

Tutorial sul Multiboot

13 marzo 2008

0.1 Introduzione

Un pò di tempo fa sul nostro DreamOS abbiamo deciso di lasciar perdere i vari bootloader scritti da noi e realizzare il supporto per GRUB <http://gnu.grub.org>. Per fare questo è necessario un header secondo le specifiche Multiboot. Le specifiche Multiboot originali si trovano su http://www.gnu.org/software/grub/manual/multiboot/html_node/index.html.

Indice

0.1	Introduzione	1
0.2	Header Multiboot	1
0.2.1	Makefile	3
0.2.2	Mbchk	3
0.2.3	Immagine di GRUB	4
0.3	DreamOS	4
0.3.1	Avviare DreamOS dalla shell di GRUB	4
0.3.2	Testare una modifica al sorgente	5

0.2 Header Multiboot

In DreamOS, il nostro header Multiboot è stato inserito nel file *boot/multicatcher.S*

```
. *****
;
; * * * * *          MultiCatcher          * * * * *
;
; * Author: shainer                                     *
; * Description: the Multiboot header                  *
; *****
;
```

```

; Multiboot declarations
MBOOT_PAGE_ALIGN    equ 1          ; Page aligning
MBOOT_MEM_INFO      equ 1<<1      ; Provide your kernel with
memory info
MBOOT_HEADER_MAGIC   equ 0x1BADB002 ; Multiboot Magic Number
; NOTE: We do not use MBOOT_AOUT_KLUDGE. It means that
GRUB does not
; pass us a symbol table.
MBOOT_HEADER_FLAGS   equ MBOOT_PAGE_ALIGN | MBOOT_MEM_INFO
MBOOT_CHECKSUM       equ -(MBOOT_HEADER_MAGIC +
MBOOT_HEADER_FLAGS)

[BITS 32]

[GLOBAL mboot]

mboot:
    dd MBOOT_HEADER_MAGIC
    dd MBOOT_HEADER_FLAGS          ; How GRUB should load your file
/ settings
    dd MBOOT_CHECKSUM              ; Check the other values

    dd mboot
    dd start                      ; Kernel start point

[GLOBAL start]
[EXTERN _start]                  ; The start point of our C code

start:
    push    ebx                  ; Load multiboot header

; Real execution
    cli                          ; Disable interrupts. x11-pc-it.map
    call _start                  ; main() procedure, located in
kernel.c
    jmp $                        ; Enter an infinite loop to freeze the
OS after the boot

```

Andiamo ora ad analizzare le parti principali. L'header deve essere contenuto nei primi 4kb dell'immagine.

MBOOT_HEADER_MAGIC: questa costante contiene il 'numero magico' che identifica l'header, 0x1BADB002

MBOOT_HEADER_FLAGS: questi sono i vari flags che decidono alcuni dei comportamenti di grub nei confronti del vostro OS. Ci sono molti flag (1= opzione attivata, 0= opzione disattivata) che spesso rimandano ad altre informazioni nella struttura. Mi limiterò a segnalare quelli usati nel mio

header, ovvero `MBOOT_MEM_INFO` e `MBOOT_PAGE_ALIGN`. Il primo abilita il passaggio di informazioni sulla memoria da parte di GRUB, il secondo allinea a 4kb (una pagina) i vari moduli di boot.

MBOOT_CHECKSUM: semplicemente un modo per verificare se l'header è corretto.

La parte successiva all'etichetta *mboot* (dichiarata `GLOBAL`) si occupa di ridefinire le stesse costanti; dopo il caricamento, si disattivano gli interrupt e si chiama la funzione principale del kernel (definita in *kernel.c*) per passarle il controllo.

0.2.1 Makefile

Dopo aver scritto l'header, è ora di compilare e linkare questo file. Le righe che ci interessano sono:

```
bl.img : boot/multicatcher.S
nasm -f elf ./boot/multicatcher.S -o bl.img
```

La prima riga secondo la sintassi di make definisce il nome del binario compilato e i file dal quale esso dipende (in questo caso solo uno). La seconda lo compila in formato ELF (supportato dal Multiboot). In questo modo *bl.img* contiene il nostro header compilato, ora questa riga si occupa di linkarlo al resto del codice, creando così l'immagine finale *kernel.bin*, che poi viene rinominato in *dreamos.img*.

```
kernel.bin: $(OBJ)

ld -static -oformat elf32-i386 -output=kernel.bin -script=kernel.lds bl.img $(OBJ)
-Ttext 0x100000 -Map kernel.map
```

Nota: alcune versioni di NASM mi danno problemi. In alternativa è possibile usare yasm: le opzioni rimangono identiche ed è anche più leggero. Disponibile sui repository APT.

0.2.2 Mbchk

Può risultare scomodo dover riavviare tutte le volte per provare se il proprio header contiene errori. C'è a disposizione l'utilità *mbchk*: chiamata con l'immagine del kernel subito dopo, esegue un rapido controllo dell'header alla ricerca di eventuali imprecisioni. Tuttavia non è in grado di trovare errori che dipendono ad esempio da altre parti del sistema operativo, dal formato sbagliato, da una chiamata a `ld` scorretta; prima di considerare il Multiboot completato bisogna fare delle prove di avvio con GRUB. I metodi sono due: inserire una voce relativa all'OS nel proprio *menu.lst* e avviare il vostro OS da macchina reale. Oppure, se lo ritenete scomodo (soprattutto per i test successivi) potete creare un'immagine di GRUB e ci penserà il vostro emulatore a far funzionare il tutto.

0.2.3 Immagine di GRUB

Prima di tutto, create un'immagine riempita di zeri della dimensione di un floppy (1.44 MB).

```
dd if=/dev/zero of=grub.img bs=1024 count=1440
```

Poi associatela ad un device di loop.

```
losetup /dev/loop1 grub.img
```

Creare un filesystem di tipo EXT2 sul device.

```
mkfs /dev/loop1
```

Montate la nuova immagine in una directory qualsiasi.

```
mount -o loop /dev/loop1 /mnt/dreamos
```

Create una directory apposita per grub (è un passaggio del tutto opzionale, ma a me piace metterlo per completezza e per rendere il tutto più ordinato).

```
mkdir -p /mnt/dreamos/boot/grub
```

Copiate i due stage fondamentali di GRUB (i percorsi qui possono cambiare).

```
cp /boot/grub/stage1 /boot/grub/stage2 /mnt/dreamos/boot/grub
```

Copiate qualsiasi altra cosa desiderate inserire nell'immagine (è il momento di copiare il binario del vostro kernel, ad esempio, o creare un grub.conf). Quando avete terminato, smontate il device:

```
umount /mnt/dreamos
```

Avviate grub e digitate i comandi nella sua shell.

```
grub -device-map=/dev/null
```

```
grub> device (fd0) /dev/loop1
```

```
grub> root (fd0)
```

```
grub> setup (fd0)
```

```
grub> quit
```

Staccate il dispositivo.

```
losetup -d /dev/loop1
```

L'immagine adesso è pronta! Per modificare qualsiasi cosa al suo interno, tocca rimontarla e rimontarla come visto in precedenza.

0.3 DreamOS

0.3.1 Avviare DreamOS dalla shell di GRUB

Per ora non abbiamo ancora GRUB in modalità grafica, ma solo la shell. Una volta avviato *grub.img* con il vostro emulatore, i comandi per avviare dreamos sono:

```
root (fd0)
```

```
kernel /boot/grub/dreamos.img
```

```
boot
```

Nota per chi usa un portatile: ho notato che Bochs non riconosce la normale posizione del carattere /, restituendo un bel punto interrogativo. Questo con il layout italiano: per ovviare al problema ho usato il tasto funzione + il tasto su cui in basso era segnato il carattere / (per me era il trattino, per altri potrebbe cambiare).

0.3.2 Testare una modifica al sorgente

Dopo aver digitato *make* per compilare:

```
mount -o loop /dev/loop1 boot/grub.img  
cp dreamos.img /mnt/dreamos/boot/grub/dreamos.img  
umount /mnt/dreamos
```

Dopo umount, *grub.img* contiene la versione aggiornata di DreamOS :)

Spero che la guida in generale vi sia piaciuta e vi abbia chiarito le idee sul Multiboot e su come provare DreamOS adesso che non usiamo più il vecchio DreamCatcher (che rimane comunque nei repository). Per qualsiasi domanda chiedete pure sul nostro forum (forum.dreamos.eu).