ADE

North Star ADVANTAGE Z80 Computer Emulator.

Series 5.   6th February 2019

TABLE OF CONTENTS

0.TABLE OF CONTENTS

ADE APPENDICES

1. INTRODUCTORY INFORMATION

1.1 Overview

ADE emulates the early 1980s North Star Advantage Z80 Computer.

ADE uses disk-image files which may contain any of North Star Computers' Disk Operating Systems of the period which were designed for the graphical hardware of the North Star Advantage: North Star GDOS, CP/M, etc.

The Advantage used a double-density-controller which could access both sides of the disk and used 512-byte sectors giving 350K of storage. The Advantage had space for two floppy-disk drives. Later, hard-disk capability was added to the North Star Advantage by replacing one of the floppy-disk drives with a 5-inch hard drive. The hard drive was available in several capacities from 5 megabytes up to 30 megabytes.

ADE contains 64K of RAM, a Z80 microprocessor emulator, and a keyboard-handler. There is also a behind-the-scenes monitor which emulates the operator's interaction with the hardware, such as inserting or removing floppy disks, and organizing the interaction between the host linux machine and the virtual Z80 machine.

Other parts of the emulator are concerned with the North Star Advantage-specific components, such as the data and control ports of the serial and parallel I/O, the double-density floppy-disk controller with its boot-up PROM, the fixed-disk controllers and the bit-mapped display screen.

1.2 ATTRIBUTIONS FOR OTHERS' CODE in ADE

ADE's Z80 emulation code pretty much comes from yaze, a CP/M emulator written by Frank Cringle. Most of the screen-display code was obtained from Gingold and Wolman’s TRS80 Emulator. North Star specific amendments such as memory-mapped floppy-disk I/O, and a few other additions such as Mode 2 interrupt code were made by Jack Strangio.

ADE's Z80 disassembly code comes from Marat Fayzullin's 1999 DAsm code with some local alterations.

The rest of ADE cannot be blamed on anyone else but myself.
Jack Strangio, January 2019

1.3 THANKS

I have only the greatest appreciation for all those who have helped me in my rather idiosyncratic quest to write emulators of the North Star Computers' Horizon and Advantage hardware. The Horizon was my first computer which took more than 40 hours to build during the course of several weeks in late 1978. The thousands of solder-joints literally burnt-out a new soldering iron. It says a lot for the quality of the instruction manual that most of the time I really had no idea what each step did but at the end (once my half-dozen wiring mistakes were fixed) I had assembled a computer which worked perfectly.

I'd like to mention a few of the people who have generously helped me:

Dave Dunfield, who gave me a lot of help in many different areas. He also had quite a few North Star floppy disk-image files that I have rummaged though over the years.

The Late Don Maslin, who got me started on the double-density floppy work by transferring a lot of data from my old 10-sectored disks to disk-image files.

Martin Brown, who helped me along the way with scanning old Disk-Controller manuals, without which I was more clueless than usual.

Howard Harte , whose regard for old computers means he has taken the trouble to maintain lots of North Star Manuals:

http://www.hartetechnologies.com/manuals/Northstar/

Bitsavers.org. (http://www.bitsavers.org/bits/NorthStar/). Thanks to them, there are still quite a few disk-image files around for the North Star Horizon.

Allison Parent, for indicating where I could get hold of information regarding the HD5X controller board.

Jon Hales, whose encouragement got me working again when I had struck a brick wall with the Advantage floppy disk controller code and my enthusiasm for digging down amongst the Advantage's technical documentation was seriously flagging, and who was very generous with his time and extra Advantage Software.

1.4 FLOPPY DISKS AND A HARD DISK SUPPLIED WITH ADE

Several floppy disks are supplied with ADE to get you up and going quickly. They are stored in the 'disks' subdirectory. Generally speaking, the floppy-images supplied with ADE are from true North Star Advantage floppies. We emulator nerds take pride in using either the original software disks themselves, or disk-images which are exact copies of those.

Unfortunately, I only have one sample of Advantage’s GDOS/GHDOS and that seems to be a copy of a very early release. At one point when it boots up GHDOS, it checks the version of the hard-disk and expects it to be version ‘1.0’. As far as I know there was only one hard drive supplied by North Star which was classed as version 1.0, and that was the SG5A (5 megabytes). All other ensuing North Star hard drives are classed as Version 2.0.

Consequently, I was forced to restrict the use of the DOSGBASI.NSI (ghdos 2.0.0) boot floppy to working with a version 1.0 five megabyte hard drive (supplied as SG5AX.NHD).

Some of these archive disks have been renamed to allow their uses to be self-explanatory. Non-official disk names are designated in lower-case.

adv_cpm_120_wm.nsi	North Star CP/M 1.2.0 Boot Disk for Hard Disks
adv_ghdos_200.nsi	North Star GDOS 2.0.0 Boot Disk
DEMODIAG.NSI	North Star Graphics Demo and Diagnostics
hd_supplement_disk.nsi	North Star Hard Drive Formatting and Testing

Two hard-disks are supplied as samples. One is a Version 2.0 ten megabyte hard disk, (cm10ex.nhd). It has been set up have several 1-megabyte CP/M virtual disks, ranging from CPMA to CPMH, and provision for the two floppy CP/M drives, I: and J: which are used for transferring files to and from the larger virtual disks. When ADE is booted with the CP/M Boot Disk, it will take you to the A: directory on the hard-disk on startup.

The other hard drive is a smaller 5-megabyte Version 1.0 hard drive (sg5ax.nhd). That is useful enough to use with the restrictive adv_ghdos_200.nsi GHDOS 2.0.0 floppy boot-disk.

Larger hard drives are more useful in practice, but these two hard drive images should give a good feel for the emulator.

There are two ready-prepared scripts ‘ac’ and ‘ad’. Running ‘ac’ will set up a CP/M boot disk and a 10-megabyte hard drive and run that configuration, running ‘ad’ will set up the GHDOS boot disk and the 5-megabyte hard-drive and run that configuration. These two scripts should be run from the command-line within the source/executables file directory. They may be amended and then run from elsewhere if wanted.

1.5 LIST OF SCREEN CAPTURES

ADE was originally ncurses-based. Subsequently the display was rewritten to use the X Windowing System’s Xlib. The current version of ADE still shows it’s cobbled-together character by even using ‘faked’ ncurses functions at various places .

ADE looks like a typical "green-screen" terminal of the 70's-80's period.

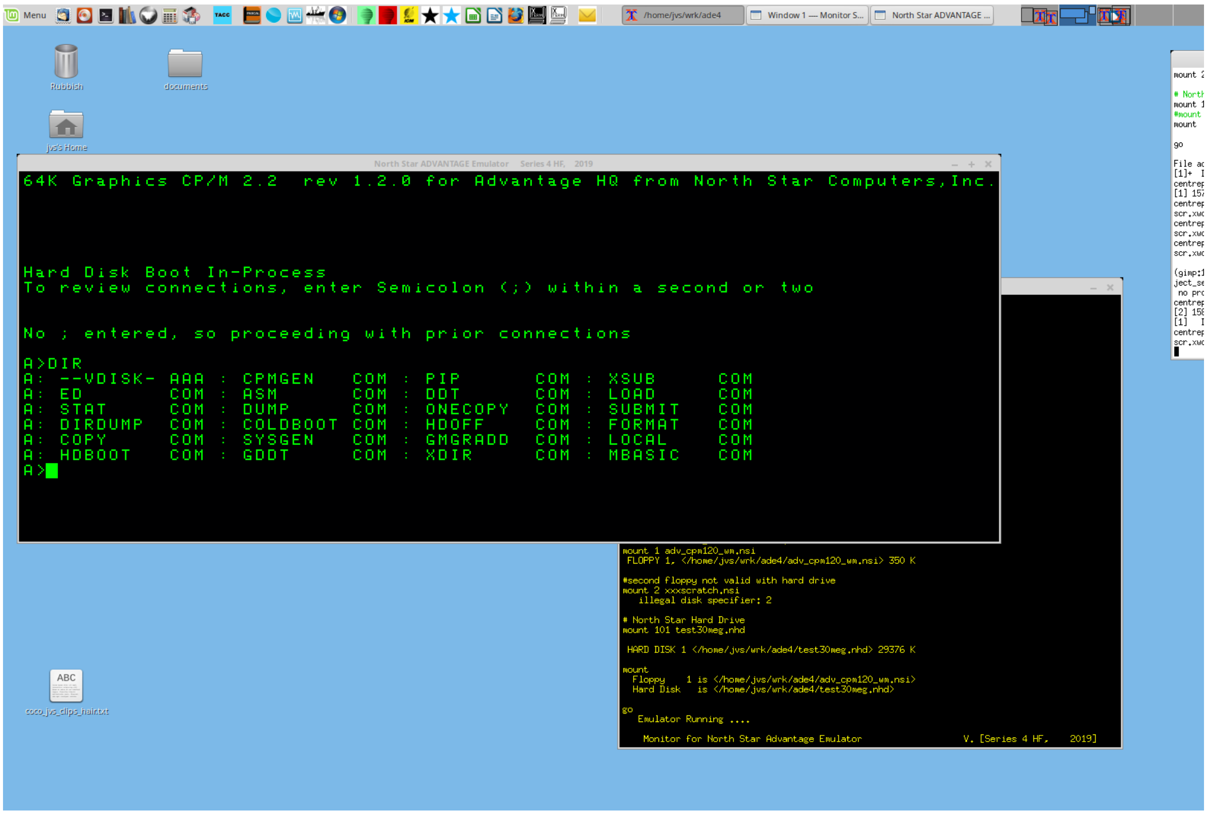
When you hit the default interrupt key (F3), you will be shown the "control console" terminal. This will display as an "amber screen" terminal.

To get back to the running emulator, type in the "go" command when you have finished with the control console.

Other screen shots are included later in the User manual to show specific details if necessary.

HDOS running in 25x80 character screen format. The status line at the bottom of the monitor screen displays the Name of the emulator, then some version information. Other versioning information is given in the title of the main Advantage screen.

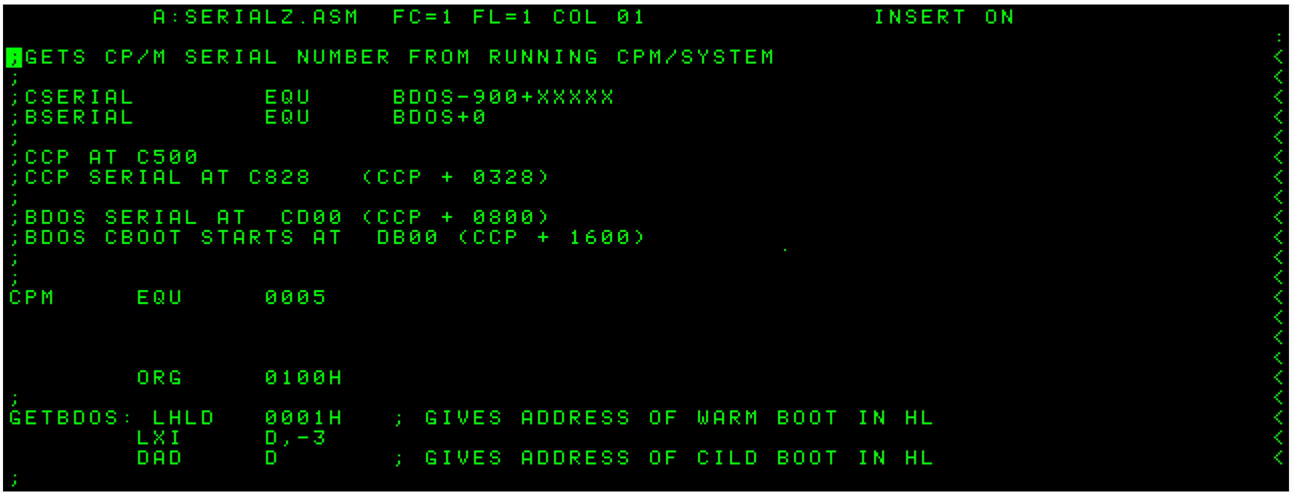
Fig 1. ADE splash screen followed by CP/M running.



```
64K Graphics CP/M 2.2  rev 1.2.0 for Advantage HQ from North Star Computers,Inc.



Hard Disk Boot In-Process
To review connections, enter Semicolon (;) within a second or two

No ; entered, so proceeding with prior connections

A>DIR
A: --VDISK- AAA : CPMGEN   COM : PIP      COM : XSUB     COM
A: ED       COM : ASM      COM : DDT      COM : LOAD     COM
A: STAT     COM : DUMP     COM : ONECOPY  COM : SUBMIT   COM
A: DIRDUMP  COM : COLDBOOT COM : HDOFF    COM : FORMAT   COM
A: COPY     COM : SYSGEN   COM : GMGRADD  COM : LOCAL    COM
A: HDBOOT   COM : GDDT     COM : XDIR     COM : MBASIC   COM
A>
```

```
mount 1 adv_cpm120_wm.nsi
 FLOPPY 1, </home/jvs/wrk/ade4/adv_cpm120_wm.nsi> 350 K

#second floppy not valid with hard drive
mount 2 xxxscratch.nsi
   illegal disk specifier: 2

# North Star Hard Drive
mount 101 test30meg.nhd

 HARD DISK 1 </home/jvs/wrk/ade4/test30meg.nhd> 29376 K

mount
  Floppy    1 is </home/jvs/wrk/ade4/adv_cpm120_wm.nsi>
  Hard Disk   is </home/jvs/wrk/ade4/test30meg.nhd>

go
   Emulator Running ....

    Monitor for North Star Advantage Emulator                 V. [Series 4 HF,    2019]
```

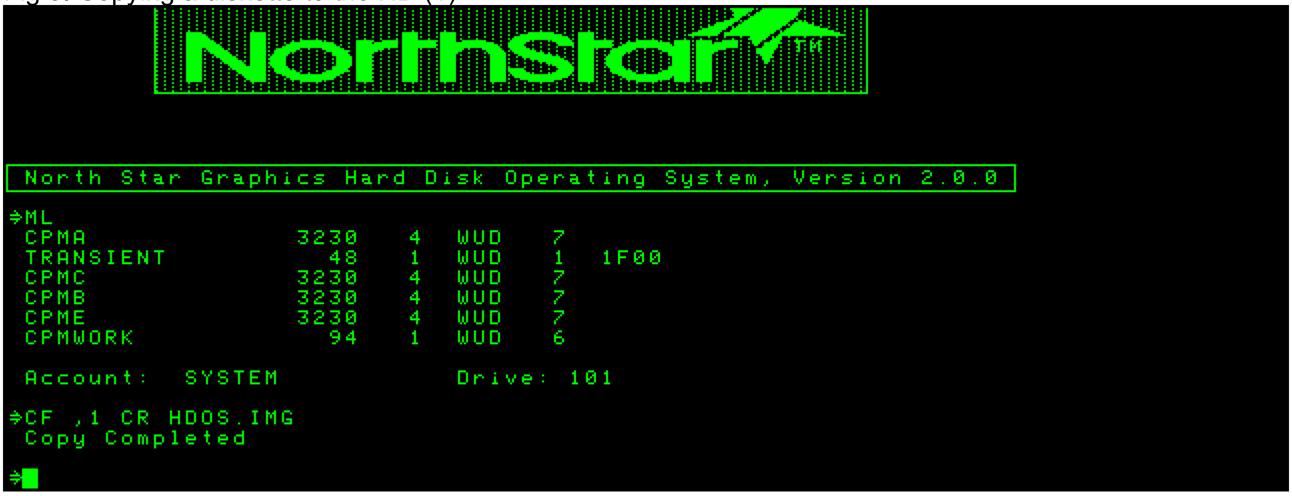
Fig 2. A version of Word Star running.



```
         A:SERIALZ.ASM  FC=1 FL=1 COL 01                 INSERT ON

;GETS CP/M SERIAL NUMBER FROM RUNNING CPM/SYSTEM
;
;CSERIAL        EQU     BDOS-900+XXXXX
;BSERIAL        EQU     BDOS+0
;
;CCP AT C500
;CCP SERIAL AT C828   (CCP + 0328)
;
;BDOS SERIAL AT  CD00 (CCP + 0800)
;BDOS CBOOT STARTS AT  DB00 (CCP + 1600)
;
;
CPM     EQU     0005


        ORG     0100H
;
GETBDOS: LHLD   0001H   ; GIVES ADDRESS OF WARM BOOT IN HL
        LXI     D,-3
        DAD     D       ; GIVES ADDRESS OF CILD BOOT IN HL
;
```

1.6 COPYING A DISKETTE TO a FLOPPY IMAGE-FILE ON A GHDOS HARD DRIVE AND THEN COPYING THAT DISK_IMAGE BACK TO CLONE OTHER FLOPPIES.

Fig 3. Copying a diskette to the HD (1)



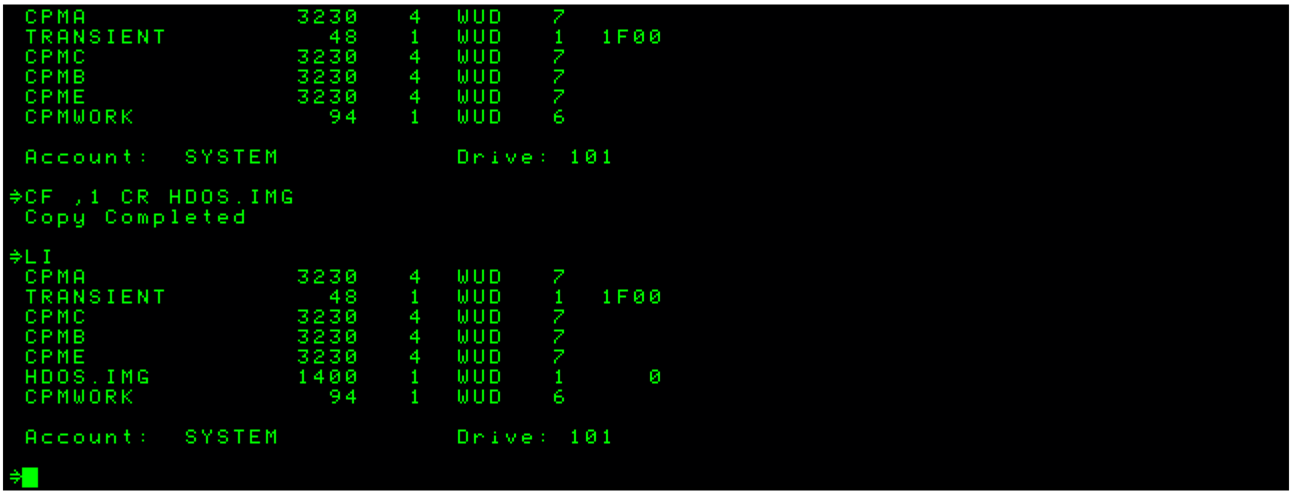
```
 North Star Graphics Hard Disk Operating System, Version 2.0.0

⇒ML
 CPMA              3230   4  WUD   7
 TRANSIENT           48   1  WUD   1  1F00
 CPMC              3230   4  WUD   7
 CPMB              3230   4  WUD   7
 CPME              3230   4  WUD   7
 CPMWORK             94   1  WUD   6

 Account:  SYSTEM              Drive: 101

⇒CF ,1 CR HDOS.IMG
 Copy Completed

⇒
```

Copy the diskette from Floppy Drive number 1 to the GHDOS Hard Drive with

```
CF  ,1  CR  HDOS.IMG
```

After it's completed the 1400-block (350K) HDOS.IMG diskette image-file is available. That could be transferred over the net as a North Star floppy-image, or used to clone further floppies.

Fig 4. Copying a diskette to the HD (2)



```
 CPMA              3230   4  WUD   7
 TRANSIENT           48   1  WUD   1  1F00
 CPMC              3230   4  WUD   7
 CPMB              3230   4  WUD   7
 CPME              3230   4  WUD   7
 CPMWORK             94   1  WUD   6

 Account:  SYSTEM              Drive: 101

⇒CF ,1 CR HDOS.IMG
 Copy Completed

⇒LI
 CPMA              3230   4  WUD   7
 TRANSIENT           48   1  WUD   1  1F00
 CPMC              3230   4  WUD   7
 CPMB              3230   4  WUD   7
 CPME              3230   4  WUD   7
 HDOS.IMG          1400   1  WUD   1      0
 CPMWORK             94   1  WUD   6

 Account:  SYSTEM              Drive: 101

⇒
```

To clone further floppies, change the diskette in the floppy drive. In ADE this is done by unmounting the diskette in 'floppy drive 1' with

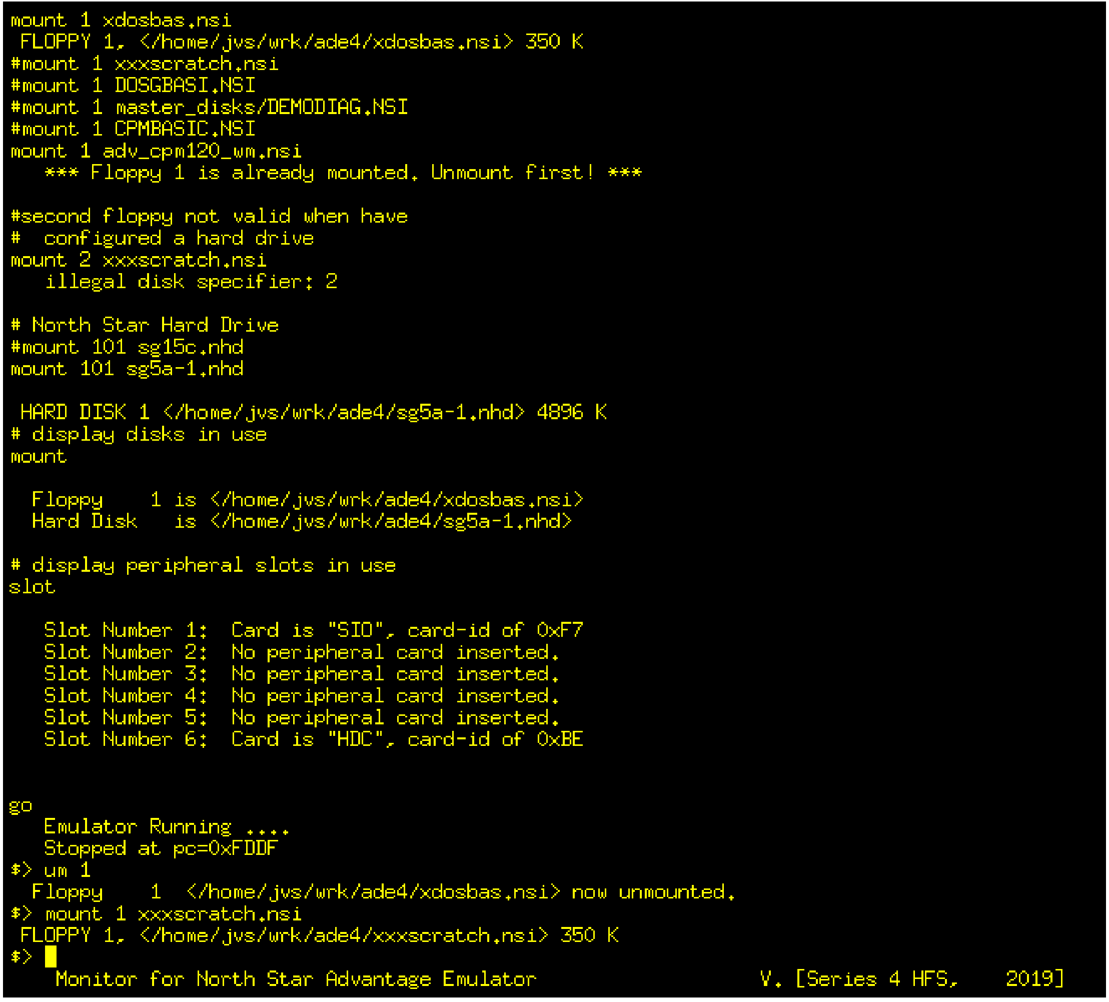
```
umount 1    (Or even  just " um 1 ")
```

and then inserting another diskette image-file into the drive with

```
mount 1 xxxscratch.nsi
```

and then returning to ADE itself with a 'go' command on the monitor command-line. (See image on next page.)

Fig 5. Copying a diskette to the HD (3)



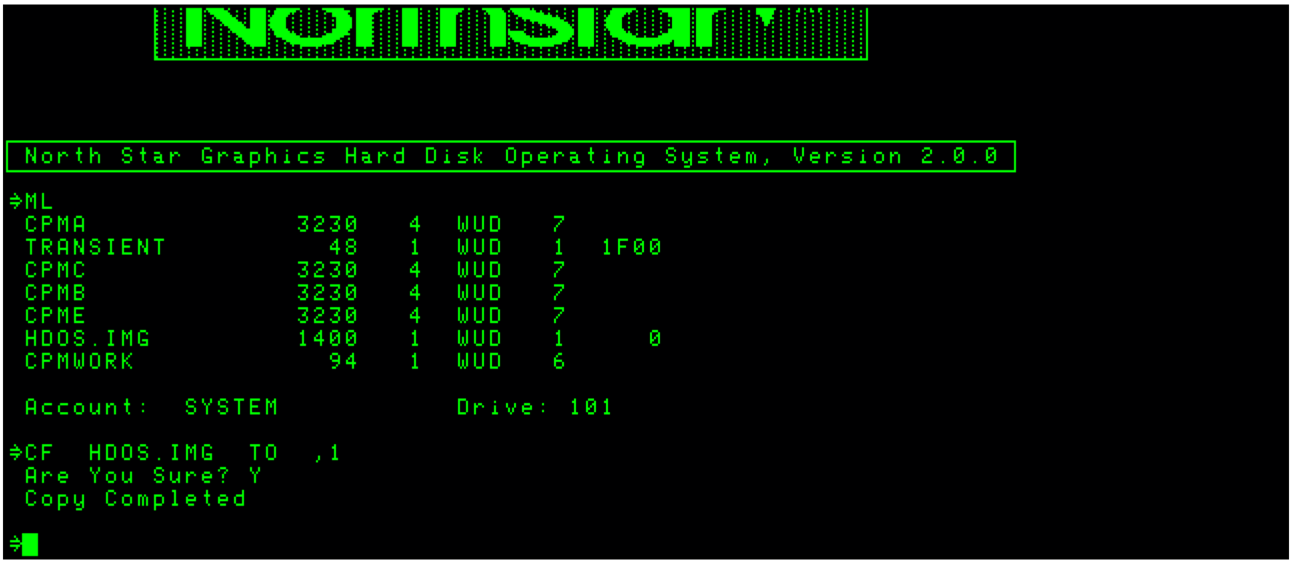
```
mount 1 xdosbas.nsi
 FLOPPY 1, </home/jvs/wrk/ade4/xdosbas.nsi> 350 K
#mount 1 xxxscratch.nsi
#mount 1 DOSGBASI.NSI
#mount 1 master_disks/DEMODIAG.NSI
#mount 1 CPMBASIC.NSI
mount 1 adv_cpm120_wm.nsi
   *** Floppy 1 is already mounted. Unmount first! ***

#second floppy not valid when have
#  configured a hard drive
mount 2 xxxscratch.nsi
   illegal disk specifier: 2

# North Star Hard Drive
#mount 101 sg15c.nhd
mount 101 sg5a-1.nhd

 HARD DISK 1 </home/jvs/wrk/ade4/sg5a-1.nhd> 4896 K
# display disks in use
mount

  Floppy    1 is </home/jvs/wrk/ade4/xdosbas.nsi>
  Hard Disk   is </home/jvs/wrk/ade4/sg5a-1.nhd>

# display peripheral slots in use
slot

   Slot Number 1:  Card is "SIO", card-id of 0xF7
   Slot Number 2:  No peripheral card inserted.
   Slot Number 3:  No peripheral card inserted.
   Slot Number 4:  No peripheral card inserted.
   Slot Number 5:  No peripheral card inserted.
   Slot Number 6:  Card is "HDC", card-id of 0xBE


go
   Emulator Running ....
   Stopped at pc=0xFDDF
$> um 1
  Floppy    1  </home/jvs/wrk/ade4/xdosbas.nsi> now unmounted.
$> mount 1 xxxscratch.nsi
 FLOPPY 1, </home/jvs/wrk/ade4/xxxscratch.nsi> 350 K
$> 
    Monitor for North Star Advantage Emulator                   V. [Series 4 HFS,    2019]
```

After that copy the diskette image-file on the hard-drive to another diskette with

CF HDOS.IMG TO ,1

After confirming the command which will completely change the diskette's contents, the job proceeds to completion.

Fig 6. Copying a diskette to the HD (4)



```
 North Star Graphics Hard Disk Operating System, Version 2.0.0

$ML
 CPMA              3230    4   WUD    7
 TRANSIENT           48    1   WUD    1   1F00
 CPMC              3230    4   WUD    7
 CPMB              3230    4   WUD    7
 CPME              3230    4   WUD    7
 HDOS.IMG          1400    1   WUD    1      0
 CPMWORK             94    1   WUD    6

 Account:   SYSTEM             Drive: 101

$CF   HDOS.IMG   TO   ,1
 Are You Sure? Y
 Copy Completed

$
```

Fig 3. Copying a diskette to the HD (1)

1.7 CONFIGURATION FILES

1.7 LOCATION OF ADE CONFIGURATION FILES:

~/aderc

A user's default configuration file in the /home directory.

You can also use **aderc** as a program configuration file in the current working directory.

$PWD/aderc

This location is the best for general use

1.8 PRECEDENCE OF CONFIGURATION FILES

ADE will look for a 'aderc' file first in the current-working-directory. If it isn't found there it will look for it in the user's $HOME directory. The configuration file is a set of control console commands placed in a file at a pre-specified location. Generally, the most useful and specific file is the 'aderc' file placed in the current working directory or in the user's home directory.

1.9 USER CONFIGURATION FILES: aderc

A more complex user-level configuration file in the current-working-directory. See the next page for a complete copy of a real file:

1.10 Complete Sample Real-World Configuration File

```
####### ADERC SAMPLE FILE ####

#capslock on
caps on

delay off

# specify our log file locations
# care - lots of debug output can be produced
log xlog
screenlog /tmp/screenlog

#                     Debugging BitMap
# one or more values added to allow less or more debugging
# information to be logged. default log file is called 'xlog'

#define DISASS           0x01
#define MOTHERBOARD      0x02
#define DEV              0x04
#define MEM              0x08
#define FDC              0x10
#define HDC              0x20
#define BIOS_EMULATE     0x40
#define CMD/STATUS      0x080
#define TERM            0x100
#define CLI             0x200
#define QUEUE           0x400
#define X11            0x0800
#define XEVENT         0x1000
#define KEYB           0x2000
#define INFO           0x4000
#define TRAP          0x08000
#define ALL           0x10000

#brkdbg 29

debug a9

#break 0100
# serial2 and parallel port
attach s2o out_serial
attach s2i in_serial
attach plo out_parallel

#peripherals in slots
slot 6 hdc
slot 4 pio
slot 1 sio

# North Star Advantage Floppy
mount 1 xdosbas.nsi
#mount 1 xxxscratch.nsi

mount 1 adv_cpm120_wm.nsi

mount 2 xxxscratch.nsi

# North Star Hard Drive
mount 101 sg5a-1.nhd
# display disks in use
mount
# display peripheral slots in use
slot

go
```

1.11 ADVANTAGE MACHINE CONFIGURATION

1.11 Slots and Peripheral cards

The North Star Advantage is normally set up so that the CP/M I/O Peripheral Card devices are allocated to slot numbers 1 and 2. Slot number one contains the device specified as the 'LIST' device. This could be either a parallel or a serial printer, according to whether a parallel card (PIO) or a serial card (SIO) is inserted into Slot Number 1.

The 'READER' and 'PUNCH' devices are allocated to Slot Number 2. This pretty much requires these devices to be managed by a serial device, and the peripheral card in Slot Number 2 is to be a serial card (SIO).

The North Star software assumes that the Peripheral Device cards will be inserted in the slots as specified above.

The Hard Disk Controller card is not allocated to any specific Slot Number, as far as I know. But I believe I have seen somewhere in the documentation that it's normally in Slot 6.

In summary, ADE when started has the following configuration:

Slot 1	PIO Card
Slot 2	SIO Card
Slot 6	HDC Card

Each peripheral card has a numeric card-ID so that the Advantage can tell what sort of card is inserted into any particular slot.

HD Controller Card	(Later Revision)	BF hex
HD Controller Card	(Early Revision)	BE hex
SIO Serial Card		F7 hex
PIO Parallel Card		DB hex

The supplied 'Makefile' used to generate ADE from source is configured to automatically include the source code which emulates the three peripheral cards when recompiling. The cards pre-inserted into the card-slots may be changed manually by using the monitor 'slot' command if deemed necessary.

As supplied, the ADE emulation will produce a 'North Star Advantage' which contains 2 floppy drives, 1 hard drive, 1 SIO card, and 1 PIO card. This was not really the standard configuration, as we have taken up the option of having two floppies as well as a hard drive. The original Advantage did not have the physical space inside the cabinet for doing that.

Technical Note: The floppy controller has the capacity to control four floppy drives. However, in a short-sighted decision when planning the Advantage and its hardware, the two drive-select bits in the Drive Control Register, which could have been used to designate 4 separate drives, can only designate two floppies, depending on whether bit 0 (floppy unit 1) or bit 1 (floppy unit 2) is high. Bits 2 and 3 are reserved, and also could have been used to designate an extra two floppies in a continuation of the drive-select rule. But that never happened.

2. THE CONTROL CONSOLE

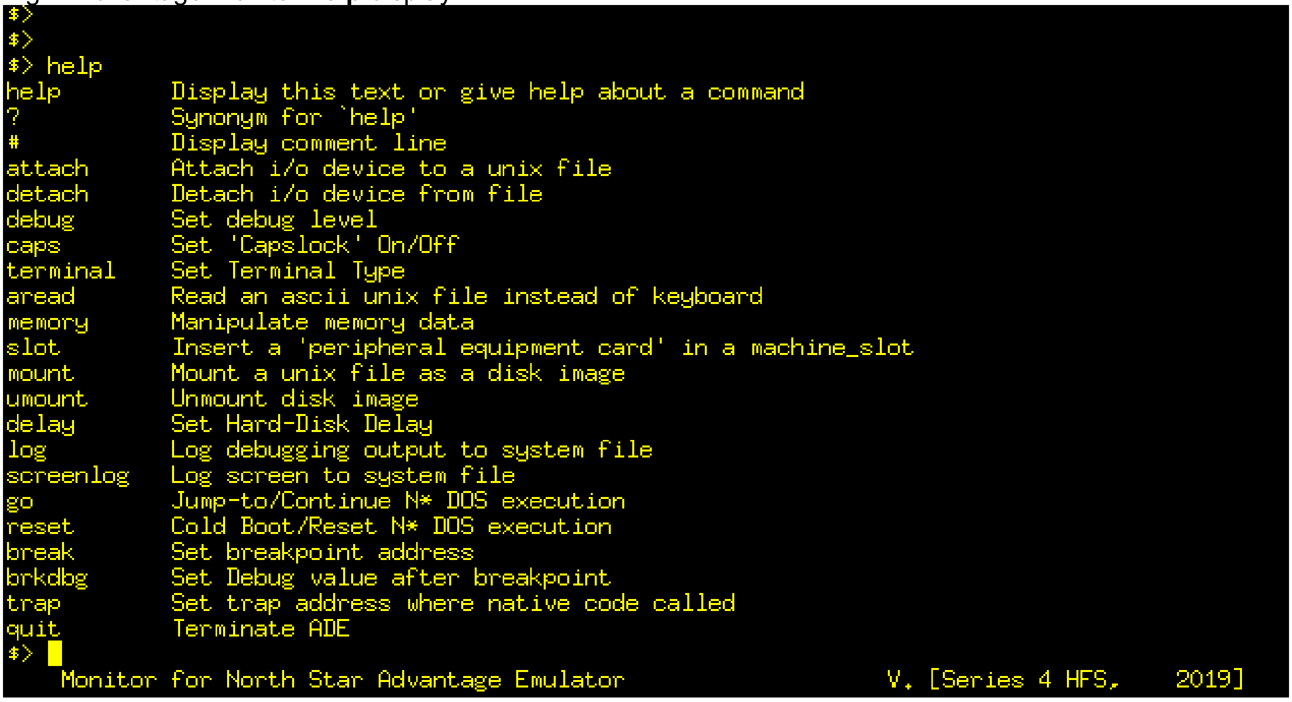
The control console is used to do the nuts-n-bolts stuff behind the workings of the emulator, such as 'inserting' floppy disks, changing the CapsLock, altering the keyboard delay, altering which key will be used as the interrupt-key, etc.

IMPORTANT NOTE:
The control console ('monitor') is being phased out. The number of commands available will gradually be reduced and be replaced by Function Key activities, or by GUI activities. See also Section 3, 'Function Keys and GUI Controls'.

2.1 help   Show commands and brief synopsis of how they are used.

help <command>   displays more information about <command

Fig 7. Advantage Monitor **help** display



```
$>
$>
$> help
help        Display this text or give help about a command
?           Synonym for `help'
#           Display comment line
attach      Attach i/o device to a unix file
detach      Detach i/o device from file
debug       Set debug level
caps        Set 'Capslock' On/Off
terminal    Set Terminal Type
aread       Read an ascii unix file instead of keyboard
memory      Manipulate memory data
slot        Insert a 'peripheral equipment card' in a machine_slot
mount       Mount a unix file as a disk image
umount      Unmount disk image
delay       Set Hard-Disk Delay
log         Log debugging output to system file
screenlog   Log screen to system file
go          Jump-to/Continue N* DOS execution
reset       Cold Boot/Reset N* DOS execution
break       Set breakpoint address
brkdbg      Set Debug value after breakpoint
trap        Set trap address where native code called
quit        Terminate ADE
$> █
    Monitor for North Star Advantage Emulator                  V. [Series 4 HFS,    2019]
```

2.2 ADE CONTROL CONSOLE INPUT-OUTPUT CONTROL

2.2 attach   Attach I/O device to a unix file.

attach <iodev> <file>

attach slo /tmp/printout

Without arguments, attach lists the current attachments.

With arguments, attach attaches <iodev> to the <unix file>, where <iodev> is one of sli, slo, pli, plo. These are the North Star Advantage I/O interfaces where sli = serial-in, slo = serial-out, pli = parallel-in, plo = parallel-out.

2.3 detach   Detach I/O device from file.

detach <iodev>

detach slo

detach closes the unix file attached to <iodev>. See **attach**

2.4 <u>aread</u> Read in an ASCII unix file instead of keying in the characters from the keyboard.

aread <text file>

aread /tmp/program1

The linefeed character (0A H) is converted to a carriage-return character (0D H) automatically. Note that aread can supply 'keyboard' input faster than a lot of programs (e.g. Word Star) can handle gracefully.

2.5 <u>caps</u> Toggle 'Capslock' On/Off.

caps [on | off]
caps

When ADE starts, the capslock is 'Off'. NSDOS requires the capslock to be toggled on. This can be achieved either by the actual CAPSLOCK key or by the use of this command. I usually put the 'caps on' line in the configuration file.

2.6 <u>delay</u> Toggle a delay into the Hard-Drive Controller subsystem.

delay [on | off]
delay

The delay in the hard-drive subsystem will slow down the operation of the emulator. It is generally only used in a very few required places when the emulator outruns the user. e.g. when the system is looking for a ';' character to signal the need to edit CP/M Connections on the hard-drive and the emulator gets to the following prompt before the user can react.

The HDC delay should almost always be set to 'off'.

2.7 ADE DISK MANAGEMENT

2.7 <u>mount</u> Mount a unix file as North Star disk image.

mount
mount [-r] <drive> <file>

mount 1 /home/fred/cpm_22.nsi

mount without arguments merely lists the mount tables.

```
$>mount
   Floppy     1 is </a/1/northstar/WRK_CPMBOOT_DRI.NSI>
   Floppy     2 is   ** not mounted. **
   Hard Disk 1 is </a/1/northstar/30mb101.nhd>
```

mount with arguments mounts <file> as N* DOS floppy disk <drive>.

<file> must contain a North Star filesystem disk-image. If '-r' option is used, then <file> is mounted READ-ONLY. The '-r' option is not active with hard disks.

The unit numbers for floppy disks and hard disks follow the North Star HDOS convention. Units 1 and 2 are floppy disks, unit 101 is the hard disk.

2.8 umount Unmount a North Star disk-image.

umount <drive number>

umount 2

umount closes the file associated with disk drive <drive number> and frees the resources.

2.9 slot Install a Peripheral Card in an Advantage Slot.

slot <no arguments following>

slot

Slot with no arguments merely gives a list of the six available slots of the North Star Advantage and whether or not any peripheral cards are installed.

slot <slot number>

slot 2

Slot followed by just a slot-number as argument will uninstall any peripheral card installed in that slot.

slot <slot number> <peripheral card type>

slot 3 pio

Slot with two arguments, the slot number and a peripheral type, will install a peripheral card of the type given into the specific slot.

There are six slots available, 1 to 6, and three peripheral types available for the emulator to use, being:

HDC	Hard Disk Controller
SIO	Serial I/O Card
PIO	Parallel I/O Card

Both upper and lower case can be used for the peripheral card-type.

2.10 ADE DEVELOPMENT ASSISTANCE

2.10 memory   Manipulate memory in the North Star Horizon virtual machine.

This subsystem has usage similar to CP/M 'DDT' or MSDOS 'DEBUG'

Commands:

Upper or lower case commands are accepted
<xxx> is required parameter
[xxx] is optional parameter

C <start address> <finish address> <start of compared block>
c 1a00 2000 2a00

Compare two equal-length blocks of memory. Only the bytes which are different will be displayed with location and values.

D [start address] [finish address]
d 0 12FF

Display the block of memory selected, showing bytes as hexadecimal and ASCII. If no start and end address specified, the command will continue for 100 H bytes from where it ended last.

E <start address>
E 2CFF

Examine/change values at memory locations. The operation is stopped when no new value is entered, just a plain 'enter'.

F <start address> <finish address> <fill byte>
f 1000 2000 55

Fill a block of memory with byte-value specified by <fill byte>.

H <value> <value>
h 1267 abcd

Hex arithmetic results of the addition of two values and the subtraction of the second value from the first value.

L [load address]
l 2a00

Load the file (previously specified by the 'N' command) into memory. If a load-address is not specified the file will be loaded into location 0000 H.

M <source start address> <source end> <destination>
M 4d00 5000 6d00

Move the block of memory specified by the block's start and end into memory beginning at the destination address.

N <file name>
N xtest.bin.bas

Change active file-name which specifies which unix file will be used for 'load' and 'write' operations.

Q

Quit from the memory subsystem back to the emulator's control console.

S <start address> <end address> "string"
S 0100 4fff "North Star"

Search for string delimited by quotes (") within memory block specified.

S <start address> <end address> byte.byte...
S 0100 4FFF 38.4F.4D.60

Search for a list of bytes specified in hex. and joined by dots.

W [number of bytes in hex]
w 5c00

Write to the disk file previously specified by the 'N' command. Write the number of bytes specified.

2.11 debug Set Debug Level.

debug [<hex level>]

debug c3

debug with a hexadecimal numeric argument sets the debug level silently to the level specified. (0 = no debugging logged).

debug

debug with no argument displays the current bit-mapped level of debug logging to the **log** file, displays the current settings, then asks for a selection.   (See image below on Page XXX)

2.12 log log the debug information to unix disk file.

log <file>

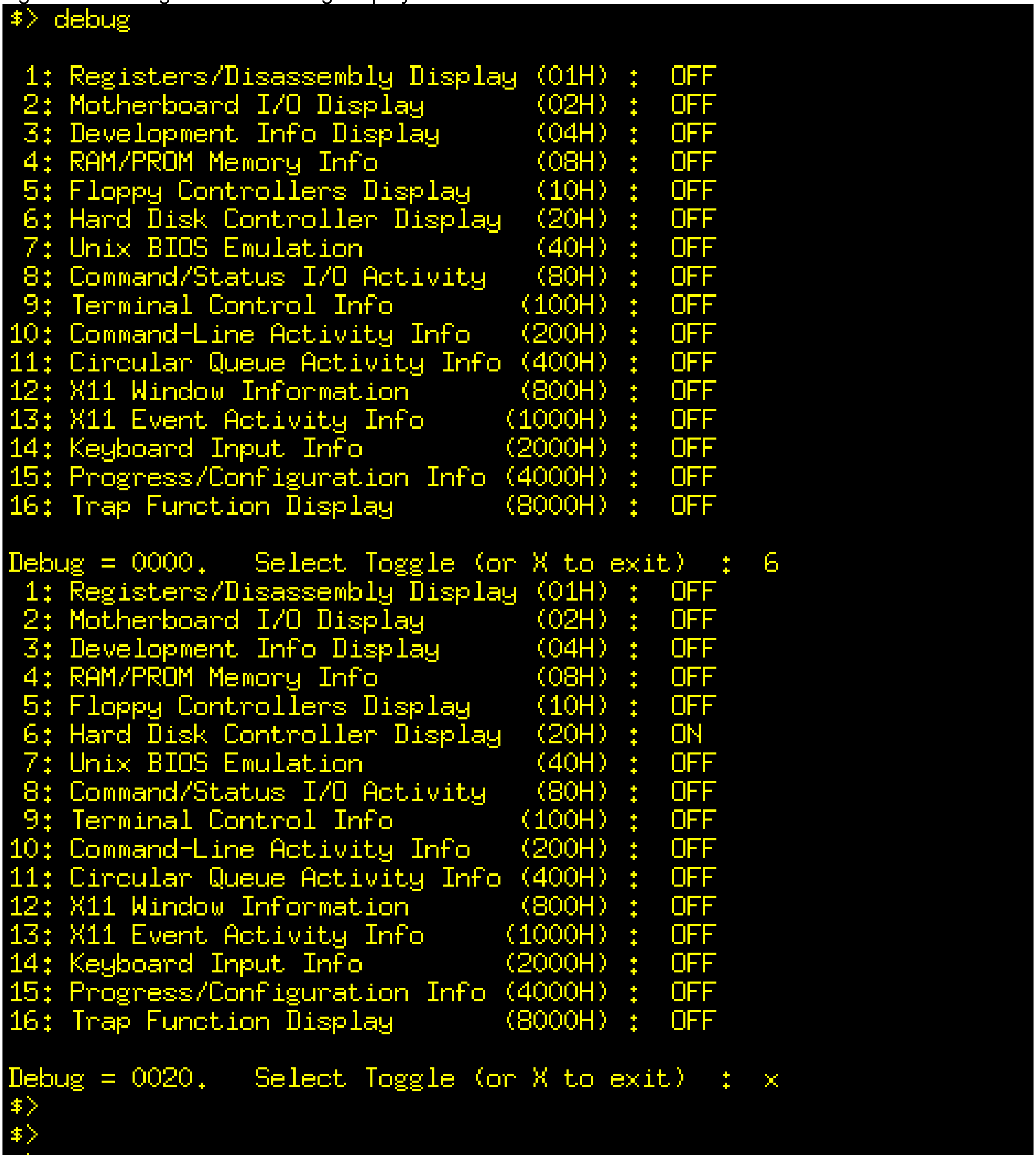
log xlog

sends debugging output to the specified unix file.

Take care, because the quantity of information sent to the log file can reach the maximum size (2 Gig in 32-bit systems, whole disk or filesystem in 64-bit systems) within a fairly short time.

Unless you're doing development on the North Star Emulator itself, it probably will not be useful to use any debugging or logging at all.

Fig 8.   Advantage Monitor Debug Display



```
$> debug

 1: Registers/Disassembly Display (01H) :  OFF
 2: Motherboard I/O Display       (02H) :  OFF
 3: Development Info Display      (04H) :  OFF
 4: RAM/PROM Memory Info          (08H) :  OFF
 5: Floppy Controllers Display    (10H) :  OFF
 6: Hard Disk Controller Display  (20H) :  OFF
 7: Unix BIOS Emulation           (40H) :  OFF
 8: Command/Status I/O Activity   (80H) :  OFF
 9: Terminal Control Info        (100H) :  OFF
10: Command-Line Activity Info   (200H) :  OFF
11: Circular Queue Activity Info (400H) :  OFF
12: X11 Window Information       (800H) :  OFF
13: X11 Event Activity Info     (1000H) :  OFF
14: Keyboard Input Info         (2000H) :  OFF
15: Progress/Configuration Info (4000H) :  OFF
16: Trap Function Display       (8000H) :  OFF

Debug = 0000.   Select Toggle (or X to exit)  :  6
 1: Registers/Disassembly Display (01H) :  OFF
 2: Motherboard I/O Display       (02H) :  OFF
 3: Development Info Display      (04H) :  OFF
 4: RAM/PROM Memory Info          (08H) :  OFF
 5: Floppy Controllers Display    (10H) :  OFF
 6: Hard Disk Controller Display  (20H) :  ON
 7: Unix BIOS Emulation           (40H) :  OFF
 8: Command/Status I/O Activity   (80H) :  OFF
 9: Terminal Control Info        (100H) :  OFF
10: Command-Line Activity Info   (200H) :  OFF
11: Circular Queue Activity Info (400H) :  OFF
12: X11 Window Information       (800H) :  OFF
13: X11 Event Activity Info     (1000H) :  OFF
14: Keyboard Input Info         (2000H) :  OFF
15: Progress/Configuration Info (4000H) :  OFF
16: Trap Function Display       (8000H) :  OFF

Debug = 0020.   Select Toggle (or X to exit)  :  x
$>
$>
```

All debugging output is sent to the **log** file (See 'log' below).

2.13 screenlog log the screen output to unix disk file.

screenlog <file>

screenlog /tmp/screen.txt

sends all screen output to the specified unix file. This can be handy if output scrolls off the top of the screen before you can read it.

2.14 break Set a breakpoint address to stop the emulator and return to the control console prompt.

break <4-hex_digit address>

break 0a5c

2.15 brkdbg Sets an automatic debug level after a 'break' program stop.

brkdbg <debug level>

brkdbg a9

2.16 trap Set a trap address to stop the emulator, perform a user-specified unix operation, return to the emulator and continue.

trap <4-hex_digit address>

trap 2a00

A dummy function is included in the emulator source (trap.c) which merely prints the trap address and the register values. The trap function could be used to access parts of the host unix system or perform any other required operation.

2.17 PROGRAM FLOW CONTROL

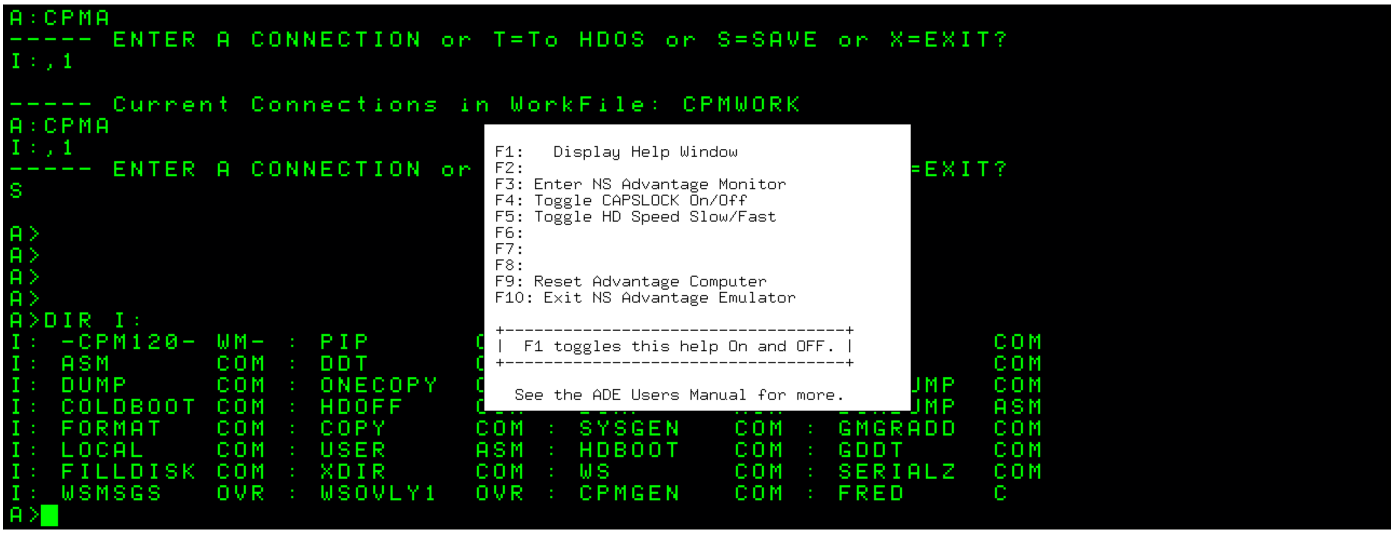
2.17 go [new address] Start or continue North Star Horizon code execution.

2.18 reset Reset. NO LONGER AVAILABLE. See Function Key F9.

2.19 quit Terminate ADE. Stop running, close files, close windows.

3. FUNCTION KEYS AND GUI CONTROLS

3.1 Function Key F1 – Help



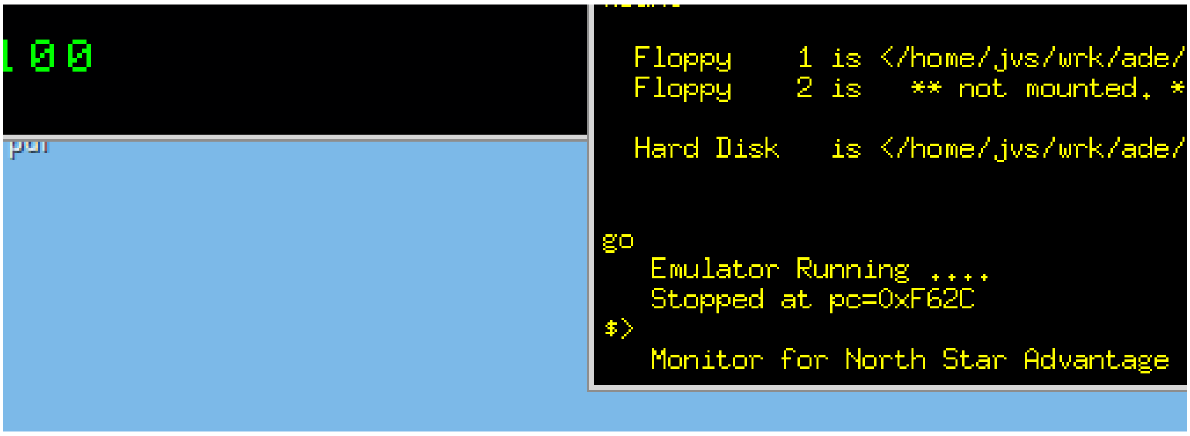
```
A:CPMA
----- ENTER A CONNECTION or T=To HDOS or S=SAVE or X=EXIT?
I:,1

----- Current Connections in WorkFile: CPMWORK
A:CPMA
I:,1
----- ENTER A CONNECTION or                                =EXIT?
S

A>
A>
A>
A>
A>DIR I:
I: -CPM120- WM-      : PIP      C                                      COM
I: ASM      COM      : DDT      C                                      COM
I: DUMP     COM      : ONECOPY  C                             JMP      COM
I: COLDBOOT COM      : HDOFF    C                             JMP      ASM
I: FORMAT   COM      : COPY     COM : SYSGEN   COM : GMGRADD  COM
I: LOCAL    COM      : USER     ASM : HDBOOT   COM : GDDT     COM
I: FILLDISK COM      : XDIR     COM : WS       COM : SERIALZ  COM
I: WSMSGS   OVR      : WSOVLY1  OVR : CPMGEN   COM : FRED     C
A>
```

```
F1:   Display Help Window
F2:
F3: Enter NS Advantage Monitor
F4: Toggle CAPSLOCK On/Off
F5: Toggle HD Speed Slow/Fast
F6:
F7:
F8:
F9: Reset Advantage Computer
F10: Exit NS Advantage Emulator

+----------------------------------+
|  F1 toggles this help On and OFF. |
+----------------------------------+

  See the ADE Users Manual for more.
```

The F1 function key acts as a toggle to display or hide this Help Screen

3.2 Function Key F2 – not yet implemented

3.3 Function Key F3 – Pause Emulator, go to Monitor screen



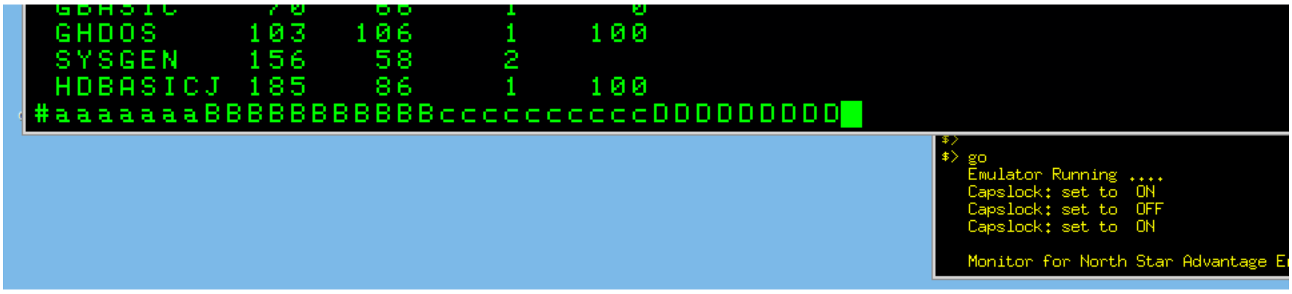
```
 Floppy    1 is </home/jvs/wrk/ade/
 Floppy    2 is    ** not mounted. *

 Hard Disk   is </home/jvs/wrk/ade/


go
   Emulator Running ....
   Stopped at pc=0xF62C
$>
   Monitor for North Star Advantage
```

The F3 function key will halt the emulator, the PC address where execution was ended is displayed as a hex number. The key input focus is transferred to the Monitor screen for commands input.

The input prompt is ‘$> ‘.

This screen has editing features on its command-line. Up and down arrow-keys will take the input focus to several previous commands. Left and right arrow-keys will move the cursor along the line.

3.4 Function Key F4 – Toggle the CapsLock.



```
 GBASIC     70    66     1     0
 GHDOS     103   106     1   100
 SYSGEN    156    58     2
 HDBASICJ  185    86     1   100
#aaaaaaaBBBBBBBBBBBccccccccccDDDDDDDDD
```

```
$>
$> go
   Emulator Running ....
   Capslock: set to  ON
   Capslock: set to  OFF
   Capslock: set to  ON

   Monitor for North Star Advantage E
```

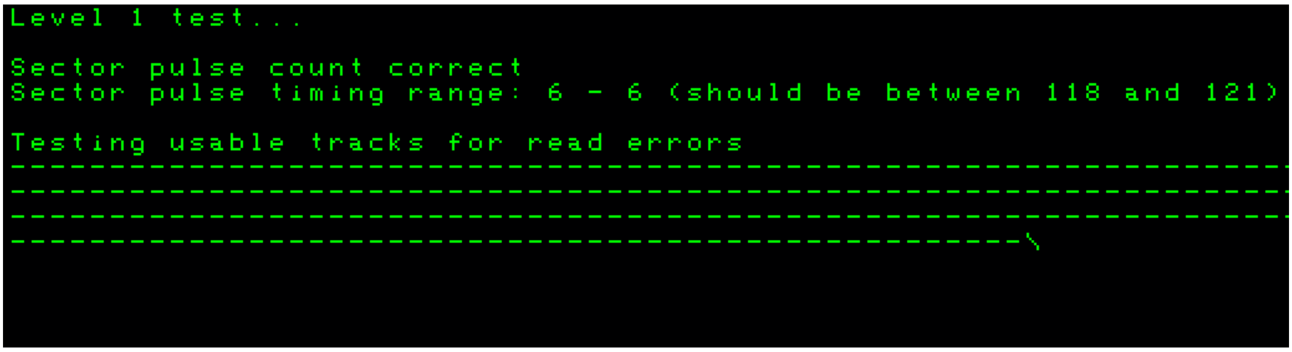
The F4 function key toggles the CapsLock. Many of the older Operating Systems will not recognise the use of lower-case characters. While one can use the actual Caps Lock Key to turn on the CapsLock, it would also turn on upper-case for the host Operating System as well. This can be a nuisance.

The action of the F4 Function Key is displayed in the Monitor screen as shown above.

In the screen shot above, the input line starts off as lower-case (CapsLock OFF) and then toggled with the F4 key with each succeeding letter change.

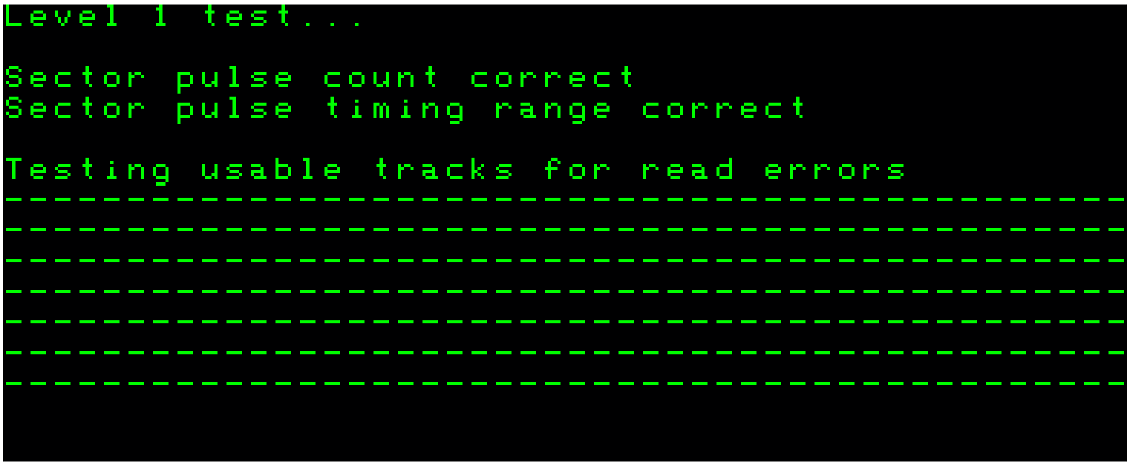
3.5 Function Key F5 – Toggle the Hard Disk Speed 'Fast' or 'Slow'

ADE's emulation of the floppy disk drives and the hard disk drives is extremely fast!



```
Level 1 test...

Sector pulse count correct
Sector pulse timing range: 6 - 6 (should be between 118 and 121)

Testing usable tracks for read errors
----------------------------------------------------------------
----------------------------------------------------------------
----------------------------------------------------------------
---------------------------------------------------\
```

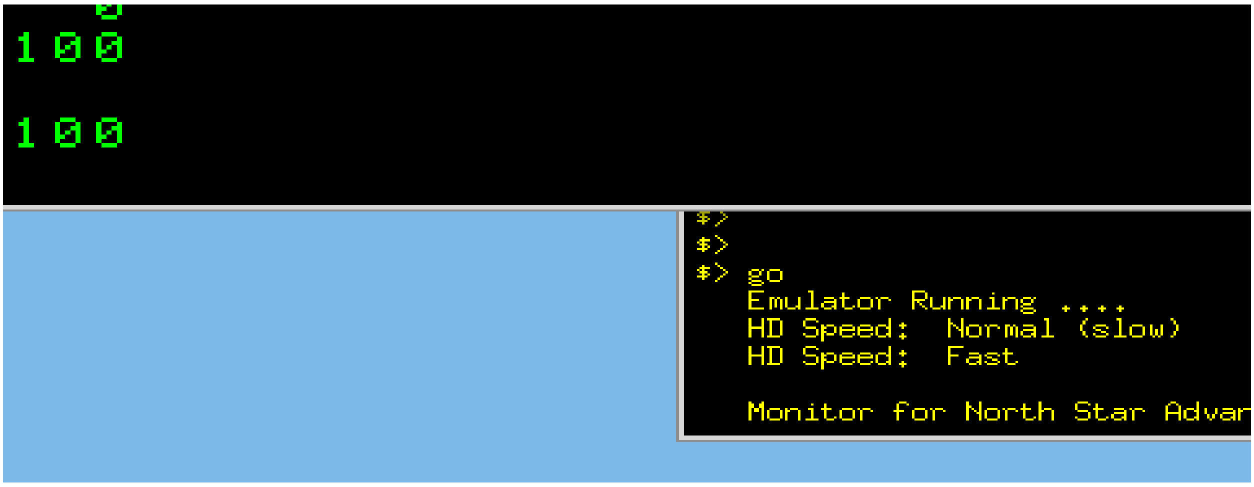
The hard drive is usually 20 times faster than the emulated 'correct speed'.



```
Level 1 test...

Sector pulse count correct
Sector pulse timing range correct

Testing usable tracks for read errors
----------------------------------------------
----------------------------------------------
----------------------------------------------
----------------------------------------------
----------------------------------------------
----------------------------------------------
----------------------------------------------
```

Except that the ‘correct speed’ is itself about 5 times faster than the REAL ‘correct speed’. The hard drive is deliberately slowed down even more, so that the period allowed for entering a ‘;’ to enable editing of the CP/M virtual disk configuration is increased from about half a second to about 3 seconds. It is suggested that when you need to adjust the CP/M virtual-disk configuration, that the ‘normal/slow’ speed is toggled on at the ‘LOAD SYSTEM’ screen, then the ‘fast’ speed is toggled back on, and left on, at the ‘A> ‘ prompt. The default speed is ‘Fast’.

It is very noticeable if the F5 key is toggled while the HD Supplement Disk hard drive tests are in action.



```
100

100
```

```
$>
$>
$> go
   Emulator Running ....
   HD Speed:  Normal (slow)
   HD Speed:  Fast

   Monitor for North Star Advar
```

As with the F4 CapsLock key above, the action of the F5 toggle to speed up or slow down the hard drives is notified in the Monitor screen.

3.6 Function Key F6 – not yet implemented

3.7 Function Key F7 – not yet implemented

3.8 Function Key F8 – not yet implemented

3.9 Function Key F9 – Reset The Emulator (Reboot)

Using the F9 function key will cause an immediate reboot to the ‘LOAD SYSTEM’ screen.

3.10 Function Key F10 – Exit The Emulator

Using the F10 function key causes the emulator to stop working and to close down. Depending on how much cleaning up, like finishing writing to logging files, needs doing, it may take several seconds for the Advantage window and the Monitor window to close.

3.11 Function Key F11 – not yet implemented

3.12 Function Key F12 – not yet implemented

4. HELPER PROGRAMS

4.1 ADE tools

4.1 mkhd (make hard-disk-image file)

mkhd is used to produce ADE hard-disk image files. The smallest of the images of the North Star 'standard' hard-disk types (as included in the HD5XTEST program) is 5 megabytes, the largest is 30 megabytes.

A typical example session with mkhd is shown (user input in **bold**):

```
centrepoint [jvs] /home/jvs/wrk/ade/north_star_tools [sdev*] > mkhd


                         === mkhd ===
                         Version  3.0

Prepares a "Standard" 5-inch Hard-Disk Imagefile for use with North Star
Horizon Emulator (nse) and Advantage Emulator (ade) running HDOS.

Disk-image sizes available range from 5 MB to 30MB.

                                  Usable    Usable  Shipping  Total   Total
No.   Type   Rev. Cylinders Heads Sectors  Capacity Cylinder Sectors Capacity

 1    SG5A   1.0    153      4     9792     4.90 M    153      9792   4.90 M
 2    TN5A   2.0    153      4     9792     4.90 M    153      9792   4.90 M
 3    MS5B   2.0    306      2     9792     4.90 M    336     10752   5.38 M
 4    RD5B   2.0    306      2     9792     4.90 M    319     10208   5.10 M
 5    SG5B   2.0    306      2     9792     4.90 M    306      9792   4.90 M
 6    TN5B   2.0    306      2     9792     4.90 M    306      9792   4.90 M
 7   CM10E   2.0    612      2    19584     9.79 M    650     20800  10.40 M
 8   MS10E   2.0    612      2    19584     9.79 M    656     20992  10.50 M
 9   CM15C   2.0    306      6    29376    14.69 M    306     29376  14.69 M
10   SG15C   2.0    306      6    29376    14.69 M    306     29376  14.69 M
11   RD15C   2.0    306      6    29376    14.69 M    319     30624  15.31 M
12   TN15C   2.0    306      6    29376    14.69 M    306     29376  14.69 M
13   MS15D   2.0    480      4    30720    15.36 M    522     33408  16.70 M
14   MS15E   2.0    459      4    29376    14.69 M    522     33408  16.70 M
15   CM20E   2.0    612      4    39168    19.58 M    650     41600  20.80 M
16   MS20E   2.0    612      4    39168    19.58 M    656     41984  20.99 M
17   RD20E   2.0    612      4    39168    19.58 M    639     40896  20.45 M
18   MS30D   2.0    459      8    58752    29.38 M    522     66816  33.41 M
19   CM30E   2.0    612      6    58752    29.38 M    650     62400  31.20 M
20   MS30E   2.0    612      6    58752    29.38 M    656     62976  31.49 M
21   RD30E   2.0    612      6    58752    29.38 M    639     61344  30.67 M


   Select ( '0' to exit) : 18


     Type: MS30D disk:  29.38 M usable capacity.   ---- Is that correct? y

    creating disk-image type MS30D, 29.38 M.

    Enter file name for this disk: thirty-megs.nhd

Disk ImageFile:  /home/jvs/wrk/ade/north_star_tools/thirty-megs.nhd    requested.


Disk ImageFile: '/home/jvs/wrk/ade/north_star_tools/thirty-megs.nhd'   created OK.

Do you want to include the SYSTEM account and TRANSIENT file? (Y/n) n

Done.
```

I suggest the use of the .NHD extension for these North Star Hard-Disk Image files. This extension, like most, is probably already in use elsewhere but is unlikely to be confused with our usage.

4.2 nshdls (nshd list directory)

nshdls   <North Star hard-disk-image>

nshdls disks/SG5A-1.NHD

nshdls displays lots of information from the file-entries in the North Star Hard Disk index. Because the position of the file-entries within the directory depends on a hash function, the directories always seem jumbled – this was a trade-off way back in the days of slow hard-disks.

There are two options to **nshdls**:

-d Show deleted files as well as existing files
-l (long) Show more details about the directory and hard drive

```
centrepoint [jvs] /home/jvs/wrk/ade > nshdls -d disks/cm10ex.nhd


FileName  Size:Sectors---Blocks     Account    Filetype       Description


BUSINESS          0       0 blk     - - -          0     ACCOUNT (02) "BUSINESS"
CPMA           2031    4062 blk     SYSTEM         7     CP/M Virtual Disk File
TRANSIENT         0       0 blk     SYSTEM         1     NS **  DELETED  **
CPMB           2031    4062 blk     SYSTEM         7     CP/M Virtual Disk File
CPMC           2031    4062 blk     SYSTEM         7     CP/M Virtual Disk File
CPMD           2031    4062 blk     SYSTEM         7     CP/M Virtual Disk File
CPME           2031    4062 blk     SYSTEM         7     CP/M Virtual Disk File
CPMF           2031    4062 blk     SYSTEM         7     CP/M Virtual Disk File
CPMG           2031    4062 blk     SYSTEM         7     CP/M Virtual Disk File
CPMH           2031    4062 blk     SYSTEM         7     CP/M Virtual Disk File
SYSTEM            0       0 blk     - - -          0     ACCOUNT (01) "SYSTEM"
CPMWORK          47      94 blk     SYSTEM         6     CP/M Workfile
LEDGER          250     500 blk     BUSINESS       3     HBASIC data file
```

4.3 nshdbm (nshd bitmap)

nshdbm <North Star hard-disk-image>

nshdbm disks/SG5A-1.NHD

nshdbm gives a slightly more compact set of information than the long North Star Hard Directory Listing and also shows a graphical representation of the usage of the hard disk's file-space.

```
 /tmp/nse_120805/disks > nshdbm SG5A-1.NHD


           Bitmap and Directory for North Star Hard Disk Image "SG5A-1.NHD"  (4.896 Megabytes)

Total DIBs = 612 (maximum usable = 602, as also uses 1 DIB for System Track, 8 DIBs for File-Directory, 1 DIB for Test Track)
                              [ 1 DIB is 16 sectors unless a power-of-2 factor is applied ]


Dir    Dir              File            FileSize   Bytes  dib/dib    ndib0    ndib0     File  Type       File Type
Offset Type  Accnt  --  Name  -- Attr Sectors Blocks inLast Num/Sects Address Img_Offset  Type  Data       Description
_____________________________________________________________________________________________________________________________
011A00  1    0001  CPMA          80    1562  3124    200    6  64     0020   00040000     7   0001.00  CP/M Virtual Disk File
012000  1    0001  TRANSIENT     80      24    48    200    4  16     0003   00006000     1   1F00.00  Executable, TypeData=GO
012A00  1    0001  CPMB          80    1562  3124    200    6  64     0084   00108000     7   0001.00  CP/M Virtual Disk File
013400  1    0001  DT            80       2     4    200    4  16     0005   0000A000     1   5000.01  Executable, TypeData=GO
014400  1    0001  BACKUP        80      31    62    200    4  16     0006   0000C000     1   2600.00  Executable, TypeData=GO
016E00  1    0001  CLEAN         80       9    18    200    4  16     0013   00026000     2   0010.00  HBASIC program
017000  1    0001  CK            80       2     4    200    4  16     0010   00020000     1   5000.01  Executable, TypeData=GO
017020  1    0001  CO            80       4     8    200    4  16     0011   00022000     1   5000.01  Executable, TypeData=GO
017400  1    0001  RECMAIN       80      15    30    200    4  16     0012   00024000     2   001D.00  HBASIC program
01A600  1    0001  RECOVERS      80      24    48    200    4  16     0014   00028000     2   002E.00  HBASIC program
01C400  1    0001  RECEXP        80       3     6    200    4  16     0016   0002C000     2   0003.00  HBASIC program
01CC00  1    0001  BAKEXP        80       3     6    200    4  16     0017   0002E000     2   0005.00  HBASIC program
01DE00  1    0000  SYSTEM        01       0     0    000    0   1     0000   00000000     0   0000.00  ACCOUNT directory entry
01DE20  1    0001  CPMWORK       80      47    94    200    4  16     0001   00002000     6   0000.00  CP/M Workfile
01E400  1    0001  RECOVER       80      31    62    200    4  16     001A   00034000     1   2600.00  Executable, TypeData=GO
01EC00  1    0001  HBASIC        80      32    64    200    4  16     001C   00038000     1   2600.00  Executable, TypeData=GO
01F200  1    0001  BACKUPS       80      24    48    200    4  16     0018   00030000     2   0030.00  HBASIC program


128 DIBs per line. 'S' = System, 'D' = Directory, 'X' = TestTrack, '.' = Unallocated, 'o' = Allocated, 'U' = Multiple Allocation

SoooooooDDDDDDDDoooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo........................
................................................................................................................................
................................................................................................................................
...................................................................................................X
nullius [jvs] /tmp/nse/disks >
```

4.4 nshdcp (nshd copy file to unix)

nshdcp <North Star hard-disk-image> <Filename>

nshdcp SG5A-1.NHD HBASIC

nshdcp extracts a North Star HDOS file from the North Star Hard Disk image. The filename to be extracted is case-sensitive, although the huge majority of HDOS filenames are upper-case only.

Note that any CP/M files are contained within CP/M virtual disks which are large HDOS files. **nshdcp** will only extract the virtual disk file itself, rather than any indivual CP/M file contained within the virtual-disk file.

4.5 unskew-hd-image

unskew-hdimage <North Star Hard Disk Image> <unskewed image file>
OR
unskew-hd-image <unskewed image file> <North Star Hard Disk Image>

unskew-hd-image SG5A-1.NHD image-plain-a

unskew-hd-image can be dangerous to your hard-disk image-files. **Be careful!** It will be used mainly if you are trying to resurrect portions of files which have been lost by removing the interleaving of the sectors and giving a flat file with everything in correct order.

4.6 nsfilecalc (calculate filesizes in terms of NSDOS 256-byte 'blocks')

ns*filecalc*

```
  nullius [jvs] /tmp/nse/disks > nsfilecalc

             North Star DOS/HDOS File-Size Calculator
                   copyright 2012 Jack Strangio

A North Star Floppy Disk file is restricted to a maximum length of 66
tracks on a DQ disk, or 660 sectors, 1320 blocks, 330 kilobytes.

A North Star Hard-Disk file is made from 'hunks' containing multiple
sectors.  These 'hunks' were originally so-named by North Star, but later
this name was changed to 'DIBs'.

Each DIB ('Data Incremental Block', similar to 'clusters', 'extents', etc.
in other operating systems) contains a multiple of 16 sectors.  There can be
a maximum of 128 DIBs per file.

Since this could really restrict the maximum size of a file, a power-of-2
factor can be applied to 16 giving 16, 32, 64, 128, or even up to 256 sectors
per DIB.  Consequently, it becomes possible to produce a file which can go up
to the maximum allowable file-size on a hard-disk: 65,535 blocks, 32,768
sectors or 16.384 megabytes.

Each file contains its own internal DIB-directory, which takes up the first
sector of the file itself.  Keep this 'loss' of the first file sector in
mind when creating your files on the hard-disk.  The Hard-Disk Directory (or
Index) merely tells HDOS where the file's first sector with its
DIB-directory is located upon the hard-drive.


 Bytes                                  (1)
 North Star Blocks (256-byte)           (2)
 Hard-Disk Sectors (512-byte)           (3)
 North Star DIBs ('clusters','extents')  (4)
 Kilobytes (1024 bytes)                 (5)
 Megabytes (1000x1024 bytes)            (6)


         Select Units: ('0' to quit)  6
                 Enter Value wanted : 3
```

```
File is:  3072000 bytes, 12000 blocks, 6000 sectors, 94 DIBs, allocation factor = 4, 3000.0 KB

    HDOS Command Line:     CR FILENAME[[,ACCOUNT],DISK_UNIT]  12000  4

****    That size of file has unused sectors in the last DIB.    ****

      If all sectors of the last DIB were to be included, the file's size would then become:

          3079680 bytes, 12030 blocks, 6015 sectors, 94 DIBs, allocation factor = 4, 3007.5 KB

    HDOS Command Line:     CR FILENAME[[,ACCOUNT],DISK_UNIT]  12030  4

nullius [jvs] /tmp/nse/disks >
```

nsfilecalc will notify you whether the file-size you have requested will not completely fill a DIB. If there is unused space left in the allocated disk area you may, if you want, increase the size requested up to the end of the last DIB.

4.7 nsfd2u (copy NSDOS file from floppy-disk to unix)

nsfd2u <NSDOS disk-image>

nsfd2u   D04B01.NSI

nsfd2u reads the files off a double-density North Star DOS disk image file and creates copies of those files in the unix file space.

The unix filenames will have the format of <Name of File>_<FileType>[_Go-Address]. The Go-address will only be used with a file of Type 1 (executable).

example 1.
The M5700 executable file is Type 1 and has a Go-Address of 5700 H; this has a unix file name of M5700_1_5700

example 2.
The BASIC program called OTHELLO is Type 2 (BASIC Program) and not being a executable Type 1 will have no Go-Address; this has a unix file name of OTHELLO_2

4.8 u2nsfd (copy file from unix to NSDOS floppy-disk)

u2nsfd <unix file> <NSDOS disk-image>

u2nsfd M5700_1_5700   MYDOSDISK.NSI

u2nsfd will copy a file from the unix file space onto a double-density North Star DOS disk image file.

If the above filename format (as in nsfd2u) is used for the North Star DOS filename in the unix file space, then the file will be added to the NSDOS disk directory complete with Type attributes and Go-Address if applicable. If the NSDOS directory already has a file of the same name, the new file will replace the earlier file.

If the above filename format is not used, the file-type defaults to Type 0 (undefined). This can then be altered using the TY command in NSDOS:

TY <filename>   <File-Type>   [Go-Address]

4.9 compact

compact <NSDOS disk-image>

compact   MYDOSDISK.NSI

compact will 'compact' a North Star DOS disk image file. It will act similar to a defragmenting of the disk-image file by moving all files towards the beginning of the disk, eliminating any unused space between the files where previously deleted files once were.

3.10 nsfdls (NS floppy-disk list directory)

```
nsfdls <NSDOS disk-image>

nsfdls   MYDOSDISK.NSI
```

nsfdls lists the directory of the floppy-disk image file in the same format as the LI in NSDOS.

4.11 mkfs.ns

```
mkfs.ns   [-s]   <disk-image filename>

mkfs.ns   -s    MYSSSDDISK.NSI
```

mkfs.ns creates an empty North Star DOS formatted floppy-disk image. It can produce either single-sided, single-density disk-images (88K) or double-sided, double-density disk-images (350K). The default size is 350K, if you use the '-s' option an 88K disk-image is produced. The first 8 characters of the filename are used as the disk-label.

4.12 jdz80 (Z80 disassembler)

jdz80 is a slightly improved version of Marat Fayzullin's 1999 DAsm, in which relative jump destination addresses are calculated and displayed rather than just displaying the relative jump offsets.

4.13 OTHER TOOLS

4.13 cpmtools

Life is simpler with cpmtools-2.7 (or later) which can be obtained from most linux repositories. This set of utilities can be used to copy files directly between North Star CP/M disk-images and the unix/linux file space. It will be necessary to add the following disk definitions to the cpmtools config-file **diskdefs** which is usually at /etc/cpmtools/diskdefs.

```
diskdef nsfd
  seclen 512
  tracks 70
  sectrk 10
  blocksize 2048
  maxdir 64
  skew 5
  boottrk 2
  os 2.2
end


diskdef nshd4
  seclen 512
  tracks 512
  sectrk 16
  blocksize 4096
  maxdir 256
  skew 0
  boottrk 0
  os 2.2
end
```

The added disk-definitions will enable cpmtools to understand the North Star CP/M disk formats, both the floppy-disk images and the larger CP/M Virtual Disk Images on the hard disk. (Note that you will need to copy the hard-disk CP/M Virtual Disk image-file off from the hard disk image-file by using the **nshdcp** program before you can start to use the cpmtools with it.)

The utilities in cpmtools include:

cpmls	list files in the North Star CP/M disk-image
cpmcp	copy files to and from the North Star CP/M disk-image
cpmrm	delete files from the North Star CP/M disk image
mkfs.cpm	prepare stub disk for CP/M. In my experience, this does not work properly. Instead, use mkfs.ns to produce an NSDOS disk then FORMAT it for CP/M.

4.14 screenlog

screenlog is not a tool as such but a record of ADE's output.

5 VARIOUS.

5.1 OTHER FILES REQUIRED

Various floppy-disk image files:

These are available from various sources. Most of them have a .nsi extension.

5.2 COMPILING LIBRARIES REQUIRED

The linux libraries required are libX11.

5.3 VARIOUS USEFUL MANUALS

Most of the manuals are available from http://www.hartetechnologies.com/manuals/Northstar/ or from http://itelsoft.com.au.

Probably the most useful are:

North Star Advantage Technical Manual
North Star Advantage User Manual
North Star Advantage Graphics Manuals
North Star DOS Rev 5
North Star BASIC Version 6
North Star Advantage Emulator (ADE) User Guide (this manual)
North Star Hard Disk Operating System Manual
North Star CPM 2.2 Manual
North Star CPM 2.2 Preface to the Addendum
North Star CPM 2.2 Addendum

These are all included in the 'documentation' directory

5.4 BUGS

I feel I have got many bugs out which makes ADE very usable. But there are still a few to go, apart from the things that could be done to make ADE not quite so rough-edged. I am the very first to admit that ADE is currently at the “proof of concept” level. It certainly is not anywhere near as elegant as I would like, and the fault-lines between the several programs that ADE is based upon are still very clearly visible. Please inform me of any bugs that you discover. Email me at: jackstrangio@yahoo.com

5.5 TODOs

More realistic emulation of Parallel I/O. A GUI monitor instead of the text-only monitor. There is also a list in the TODO file.

5.6 AUTHOR and SUPPORT

Jack Strangio <jackstrangio@yahoo.com>

Website: http://itelsoft.com.au

APPENDIX A.

NORTH STAR HARD-DISK DATA FORMAT

DATA LAYOUT ON HARD DISK.

A hard-disk drive is actually a set of spinning disks (or platters). For each platter there are two heads, one above and one below the platter. Thus a hard-disk drive with two platters has four heads, and each head reads and writes on a separate 'surface'. Because all the heads are moved as a single unit from track to track on the platters, the set of tracks being read from is called a 'cylinder', so, in this case, there would be four tracks within each cylinder.

STRUCTURE OF SINGLE TRACK

Each North Star hard disk track consists of 16 sectors. Each sector has its own set of data fields. As the platter spins the disk-drive electronics supply pulses which specify when the first sector of the set of 16 sectors is reached by the read/write head (the index pulse), and when the start of each sector begins(the sector pulse). The index pulse is not retained by the North Star Hard-Disk Controller, but the sector pulse is latched on and is turned off by the Hard-Disk Controller itself.

STRUCTURE OF THE WRITTEN DISK SECTOR

When the sector-pulse is received from the hard-drive by the hard-disk controller, the controller waits a short period then begins sending a stream of zero bytes (00 H). This is to cushion variations in speed of the physical drive. After a enough time has passed, a Sync Byte (01 H) is sent to the hard drive to signify the actual start of the data to write on the disk sector.

The first set of real data written is the Sector-Label Header field, this is a set of nine bytes which identify which sector is being written. This information is later used when reading the disk, to ensure that the data being read is from the sector desired and not another sector.

The next data field contains the 512-bytes of data or program we want to store.

The last data field contains CRC information to ensure that the data has been written cleanly. If the data read back from the disk-sector does not match the store CRC value, there has been corruption of the data.

STRUCTURE OF THE SECTOR-LABEL HEADER FIELD

Example:

```
  PHY  CYL  HED  LSl  LSh  STl  STh  CRC  CRC~
  05   0C   83   BD   04   B0   04   09   F6
```

In typical North Star Computers fashion, the sector ID label is not that as suggested by Shugart in the ST506 protocol, but one which was designed by North Star themselves. However there are similarities.

Byte 1: PHYSICAL SECTOR

The lower 4 bits (Bits 0-3) are used to specify the physical sector on the track. The physical sector is the one calculated by skewing the reads to improve reading/writing speeds. The physical sector is calculated by adding 8 to the ODD logical sectors: logical sector 1 is at physical sector 9, logical sector 15 is at physical sector 7.

Bits 4 and 5 contain the 2-bit overflow of the CYLINDER byte (Byte 2) which then gives the CYLINDER byte a total of 10 bits which allows a maximum of 1024 cylinders

Byte 2: CYLINDER

This byte plus the extra 2 bits specified in Byte 1 allow 1024 cylinders.

Byte 3: Surface (Head Number)

The lower 3 bits are used to specify which head is selected.
The high bit (Bit 7) may used to specify whether the sector is write-protected or not.

Bytes 4 - 5:   LOGICAL SECTOR NUMBER

These bytes contain the logical sector-number on the hard-drive. This number may differ from the physical sector number because of the skewing described above.

Bytes 6 - 7:   SHIFTED TRACK NUMBER

These bytes contain the logical sector-number on the drive modulo 16. This can be thought of as either the disk-address of sector 0 on the track, or the 12 bits of the track number shifted up 4 bits. This supplies the physical sector address quite simply by adding the PHYSICAL sector in Byte 1 to this up-shifted track number.

example: (In hex numbers as it makes it easier to see.)

```
Logical sector : 04BD H
Track Number   : 004B H
Shifted Track  : 04B0 H
PHYSICAL       :   05 H
```

Physical Sector: 04B5 H

Byte 8: CRC SUM

This byte contains the lower 8 bits of the total obtained by adding all 7 previous bytes.

Byte 9: CRC BYTE COMPLEMENT

This byte contains the complemented CRC byte. (The sum of Byte 8 and Byte 9 is always FF H)

FURTHER EXAMPLE:

```
 PHY  CYL  HED  LSl  LSh  STl  STh  CRC  CRC~
  25   52   80   CD   DE   C0   DE   40   BF
```

Physical Sector: 5   (From Bits 0-3 of PHY)

Cylinder   :   52 H (From CYL) + 0200 H (From Bits 4 & 5 of PHY) = 0252 H = 594 (Dec.)

Head   :   0   (From Bits 0-2) of HED

Logical Sector :   DECD H   = 57037 (Dec.)

Physical Sector: 5 (From PHY) + DEC0 H (From Shifted Track) = DEC5 H = 57029 (Dec.)

CRC   :   25 H + 52 H + 80 H + CD H + DE H + C0 H + DE H = 440 H = 40H

CRC~   :   40 H complemented = BF H (or BF H + 40 H = FF H)

APPENDIX B.

ADE's HARD-DISK IMAGE FILE STRUCTURE

The hard-disk image structure's size varies according to the number of sectors which were in the original physical hard disk.

The sectors are laid out as in physical sectors, rather than logical sectors. This means the sectors in the disk-image are interleaved, just as they are on the physical disk. There is an unskewing utility in the nse_tools directory, but I don't think this would ever be used by most users of ADE.

NOTE: Validation that the file is truly a North Star Emulator hard-disk image depends solely on the presence of the North Star 'magic' bytes (00 H, FF H) at the start of the first sector of the hard disk image-file. This first sector is North Star's "Hard Disk Label" and contains much information about the size and layout of the hard disk.

If the two validation bytes are not found, ADE will not mount the file at all. While this means that a hard disk image file may become unusable very occasionally, it serves to guard against unwanted accidental damage to other types of files. If warranted, further tests for disk image validity may be included in later versions of ADE.

For producing ADE hard-disk image files of the 'standard' hard disks used by North Star Computers, see under ADE Tools, **mkhd.**