

JUKE-BOX ROM CREATOR SCRIPT PACK

by P-LAB 2020

Questo pack contiene una serie di script utili per realizzare ROM personalizzate (custom-ROM) con i propri programmi o giochi preferiti.

Analogamente a quanto specificato nel manuale d'uso della *Apple-1 Juke-box* in questo documento il termine ROM indica genericamente dispositivi programmabili tipo EPROM/EEPROM/FLASH/OTP etc.

I file prodotti da questi script saranno pronti per essere utilizzati con il vostro programmatore di dispositivi preferito.

L'uso di questi script presuppone una certa dimestichezza con il terminale a caratteri di Linux.

È suggerito, ma non indispensabile, l'uso di utility tipo *Midnight Commander* per evitare la digitazione di lunghi comandi e il conseguente rischio di errori.

Gli script sono stati scritti in BASH per Linux e possono essere liberamente modificati.

1. Requisiti per il funzionamento

Per funzionare correttamente gli script hanno bisogno dei seguenti programmi:

- bc
- xxd
- asci2binary

Se non presenti nel sistema vanno installati con il comando:

```
sudo apt-get install bc xxd asci2binary
```

Gli script e tutte le directory di servizio vanno copiati nella directory chiamata JUKEBOX_ROM_CREATOR all'interno della vostra *home directory*.

2. Descrizione del funzionamento

Questo pack è in grado di interpretare e utilizzare il formato e la nomenclatura utilizzati dal programma CIDERPRESS e utilizzati anche dalla scheda CFFA1, ossia:

```
Name#TypeStartaddress
```

Dove troviamo:

- Name è il nome del file,
- # è un separatore,
- Type è il tipo di file: 06 per il tipo di file binario e F1 se si tratta di un programma BASIC,
- Startaddress è l'indirizzo di memoria (esadecimale) ove il file deve iniziare ad essere copiato.

Esempi:

BASIC#06E000 --> il programma si chiama BASIC, è un programma binario (06), va copiato in memoria dall'indirizzo 0xE000 in avanti.

STARTREK#F10300 --> il programma si chiama STARTREK, è un programma BASIC (F1) e va copiato in memoria dall'indirizzo 0x0300 in poi.

I files/programmi/giochi che desideriamo portare sulla nostra custom-ROM devono essere dapprima elaborati dallo script chiamato **1-stripper.sh**.

Esso rimuove le informazioni non strettamente necessarie affinché l'efficienza della memorizzazione sulla ROM sia massima e crea inoltre un file di supporto che verrà utilizzato in seguito.

Una volta fatto ciò il secondo script **2-packer.sh** si occuperà, dopo aver posto qualche domanda, di creare il file binario (o più file binari) che andranno poi scritti sulla vostra custom-ROM.

3. Struttura delle directory

Le directory sono strutturate come segue:

- 1-programs qui è dove vanno copiati manualmente i programmi e i giochi estratti dalla directory 9-archive (o da altre fonti) che si desidera includere nella propria custom-ROM.
- 2-stripped qui vengono depositati i programmi e i loro file accessori dopo l'esecuzione dello script 1-stripper.sh.
- 3-topack qui vanno copiati manualmente tutti i files provenienti dalla directory 2-stripped affinché vengano elaborati dal programma di packing.
- 4-packed qui vengono depositati i file binari della dimensione prescelta (16 kB o 32 kB) dopo l'elaborazione da parte del programma di packing, unitamente a un file con l'elenco dei contenuti.
- 9-archive questa directory contiene una serie di programmi per Apple-1, divisi per categorie.

4. Esempio di creazione di una custom-ROM

Supponiamo di volere creare una custom-ROM contenente il programma STARTREK.

Esso è un programma BASIC, quindi anche il programma BASIC dovrà necessariamente essere incluso nella custom-ROM. Nell'esempio seguente i comandi da dare sono scritti in **grassetto**.

Aprire un terminale e posizionarsi nella cartella di lavoro:

```
cd ~/JUKEBOX_ROM_CREATOR
```

Copiare ora i due programmi desiderati dall'archivio alla directory di partenza:

```
cp ./9-archive/BASIC/BASIC#06e000 ./1-programs  
cp ./9-archive/BASIC/STARTREK#f10300 ./1-programs
```

Ora eseguire il programma di preparazione dei files.

```
./1-stripper.sh
```

Compariranno a schermo una serie di informazioni di servizio, a prova del fatto che lo script sta girando:

```
found:
BASIC#06e000
BASIC--- BAS=0 LEN=4096 LEN_HEX=1000 LEN_HI=10 LEN_LO=00 START_HEX=E000
START_HI=E0 START_LO=00
found:
STARTREK#f10300
STARTREK BAS=1 LEN=3328 LEN_HEX=0D00 LEN_HI=0D LEN_LO=00 START_HEX=0300
START_HI=03 START_LO=00
Slicing: REAL_BASIC_START=0430 BYTES TO SLICE=305
          NEW_LEN=3024 NEW_LEN_HEX=0BD0 NEW_LEN_HI=0B NEW_LEN_LO=D0
Strip completed.
```

Al termine dell'esecuzione è possibile dare un'occhiata, se lo si desidera, alla cartella di destinazione per verificare che tutto abbia funzionato: per ogni file di partenza ci devono essere due files: uno con estensione .bin e uno con estensione .pat:

```
ls -la ./2-stripped/
```

```
total 24
drwxrwxrwx 2 root root 4096 ago  3 11:26 .
drwxrwxrwx 7 root root 4096 ago  3 10:59 ..
-rwxr-xr-x 1 pi  pi  4096 ago  3 11:26 BASIC---.bin
-rw-r--r-- 1 pi  pi   15 ago  3 11:26 BASIC---.pat
-rw-r--r-- 1 pi  pi  3206 ago  3 11:26 STARTREK.bin
-rw-r--r-- 1 pi  pi   15 ago  3 11:26 STARTREK.pat
```

Nell'esempio soprastante, sono stati creati tutti i files necessari agli step successivi.

Nota sui nomi dei files: se la lunghezza del nome del file originale è superiore a 8 caratteri il nome non sarà troncato ai primi 8 caratteri, ma sarà composto dai primi 7 caratteri del nome originale, più l'ultimo del nome originale.

Questa operazione viene fatta per limitare l'ambiguità che si verrebbe a creare con programmi aventi nomi più lunghi di 8 caratteri ma con la stessa radice (es. STARTREK e STARTREK2003 → STARTREK e STARTRE3 invece di STARTREK e STARTREK).

Copiare ora manualmente questi files nella cartella 3-topack:

```
cp ./2-stripped/* ./3-topack
```

Il prossimo passo è lanciare il programma di creazione del file binario vero e proprio, che conterrà anche il programma di gestione e tutto ciò che serve al suo funzionamento. Eseguire dunque:

```
./2-packer.sh
```

Il programma chiederà che tipo di prefisso si vorrà dare al nome del file binario (o dei file binari) che verranno creati. Nell'esempio si è scelto "MYROM_".

```
Filename prefix for output files ?
```

```
MYROM_
```

Ora il programma chiede come deve essere mappato dal punto di vista fisico il file che verrà creato: 16 kB (16384 bytes) oppure 32 kB (32768 bytes)

```
ROM file size in bytes [16384/32768] ?  
32768
```

Ora lo script ricerca e sceglie automaticamente la migliore allocazione per tutti i file scelti e procede alla creazione del file binario:

```
Starting loop 0  
analyzing: BASIC---.bin
```

```
PRG#1 BASIC---.bin FITS  
analyzing: STARTREK.bin
```

```
PRG#2 STARTREK.bin FITS  
24442+0 records in  
24442+0 records out  
24442 bytes (24 kB, 24 KiB) copied, 0,0494592 s, 494 kB/s  
226+0 records in  
226+0 records out  
226 bytes copied, 0,000362394 s, 624 kB/s  
32 kB EPROM detected: 32768 bytes  
HEADER=F1 -- BASIC PROGRAM DETECTED -- PROGRAM LENGTH=0BD0 -- NEXT START  
ADDRESS=4C86 -- Next PAT=31759  
HEADER=FE -- BINARY PROGRAM DETECTED -- PROGRAM LENGTH=1000 -- NEXT START  
ADDRESS=5C86 -- Next PAT=31774  
HEADER=FF
```

```
32 kB file detected. Flipping banks...  
PAT Completed.  
Loop 0 completed.  
All completed.
```

Se si sono selezionati molti files verranno effettuati molti cicli. Al termine dell'esecuzione i files saranno depositati nella cartella 4-packed:

```
ls -la ./4-packed/
```

```
total 44  
drwxrwxrwx 2 root root 4096 ago 3 11:27 .  
drwxrwxrwx 7 root root 4096 ago 3 10:59 ..  
-rw-r--r-- 1 pi pi 32768 ago 3 11:27 MYROM_0.BIN  
-rw-r--r-- 1 pi pi 19 ago 3 11:27 MYROM_0.txt
```

Il file MYROM_0.BIN è quello che dovrà essere usato per programmare la ROM. Il file MYROM_0.txt contiene la lista dei programmi contenuti nel file binario. Per vederne il contenuto sarà sufficiente usare il comando `cat`:

```
cat ./4-packed/MYROM_0.txt
```

```
1 BASIC  
2 STARTREK
```

Se per la propria raccolta si selezionano più files di quanti possano essere contenuti nella dimensione fisica scelta (16 kB o 32 kB) verranno creati tutti i files necessari, aggiungendo un suffisso numerico crescente. Esempio:

```
ls -la ./4-packed/
```

```
total 44
drwxrwxrwx 2 root root 4096 ago 3 11:27 .
drwxrwxrwx 7 root root 4096 ago 3 10:59 ..
-rw-r--r-- 1 pi pi 32768 ago 3 11:27 MYROM_0.BIN
-rw-r--r-- 1 pi pi 52 ago 3 11:27 MYROM_0.txt
-rw-r--r-- 1 pi pi 32768 ago 3 11:27 MYROM_1.BIN
-rw-r--r-- 1 pi pi 59 ago 3 11:27 MYROM_1.txt
-rw-r--r-- 1 pi pi 32768 ago 3 11:27 MYROM_2.BIN
-rw-r--r-- 1 pi pi 81 ago 3 11:27 MYROM_2.txt
-rw-r--r-- 1 pi pi 32768 ago 3 11:27 MYROM_3.BIN
-rw-r--r-- 1 pi pi 136 ago 3 11:27 MYROM_3.txt
```

I files .bin potranno essere copiati singolarmente in quattro ROM da 32 kB (tipo 27c256), oppure essere concatenati manualmente tramite comando `cat` per formare un file da 128 kB (131072 bytes) ed essere programmato su una ROM di dimensione adeguata:

```
cat MYROM_0.BIN MYROM_1.BIN MYROM_2.BIN MYROM_3.BIN > MYROM_TOTAL.BIN
```

Il file `MYROM_TOTAL.BIN` avrà una dimensione di 131072 bytes.

Se i files prodotti hanno dimensione complessiva inferiore a quella della ROM a disposizione (es. i miei giochi preferiti occupano solo 32 kBytes ma nel mio cassetto ho solo ROM da 64 kBytes) è preferibile copiare due volte il file da 32 kBytes in modo da non lasciare parti vuote. Parti vuote o no allocate possono causare il blocco del computer Apple-1.

5. Compatibilità

- Tutti i programmi in formato binario sono compatibili, purché non eccedano i limiti di RAM/ROM descritti nel manuale d'uso della *Apple-1 Juke-Box*.
- Solo i programmi BASIC scritti con APPLE-1 INTEGER BASIC sono compatibili con questi script.
- La dimensione massima di supporto gestibile da *Apple-1 Juke-Box* è di 4 Mbit (512 kBytes), equivalenti ad un supporto tipo 27/29c040.