaltfläche (den habe ich mit einem roten Kreis hervorgehoben) aufgerufen werden:

Immer bevor ein Screencapture vom Mobiltelefon erstellt werden soll, muss mit einem Klick auf die Schaltfläche „Refresh“ (blau eingefärbte) die Vorschau aktualisiert werden.

Mit einem Klick auf „Save“ kann der Screencapture gespeichert werden. Hier ein paar Beispiele („USB-Debugging“ + „Aktiv lassen“ aktivieren):



Wie man die beiden Anwendungen beendet, sollte selbsterklärend sein!?

ADB

Per Android Debug Bridge eine Kommandozeile auf dem Mobiltelefon öffnen:

```
user@Santoku:~$ adb shell
```

MTD

Eine grobe Übersicht über die „Partitionierung“ des Flash-Speichers ausgeben lassen (oldschool, deprecated aber kompakt):

```
$ cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00180000 00020000 "pds"
mtd1: 00060000 00020000 "cid"
mtd2: 00060000 00020000 "misc"
mtd3: 00380000 00020000 "boot"
mtd4: 00480000 00020000 "recovery"
mtd5: 008c0000 00020000 "cdrom"
mtd6: 0afa0000 00020000 "system"
mtd7: 06a00000 00020000 "cache"
mtd8: 0c520000 00020000 "userdata"
mtd9: 00180000 00020000 "cust"
mtd10: 00200000 00020000 "kpanic"
```

Daten des Benutzers sollten sich ausschließlich in „userdata“ (mtd8) und „cache“ (mtd7) befinden!?

Es geht auch etwas umfangreicher (newworld, aktuell, ausführlich aber umständlicher):

```
$ for int in 0 1 2 3 4 5 6 7 8 9 10 ; do for file in uevent dev type flags
size erasesize writesize subpagesize oobsize numeraseregions name ; do echo
"/sys/class/mtd/mtd${int}/${file}:" ; cat "/sys/class/mtd/mtd${int}/${file}"
; done ; echo ; done
/sys/class/mtd/mtd0/uevent:
MAJOR=90
MINOR=0
DEVNAME=mtd0
DEVTYPE=mtd
/sys/class/mtd/mtd0/dev:
90:0
/sys/class/mtd/mtd0/type:
nand
/sys/class/mtd/mtd0/flags:
0x400
/sys/class/mtd/mtd0/size:
1572864
/sys/class/mtd/mtd0/erasesize:
131072
/sys/class/mtd/mtd0/writesize:
2048
/sys/class/mtd/mtd0/subpagesize:
512
/sys/class/mtd/mtd0/oobsize:
64
/sys/class/mtd/mtd0/numeraseregions:
0
/sys/class/mtd/mtd0/name:
pds

/sys/class/mtd/mtd1/uevent:
MAJOR=90
MINOR=2
DEVNAME=mtd1
DEVTYPE=mtd
/sys/class/mtd/mtd1/dev:
90:2
/sys/class/mtd/mtd1/type:
nand
/sys/class/mtd/mtd1/flags:
0x400
/sys/class/mtd/mtd1/size:
393216
/sys/class/mtd/mtd1/erasesize:
131072
/sys/class/mtd/mtd1/writesize:
2048
/sys/class/mtd/mtd1/subpagesize:
512
```

```
/sys/class/mtd/mtd1/oobsize:
64
/sys/class/mtd/mtd1/numeraseregions:
0
/sys/class/mtd/mtd1/name:
cid

/sys/class/mtd/mtd2/uevent:
MAJOR=90
MINOR=4
DEVNAME=mtd2
DEVTYPE=mtd
/sys/class/mtd/mtd2/dev:
90:4
/sys/class/mtd/mtd2/type:
nand
/sys/class/mtd/mtd2/flags:
0x400
/sys/class/mtd/mtd2/size:
393216
/sys/class/mtd/mtd2/erasesize:
131072
/sys/class/mtd/mtd2/writesize:
2048
/sys/class/mtd/mtd2/subpagesize:
512
/sys/class/mtd/mtd2/oobsize:
64
/sys/class/mtd/mtd2/numeraseregions:
0
/sys/class/mtd/mtd2/name:
misc

/sys/class/mtd/mtd3/uevent:
MAJOR=90
MINOR=6
DEVNAME=mtd3
DEVTYPE=mtd
/sys/class/mtd/mtd3/dev:
90:6
/sys/class/mtd/mtd3/type:
nand
/sys/class/mtd/mtd3/flags:
0x0
/sys/class/mtd/mtd3/size:
3670016
/sys/class/mtd/mtd3/erasesize:
131072
/sys/class/mtd/mtd3/writesize:
2048
/sys/class/mtd/mtd3/subpagesize:
```

```
512
/sys/class/mtd/mtd3/oobsize:
64
/sys/class/mtd/mtd3/numeraseregions:
0
/sys/class/mtd/mtd3/name:
boot

/sys/class/mtd/mtd4/uevent:
MAJOR=90
MINOR=8
DEVNAME=mtd4
DEVTYPE=mtd
/sys/class/mtd/mtd4/dev:
90:8
/sys/class/mtd/mtd4/type:
nand
/sys/class/mtd/mtd4/flags:
0x400
/sys/class/mtd/mtd4/size:
4718592
/sys/class/mtd/mtd4/erasesize:
131072
/sys/class/mtd/mtd4/writesize:
2048
/sys/class/mtd/mtd4/subpagesize:
512
/sys/class/mtd/mtd4/oobsize:
64
/sys/class/mtd/mtd4/numeraseregions:
0
/sys/class/mtd/mtd4/name:
recovery

/sys/class/mtd/mtd5/uevent:
MAJOR=90
MINOR=10
DEVNAME=mtd5
DEVTYPE=mtd
/sys/class/mtd/mtd5/dev:
90:10
/sys/class/mtd/mtd5/type:
nand
/sys/class/mtd/mtd5/flags:
0x400
/sys/class/mtd/mtd5/size:
9175040
/sys/class/mtd/mtd5/erasesize:
131072
/sys/class/mtd/mtd5/writesize:
2048
```

```
/sys/class/mtd/mtd5/subpagesize:
512
/sys/class/mtd/mtd5/oobsize:
64
/sys/class/mtd/mtd5/numeraseregions:
0
/sys/class/mtd/mtd5/name:
cdrom

/sys/class/mtd/mtd6/uevent:
MAJOR=90
MINOR=12
DEVNAME=mtd6
DEVTYPE=mtd
/sys/class/mtd/mtd6/dev:
90:12
/sys/class/mtd/mtd6/type:
nand
/sys/class/mtd/mtd6/flags:
0x400
/sys/class/mtd/mtd6/size:
184156160
/sys/class/mtd/mtd6/erasesize:
131072
/sys/class/mtd/mtd6/writesize:
2048
/sys/class/mtd/mtd6/subpagesize:
512
/sys/class/mtd/mtd6/oobsize:
64
/sys/class/mtd/mtd6/numeraseregions:
0
/sys/class/mtd/mtd6/name:
system

/sys/class/mtd/mtd7/uevent:
MAJOR=90
MINOR=14
DEVNAME=mtd7
DEVTYPE=mtd
/sys/class/mtd/mtd7/dev:
90:14
/sys/class/mtd/mtd7/type:
nand
/sys/class/mtd/mtd7/flags:
0x400
/sys/class/mtd/mtd7/size:
111149056
/sys/class/mtd/mtd7/erasesize:
131072
/sys/class/mtd/mtd7/writesize:
```

```
2048
/sys/class/mtd/mtd7/subpagesize:
512
/sys/class/mtd/mtd7/oobsize:
64
/sys/class/mtd/mtd7/numeraseregions:
0
/sys/class/mtd/mtd7/name:
cache

/sys/class/mtd/mtd8/uevent:
MAJOR=90
MINOR=16
DEVNAME=mtd8
DEVTYPE=mtd
/sys/class/mtd/mtd8/dev:
90:16
/sys/class/mtd/mtd8/type:
nand
/sys/class/mtd/mtd8/flags:
0x400
/sys/class/mtd/mtd8/size:
206700544
/sys/class/mtd/mtd8/erasesize:
131072
/sys/class/mtd/mtd8/writesize:
2048
/sys/class/mtd/mtd8/subpagesize:
512
/sys/class/mtd/mtd8/oobsize:
64
/sys/class/mtd/mtd8/numeraseregions:
0
/sys/class/mtd/mtd8/name:
userdata

/sys/class/mtd/mtd9/uevent:
MAJOR=90
MINOR=18
DEVNAME=mtd9
DEVTYPE=mtd
/sys/class/mtd/mtd9/dev:
90:18
/sys/class/mtd/mtd9/type:
nand
/sys/class/mtd/mtd9/flags:
0x400
/sys/class/mtd/mtd9/size:
1572864
/sys/class/mtd/mtd9/erasesize:
131072
```

```
/sys/class/mtd/mtd9/writesize:
2048
/sys/class/mtd/mtd9/subpagesize:
512
/sys/class/mtd/mtd9/oobsize:
64
/sys/class/mtd/mtd9/numeraseregions:
0
/sys/class/mtd/mtd9/name:
cust

/sys/class/mtd/mtd10/uevent:
MAJOR=90
MINOR=20
DEVNAME=mtd10
DEVTYPE=mtd
/sys/class/mtd/mtd10/dev:
90:20
/sys/class/mtd/mtd10/type:
nand
/sys/class/mtd/mtd10/flags:
0x400
/sys/class/mtd/mtd10/size:
2097152
/sys/class/mtd/mtd10/erasesize:
131072
/sys/class/mtd/mtd10/writesize:
2048
/sys/class/mtd/mtd10/subpagesize:
512
/sys/class/mtd/mtd10/oobsize:
64
/sys/class/mtd/mtd10/numeraseregions:
0
/sys/class/mtd/mtd10/name:
kpanic
```

mount

```
$ mount
rootfs / rootfs ro,relatime 0 0
tmpfs /dev tmpfs rw,relatime,mode=755 0 0
devpts /dev/pts devpts rw,relatime,mode=600 0 0
proc /proc proc rw,relatime 0 0
sysfs /sys sysfs rw,relatime 0 0
none /acct cgroup rw,relatime,cpuacct 0 0
tmpfs /mnt/asec tmpfs rw,relatime,mode=755,gid=1000 0 0
none /dev/cpuctl cgroup rw,relatime,cpu 0 0
/dev/block/mtdblock6 /system yaffs2 ro,relatime 0 0
/dev/block/mtdblock8 /data yaffs2 rw,nosuid,nodev,relatime 0 0
```

```
/dev/block/mtdblock7 /cache yaffs2 rw,nosuid,nodev,relatime 0 0
/dev/block/mtdblock5 /cdrom yaffs2 rw,relatime 0 0
tmpfs /tmp tmpfs rw,relatime,size=2048k 0 0
/dev/block/mtdblock0 /pds yaffs2 rw,nosuid,nodev,relatime 0 0
```

In älteren Android-Versionen (hier 2.2.1) findet das extra für Flash-Speicher entwickelte YAFFS2 Anwendung.

YAFFS(2)

```
$ cat /proc/yaffs
YAFFS built:Jan 19 2011 00:32:38
$Id$
$Id$

Device 0 "system"
startBlock......... 0
endBlock........... 1404
totalBytesPerChunk. 2048
nDataBytesPerChunk. 2048
chunkGroupBits..... 0
chunkGroupSize..... 1
nErasedBlocks...... 93
nReservedBlocks.... 5
blocksInCheckpoint. 3
nTnodesCreated..... 6700
nFreeTnodes........ 56
nObjectsCreated.... 1300
nFreeObjects....... 38
nFreeChunks........ 6336
nPageWrites........ 0
nPageReads......... 38640
nBlockErasures..... 0
nGCCopies.......... 0
garbageCollections. 0
passiveGCs......... 0
nRetriedWrites..... 0
nShortOpCaches..... 10
nRetireBlocks...... 0
eccFixed........... 0
eccUnfixed......... 0
tagsEccFixed....... 0
tagsEccUnfixed..... 0
cacheHits.......... 0
nDeletedFiles...... 0
nUnlinkedFiles..... 0
nBackgroudDeletions 0
useNANDECC......... 1
isYaffs2........... 1
inbandTags......... 0
```

```
doesTagsEcc........ 1

Device 1 "userdata"
startBlock......... 0
endBlock........... 1576
totalBytesPerChunk. 2048
nDataBytesPerChunk. 2048
chunkGroupBits..... 0
chunkGroupSize..... 1
nErasedBlocks...... 11
nReservedBlocks.... 5
blocksInCheckpoint. 0
nTnodesCreated..... 2300
nFreeTnodes........ 169
nObjectsCreated.... 1000
nFreeObjects....... 168
nFreeChunks........ 75569
nPageWrites........ 19674
nPageReads......... 28191
nBlockErasures..... 205
nGCCopies.......... 1927
garbageCollections. 101
passiveGCs......... 0
nRetriedWrites..... 0
nShortOpCaches..... 10
nRetireBlocks...... 0
eccFixed........... 0
eccUnfixed......... 0
tagsEccFixed....... 0
tagsEccUnfixed..... 0
cacheHits.......... 199
nDeletedFiles...... 0
nUnlinkedFiles..... 1100
nBackgroudDeletions 0
useNANDECC......... 1
isYaffs2........... 1
inbandTags......... 0
doesTagsEcc........ 1

Device 2 "cache"
startBlock......... 0
endBlock........... 847
totalBytesPerChunk. 2048
nDataBytesPerChunk. 2048
chunkGroupBits..... 0
chunkGroupSize..... 1
nErasedBlocks...... 846
nReservedBlocks.... 5
blocksInCheckpoint. 1
nTnodesCreated..... 100
nFreeTnodes........ 99
```

```
nObjectsCreated.... 100
nFreeObjects....... 94
nFreeChunks........ 54203
nPageWrites........ 14
nPageReads......... 9
nBlockErasures..... 1
nGCCopies.......... 0
garbageCollections. 0
passiveGCs......... 0
nRetriedWrites..... 0
nShortOpCaches..... 10
nRetireBlocks...... 0
eccFixed........... 0
eccUnfixed......... 0
tagsEccFixed....... 0
tagsEccUnfixed..... 0
cacheHits.......... 0
nDeletedFiles...... 0
nUnlinkedFiles..... 0
nBackgroudDeletions 0
useNANDECC......... 1
isYaffs2........... 1
inbandTags......... 0
doesTagsEcc........ 1

Device 3 "cdrom"
startBlock......... 0
endBlock........... 69
totalBytesPerChunk. 2048
nDataBytesPerChunk. 2048
chunkGroupBits..... 0
chunkGroupSize..... 1
nErasedBlocks...... 10
nReservedBlocks.... 5
blocksInCheckpoint. 1
nTnodesCreated..... 300
nFreeTnodes........ 31
nObjectsCreated.... 100
nFreeObjects....... 95
nFreeChunks........ 676
nPageWrites........ 0
nPageReads......... 0
nBlockErasures..... 0
nGCCopies.......... 0
garbageCollections. 0
passiveGCs......... 0
nRetriedWrites..... 0
nShortOpCaches..... 10
nRetireBlocks...... 0
eccFixed........... 0
eccUnfixed......... 0
```

```
tagsEccFixed....... 0
tagsEccUnfixed..... 0
cacheHits.......... 0
nDeletedFiles...... 0
nUnlinkedFiles..... 0
nBackgroudDeletions 0
useNANDECC......... 1
isYaffs2........... 1
inbandTags......... 0
doesTagsEcc........ 1

Device 4 "pds"
startBlock......... 0
endBlock........... 11
totalBytesPerChunk. 2048
nDataBytesPerChunk. 2048
chunkGroupBits..... 0
chunkGroupSize..... 1
nErasedBlocks...... 9
nReservedBlocks.... 5
blocksInCheckpoint. 1
nTnodesCreated..... 100
nFreeTnodes........ 64
nObjectsCreated.... 100
nFreeObjects....... 55
nFreeChunks........ 585
nPageWrites........ 194
nPageReads......... 248
nBlockErasures..... 4
nGCCopies.......... 175
garbageCollections. 3
passiveGCs......... 0
nRetriedWrites..... 0
nShortOpCaches..... 10
nRetireBlocks...... 0
eccFixed........... 0
eccUnfixed......... 0
tagsEccFixed....... 0
tagsEccUnfixed..... 0
cacheHits.......... 0
nDeletedFiles...... 0
nUnlinkedFiles..... 13
nBackgroudDeletions 0
useNANDECC......... 1
isYaffs2........... 1
inbandTags......... 0
```

Ein einhängen unter Zuhilfenahme „der“ Referenzimplementierung inkl. nachträglicher Installation der „mtd-utils“ per apt-get/aptitude war leider erfolglos! ☹

Zum Glück unterstützt Sleuthkit (sowie Autopsy) seit geraumer Zeit YAFFS2!

root Android

Root-Rechte erlangt man nach der Installation der USB-Treiber am einfachsten mit Hilfe von Super One Click!

(Leider ist hierfür ein M$ Windoof-System von Nöten… aber wozu gibt es VMware Fusion/Oracle VirtualBox!?)

(Allerdings ist die busybox in Super One Click schon etwas älter und enthält noch kein „nanddump“!)

```
user@Santoku:~$ adb shell
```

Wenn der Bootloader den Kernel läd, sagt er diesem auch, wie die einzelnen Bereiche im Flash-Speicher eingehängt werden sollen:

```
$ su
# cat /proc/cmdline
console=/dev/null console=ttyMTD10 rw mem=244M@0x80C00000 init=/init ip=off
brdrev=P2A androidboot.bootloader=0x0000 mtdparts=omap2-
nand.0:1536k@2176k(pds),384k@4480k(cid),384k@7424k(misc),3584k(boot)ro,4608k
@15232k(recovery),8960k(cdrom),179840k@29184k(system),106m@209408k(cache),20
1856k(userdata),1536k(cust),2m@521728k(kpanic)
```

"wipe" userdate/cache

Nach dem Zurücksetzen (formatieren) des Gerätes sah die Flash-Nutzung wie folgt aus:

```
$ busybox df -h
Filesystem                Size      Used Available Use% Mounted on
tmpfs                   113.1M         0    113.1M   0% /dev
tmpfs                   113.1M         0    113.1M   0% /mnt/asec
/dev/block/mtdblock6    175.6M    164.3M     11.4M  94% /system
/dev/block/mtdblock8    197.1M     49.8M    147.3M  25% /data
/dev/block/mtdblock7    106.0M      1.1M    104.9M   1% /cache
/dev/block/mtdblock5      8.8M      8.4M    328.0K  96% /cdrom
tmpfs                     2.0M     28.0K      2.0M   1% /tmp
/dev/block/mtdblock0      1.5M      1.4M    144.0K  91% /pds
```

Um ein Mobiltelefon wirklich forensisch zu sichern, sollte man die Recovery-„Partition“ mit einem um Root-Rechte, netcat und nanddump ergänzten Image flashen und über diese angepasste Recovery-„Partition“ eine Sicherung durchführen!