

Puntatori (SNES): questi sconosciuti...
Versione Beta
Copyright 2001 drzork4 (drzork4@hotmail.com)

=====
Versione

=====
Beta - Primo abbozzo per il mio grande amico (nonchè allievo ;))
Ombra.

=====
Cosa manca?

- =====
- Gestione dei Banchi di Testo
- Uso dei 'puntatori' dei Banchi di Testo
- Muovere il testo all'interno di una ROM
- Espansione di un Banco di Testo

=====
Indice

=====
Capitolo 1 - Introduzione
Capitolo 2 - I programmi
Capitolo 3 - Che cos'è un 'puntatore'?
Capitolo 4 - La nascita del 'puntatore'
Capitolo 5 - Come funziona un 'puntatore'
Capitolo 6 - Organizzazione del testo in una ROM
-Paragrafo 6.1 - Testo a lunghezza 'variabile'
-Paragrafo 6.2 - Testo a lunghezza 'fissa'
-Paragrafo 6.3 - Testo 'sparso' nella ROM
Capitolo 7 - \$Header\$ o 'colpo di testa'?
Capitolo 8 - 'Puntatori' SNES: finalmente...
-Paragrafo 8.1 - Tavola dei 'puntatori'
Capitolo 9 - Utilizziamo i 'puntatori'
-Paragrafo 9.1 - Stringa di testo
-Paragrafo 9.2 - Oggetti

=====
1 - Introduzione
=====

Visto che ci ho preso piacere a scrivere queste piccole lezioni, mi sono deciso a scrivere un'altro piccolo documento tanto per chiarire uno degli aspetti più importanti per la traduzione di una ROM: i 'puntatori'. Molte volte, leggendo i testi di grandi hacker inglesi, ci pervade un grande senso di impotenza e molte volte lasciamo la lettura a metà per paura di continuare a non capire quello che ad una osservazione più attenta poi risulta molto semplice e anche molto utile.

Prima di iniziare però devo fare alcune premesse molto importanti!!! In primus (grande il mio latino maccheronico... ;)) vorrei scusarmi già da adesso per gli eventuali errori logici che incontrerete nella lettura e vi prego di mandare ad ogni kazzata che scrivo mail con parolacce e accuse di vario genere...

Poi vorrei precisare che il testo è ALTAMENTE SCONSIGLIATO a chi di ROM non ne capisce un granchè o che si è avvicinato da poco al mondo dell'emulazione. Per comprendere a pieno il contenuto bisogna avere delle conoscenze base, il minimo indispensabile:

- sapere come è fatta una ROM;
- conoscere bene i programmi che a presso andrò ad elencare;
- avere molta molta ESPERIENZA e soprattutto PAZIENZA!!!

Questo è quello che chiedo. Ora possiamo iniziare...

=====
2 - I programmi
=====

Sono pochi i programmi base che ci servono per questo tipo di operazione.

Prima di tutto ci occorre una ROM fresca fresca. Mi raccomando: prima di cominciare a lavorare accertarsi che sia perfettamente funzionante se no si rischiano notti insonni... ;) .

Naturalmente l'emulatore!!! Ci servirà per vedere se effettivamente quello che facciamo è giusto o sbagliato. Io consiglio vivamente lo ZSNES, perchè è molto stabile e farà girare un numero enorme di ROMs. (ci sono troppo affezionato per non consigliarvelo!!!)

Poi un editor HEX in grado di ricercare sia il testo che (ed è indispensabile) i byte esadecimali. Non ho nessuna preferenza...uno vale l'altro in questo caso. L'operazione importante è quella di ricerca. Comunque tra i migliori ci sono il Thingy (storico!), il Translhextion (pieno zeppo di funzioni!) e lo Snedit (per i veri hacker!).

Penultima, una bella calcolatrice sotto mano con cui farsi i conti con i famigerati numeri esadecimali. Mi raccomando non una qualunque...

E come ultima cosa...TANTA TANTA TANTISSIMA PAZIENZA!!!

=====
3 - Che cos'è un 'puntatore'?
=====

Bene, per capire cosa sono questi 'cosi' torniamo indietro nel tempo, ai tempi delle medie e dei primi anni di liceo...

La vostra professoressa di matematica cercava di spiegare a voi, hacker in erba ;) le basi di linguaggi ad alto livello, non riuscendo però a soddisfare la vostra fame di sapere.

Uno dei linguaggi che mi rimarrà per sempre nel cuore è il Basic con cui ho iniziato la mia carriera e di cui ancora conservo tanti ricordi... ^sigh^

Lasciando perdere il passato, è dal basic che voglio iniziare per spiegare cos'è un puntatore.

Vi ricordate il mitico comando GOTO che tantissime volte avrete usato nei vostri piccoli programmi sul C64? Bene il 'coso' può essere paragonato ad un GOTO... ad esempio:

```
10 PRINT "Come ti chiami?"  
20 GOTO 10
```

La riga 20 è un puntatore vero e proprio. Infatti non appena viene eseguita, rimanda il 'processo' alla riga 10. In poche parole un puntatore dà l'informazione alla CPU su dove jumpare (saltare) e può essere intesa quasi come un comando: VAI LA'!!!

Nel caso specifico dello SNES (ma anche del NES) i puntatori ci permettono di cambiare la disposizione del testo nello schermo e di sfruttare (nei limiti del possibile) lo spazio vuoto contenuto nella ROM o quello ricavato da un'eventuale espansione (vedere "Espansione fisica di una ROM").

Ma voi mi chiederete: sì, abbiamo capito cosa significa puntatore ma che cos'è effettivamente? Dove si trova nella ROM?

Prima di rispondere a queste domande, dovrete conoscere cosa c'è in una ROM e come sono organizzati i dati che vi interessano. Quello che vi posso dire adesso (ed è già tanto) è che un 'pointer' corrisponde di solito a due valori esadecimali posti molto vicino al testo che

normalmente traducete...

=====

4 - La nascita del 'puntatore'

=====

Questa piccola sezione non è di necessaria importanza per poter imparare ad usare i pointer, quindi se avete fretta di iniziare siete autorizzati a saltare ed andare avanti. Se invece volete aumentare la vostra cultura generale non vi resta altro che leggere.

I puntatori non fanno parte integrante della cartuccia originale del gioco. Infatti se andate a controllare quello che c'è nella 'cassetta' :) e lo andate a confrontare con la vostra ROM (intendo il file), vi accorgete della grossa differenza di contenuto: MANCANO I POINTER!!! Non è una magia! Quando il gioco viene dumpato dalla cassetta originale al computer e trasformato in file, il così detto 'dumper' inserisce questi 'puntatori' che servono a guidare l'emulatore nel contenuto della ROM, e indirizzarlo nella giusta direzione.

Per questo sapendo usare i 'puntatori' si possono fare delle modifiche molto utili e rendere una traduzione ancora più completa.

=====

5 - Come funziona un 'puntatore'

=====

Come abbiamo già detto un 'puntatore' serve a 'puntare' la CPU (meglio, l'emulatore) in una precisa posizione della ROM e a far leggere una stringa di testo dal punto 'segnato' fino al byte <END>.

Per chiarire il funzionamento dei pointers è meglio fare un'esempio pratico. Immaginiamo di avere questa serie di parole:

```
Pozione<FF>Tenda<FF>Spada<FF>      [esempio 1]
```

^

^

^

Il simbolo '^' stà proprio ad indicare dove è posizionato il pointer e dove effettivamente rimanda nella ROM. Il byte <FF> sta ad indicare la fine della stringa...

Quindi, ogni volta che l'emulatore ha bisogno di quella stringa di testo, viene rimandato dal 'puntatore' in quella zona. Immaginiamo di aprire il menù degli Oggetti di un qualsiasi RPG. Supponiamo ancora di avere un solo oggetto: una pozione. Bene prima di aprire il menù l'emulatore vada a trovare le varie parti che lo compongono tra cui c'è anche la stringa 'Pozione'. Viene jumpato quindi grazie ai suoi 'fedeli amici' nel punto esatto in cui si trova la parte interessata e comincia a leggere finchè non trova l'altro amico, il byte fine <FF>, che gli dice di fermarsi. Ecco che compare la schermata:

```
ò-----ò
| Oggetti                               |
| > Pozione                             1 |      [schermata 1]
| Esci                                   |
|-----ò
```

Proviamo a scambiare di posizione, ad esempio, 'Pozione' con 'Tenda':

```
Tenda<FF>Pozione<FF>Spada<FF>      [esempio 2]
```

^

^

^

Come vediamo il 'puntatore', pur avendo cambiato il testo, non si sposta. Ora immaginiamo di avere come oggetti sia 'Pozione' che 'Tenda' e vediamo cosa succede facendo partire il gioco con questa modifica:

```

ò-----ò
|Oggetti                               |
|> Tenda                               1| [schermata 2]
| zione                                1|
| Esci                                  |
|-----ò

```

Come si può notare l'oggetto 'Tenda' è perfettamente leggibile. Questo non si può dire per 'Pozione' di cui si legge soltanto 'zione'... La spiegazione è semplice. Guardando [esempio 2] ci accorgiamo che l'emulatore comincia a leggere dal punto in cui è jumpato dal puntatore e cioè proprio dalla 'z'...

Dobbiamo imparare allora a spostare il pointer dove ci conviene !!! (Altri esempi sul funzionamento dei pointer nel Capitolo 6)

=====

6 - Organizzazione del testo nella ROM

=====

Aperto l'hex editor e trovata la TBL, ci accorgiamo che il testo nella ROM è organizzato in quelli che potremmo definire 'gruppi di frasi' ma che più correttamente si definiscono 'blocchi di testo'. In questi serbatoi così importanti per noi ;) il testo di solito è organizzato in modi differenti, che ora andremo a specificare.

(6.1) Testo a lunghezza 'variabile'

Nel blocco il testo è diviso in stringhe ordinate nell'ordine in cui compaiono nel gioco (o quasi). Le varie stringhe sono divise da un valore sempre uguale che ne indica la fine (e che cambia da ROM a ROM). Nelle TBLs create per Thingy questo valore viene indicato con uno slash, '\', per rendere più facile la traduzione. Codeste ROM sono organizzate secondo un sistema di 'puntatori' che potremmo definire STANDARD. Questo vale per la maggior parte dei giochi ma ci sono delle eccezioni. Nello SNES non sempre si trovano ROM di questo tipo...

Ad esempio ce ne sono alcune che non utilizzano un solo valore per indicare la fine della stringa, ma più di uno, che rende la traduzione del gioco più complicata e soprattutto molto più antipatica.

Ma non è finita !!! Alcune DEVIL ROM (come oserei chiamarle) non utilizzano affatto i 'puntatori' e i byte '\' possono essere spostati a piacimento. Ad un primo impatto potrebbe sembrare un semplificazione per la traduzione ma se si vuole dumpare il testo, beh, è molto complicato non sapendo dove effettivamente inizia e finisce dove una stringa. Quindi state attenti!!!

Del primo caso conosco soltanto Seiken Densetsu 3 che utilizza 4-5 byte come <END>; mentre del secondo ho saputo che Romanting Saga 3 è una delle uniche.

Per far comprendere meglio facciamo degli esempi.

Ranma 1/2 RPG --> Formato Standard

Ecco come compare nell'editor (riporto soltanto una parte):

```

69 6c 6c 61 67 65 2e ff 4d 61 6e 3a fc 54 68 65 illage.\Man:*The
20 53 74 61 72 20 57 69 6c 6c 6f 77 20 63 61 74 Star Willow cat
20 6c 75 72 65 fc 69 73 20 68 69 64 64 65 6e 20 lure*is hidden
69 6e 20 74 68 65 20 43 61 76 65 20 6f 66 fc 41 in the Cave of*A
6e 63 69 65 6e 74 73 2e ff 4d 61 6e 3a fc 50 6c ncients.\Man:*Pl
      ^^          ^^

```

Come ben si nota il byte FF è la fine della stringa. Proviamo a scambiare ad esempio i due byte segnati sopra:

```

69 6c 6c 61 67 65 2e ff 4d 61 6e 3a fc 54 68 65 illage.\Man:*The

```


00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000

Questo spazio non è inutile... L'header ci sarà tantissimo utile sia per i pointer sia per lo spostamento dei blocchi di testo.

Le informazioni contenute però non sono di vitale importanza per il suo funzionamento.

Qualche volta si possono trovare ROM senza l'header. A quanto ho letto è sempre meglio inserirlo, anche manualmente, perchè alcuni emulatori potrebbero fare problemi a causa della sua mancanza.

=====

8 - 'Puntatori' SNES: finalmente...

=====

Eccoci qui !!! Mi raccomando leggete attentamente perchè dopo tante premesse e spiegazioni varie, è questa la parte più importante di tutto il DOC. Ora, per trovare il puntatore dobbiamo prima sapere l'inizio del blocco di testo e il byte che corrisponde alla fine di una stringa. Questo è vostro compito...

Trovato l'inizio del blocco e il byte <END> possiamo cominciare con la lettura dell'offset, ovvero dell'indirizzo dove è situata la stringa di cui dobbiamo trovare il pointer.

Immaginiamo che questa sia all'indirizzo 12345. Ecco i vari passaggi:

- 1) Sottraiamo i 200 bytes del HEADER. es. $19345 - 200 = 19145$
- 2) Prendiamo le ultime 2 coppie di numeri scartando tutto il resto.
es. $12145 \rightarrow 9145$
- 3) Dividi i due numeri esadecimali. es. $91 \ 45$
- 4) Inverti l'ordine. es. $45 \ 91$

4591 è il nostro puntatore !!! Ora cercate questi due numeretti magici dall'indirizzo da dove avete ricavato il pointer e cercate nella porzione di testo immediatamente superiore (di solito non è molto lontano dal blocco di testo anzi di solito è proprio sopra...). Trovato la posizione del puntatore nella ROM vi accorgete che a seguire ci saranno gli altri 'puntatori' del blocco. Il primo valore del pointer (45) cambierà sempre in crescendo (45, 4A, 53, ...).

es. $45 \ 91 \ 50 \ 91 \ A1 \ 91 \ .. \ .. \ .. \ .. \ 12 \ 92 \ .. \ .. \ .. \ ..$

Come vi sarete accorti anche il secondo numero del nostro puntatore (es. 91) è CAMBIATO.

Il motivo dovrebbe essere abbastanza chiaro... spero! ^_^

Una cosa IMPORTANTISSIMA si deve ricordare: il valore del 'puntatore' deve essere compreso tra i valori HEX 8000 e FFFF.

Ad esempio: prendiamo offset 16200.

- 1) Sottraiamo i 200 bytes del HEADER. es. $16200 - 200 = 16000$
- 2) Prendiamo le ultime 2 coppie di numeri scartando tutto il resto.
es. $16000 \rightarrow 6000$
- 3) Il valore più piccolo di 8000, allora aggiungiamoglielo.
es. $6000 + 8000 = E000$
- 4) Dividi i due numeri esadecimali. es. $E0 \ 00$
- 5) Inverti l'ordine. es. $00 \ E0$

00E0 è il pointer che stavamo cercando. Basta trovarlo ed il gioco è fatto !!!

8.1 Tavola dei 'puntatori'

A me piace più chiamarla più 'pointer's table'. Comunque l'importante è capire che cos'è !!! ;)

Quando cerchiamo il puntatore di una stringa ci accorgiamo, dopo averlo trovato, che prima e dopo ci sono altri 'puntatori' che corrispondono alle stringhe precedenti e seguenti a quella che a noi interessava. Questa grande SCOPERTA può portare ad una facile conclusione: basta che troviamo il primo pointer della prima stringa del blocco e il gioco è fatto.....BRAVISSIMI !!! ^_^

Ma tanto per fare il guastafeste devo dare un avvertimento: non è detto infatti che i 'puntatori' siano messi sempre seguendo l'ordine del blocco. Infatti a volte mi è capitato di incontrare delle ROM in cui i 'puntatori' erano messi quasi in un ordine casuale e anche programmoni per il dumping come GIZMO non sono riusciti ad aiutarmi molto. *sigh*

=====

9 - Utilizziamo i 'puntatori'

=====

Ora che ci sappiamo trovare un puntatore, vediamo a cosa serve ! o_o
Ma soprattutto come possiamo sfruttare in maniera efficiente questo valido strumento. ^_^

Bene, i pointer sono di un'utilità mostruosa soprattutto per chi ha in mente di fare una traduzione pulita e sfruttare ogni minima risorsa di spazio contenuta nella ROM.

Possiamo riassumere i vari casi in 5 base.

9.1 Stringa di testo

Con i puntatori possiamo cambiare la lunghezza di una stringa e sfruttare lo spazio che abbiamo guadagnato da un'altra parte. Ad esempio abbiamo le seguenti stringhe:

```
      *(1)      *(2)
0xxxxxh Ciao!!!\Ci si vede!
```

Vogliamo togliere due dei tre punti esclamativi in modo da guadagnare byte da aggiungere alla frase seguente. Benissimo niente di più facile ora che conosciamo i pointer !!!

Troviamo il 'puntatore' (2) e andiamo a trovare la posizione esatta. Modifichiamo seguendo il nostro scopo. Quindi diventa:

```
      *(1)      *(2)
0xxxxxh Ciao!\Ci si vede!!
              ^(3)
```

Questa modifica così kazzona richiede del lavoro. Troviamo il nuovo puntatore, ovvero il (3) dell'esempio. Andiamo a sostituire il vecchio pointer con il nuovo ed il gioco è FATTO !!!

9.2 Oggetti

Alcune volte, in alcuni RPG, troviamo item, magie, mostri, ecc. non con il testo a lunghezza fissa, ma bensì che sfrutta a pieno i puntatori. Quando un hacker riceve questa grazia....

(lascio immaginare ;))

In questa piccola sezione voglio riprendere soltanto un consiglio che potrete facilmente trovare nella guida di Gidheon Zhi degli AGTP. Se vi capita un caso di questo tipo:

Potion<END>BigPotion<END>

La cosa più immediata sarebbe tradurre, cambiare i puntatori e tanta fortuna. Alla fine si avrebbe un risultato come questo:

```
*1          *2
Pozione<END>GranPozione<END>
```

Ma perchè non sfruttare questo nuovo strumento acquisito?
Tutte e due gli oggetti hanno una parte comune, 'Pozione', quindi
potremmo cambiare i pointer in questo modo:

```
*2 *1
GranPozione<END>
```

Il risultato è lo stesso ma abbiamo risparmiato un bel pò di spazio !!

=====

Questo per ora è tutto...non ce la faccio più !!! Ho gli esami ed
invece di studiare scrivo tutte queste cose per voi !!!
Spero di essere stato abbastanza chiaro...non ho niente da scrivere...
ciao

drzork4