THE LOST "TRICKS" [PART 1]
BY MIKE WHITE

LAST TIME WE LOOKED AT THE MULTIPLE LOADING TECHNIQUE USED IN "DUNGEONS OF DRACULA" AND I ASKED IF YOU COULD MAKE IMPROVEMENTS. WELL, MIKE PEACE DID IT THIS WAY TO HOPEFULLY STOP COPYING OF HIS PROGRAMS RATHER THAN STRIVE FOR EFFICENCY. SO IT'S NOT ,TOO, DIFFICULT TO MAKE IMPROVEMENTS PROVIDING THAT WE DISCARD THE COPY GAURD ASPECT <u>ENTIRELY</u>! IN FACT, I DIDN'T WRITE THIS UP BACK IN THE 1980'S BECAUSE MIKE ASKED ME NOT TO! BUT NOW THAT **WAVEMAKERS** TAPES ARE "PUBLIC DOMAIN" I THOUGHT THAT THE PROMISSED BUT MISSING LESSON FROM "TRICKS OF THE TRADE" COULD NOW BE TOLD (SEE ARCADIAN VOL.7 PG.16 OR NIAGARA BUG BULLETIN VOL.2 PG.27). MIKE PEACE WAS A BASIC PROGRAMING GENIUS AND A TERRIFIC ENTERTAINER AS WELL!!

WE BEGIN BY LOADING THE MAIN PROGRAM, USING:

```
ENTER> CLEAR ;PRINT ;:INPUT [GO] (AND PLAY THE TAPE)
```

(SEE NIAGARA BUG BULLETIN VOL.2 PG.38 OR ARCADIAN VOL.7 PG.21) FOR THE EXPLAINATION. THEN KEY IN:

```
ENTER> %(-22988)=-256;%(-24565)=5;A=21543 [GO]
```

THE 1ST POKE REPLACES THE "END OF TEXT MARKER" THAT MIKE TOOK OUT TO FURTHER COPY GAURD HIS PROGRAM! I ALREADY EXPLAINED THE 2ND POKE LAST TIME. YOU CAN NOW EASILY ELIMINATE LINES 1 & 5, AND REPLACE THEM WITH:

```
>21543 :INPUT %(R)
```

THIS "TIME SHARES" THE VALUE OF THE "A" VARIABLE, AS WE NEED IT TO CORRECT A PRINTING PROBLEM AS WELL! THEN TAPE THE MAIN PROGRAM LIKE SO:

```
ENTER> PRINT ';CLEAR ;CY=9;PRINT " D0N0T ST0P THE REC0RDER";GOTO A';
     %(20170)=A;:PRINT %(16384),1937 [REC]
```

(THE POKE GENERATES THE CONTRACTION) THE 1ST BURST THEN BECOMES:

```
;PRINT " THANKS F0R Y0UR SUPP0RT";PRINT ;CX=0;PRINT "MIKE PEACE";
     BC=&(28)x8-1;&(9)=-1;GOTO A
```

THIS COULD BE KEYED IN AND SAVED FROM A SEPARATE PROGRAM, JUST LIKE BEFORE. THE 2ND BURST WOULD LOOK EXACTLY LIKE THE OLD ONE, EXCEPT THAT THE "R=20200;RUN " AT THE END, BECOMES "GOTO A"! TO CREATE THE 3RD BURST, YOU'LL HAVE TO GET THE MACHINE CODE TO %(20200) BY RELOADING THE ORIGINAL TAPE (NORMALLY) AND [HALT] AT THE "PLAYERS" SCREEN. THEN SAVE BY USING:

```
ENTER> PRINT "    ;R=20200;RUN ";:PRINT %(20160),95 [REC]
```

NOTE: 4 SPACES ARE IN THE QUOTE! YOU NOW HAVE 18 BYTES SAVED IN THE BASIC TEXT AREA, AND THE "B" VARIABLE ISN'T EVEN NEEDED! THE BURSTS DID ALL THE ENTERING INTO THE PROGRAM! IF YOU TRIED ">21543 :INPUT %(R);RUN " AND LEFT THE "R=20200;" IN THE 2ND BURST INSTEAD OF MOVING IT TO THE 3RD, THE COMPUTER NEVER LEAVES THE LINE INPUT BUFFER! ANYONE KNOW WHY?

BESIDES MULTIPLE LOADING IMPROVEMENTS, THERE'S A FEW CORRECTIONS THAT NEED CLEARED UP! FIRST OFF, IF ANYONE OUT THERE IS TRYING TO BUILD A HOMEMADE MEMORY EXPANSION FROM THE DRAWING THAT I SUPPLIED TO BALLY ALLEY.COM, THE RETURN LINE TO PIN 49 OF THE 50 PIN CONNECTOR (BUZOFF) NEEDS TO BE JUMPED TO PINS 47 (CASEN) AND 50 (SYSEN) ALSO! THIS IS SO THE READ FUNCTION WON'T TRY TO READ THE "ON BOARD" ROM, OR THE CARTRIDGE SLOT WHEN IT ACCESSES EXTERNAL MEMORY ABOVE 8000 HEX! THE 6000 & 7000 HEX AREAS SHOULD HAVE BEEN WORKING ALREADY. I HAD THE WRITE FUNCTION CORRECT, BUT NOT THE READ! SORRY.

IN MY TUTORIAL "FILE SEARCH II" I LEFT OUT THE FACT THAT IF YOU OPT FOR ALL 10 SLAVE PROGRAMS AND CHANGE "G<1" TO "G<0" AS I SUGGESTED, YOU'LL ALSO NEED TO CHANGE "%(20111)÷10-8" TO "%(20111)÷10-9" IN BOTH PLACES (IN AB), AND

"%(27808)÷10-8" BECOMES "%(27808)÷10-9" IN BRB AS WELL!! THE "LAST LINE NUMBER" IS 180 WITH 10 SLAVE PROGRAMS. THEREFORE %(20111) IN AB, OR %(27808) IN BRB EQUALS 180, AND 180÷10-8=10. STARTING THE LOOP AT 0 MEANS IT RUNS 11 TIMES, NOT 10! ALSO, ADD A QUOTE MARK AND A COMMA TO THE END OF LINE 30 TO SUPPRESS SCROLLING AS 10 ITEMS FILL THE WHOLE SCREEN!

THE LISTING FOR "REPACK" IN THE TUTORIAL OF THAT NAME HAD A REAL BONE HEADED MISTAKE IN IT! THE ADDRESS FOR THE "H" VARIABLE IN AB IS %(20016), NOT %(20018)! (SEE ARCADIAN VOL.5 PG.59). EITHER THE "B=20018" IN LINE 20 NEEDS TO BE "B=20016", OR CHANGE LINE 170 TO:

```
>170 IF KPIF D:PRINT I,18
```

THIS TAPES 18 VARIABLES "I" THROUGH "Z" TO BE RELOADED LATER INTO "H" THROUGH "Y". IF THAT'S ,TOO, CONFUSING, JUST CHANGE LINE 20 INSTEAD.

I WROTE "BEYOND REPACK" WHILE WORKING ON THE "MASTER TAPE LISTING" AND NOT ONLY FAILED TO CORRECT PAST MISTAKES, BUT ADDED MORE!!! IN THE "REPACK FAST" LISTING THE ABOVE FLAW SHOWS UP AS "D=20018" IN LINE 60. EITHER CHANGE THIS TO "D=20016", OR CHANGE LINE 100 TO RESEMBLE LINE 170 (SHOWN ABOVE). BESIDES THAT, LINE 10 IS MISSING A QUOTE MARK THAT SHOULD BE FOLLOWING THE WORD "TEXT". THE MACHINE IS SUPPOSED TO LOAD THE OBJECT PROGRAM INTO MEMORY BEGINNING AT %(A), NOT PRINT ";:INPUT %(A)" ON THE SCREEN! ALSO, ONE OF THE MACHINE CODE NUMBERS SHOULD BE "-6630" NOT "-6330"! I COPIED IT WRONG. OOPS!

AND IF THAT WASN'T BAD ENOUGH, THE DISASSEMBLED AB CARTRIDGE LOADER LISTING IS MISSING THE 2 BYTE INSTRUCTION:

```
     ED  IM2
     5E  (SET INT. MODE)
```

THIS IS SUPPOSED TO FOLLOW THE [END CALLS] INSTRUCTION (02), AND SETS THE Z80 TO IM2 (INTERRUPT MODE 2). YOU'LL HAVE TO MOVE THE BEGINNING OF THIS ROUTINE UP 2 BYTES TO FIT IT IN, AND CHANGE %(!2001) TO !C7. IM2 WAS THE MODE OF CHOICE FOR THE BALLY SYSTEM AS IT USES AN INDIRECT ADDRESSING SCHEME. WHEN THE SCREEN INTERRUPT OCCURS, THE Z80 FINISHES THE CURRENT INSTRUCTION, RESETS THE INTERRUPT FLIP-FLOPS (DISABLING ANY FURTHER INTERRUPTS), PUSHES THE "PC" REGISTER (THE Z80'S OWN PROGRAM COUNTER) TO THE STACK, AND RELOADS IT WITH AN ADDRESS THAT IT GETS FROM MEMORY [AT THE LOCATION IDENTIFIED BY THE 2 PART (2 BYTE) "VECTOR" THAT IT STORED DURING INITIALIZATION!] THE HIGH ORDER BYTE IS KEPT IN THE Z80'S OWN "I" (FOR INTERRUPT) REGISTER. THE LOW ORDER BYTE GETS RETURNED ON THE DATA BUSS FROM OUTPUT &(13) WHERE IT IS KEPT (IN THE ADDRESS CHIP). THE ADDRESS CHIP KNEW TO STORE AND RETURN THIS BYTE BECAUSE WE ALSO SET IT TO IM2 WHEN WE CALLED "SETOUT" (RST 38 SUB #22+1)! THE VALUE OF !18 WAS SENT TO OUTPUT &(14)! BIT 3 (08) SETS THE ADDRESS CHIP INTO IM2 MODE. IF BIT 2=0 THE ADDRESS CHIP WILL KEEP SIGNALING FOR AN INTERRUPT IF THE Z80 IS BUSY. IF BIT 2=1 THE ADDRESS CHIP ONLY SIGNALS ONCE! HENSEFORTH, THIS IS NOT USED IN THIS SYSTEM! BITS 1 & 0 DO THE SAME FOR THE LIGHT PEN INTERRUPT, BUT NOWHERE IS IT TOLD WHAT BIT 4 DOES! IF ANYONE KNOWS, PLEASE CONTACT US!

THE ONLY PART LEFT IN THE INITIALIZATION (BESIDES THE "EI" [FOR ENABLE INTERRUPTS]) IS TO TELL THE ADDRESS CHIP WHEN TO CALL FOR AN INTERRUPT! THIS IS SENT TO OUTPUT &(15) AS THE # OF LINES TO SCAN BEFORE AN INTERRUPT IS CALLED, MORE ON THIS NEXT TIME! THE VALUES USED IN THE AB BASICART LOADER CAME FROM A DISASSEMBLY OF AB, AND ARE AB'S OWN VALUES! IF ANYONE WANTS TO SEE A BASICART LOADER FOR BB YOU'LL HAVE TO ASK ME. I DID ONE BY SPECIAL REQUEST BACK IN THE 1980'S FOR AMORTIZATION BY **W&W SOFTWARE** (BOB WEBER).

IN BRB, THE ROUTINE LOOKS A LOT DIFFERENT THAN THE AB BASICART LOADER I SHOWED LAST TIME BECAUSE OF THE DIFFERENCES MENTIONED LAST TIME AS WELL. THE "ON BOARD MENU" WAS USED TO ACCESS TWO 4K PROGRAMS IN ONE 8K ROM! BUT, TO UNDERSTAND "MENU" WE MUST FIRST LOOK AT THE OPENING BYTES OF THE "ON BOARD ROM". WHEN WE HIT THE [RESET] BUTTON THE "PC" REGISTER (AS MENTIONED ABOVE) GETS CLEARED OUT! THIS DOES THE SAME AS A "JP,0000" OR "RST 00" INSTRUCTION, WHICH OCCURS IN THE COPY GAURD THAT L&M USED ON SOME OF THIER TAPES WHERE THE MACHINE RUNS INTO A "RST 00" COMMAND WHEN THE [HALT] KEY IS PRESSED! (SEE NIAGARA BUG BULLETIN VOL.2 PG.38 OR ARCADIAN VOL.7 PG.21) WHERE I SHOWED THE EASY WAY TO DO IT! WHEN THE COMPUTER STARTS BACK UP, IT RUNS THIS:

```
!0000 00 NOP <COLD START>
!0001 F3 DI
!0002 AF XOR A
!0003 D3 OUT (08),A
!0004 08 [LOW RES MODE]
!0005 C3 JP,0C61
!0006 61 [TO COLD]
!0007 0C [START TEST]
!0008 C3 JP,2007
!0009 07 [RST 08 JUMP]
!000A 20 [TO CARTRIDGE]
  ↓
!0010 C3 JP,200A
!0011 0A [RST 10 JUMP]
!0012 20 [TO CARTRIDGE]
!0013 06 [TO &(7)] <COLOR TABLE>
!0014 FA [TO &(6)]
!0015 07 [TO &(5)]
!0016 62 [TO &(4)]
!0017 FF [TO &(3)]
!0018 C3 JP,200D [TO &(2)]
!0019 0D [RST 18 JUMP] [TO &(1)]
!001A 20 [TO CARTRIDGE] [TO &(0)]
  ↓
!0020 C3 JP,2010
!0021 10 [RST 20 JUMP]
!0022 20 [TO CARTRIDGE]
  ↓
!0028 C3 JP,2013
!0029 13 [RST 28 JUMP]
!002A 20 [TO CARTRIDGE]
  ↓
!0030 C3 JP,2016
!0031 16 [RST 30 JUMP]
!0032 20 [TO CARTRIDGE]
  ↓
!0C61 3A LD A,(2000)<COLD>
!0C62 00              <START>
!0C63 20              <TEST>
!0C64 FE CP C3
!0C65 C3
!0C66 CA JP Z,2000
!0C67 00 [COLD START]
!0C68 20 [TO CARTRIDGE]
!0C69 31 LD SP,4FCE
!0C6A CE [SET STACK]
!0C6B 4F [POINTER]
!0C6C FF RST 38
!0C6D 1B [FILL+1]
!0C6E CE [STARTING]
!0C6F 4F [ADDRESS]
!0C70 32 [BYTES TO FILL]
!0C71 00
!0C72 00 [WITH THIS]
!0C73 32 LD (0FFF),A
!0C74 FF [CLEAR MAGIC]
!0C75 0F [WASTE BYTE]
!0C76 3D DEC A
!0C77 32 LD (4FEC),A
!0C78 EC [TIME OUT]
!0C79 4F [COUNTER]
!0C7A FF RST 38
!0C7B 00 [INTP C]
```

```
!0C7C 15 [E MUSIC]
!0C7D 17 [SET OUT+1]
!0C7E BF [TO &(10)]
!0C7F 29 [TO &(9)]
!0C80 08 [TO &(14)]
!0C81 19 [COL SET+1]
!0C82 13 [COLOR TABLE]
!0C83 00 [ADDRESS]
!0C84 0F [ACT INT]
!0C85 02 [X INT C]
!0C86 11 LD DE,0DF3
!0C87 F3 [STRING]
!0C88 0D [ADDRESS]
!0C89 21 LD HL,2000
!0C8A 00 [USER]
!0C8B 20 [LINK]
!0C8C 7E LD A,(HL)
!0C8D 23 INC HL
!0C8E FE CP 55
!0C8F 55 [USER]
!0C90 28 JR Z,03
!0C91 03 [TO "MENU"]
!0C92 21 LD HL,0218
!0C93 18 [TO LINK]
!0C94 02 [FOR GUNFIGHT]
!0C95 FF RST 38
!0C96 4A ["MENU"]
  ↓
!0DF3 53 S <STRING>
!0DF4 45 E
!0DF5 4C L
!0DF6 45 E
!0DF7 43 C
!0DF8 54 T
!0DF9 20 [SPACE]
!0DFA 47 G
!0DFB 41 A
!0DFC 4D M
!0DFD 45 E
!0DFE 67 (CHANGE E,D,C)
!0DFF 08 (HOR. POS.)
!0E00 58 (VER. POS.)
!0E01 0D (MAGIC BYTE)
!0E02 28 (
!0E03 43 C
!0E04 29 )
!0E05 20 [SPACE]
!0E06 42 B
!0E07 41 A
!0E08 4C L
!0E09 4C L
!0E0A 59 Y
!0E0B 20 [SPACE]
!0E0C 4D M
!0E0D 46 F
!0E0E 47 G
!0E0F 20 [SPACE]
!0E10 31 1
!0E11 39 9
!0E12 37 7
!0E13 38 8
!0E14 00 (END)
```

THIS BEGINS WITH THE "DI" (DISABLE INTERRUPTS) INSTRUCTION WHICH RESETS THE INTERRUPT FLIP-FLOPS (SEE ABOVE). DON'T FORGET THIS! NOTE: "EI" (ENABLE INTERRUPTS) DOES THE OPPOSITE. THEN THE ACCUMULATOR "A" IS XORED WITH ITSELF, AND THE RESULTANT 00 IS SENT TO OUTPUT &(8) WHICH LOCKS THE ARCADE IN LOW RES (160x104) MODE. THAT'S FOLLOWED BY A JUMP OVER THE RESTART "RST" TABLE. LOOK IN HERE TO SEE WHY CARTRIDGES HAVE A JUMP TABLE IN THE FIRST 24 BYTES.

EVENTS CONTINUE AT !0C61 BY INSPECTING THE FIRST BYTE OF THE CARTRIDGE SLOT FOR A "!C3" OR DIRECT JUMP! IF THIS OCCURS, THE INTERRUPTS ARE DISABLED (REMEMBER), AND THE STACK POINTER ISN'T SET YET EITHER! IN FACT, THIS IS A TOTALLY COLD START WITH A JUMP COMMAND OCCUPYING THE FIRST 3 BYTES OF THE CARTRIDGE. ALL OF THE "BASICS" DO THIS, AS WELL AS SEVERAL OTHERS!

IF NO "!C3" IS FOUND AT !2000, THE STACK POINTER GETS SET, AND A "FILL" (RST 38 SUB 26+1) CLEARS THE SCRATCHPAD. THE NEXT THING THAT HAPPENS IS A "WASTE BYTE" IS PUT THROUGH MAGIC RAM TO CLEAR THE SYSTEM, I'M NOT SURE WHY, BUT THE "BASICS" DO THIS ALSO! THEN THE "TIME OUT COUNTER" IS SET FOR SCREEN BLANKING (UPDATED BY THE SCREEN INTERRUPT SERVICE ROUTINE). NOTE: IF THE 300 BAUD INTERFACE IS CONNECTED, NO BLANKING OCCURS! ANYONE KNOW WHY?

THE INTERPRETER THEN GANGS 4 SUBROUTINES TOGETHER. "E MUSIC" SENDS ZEROS TO ALL THE SOUND PORTS STOPPING ALL SOUND. WE'VE DISCUSSED "SET OUT" AND "COL SET" BEFORE. "ACT INT" ACTIVATES (INITIALIZES) THE SCREEN INTERRUPT THAT THE CARTRIDGES, THE "ON BOARD GAMES", AND THE "MENU" USES. AFTER EXITING THE INTERPRETER, THE REGISTERS "DE" AND "HL" LOAD WITH THE REQUIRED DATA FOR "MENU". THEN THE CONTENTS OF !2000 IS CHECKED FOR A !55 VALUE. IF FOUND, "HL" REMAINS AT !2001. IF NOT (EMPTY CARTRIDGE SLOT USUALLY READS AS "!B0") "HL" RELOADS WITH !0218, WHICH IS WHERE THE FIRST OF 4 SEGMENTS OF A "LINKED LIST" BEGINS THAT BRINGS UP THE "ON BOARD GAMES". AFTER THAT, THE RST 38 "MENU" SUBROUTINE IS CALLED AND THE COMPUTER EXITS VIA THE "LINKED LIST" AND THE GAME PLAYER'S SELECTION! ONLY FROM A COLD START (AS ABOVE) DOES ANY SEGMENT NEED TO BE AT !2001! OTHERWISE THESE 6 BYTE (3 ADDRESS) SEGMENTS COULD BE ANYWHERE! THEY CONSIST OF:

```
  <LINK> ADDRESS OF NEXT SEGMENT  NOTE: 0000 ENDS LIST, AND !0218 BRANCHES TO
                  GUNFIGHT AND ETC.
<STRING> STARTING ADDRESS OF TITLE STRING  NOTE: "ASCII" CODE FOR "STR DIS"
  <JUMP> ADDRESS TO JUMP TO IF THIS TITLE IS SELECTED (EXIT "MENU")
```

IF "MENU" IS CALLED DIRECTLY (AS FROM A GAME CARTRIDGE), THE TIME OUT COUNTER AND STACK POINTER SHOULD BE SET (AS ABOVE), AND THE SCREEN INTERRUPT (RST 38 SUB 14) SHOULD BE ACTIVE. THEN USE:

```
"MENU" FF RST 38
       4B ["MENU"+1]
       ?? [TO E] (TOP OF SCREEN MESSAGE)
       ?? [TO D] (STARTING ADDRESS)
       ?? [TO L] (EXITING ADDRESS)
       ?? [TO H] (TO SELECTION)
```

NEXT TIME WE'LL EXAMINE THIS FURTHER! FOR NOW LET'S LOOK AT THE BRB 2x4 BASICART (AND MULTICART LOADER), WHICH IS:

```
!2000 55 ["MENU" FLAG] <USER>
!2001 EB [ADDRESS OF NEXT SEGMENT]
!2002 2F [OF "LINKED LIST"]
!2003 AC [ADDRESS OF #1]
!2004 3F [TITLE STRING]
!2005 88 [ADDRESS OF #1]
!2006 2F [PROGRAM LOADER]
!2007 ↓↓ [BASIC PROGRAM #1]
  ↓
!2F88 FF RST 38 <#1 PROGRAM LOADER>
!2F89 00 [INTP C]
!2F8A 07 [M CALL+1]
```

```
!2F8B 9D [ADDRESS OF]
!2F8C 2F [INTERPRETED NEST]
!2F8D 35 [STR DIS]
!2F8E 00 [HOR. POS.]
!2F8F 15 [VER. POS.]
!2F90 14 [MAGIC BYTE]
!2F91 AC [ADDRESS OF #1]
!2F92 3F [TITLE STRING]
!2F93 35 [STR DIS] <M JUMP>
!2F94 00 [HOR. POS.]
!2F95 25 [VER. POS.]
!2F96 18 [MAGIC BYTE]
```

```
!2F97 B8 [ADDRESS OF]
!2F98 2F [INSTRUCTION STRING]
!2F99 02 [X INT C]
!2F9A C3 JP,6FEA
!2F9B EA [ADDRESS OF]
!2F9C 6F [SWAP LOOP]
!2F9D 5F [MOVE+1] <INTERPRETED NEST>
!2F9E 00 [DESTINATION]
!2F9F 60 [ADDRESS]
!2FA0 00 [# OF BYTES]
!2FA1 10 [TO MOVE]
!2FA2 07 [SCORCE]
!2FA3 20 [ADDRESS]
!2FA4 1B [FILL+1]
!2FA5 00 [STARTING]
!2FA6 40 [ADDRESS]
!2FA7 00 [# OF BYTES]
!2FA8 0F [TO FILL]
!2FA9 00 [WITH THIS]
!2FAA 17 [SET OUT+1]
!2FAB C4 [TO &(10)]
!2FAC 2C [TO &(9)]
!2FAD 08 [TO &(14)]
!2FAE 19 [COL SET+1]
!2FAF E3 [ADDRESS OF]
!2FB0 2F [COLOR TABLE #1]
!2FB1 35 [STR DIS]
!2FB2 00 [HOR. POS.]
!2FB3 05 [VER. POS.]
!2FB4 18 [MAGIC BYTE]
!2FB5 DC [ADDRESS OF]
!2FB6 2F [REMOVE STRING]
!2FB7 08 [RETURN FROM NEST]
!2FB8 49 I <INSTRUCTION STRING>
!2FB9 4E N
!2FBA 53 S
!2FBB 45 E
!2FBC 52 R
!2FBD 54 T
!2FBE 6F (CHANGE E,D,C)
!2FBF 00 (HOR. POS.)
!2FC0 35 (VER. POS.)
!2FC1 1C [MAGIC BYTE]
!2FC2 42 B
!2FC3 4C L
!2FC4 55 U
!2FC5 45 E
!2FC6 20 [SPACE]
!2FC7 52 R
!2FC8 41 A
!2FC9 4D M
!2FCA 20 [SPACE]
!2FCB 42 B
!2FCC 41 A
!2FCD 53 S
!2FCE 49 I
!2FCF 43 C
!2FD0 6F (CHANGE E,D,C)
!2FD1 00 (HOR. POS.)
!2FD2 45 (VER. POS.)
!2FD3 18 (MAGIC BYTE)
!2FD4 50 P
```

```
!2FD5 52 R (INSTRUCTION STRING CONT.)
!2FD6 45 E
!2FD7 53 S
!2FD8 53 S
!2FD9 20 [SPACE]
!2FDA 3D =
!2FDB 00 (END)
!2FDC 52 R <REMOVE STRING>
!2FDD 45 E
!2FDE 4D M
!2FDF 30 0
!2FE0 56 V
!2FE1 45 E
!2FE2 00 (END)
!2FE3 E9 [TO &(7)] <COLOR TABLE #1>
!2FE4 5B [TO &(6)]
!2FE5 A5 [TO &(5)]
!2FE6 EF [TO &(4)]
!2FE7 84 [TO &(3)]
!2FE8 5A [TO &(2)]
!2FE9 06 [TO &(1)]
!2FEA 00 [TO &(0)]
!2FEB 18 [ADDRESS OF NEXT SEGMENT]
!2FEC 02 [OF "LINKED LIST"]
!2FED ?? [ADDRESS OF #2]
!2FEE 3F [TITLE STRING]
!2FEF 88 [ADDRESS OF #2]
!2FF0 3F [PROGRAM LOADER]
!2FF1 F3 DI <SWAP LOOP> (TO !6FEA)
!2FF2 DB IN A,(14)      (TO !6FEB)
!2FF3 14 [INPUT = KEY]  (TO !6FEC)
!2FF4 B7 OR A           (TO !6FED)
!2FF5 28 JR Z,FB        (TO !6FEE)
!2FF6 FB [WAIT FOR KEY] (TO !6FEF)
!2FF7 31 LD SP,6FF8     (TO !6FF0)
!2FF8 F8 [SET STACK]    (TO !6FF1)
!2FF9 6F [POINTER]      (TO !6FF2)
!2FFA 11 LD DE,6D2D     (TO !6FF3)
!2FFB 2D [SET BASIC]    (TO !6FF4)
!2FFC 6D [POINTER]      (TO !6FF5)
!2FFD CD CALL,2164      (TO !6FF6)
!2FFE 64 [START SCREEN] (TO !6FF7)
!2FFF 21 [INTERRUPT]    (TO !6FF8)
!3000 C3 JP,243E        (TO !6FF9)
!3001 3E [JUMP]         (TO !6FFA)
!3002 24 [INTO BRB]     (TO !6FFB)
!3003 00 [BASIC TEXT]   (TO !6FFC)
!3004 60 [START POINTER](TO !6FFD)
!3005 1E [BASIC TEXT]   (TO !6FFE)
!3006 6C [END POINTER]  (TO !6FFF)
!3007 ↓↓ [BASIC PROGRAM #2]
   ↓
!3F88 FF RST 38 <#2 PROGRAM LOADER>
!3F89 00 [INTP C]
!3F8A 07 [M CALL+1]
!3F8B 9D [ADDRESS OF]
!3F8C 2F [INTERPRETED NEST]
!3F8D 5F [MOVE+1]
!3F8E 00 [DESTINATION]
!3F8F 60 [ADDRESS]
!3F90 81 [# OF BYTES]
!3F91 0F [TO MOVE]
```

```
!3F92 07 [SCORCE]
!3F93 30 [ADDRESS]
!3F94 19 [COL SET+1]
!3F95 A4 [ADDRESS OF]
!3F96 3F [COLOR TABLE #2]
!3F97 17 [SET OUT+1]
!3F98 C4 [TO &(10)]
!3F99 2C [TO &(9)]
!3F9A 08 [TO &(14)]
!3F9B 35 [STR DIS]
!3F9C 00 [HOR. POS.]
!3F9D 15 [VER. POS.]
!3F9E 14 [MAGIC BYTE]
!3F9F ?? [ADDRESS OF #2]
!3FA0 3F [TITLE STRING]
!3FA1 0B [M JUMP]
!3FA2 93 [ADDRESS OF]
```

```
!3FA3 2F [M JUMP]
!3FA4 E9 [TO &(7)] <COLOR TABLE #2>
!3FA5 5B [TO &(6)]
!3FA6 A5 [TO &(5)]
!3FA7 EF [TO &(4)]
!3FA8 84 [TO &(3)]
!3FA9 5A [TO &(2)]
!3FAA 06 [TO &(1)]
!3FAB 00 [TO &(0)]
!3FAC ↓↓ [#1 TITLE STRING]
  ↓
!3F?? 00 (END)
!3F?? ↓↓ [#2 TITLE STRING]
  ↓
!3F?? 00 (END)
  ↓    ?? [EMPTY SPACE]
!3FFF ?? [FOR NEXT TIME]
```

IN EXAMINING THIS, THE FIRST SEGMENT OF THE "LINKED LIST" IS FROM !2001 TO !2006 FOLLOWING THE !55 AT !2000. THE SECOND SEGMENT IS AT !2FEB TO !2FF0 AND THAT "LINKS" TO THE "ON BOARD GAMES" AT !0218. THE CHARACTOR STRING FOR TITLE #1 AND #2 BEGINS AT !3FAC AND FOLLOWS ALL THE RULES FOR "STR DIS" (MORE ON THIS NEXT TIME).

THE 1ST PROGRAM LOADER STARTS AT !2F88 WITH A CALL TO AN "INTERPRETED NEST" VIA "M CALL" (RST 38 SUB #06+1) (SEE ARCADIAN VOL.5 PG.133). THIS BRANCHES TO !2F9D WHERE IT "MOVES" 4K OF DATA FROM !2007 TO !3006 INTO !6000 TO !6FFF! THEN IT DOES A "CLEAR SCREEN" WITH A "FILL" AND SETS THE COLORS AND ETC. WITH "SET OUT" AND "COL SET" (COLOR TABLE IS AT !2FE3). NOTE: TO CHANGE THE VALUE OF &(14), THE SCREEN INTERRUPT MUST BE DISABLED! (SEE "THE Z80 MINICOURSE" WHERE BARRY ELLERSON EXPLAINS WHY! "STR DIS" PRINTS THE WORD "REM0VE" (CHARACTOR STRING STARTS AT !2FDC) ON THE SCREEN JUST BEFORE "M RET" (RST 38 SUB 08) (SEE ARCADIAN VOL.5 PG.133) EXITS THE "INTERPRETED NEST" AND RETURNS US TO !2F8D WHERE "STR DIS" IS CALLED TWICE MORE BEFORE WE EXIT THE INTERPRETER VIA "X INTC". THE 1ST STRING (STARTING AT !3FAC) IS THE 1ST PROGRAM TITLE, AND THE 2ND STRING (STARTING AT !2FB8) PRINTS ALL THE REST. WE THEN JUMP TO !6FEA WHICH WAS COPIED INTO FROM !2FF1 BY "MOVE" EARLIER.

THE 1ST INSTRUCTION DISABLES THE INTERRUPT (FOR CARTRIDGE SWAPPING). NEXT IS A 5 BYTE LOOP THAT SIMPLY WAITS FOR A &(20) INPUT (FAR RIGHT KEYPAD ROW), AS THE "=" KEY ISN'T THE SAME KEY ON THE BASIC OVERLAY. AFTER THE CARTRIDGES ARE SWAPPED AND THE "=" KEY PRESSED, "DE" IS SET TO THE 2ND BYTE OF THE LINE INPUT BUFFER WHERE WE EXPECT TO SEE "SPACE SPACE SEMICOLON" AS THE FIRST 3 BYTES! THE STACK POINTER IS THEN SET 4 BYTES LOW SO AS NOT TO STEP ON THE JUMP INSTRUCTION WITH THE RETURN ADDRESS FROM THE "CALL !2164" WHICH IS THE SCREEN INTERRUPT INITIALIZATION ROUTINE IN BRB! UPON RETURN IT ENTERS BRB AT THE ADDRESS DETERMINED BY JOHN PERKINS AND LEROY J. FLAMM!

THE 2ND PROGRAM LOADER BEGINS AT !3F88 AND RUNS THE SAME INTERPRETED NEST (AT !2F9D) AS BEFORE, THEN CALLS "STR DIS" TO DISPLAY TITLE #2 AND "MOVE" TO "SWAP OUT" PROGRAMS TO THE 2ND BASIC PROGRAM TEXT. NOTE: THE # OF BYTES MOVED THIS TIME IS LESS (!0F81) SO AS TO LEAVE THE !6FEA THROUGH !6FFF AREA INTACT FROM THE 1ST "MOVE"! A 2ND "COL SET" AND "SET OUT" IS CALLED IN CASE A DIFFERENT COLOR SCHEME IS DESIRED (COLOR TABLE AT !3FA4 TO !3FAB), AND THEN AN "M JUMP" (RST 38 SUB #10+1) SENDS US TO !2F93 SO THAT THE INSTRUCTIONS COULD BE PRINTED AND ENTRY INTO BRB COULD BE SHARED BY BOTH PROGRAM LOADERS. IN FACT, EXCEPT FOR PRINTING THE TITLES, THE 2ND "MOVE", AND THE ALTERNATE COLORS (WHICH ARE OPTIONAL), ALL THE REST HAVE BEEN "TIME SHARED" (SEE ARCADIAN VOL.7 PG.74)!

YOU NOW HAVE ALL THE PARTS NECESSARY TO BUILD THE "4x2 AB MULTICART LOADER"! CAN YOU DO IT? NEXT TIME! KEEP BUGGIN'!