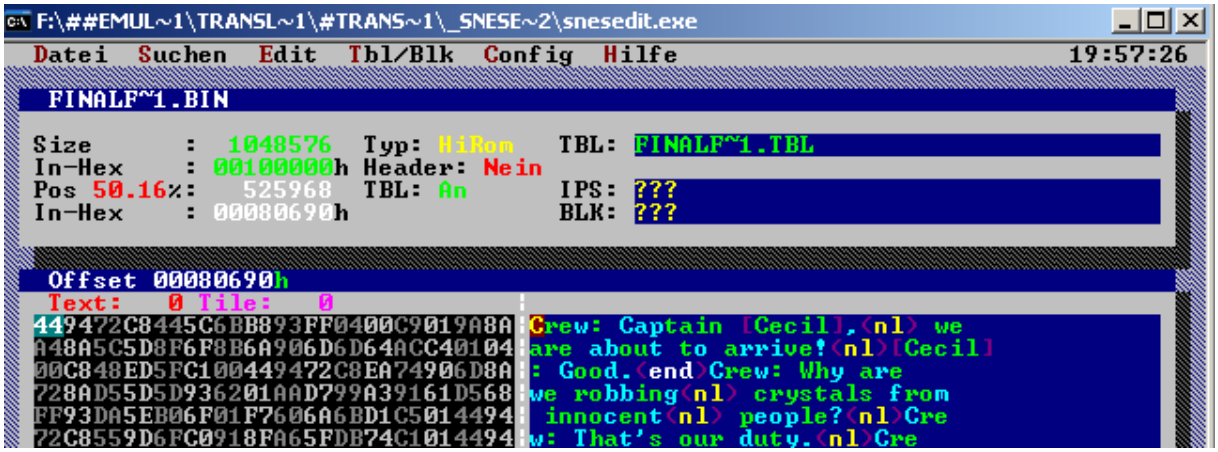
Diese Anleitung wird erläutern wie Daten in einer ROMBank verändert bzw. Daten aus einer anderen laden lässt. Den größten Vorteil als Übersetzer haben wir davon, die Daten kompakter gestalten zu können, weil teilweise völlig andere Sachen direkt nach Texten im Rom erscheinen.

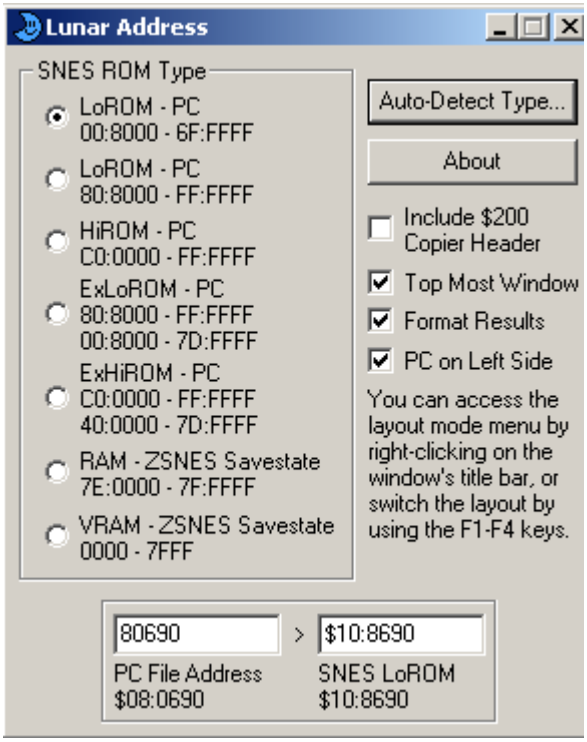
Unser Testobjekt:
Final Fantasy II (USA) <- "NoIntro" Dump, ohne Header
Und unsere Werkzeuge:
"Geiger's Snes9x Debugger"
"Lunar Adress und Lunar Expand"
SnesEdit"

Was wir verschieben:
Die ROMBank die den ersten Text im Spiel aufruft
(Zur Erinnerung: "Crew: Captain Cecil, we are about to arrive!.....")

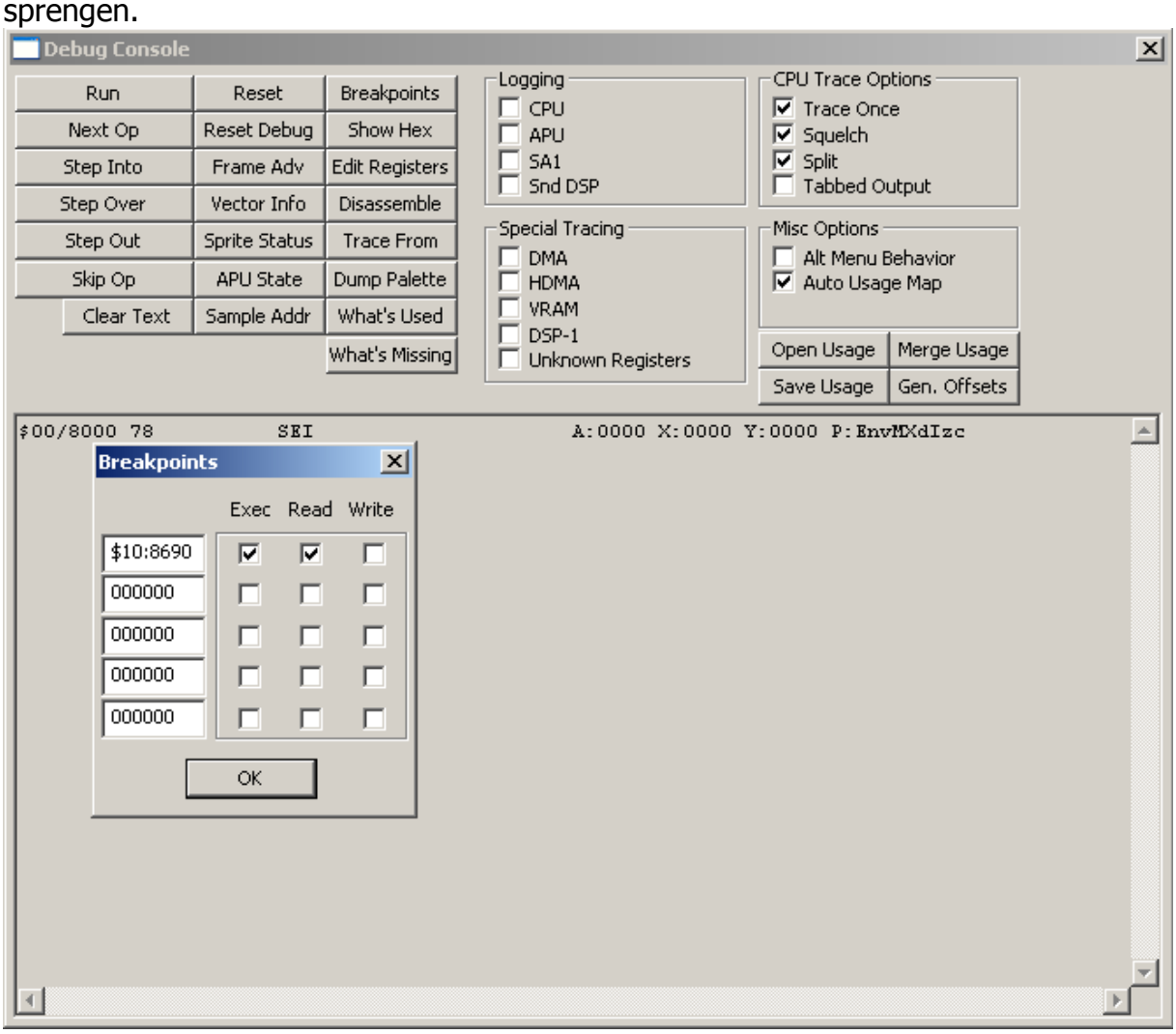


Also, hier ist unser Text, ich gehe mal davon aus dass jeder weiß wie man ihn findet und eine TBL erstellt!

Unser Text beginnt am Offset: 0x80690, damit kann der Debugger aber nichts anfangen, wir wandeln diese Adresse in eine "verwertbare" Rombank Adresse um, dazu benutzen wir "Lunar Adress", einfach die ROM laden und das Programm sollte den korrekten ROM Typ von alleine erkennen, unsere Adresse in die linke Textbox eingeben und in der rechten sollte unsere neue Adresse erscheinen -> $10:8690

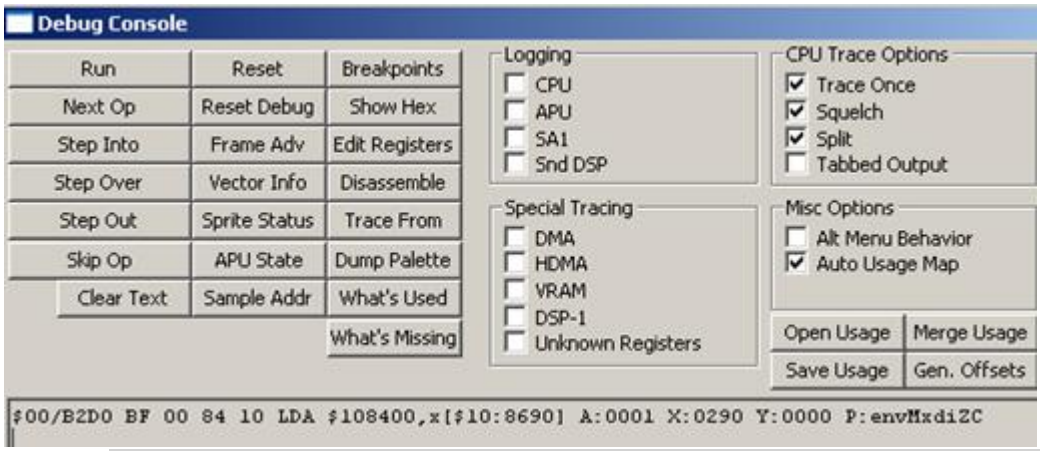


Jetzt starten wir den Snes9x Debugger, laden die ROM, die "Debug Console" sollte jetzt erscheinen: Was wir nun machen ist einen "Breakpoint" setzen. Der Debugger bekommt von uns die Adresse und den Auftrag beim ausführen oder Lesen die Emulation anzuhalten(Pausieren).
In der Debug Console auf "Breakpoints" klicken, in die erste Textbox unsere Adresse eingeben ($10:8690), Häkchen bei "Exec" und "Read" machen.
"Write" bedeutet wenn etwas in den RAM geschrieben wird. Hilfreich um die Adresse von Grafiken zu lokalisieren. Genaue Informationen würden aber hier den Rahmen sprengen.



Jetzt auf "Run" klicken und spielen, normalerweise sollte das Spiel NACH der Flugsequenz und BEVOR sich die Textbox öffnet, anhalten:

Und siehe da:



"$00/B2D0 BF 00 84 10 LDA $108400,x[$10:8690] A:0001 X:0290 Y:0000"
ist das was für uns interessant ist.
Eine genaue Betrachtung erklärt die Werte und einige Funktionen:

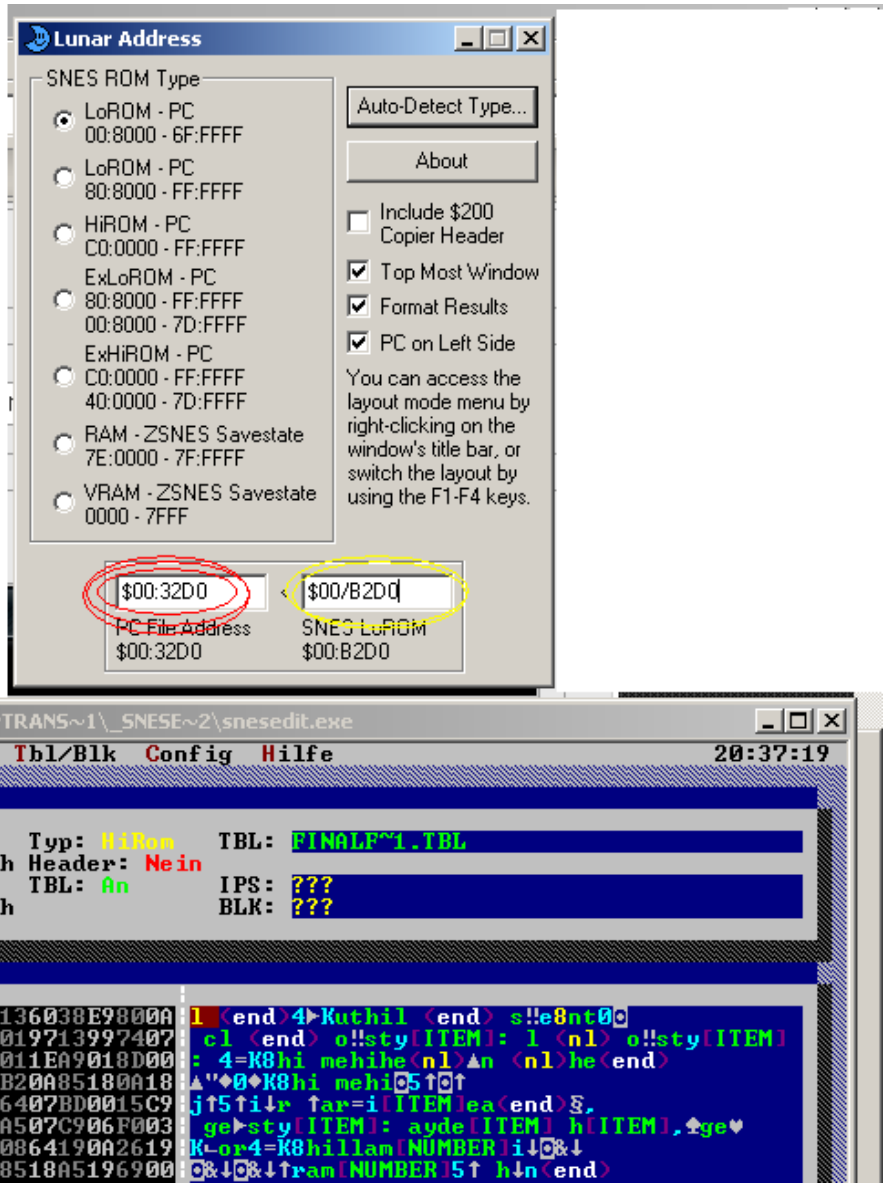
$00/B2D0 <- Unsere Adresse in der ROM von der der Aufruf gekommen ist. Wenn wir diese Adresse in eine PC-Adresse umwandeln erhalten wir: 0x32D0
BF 00 84 10 <- Wie diese Funktion als Zahlen in der ROM aussieht

LDA $108400,x[$10:8690] <- Was diese Funktion bewirkt, wird weiter unten erklärt.
A:0001 ; X:0290 ; Y:0000 <- Werte in den Registern und im Akkumulator

Zur Funktion LDA:
LaDe $108400 UND X in den Akkumulator,
Ergebniss: $10:8690 als ROM Adresse, =[0x80690]
0x80400 + X = 0x80690

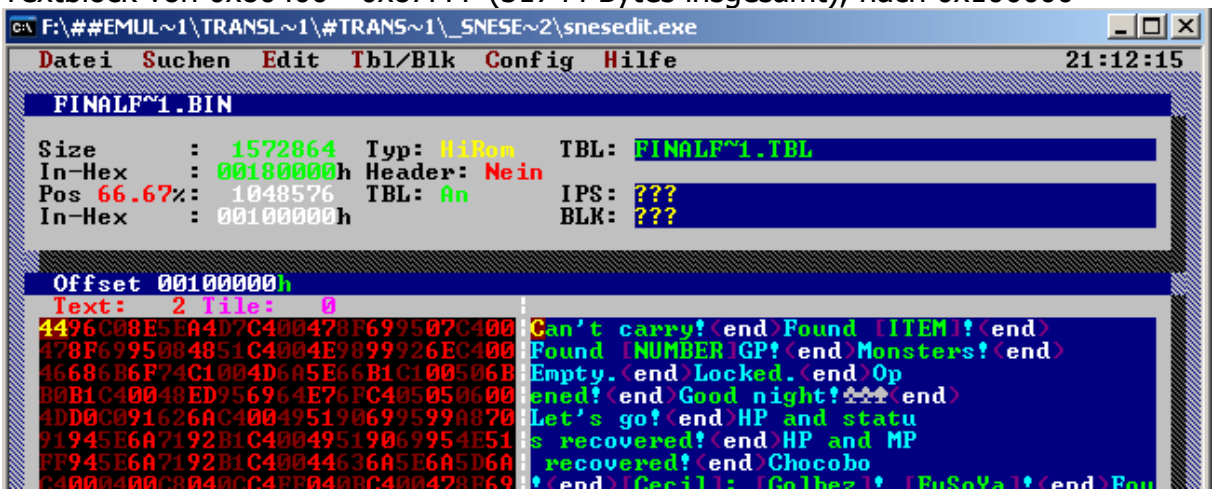
Damit haben wir unsere Routine die den Text in dieser Rombank lädt gefunden, in der ROM an Adresse: 0x32D0

Lasst uns jetzt die ROM von 8Mbit auf 12Mbit mit "Lunar Expand" erweitern, wir brauchen ja schließlich Platz



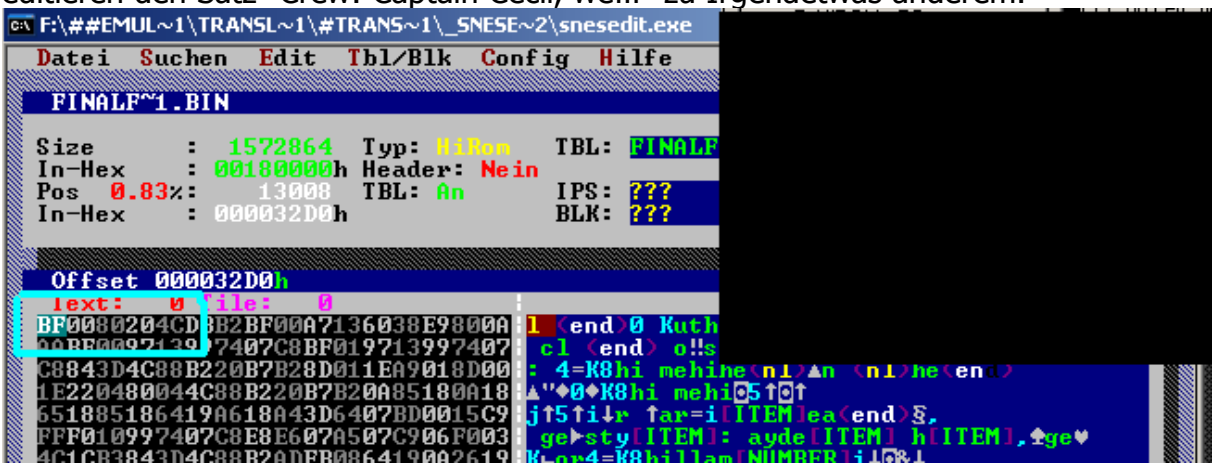
Unsere 4 neuen Rombank beginnen bei der Adresse 0x100000, die ROM Adresse ist "$20:8000"

Ihr solltet als erstes eure Daten die ihr wollt dorthin verschieben, das heißt den Textblock von 0x80400 - 0x87FFF (31744 Bytes insgesamt), nach 0x100000

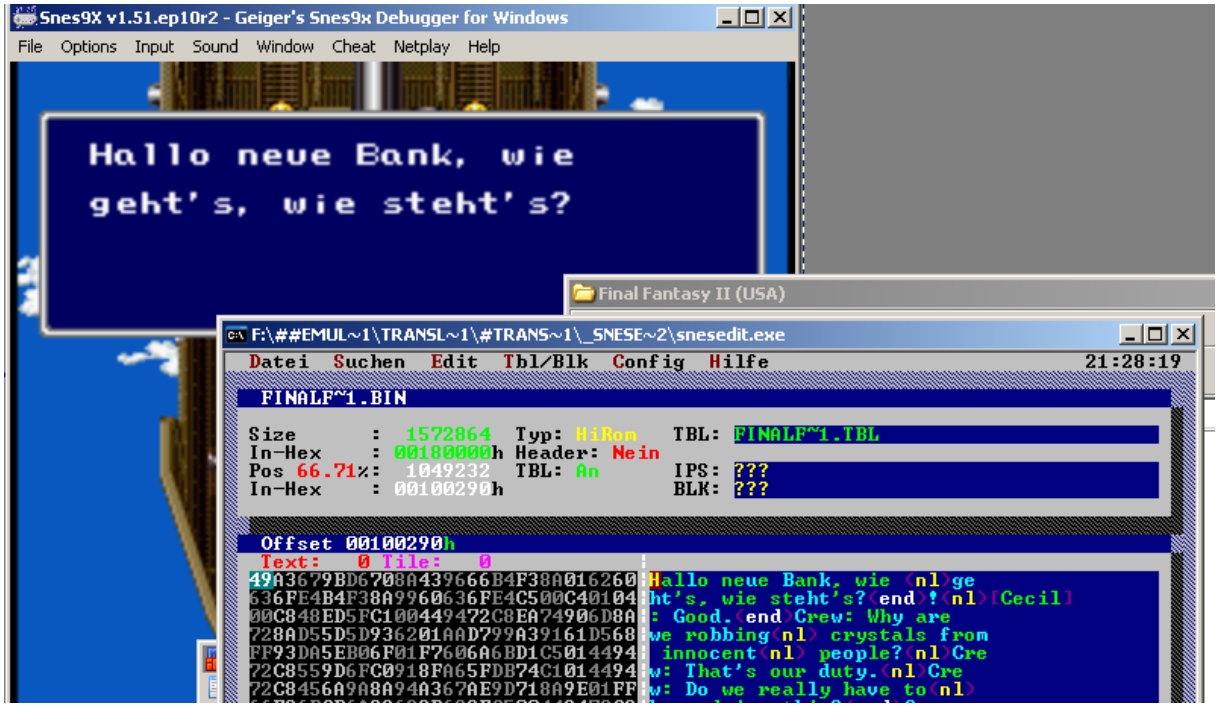


Jetzt müssen wir dem Spiel sagen von wo es den neuen Text laden soll, erinnert ihr euch? Adresse 0x32D0 in der ROM

Ändern wir doch mal "BF 00 84 10" zu "BF 00 80 20" ($20:8000 = 0x100000), editieren den Satz "Crew: Captain Cecil, we…" zu Irgendetwas anderem.



Nun schauen uns das Spiel erneut im Emulator an.



Scheint als wäre dieser kleine Hack erfolgreich gewesen, das Spiel lädt jetzt den Text von unserer neuen Rombank, als kleine Aufgabe könntet ihr ja versuchen die Pointer von diesem Textblock auch zu verschieben.

Original von ManakoAT
Verändert und als PDF umgewandelt:
blackerking