

PERSONALIZZARE UN INITRD.BIN DI UNA ROM ZAURUS
evilsocket
<http://www.evilsocket.net/>

..: Introduzione

Ultimamente mi è capitato di dover mettere una ROM Cacko sul mio palmare Sharp Zaurus SL-5500 ... ottimo sistema devo dire, davvero buono, in particolare perchè monta la root del filesystem / in una piccola zona nella ROM del palmare, e /home, ovvero tutti i dati dell'utente, compresi i programmi installati invece la monta nella scheda MMC/SD inserita .

Questa particolarità è estremamente utile poiché ci mette a disposizione tutto lo spazio che vogliamo per i nostri programmi e non saremo più legati alla piccola memoria interna del PDA e dei suoi limiti .

Ora però c'è un piccolo problema .

I sistemi Linux embedded che vanno sui PDA Zaurus, di base supportano solo schede MMC fino a 1GB, ma la mia scheda è da 2GB !!!!

Tempo fa, quando ancora usavo Open Zaurus, trovai una semplice soluzione a questo piccolo "inconveniente" ... un tale Miguel Freitas (che ho già avuto modo di ringraziare per email, ma non lo ringrazierò mai abbastanza XD) scrisse tempo fa il modulo mmc2g.o il quale, una volta messo al posto di quello che i sistemi embedded hanno di default, offre la possibilità di leggere memorie da 2GB .

Su Open Zaurus non fu un problema rimpiazzarlo, poiché tutto il filesystem era sulla ROM del PDA (al contrario di Cacko), ma su Cacko ho trovato un altro ostacolo .

Il modulo va messo al posto di

```
/lib/modules/2.4.18-rmk7-pxa3-embedix/kernel/drivers/block/sharp_mmc2g.o
```

peccato che /, essendo in ROM ed essendo di tipo "cramfs" è in sola lettura !
Come cavolo faccio a sostituire un modulo in un filesystem che posso solo leggere e NON scrivere ?

A questo punto ho capito che dovevo agire all'origine del problema, ovvero rimpiazzare il modulo nel file di installazione di Cacko che mette sulla ROM le basi del filesystem e i moduli del kernel ... quindi devo modificare initrd.bin !

Con un qualsiasi initrd.bin sarebbe semplice, ma lo stesso file di installazione è in formato cramfs, quindi non può essere modificato direttamente ... bene, andiamo a vedere come poter comunque creare un initrd.bin personalizzato con il nostro modulo per le MMC/SD da 2GB ;) .

..: Esecuzione

Do per scontato che voi abbiate i file di installazione di Cacko ROM, ovvero initrd.bin (quello che andremo a modificare) e zImage sul vostro pc (se non li avete potete tranquillamente scaricarli da <http://www.schwag.org/~crow/>) .

Prima di tutto, procediamo a creare una directory dove monteremo il file :

```
mkdir initrd_sorgente  
sudo mount initrd.bin initrd_sorgente -o loop
```

Noterete che la directory initrd_sorgente, essendo il mount point di un filesystem cramfs è in sola lettura, quindi ora dobbiamo creare un'altra directory dove copiare i contenuti del file per poi poterli modificare :

```
mkdir initrd_destinazione
cd initrd_sorgente
```

e ora copiamo i file da una directory ad un'altra

```
sudo tar -mcf - . | sudo tar -C ../initrd_destinazione -mxpf -
```

Se avete notato l'utilizzo di tar invece che del semplice comando cp, questo è perché i filesystem cramfs non possono essere copiati normalmente, ma va usato per l'appunto l'utility tar .

Ora abbiamo in initrd_destinazione tutti i file del nostro initrd.bin con la possibilità di modificarli/sostituirli, quindi procediamo a sostituire il modulo del quale parlavo prima (che potete scaricare dalla pagina <http://cambuca.ldhs.cetuc.puc-rio.br/~miguel/zaurus/>), per l'esempio che segue prendo come presupposto che nella vostra directory di lavoro abbiate la seguente struttura

```
initrd.bin (l'immagine di installazione di Cacko)
initrd_sorgente (la directory dove è montata initrd.bin)
initrd_destinazione (la directory dove abbiamo copiato i file)
mmcsd_2g.o (il modulo che avete scaricato dal sito di Miguel)
```

Procediamo a sostituire il modulo :

```
cd .. (perché dalle operazioni precedenti ci trovavamo in initrd_sorgente)

sudo cp mmcsd_2g.o initrd_destinazione/lib/modules.rom/2.4.18-rmk7-pxa3-
embedix/kernel/drivers/block/sharp_mmcsd_m.o
```

A questo punto possiamo umount-are l'initrd.bin originale, e ripulire il disco dai file che non ci servono

```
sudo umount initrd.bin
sudo rm -f initrd.bin
sudo rm -rf initrd_sorgente
```

e ricreiamoci un initrd.bin partendo dalla directory nella quale abbiamo sostituito il modulo

```
sudo mkfs.cramfs initd_destinazione initrd.bin
```

Finalmente abbiamo il nostro initrd.bin personalizzato, quindi copiamolo insieme alla zImage di Cacko ROM (qualcuno dovrebbe dirglielo che in italiano dire "caco rom" non è bello XD) nella compact flash e flashiamo il palmare zaurus seguendo la solita procedura .

Finalmente vedrete che Cacko, all'avvio del sistema rileverà la nostra mmc da 2GB in /dev/mmcda e ci darà la possibilità di partizionarla per montarci la nostra /home .

Riporto brevemente la procedura da eseguire tanto per darvi un'analisi completa dell'operazione :

```
fdisk /dev/mmcda (e createvi una partizione primaria che sarà /dev/mmcda1)
mkfs.ext2 /dev/mmcda1
exit
```

Fatto questo, al riavvio del palmare vedrete che Cacko creerà il filesystem nella partizione (ci vorrà un po di tempo dato che la mmc è grande) e finalmente avrete tutto lo spazio di cui avete bisogno per installare le vostre applicazioni ! ^^

evilsocket