Cromemco Software Update Service Note Cromix-4

Date: _____June 4, 1982_____

Product: _Cromix-L and Cromix-S_

Version.Release: _11.10_____

Date production of this version began: _____June 4, 1982 on 8"_
_____June 4, 1982 on 5"_

First serial number with this version: _____CX1612428 on 8"_
_____CX1711273A on 5"_

> [Note: Earlier serial numbers may also contain this version. All Cromemco Finished Goods stock was recopied on the above dates.]

SUMMARY

Cromix™ Operating System version 11.10 is now available. This version has new capabilities supporting Cromemco's graphics software, a new 5-1/4" hard disk system, and a new magnetic tape drive system. Several procedures involving mode characteristics have been changed or enhanced. New system calls have been added to the Cromix CDOS Simulator, corresponding to the system calls added to CDOS to permit execution of many programs which run under CP/M* Version 2.2. The complete Cromix Operating System is fully documented in the new June 1982 **Cromix Instruction Manual**, part number 023-4022.

ENHANCEMENTS

Cromix Operating System version 11.10 has color graphics capability. This version supports the Cromemco SDI Graphics Interface hardware, the Cromemco SDI Graphics Software library (SGS), and the Cromemco FontMaster package (FOMR). SGS is an all purpose library of graphics routines for drawing lines and curves and coloring areas. The FontMaster program allows users to define their own graphics character sets and to create color text images using these character sets. For details, please refer to the **SGS-2 Suds Note**, part number 023-9529, and the **SGS-3 Suds Note**, part number 023-9534, and the **Cromix Instruction Manual**.

023-9532   *CP/M is a trademark of Digital Research.

The new verion of the Cromix Operating System will support Cromemco's 8" hard disk, as well as the new 5-1/4" hard disk used in conjunction with Cromemco's new WDI-II Disk Controller board. The 5-1/4" hard disk has a capacity of 5-1/2 Mbytes. Version 2.55 or higher of the Init utility is required to initialize a hard disk interfaced by a WDI-II board.

Cromix Operating System version 11.10 will support the new Cromemco magnetic tape system (TDS-3901). This system consists of the tape drive, software, and two interface boards: an IOP, the existing I/O processing board, and a CSP, the new C-Bus serial-to-parallel board interfacing the tape drive and the IOP board.

The TDS uses the industry standard nine track formatted tape. This allows data exchange between Cromemco systems and other systems, including IBM. For backing up information, a standard 10-1/2" reel holding 2400 feet of tape can store approximately 40 Mbytes of data.

CHANGES TO CROMIX.SYS

Old version 11.05

Inode Link Damage

Inode links previously could be made incorrectly if a detached process was killed while waiting for the Quadart driver, QTTY. This has been fixed.

Tape Driver

TP is the new tape driver added to the Cromix Operating System software to support tape drive capabilities. User control of the tape drive is provided through the new version of the Mode utility provided with this release. For more information, see the discussion of **/bin/mode.bin** in the section entitled **CHANGES TO THE UTILITIES.**

Character Drivers

Modifications have been made to the TTY, QTTY, MTTY, LPT, TYP, SLPT, and QSLPT character drivers. They are discussed in the new options added to the Mode utility. Changes or additions have been made to the **modeequ.z80** equate file to correspond to the modifications to the drivers.
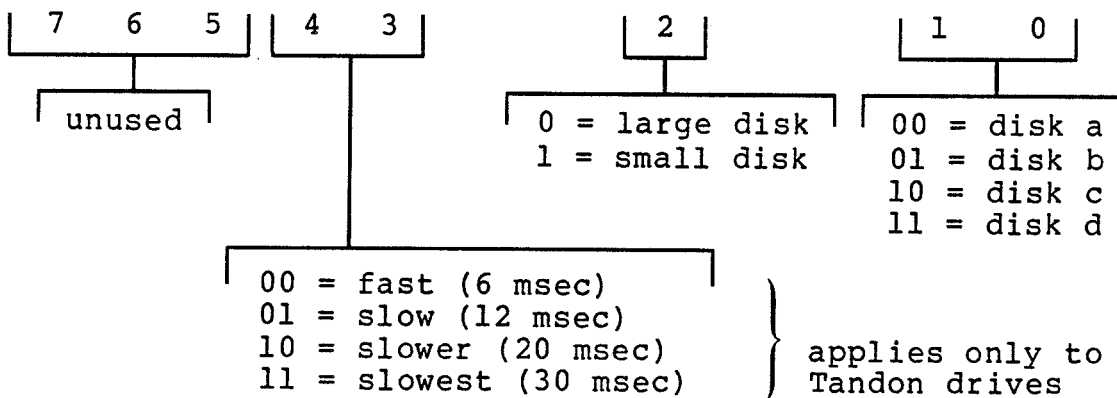
Timing Constant

The timing constant for the hard disk drivers was changed to perform more reliably with the 5-1/4" hard disk.

Home Error

In the old version, the drivers for 5" floppies were failing to seek to track 10 and try again to home whenever the floppies encountered a home error. This problem is eliminated in version 11.10.

Minor Device Numbers

Greater control over the small floppy disk drives has been added by expanding the minor device number to include seek speed. The number is derived from eight bits as follows:

```
| 7   6   5 || 4   3 |          | 2 |          | 1   0 |
     |___|         |               |               |
    unused         |         0 = large disk     00 = disk a
                   |         1 = small disk     01 = disk b
                   |                            10 = disk c
                   |                            11 = disk d
                   |
          00 = fast    (6 msec)   )
          01 = slow    (12 msec)  }  applies only to
          10 = slower  (20 msec)  )  Tandon drives
          11 = slowest (30 msec)  )
```

The speeds listed for the table above are the drive stepping rates between tracks.

The minor device numbers for small floppies are listed in the table below:

Drive   **Speed**

	Fast	Slow	Slower	Slowest
sfda	4	12	20	28
sfdb	5	13	21	29
sfdc	6	14	22	30
sfdd	7	15	23	31

The minor device numbers are listed with their corresponding devices in the **/dev** file. The new version of the Cromix

Operating System is shipped configured for small Tandon floppies at fast speed.

To configure the system for different speeds for the small floppies, you must change the minor device number for the device. This is done by using the Delete and Makdev utilities.

Be sure the devices are unmounted before beginning this procedure. Delete the **/dev** entry for the device which is to be changed using Delete. Reenter the device with its new minor number using Makdev. Refer to the **Cromix Instruction Manual**, part number 023-4022, for details on using Delete and Makdev.

Hidden Sectors

The hidden sectors are the last two sectors on the first track of the diskette. These sectors are left over from composing sectors into blocks at four sectors per block. They are not used to compose any block, so have not been previously accessible. The hidden sectors on a floppy diskette can now be read by seeking to byte 0FFFFFE00h. On large floppies, they are the 25th and 26th sectors. On small floppies, they are the 17th and 18th sectors.

Raw Mode

GTTY now turns off raw mode before asking for login name. This avoids conflicts in cases where raw mode had inadvertently been left on by a previous process.

Process Time Slice

The time slice for a process has been changed from 500 milliseconds to 100 milliseconds.

Shell

A space character is no longer required in the command line after the pipe symbol |.

In previous versions of the Cromix Operating System, CNTRL-C would sometimes abort a detached process. This has been fixed.

CHANGES TO EQUATE FILES

/equ/modeequ.z80

Definitions have been added which reflect new capabilities of the terminal and printer drivers.

/equ/tmodeequ.z80

This is a new equate file containing the definitions for the constants used by the new tape driver.

CHANGES TO THE UTILITIES

/bin/backup.bin   00.08

Old version 00.06
Backup formerly accepted diskettes initialized with either the CDOS or Cromix Operating System format. Now it accepts only diskettes initialized with the Cromix Operating System format.

Backup does not modify dump times.

/bin/dcheck.bin   00.12

Old version 00.10.
This utility has been changed due to the renaming of a routine used in both Dcheck and Boot to avoid name conflicts.

/bin/ddump.bin   RB 2.01

Direct Dump is a new utility. Ddump converts and copies data from one file or device to another. It is useful for gaining access to data stored in raw form because the input and output block sizes may be specified.

Options are:

if=pathname or -i pathname	specify input file pathname (default: standard input)
of=pathname or -o pathname	specify output file pathname (default: standard output)
ibs=n	input block size = n bytes
obs=n	output block size = n bytes

cbs=n	conversion buffer size = n bytes (default = 80 bytes)
iskip=n	skip first n input blocks before starting copy
oskip=n	skip first n output blocks before starting copy
icount=n	copy only n input blocks
conv=ascii	convert EBCDIC to ASCII
ebcdic	convert ASCII to EBCDIC
ucase	convert alphabetic characters to upper case
lcase	convert characters to lower case
strip	strip trailing blanks
nostop	do not stop processing on an error

Example: ddump if=/dev/tpl of=filel conv=ascii,lcase,strip

A more detailed explanation of Ddump is available through the Help utility. Type **help ddump.**

/bin/echo.bin   RB 00.05

Old version 00.04
Echo is now a relocatable binary program.

/bin/help.bin   RB 00.04

Old version RB 00.02
In previous versions, if a user added too many new Help files and the Help program ran out of memory while listing the directory, the program would be aborted. The program has been fixed to read in only as much as it is able and ignore the rest.

/bin/init.com   2.71

Old version 2.21
This utility has been changed to enable the initialization of 5-1/4" hard disks. Init can now label disks initialized for CDOS.

/bin/makfs.bin RB 00.13

Old version 00.11

   -r   A new option, Restore, has been added. This option creates a superblock and may be used to restore a distorted superblock. Icheck -s must be run after -r to restore the free list. The system must then be rebooted.

Use the -r option **only on Makfs version 00.13 or higher.**
Earlier versions will ignore the unrecognized option and
proceed with the entire Makfs function. **This procedure could
destroy the contents of the disk.**

A new command, **fixsb.cmd** (Fix Superblock), is now available.
This command will automatically implement the utilities to
restore the superblock. The commands which are implemented
are discussed under **/cmd/fixsb.cmd.**

The file **makfs.bin** is now relocatable binary.

/bin/mode.bin   RB 1.12

Old version RB 1.01
New options and characteristics have been added to the Mode
utility and are briefly listed below. A more detailed
explanation is available through the Help utility. Type **help
mode.**

More specific error messages have been added to the Mode
utility.

The -v option verifies by displaying mode characteristics
after changing them.

The following characteristics are tape drive specific and
must be preceded by the device name of the tape drive (i.e.,
TP1):

Block n :   This characteristic moves the tape to the nth
                    block within the current tape file. The
                    following example moves to the second block
                    within the current tape file on device TP1:

                    mode tp1 block 2

BLKSWritten: This characteristic reports the number of
                    blocks of the last tape that were written to
                    the tape device.

EOFclose: If this characteristic is set, a filemark
                    marking the end of a tapefile will
                    automatically be written on the tape when the
                    tape device is closed.

File n: This characteristic moves the tape to the nth
                    file on the tape.

Inblkln: This characteristic reports the byte length of
                    the first block of the last file read from the
                    tape device.

Outblkln n: This characteristic sets and reports the block
                   length to n bytes for files written on a tape
                   device.

REWind: This characteristic rewinds a tape.

UNLOAD: This characteristic unloads a tape.

The following characteristics refer to terminal and printer
devices TTY, QTTY, MTTY, LPT, TYP, SLPT, and QSLPT:

BINary: CBreak, RAW, and BINary are parameters of
                   terminal devices TTY, QTTY, and MTTY. If any
                   of these parameters is enabled, any read from
                   the device returns after each input character.
                   These parameters also serve to disable the
                   action of various other parameters.

Margin n: If Margin n set, a printer or terminal device
                   will emit a formfeed if it is within n lines
                   of the bottom of the page.

Break: See BINary.

DELAYcode n: This characteristic sets and reports the
                   amount of delay for TTYS, QTTYS, and MTTYS
                   after outputting the formatting characters NL,
                   TAB, CR, FF, and BS. Type **help mode** for more
                   information.

DIScard: If DIScard is set, the data area allocated for
                   a driver will be discarded as soon as the
                   device is closed. This allows another driver
                   to use the data area in cases where two
                   drivers use the same port address on an
                   interface card.

FFexpand: If FFexpand is set, a formfeed character
                   output to a printer or terminal device is
                   converted to newlines.

HUPenable: If HUPenable is set, a modem connected to a
                   QTTY or MTTY is hung up when the terminal
                   device closes.

SIGHUPall: If SIGHUPall is set, and the modem of a QTTY
                   or MTTY hangs up, the signal SIGHANGUP is sent
                   to all processes controlled by the device.

LCase: If LCase is set, terminal devices TTY, QTTY,
                   and MTTY convert upper case input characters
                   to lower case.

SIGALLchars: The combined states of the modes of
                 SIGALLchars and SIGenable determine how the
                 SIGChar-key/SIGUSER-signal will affect
                 processes. Type **help mode** for more
                 information.

TABexpand: If TABexpand is set, a tab character is
                 converted to spaces.

TANDEM: TANDEM mode is used to allow software
                 handshaking between two Cromix Operating
                 Systems connected by either TU-ART or Quadart
                 asynchronous ports.

WRAParound: If WRAParound is set, an output device will
                 print the remainder of a line extending beyond
                 the page width on a new line. If WRAParound
                 is not set, the remainder of the line will be
                 truncated.

/bin/sim.bin   00.31

Old version 00.27
New System Calls 6, 28 through 36, 40, and 162 have been
added and System Call 12 has been revised. These changes
enable the CDOS simulator to execute many programs which run
under CP/M Version 2.2. For more detailed information please
refer to the **CDOS-1 Suds Note**, part number 023-9533.

In 32K Structured Basic, a CNTRL-Z typed from the keyboard
would abort Basic. A program not yet saved could thus be
lost. This has been fixed. If CNTRL-Z is typed, an error
message will appear but the Basic session will continue.

All files previously accessed through Sim had to have
read/write access. Now a file may have only read access if
there are no programs which write to it. If an attempt is
made to write to a file which does not have write access, an
access error will be printed on the terminal (by the Cromix
Operating System) and an error will be generated to the CDOS
write system call (by **sim.bin**).

/bin/spool.bin   RB 00.12

Old version RB 00.09
The -h option now outputs a formfeed after the header.

When spool used the Stdin channel, it generated an access
error. This has been fixed. For instance, after typing
spool followed by RETURN, the user may type text on the
keyboard. This text will be printed after the user presses
CNTRL-Z (end of file).

/bin/wboot.bin   RB 00.09

Old version 00.05
The file **wboot.bin** is now relocatable binary.

/bin/version.bin   00.09

Old version 00.08
Previously, **version.bin** determined whether a program was relocatable binary (RB) by checking the header of the file, which is 256 bytes long. If the program was checking a file whose length was less than 256 bytes, it could erroneously obtain the RB characteristic information from the file header of the file leftover from a previous check. This problem has been corrected.

/cmd/fixsb.cmd

Fix Superblock is a new command file which implements the -r option of **makfs.bin**. The command file will execute only with the proper version of Makfs. It is called by typing **fixsb devname**. It will execute Makfs -r, followed by Icheck -s and Boot.

/cmd/runqd.cmd

The Runqd utility has been changed to establish the 16FDC TTY1 as the system console, where previously QTTY1 was the system console. This requires the presence of the TTY driver in **cromix.sys**. The memory required by the TTY driver reduces the number of system buffers available from the original ten to about five. If the user wishes to prevent this decrease in available system buffers, the Cromix Operating System must be reconfigured without the TTY drivers. To reconfigure the system, refer to Chapter 6 of the **Cromix Instruction Manual.**

See also **CHANGES TO MULTIPLE PROGRAMS** for related changes to other files.

/etc/sfdboot   00.08

Old version 00.04
Sfdboot has been modified so that, when a home error occurs, the drive will seek to track 10 and seek again to home. Minor device number modifications have also been made. Refer to **CHANGES TO CROMIX.SYS** for more information on these changes.

/etc/login.bin   RB 00.02

Old version RB 00.01
The new version changes the time allotment between typing the directory name and the password from 10 seconds to 30 seconds before timeout.

/gen/crogen.bin   RB 00.14

Old version RB 00.08
It is now possible to generate a **cromix.sys** which makes the
default root device the booting device. Crogen will ask a
series of questions concerning the default root device:

Default Root Device? [Yes or No]

A **no** response will end the questioning. As a result,
the operating system will ask for the major and minor
root device every time the system is booted. A **yes**
response will lead to another question:

Boot Disk? [Yes or No]

A **yes** response will end the questioning. The operating
system will automatically assign the device from which
the system was booted as the root device. A **no** response
will cause Crogen to ask two more questions:

Major number?
Minor number?

Crogen asks you to specify the root device by the major
and minor number. This is the end of the questioning.
(For more information, see **CHANGES TO MULTIPLE PROGRAMS**,
below.)

Crogen will automatically include the timer driver required
to run the operating system.

Prompts using the characters [or] now use < or >. This
change was made because the character codes which generate
the square brackets in the United States are used in other
countries to generate foreign characters.

The new version of Crogen allows generation of a Cromix
Operating System containing drivers for the SDI High
Resolution Graphics Interface and the nine track magnetic
tape drive.

CHANGES TO MULTIPLE PROGRAMS

TTY1 Console Under IOP Configuration

In a Cromix system configured for use with an IOP, the new
assignment of the system console as TTY1, discussed
previously in **/cmd/runqd.cmd,** required changes in the
following files:

/etc/iostartup.iop.cmd
/etc/ttys.iop
(Both TTY1 and QTTY1 are enabled devices.)

Default Root Device

The automatic setting of the root device to be the boot
device, discussed previously in **/gen/crogen.bin,** required in
changes in the following programs:

etc/fdboot   00.08
     Old version   00.06

etc/sfdboot 00.08
     Old version   00.04

/DEV/xxx, where **xxx** is an entry in the **/dev** file, was not
previously recognized when DEV was typed in upper case
letters.   This has been fixed, and the following programs
have been modified:

/bin/cdoscopy.bin   00.15
     Old version   00.14

/bin/dcheck.bin   00.12
     Old version   00.11

/bin/free.bin   00.09
     Old version   00.08

/bin/icheck.bin   00.15
     Old version   00.14

/bin/idump.bin   00.06
     Old version   00.05

/bin/makfs.bin   00.13
     Old version   00.12

/bin/mount.bin   00.13
     Old version   00.12

/bin/spool.bin   00.12
     Old version   00.11

/bin/unmount.bin   00.11
     Old version   00.10

/bin/wboot.bin   00.09
     Old version   00.07

RDOS 2.52

Version 02.52 of Cromemco RDOS is available and affects the Cromix Operating System with several enhancements. Briefly, this latest version allows Cromix Operating System users to boot from any floppy disk drive. A ROM-based, real time application program has the ability to boot from any of the four drives. See **Application Note: RDOS 2.52**, part number 023-9042, for details.

IOP MONITOR

Version 03.00 of the IOP Monitor ROM is now available. It offers enhancements and corrections to the original monitor, version 01.00. Users who wish to upgrade to version 03.00 should contact the Cromemco Customer Support department.

KNOWN PROBLEMS

SpellMaster™ Users

Cromix Operating System users who have been using the SpellMaster program with version 11.05 and who want to continue using their user dictionaries which have accrued useful vocabularies should continue using Cromix Operating System version 11.05. Users who want to run the SpellMaster program under version 11.10 must begin with a fresh copy of the **spellcrx.usr** dictionary in their **/usr/pkg/spell** directory after they have updated to version 11.10.

The process to replace the SpellMaster dictionary, which has a file extension of **.usr**, is explained below for those who wish to use version 11.10 of the Cromix Operating System.

First, update to Cromix Operating System version 11.10. Locate the original SpellMaster diskette. The original is on a CDOS formatted diskette. Place this diskette into any drive, for example, drive B. Now, find the pathname of the user dictionaries in your Cromix Operating System directory structure with the command

```
# find / -name "*.usr" -a -print
```

This command should normally find dictionaries in
/usr/pkg/spell only. Move to the directory in which the user
dictionary is located. For example:

> **# d /usr/pkg/spell**

Now copy the **spellcrx.usr** file from the original diskette in
drive B into the current directory. Do this by using the
Cdoscopy utility:

> **# cdoscopy -v fdb spellcrx.usr**

To summarize, the session will look like this:

> **# find / -name "*.usr" -a -print**
> /usr/pkg/spell/spelltec.usr
> **# d /usr/pkg/spell**
> **# cdoscopy -v fdb spellcrx.usr**
> **# delete spelltec.usr**
> **# ren spellcrx.usr spelltec.usr**

VERSION NUMBER SUMMARY

cromix.sys   Version 11.10

Files in /bin   **Version**

access.bin	RB	00.06
backup.bin		00.08
linker.bin		00.13
boot.bin		00.02
cdoscopy.bin		00.15
chowner.bin	RB	00.06
cmpasc.bin		00.05
compare.bin	RB	00.07
copy.bin	RB	00.10
cptree.bin		00.07
day.bin	RB	01.02
dcheck.bin		00.12
ddump.bin	RB	02.01
deltree.bin	RB	00.03
dump.bin	RB	00.10
echo.bin	RB	00.05
ed.bin		01.35
find.bin	RB	00.07
free.bin		00.09
group.bin	RB	00.01

h.bin	RB	00.04
help.bin	RB	00.04
icheck.bin		00.15
idump.bin		00.06
init.com		02.71
input.bin	RB	01.00
l.bin	RB	00.11
mail.bin		00.06
makdev.bin	RB	00.07
makfs.bin	RB	00.13
maklink.bin	RB	00.04
match.bin	RB	00.03
mode.bin	RB	01.12
mount.bin	RB	00.13
move.bin	RB	00.09
msg.bin	RB	00.08
ncheck.bin	RB	00.09
passwd.bin		00.09
patch.bin		00.03
priv.bin	RB	00.07
restore.bin		00.05
root.bin	RB	00.02
screen.bin		01.35
sim.bin		00.31
sort.bin		00.06
spool.bin	RB	00.12
tee.bin	RB	01.02
testinp.bin	RB	01.01
time.bin	RB	00.07
unmount.bin	RB	00.11
usage.bin	RB	00.06
version.bin	RB	00.09
wboot.bin	RB	00.09
who.bin	RB	00.06

Files in /dev/iop	Version	
cromix.iop	RB	11.10
ioprun.bin	RB	03.00
tape.iop	RB	11.10

Files in /etc	Version	
fdboot		00.08
login.bin	RB	00.02
sfdboot		00.08

Files in /gen	Version	
crogen.bin	RB	00.14
default.bin	RB	00.01

Date: __August 25, 1982__

Product: __CROMIX-L and CROMIX-S__

Version.Release: __11.11__

Date production of this version began: __July 16, 1982 on 8"__
__July 16, 1982 on 5"__

First serial number with this version: __CX1612536   on 8"__
__CX1711333A&B on 5"__

> [Note: Earlier serial numbers may also contain this version. All Cromemco Finished Goods stock was recopied on the above dates.]

SUMMARY

Version 11.11 of the Cromix^{TM} Operating System is now available. This version contains corrections to the previous version, version 11.10. For information on the changes made to the Cromix Operating System between versions 11.05 and 11.10, refer to the **Cromix-4 Suds Note**, part number 023-9032.

CHANGES

Changes to Cromix.sys 11.11
     Old version 11.10

The procedure for determining the system console device has been changed. As a result, the operating system will not boot if the user has a device connected at location F0h.

A value of 0 is output to port E1h, the port on the WDI-II board which stores the high byte of the 24 bit address of the hard disk DMA buffer. This permits correct execution of WDI-II DMA operations when running the Cromix Operating System on a DPU with a WDI-II board.

The system call **.getmode** now correctly returns the tape status of the tape devices. The tape driver, Tp, now marks the last file on the tape with two filemarks. The tape cannot be read beyond the last file.

023-9540

Changes to the Utilities

/bin/ddump.bin 02.02

This utility is now distributed on the Tape Drive System (TDS) software diskette shipped with the TDS. The diskette is in CDOS format. To use the TDS files on a Cromix system, load the files into the appropriate directories by executing the Install command.

/bin/sim.bin 00.34
     Old version 00.31

System Call 6 (Direct Console I/O) now accepts the control characters ^S and ^Q as input. This change was made to avoid conflict with programs which use these two control characters.

System Call 37 is now ignored instead of being unimplemented. This change allows programs which use the call to avoid being aborted when running under the Cromix CDOS Simulator. System Call 37 is now handled by the CDOS and Cromix Operating Systems in the same manner.

Keyboard I/O calls handle the setting of modes more efficiently. The improved calls are:

        System Call 1 (Read Console with Echo)
        System Call 6 (Direct Console I/O)
        System Call 10 (Input Buffered Line)
        System Call 128 (Read Console Without Echo)

The routine invoked by the Read Console Without Echo entry of the Jump table, which corresponds to the CP/M* Jump table, is also affected. Modes which are affected are:

        Echo
        Crdevice
        Raw
        Immediateecho
        Noecnl
        Escrtn

* CP/M is a registered Trademark of Digital Research, Inc.

/etc/fdboot 00.09
   Old version 00.08

A TU-ART board which powered up with interrupts pending would
send an interrupt to the Cromix Operating System before the
operating system tables had been properly set up. This would
prevent the system from booting. The boot file Fdboot has
been changed so that it does not enable interrupts before
passing control to the Cromix Operating System.

/etc/sfdboot 00.09
   Old version 00.08

The change made to Fdboot also applies to Sfdboot.

VERSION NUMBER SUMMARY

cromix.sys   Version 11.11

Files in /bin   **Version**

access.bin	RB	00.06
backup.bin		00.08
linker.bin		00.13
boot.bin		00.02
cdoscopy.bin		00.15
chowner.bin	RB	00.06
cmpasc.bin		00.05
compare.bin	RB	00.07
copy.bin	RB	00.10
cptree.bin		00.07
day.bin	RB	01.02
dcheck.bin		00.12
deltree.bin	RB	00.03
dump.bin	RB	00.10
echo.bin	RB	00.05
ed.bin		01.35
find.bin	RB	00.07
free.bin		00.09
group.bin	RB	00.01
h.bin	RB	00.04
help.bin	RB	00.04
icheck.bin		00.15
idump.bin		00.06
init.com		02.71
input.bin	RB	01.00
l.bin	RB	00.11
mail.bin		00.06
makdev.bin	RB	00.07
makfs.bin	RB	00.13
maklink.bin	RB	00.04
match.bin	RB	00.03

mode.bin	RB	01.12
mount.bin	RB	00.13
move.bin	RB	00.09
msg.bin	RB	00.08
ncheck.bin	RB	00.09
passwd.bin		00.09
patch.bin		00.03
priv.bin	RB	00.07
restore.bin		00.05
root.bin	RB	00.02
screen.bin		01.35
sim.bin		00.34
sort.bin		00.06
spool.bin	RB	00.12
tee.bin	RB	01.02
testinp.bin	RB	01.01
time.bin	RB	00.07
unmount.bin	RB	00.11
usage.bin	RB	00.06
version.bin	RB	00.09
wboot.bin	RB	00.09
who.bin	RB	00.06

Files in /dev/iop	Version	
cromix.iop	RB	11.11
ioprun.bin	RB	03.00

Files in /etc	Version	
fdboot		00.09
login.bin	RB	00.02
sfdboot		00.09

Files in /gen	Version	
crogen.bin	RB	00.14
default.bin	RB	00.01

Cromemco Software Update Service Note 68000 Cromix-1

Date: <u>August 30,1982</u>

Product: <u>CRO-DL and CRO-DS</u>

Version.Release: <u>20.05</u>

Date production of this version began: <u>Aug. 27, 1982 on 8"</u>
<u>Aug. 27, 1982 on 5"</u>

First serial number with this version: <u>1-10012</u> <u>on 8"</u>
<u>1-10070</u> <u>on 5"</u>

SUMMARY

This version of the Cromix™ Operating System is the first version configured for use with Cromemco's new Dual Processor Unit (DPU) and its associated 16 bit memory boards. The boards are the Memory Control Unit (MCU), and 265K and 512K Memory Storage Units (256MSU and 512MSU). These boards are described in the June 1982 Cromemco **68000 Board Family Instruction Manual**, part number 023-2016.

The 68000 Cromix Operating System gives you the ability to run programs for both the Z80 and 68000 microprocessors. You may thus take advantage of the power offered by the 68000 while still maintaining compatibility with the large software base established for the Z80.

This note describes the differences between the 68000 Cromix Operating System, version 20.05, and the Z80 Cromix Operating System, version 11.11. Most of the differences exist at the level of the operating system's interaction with the hardware and are not apparent at the user level. The June 1982 **Cromix Instruction Manual**, part number 023-4022, should be used as the primary reference for user operation.

To further supplement the **Cromix Instruction Manual**, copies of the following SUDS notes for the Cromix Operating System have been included in this package: **Cromix-4**, part number 023-9532, and **Cromix-5**, part number 023-9540.

Future versions of the Cromix Operating System will grow to take greater advantage of the 68000 microprocessor. Users are urged to subscribe to the Software Update Service (SUDS) to keep appraised of changes to the 68000 Cromix Operating System and to receive new releases of the software. To become a SUDS subscriber, contact your Cromemco dealer.

023-9548

1

HARDWARE CONFIGURATION

This version of the Cromix software requires at least the following boards for proper operation: a DPU, an MCU, one or more MSUs, and a disk controller board.

Any hard disk drives used with the system must be controlled by a WDI-II Winchester Disk Interface board, revision B or D. Failure to use the proper controller board will jeopardize the data on your hard disk.

NEW SOFTWARE

Crogen68.bin version RB 00.00

This utility generates a 68000 Cromix operating system. It allows you to add or delete system drivers and to select various options when you configure the system. Crogen68 is located in the **/gen** directory.

This 68000 version of Crogen operates in the same manner as the Z80 version, which is documented in the **Cromix Instruction Manual.**

Cromix.sys version 20.05

This file contains the 68000 operating system configured by Crogen68. During the boot procedure, this file is read into the system portion of memory.

Ecc.bin version RB 00.02

Ecc is a 68000 Cromix utility used with the Cromemco 256MSU and 512MSU memory boards, which have the ability to perform memory error detection and correction. This program allows a privileged user to enable or disable memory error correction hardware and to display on the console the type and location of memory errors. Error correction is disabled when the system is reset and must be normally reenabled.

Ecc has the following command formats and capabilities:

Ecc on   Turns on error correction.

Ecc off   Turns off error correction.

Ecc   Reports if error correction is off or on.

Ecc -e   Reports any errors since the last Ecc -e. If there
                are no errors, nothing is displayed on the console.

If memory errors are suspected, you may want to accumulate an
error history. The command file **/cmd/logerr.cmd** is provided
for this purpose. It periodically checks whether any memory
errors have occurred since the last check.   Errors are
written to the file **/etc/msu_errs**.

Generate an error history by typing on your console the
command

        logerr nn &

where **nn** is the desired number of seconds between each Ecc
check of the error status. The ampersand places the command
file execution in background.

In this version of the Cromix Operating System, error
correction MUST be turned off when using Hdtest or any other
program which directly accesses the hard disk (i.e., programs
which don't use Cromix system calls). Disk related Cromix
system calls disable error correcting for their duration.
Failure to turn error correction off while performing DMA to
a hard disk will jeopardize the data on the hard disk.

Init version 02.76

The 68000 Cromix software includes a new version of Init,
which has two enhancements:

The disk RPM test now works with MSUs when initializing hard
or floppy disks. MSU error correction is automatically
disabled while initializing a hard disk, preventing conflicts
between the error correction and DMA.

CNTRL-C characters are now trapped by Init, making it
impossible to interrupt Init once started. Previously, when
initializing a hard disk, the alternate track table would be
destroyed if the program received a CNTRL-C character during
initialization.

3

68000 Abort Codes

When the 68000 encounters an error condition, it generates an
error code and begins exception processing based on the type
of error. A message such as

Vector interrupt - 03 4E720002 00394E73 00000000 30740002

will be displayed on the console.   If this happens type
CNTRL-C.   In many cases, this will return control to the
operating system.   If, however, the processor was in
supervisor mode, CNTRL-C will have no affect and the system
will have to be reset.

The two digits at the far left of the numbers above, 03 are
the error code. Look up this code in the following table and
take the appropriate debugging action.

The hexadecimal numbers to the right of the error code
represent the contents of the top 16 bytes of the system
stack at the time the error was encountered.   This
information is pushed on the system stack during exception
processing and may be useful in debugging.   The first two
bytes represent the contents of the status register.   The
next long word represents the contents of the program
counter.

The following table defines run time errors generated by the
68000:

Vector number	Dec	Address Hex	Assignment
0	0	000	Reset: initial system stack pointer
1	4	004	Reset: initial program counter
2	8	008	Bus Error
3	12	00C	Address Error
4	16	010	Illegal Instruction
5	20	014	Zero Divide
6	24	018	CHK Instruction
7	28	01C	TRAPV Instruction
8	32	020	Privilege Violation
9	36	024	Trace
10	40	028	Line 1010 Emulator
11	44	02c	Line 1111 Emulator

68000 Memory Usage

The addressing capability of the 68000 and the availability of new higher-capacity memory boards has lead to a vertical arrangement of 68000 Cromix memory. That is, memory is treated as one block in which memory addresses increase from the bottom to the top of memory. Z80 Cromix memory, on the other hand, is arranged horizontally in banks of 64K bytes.
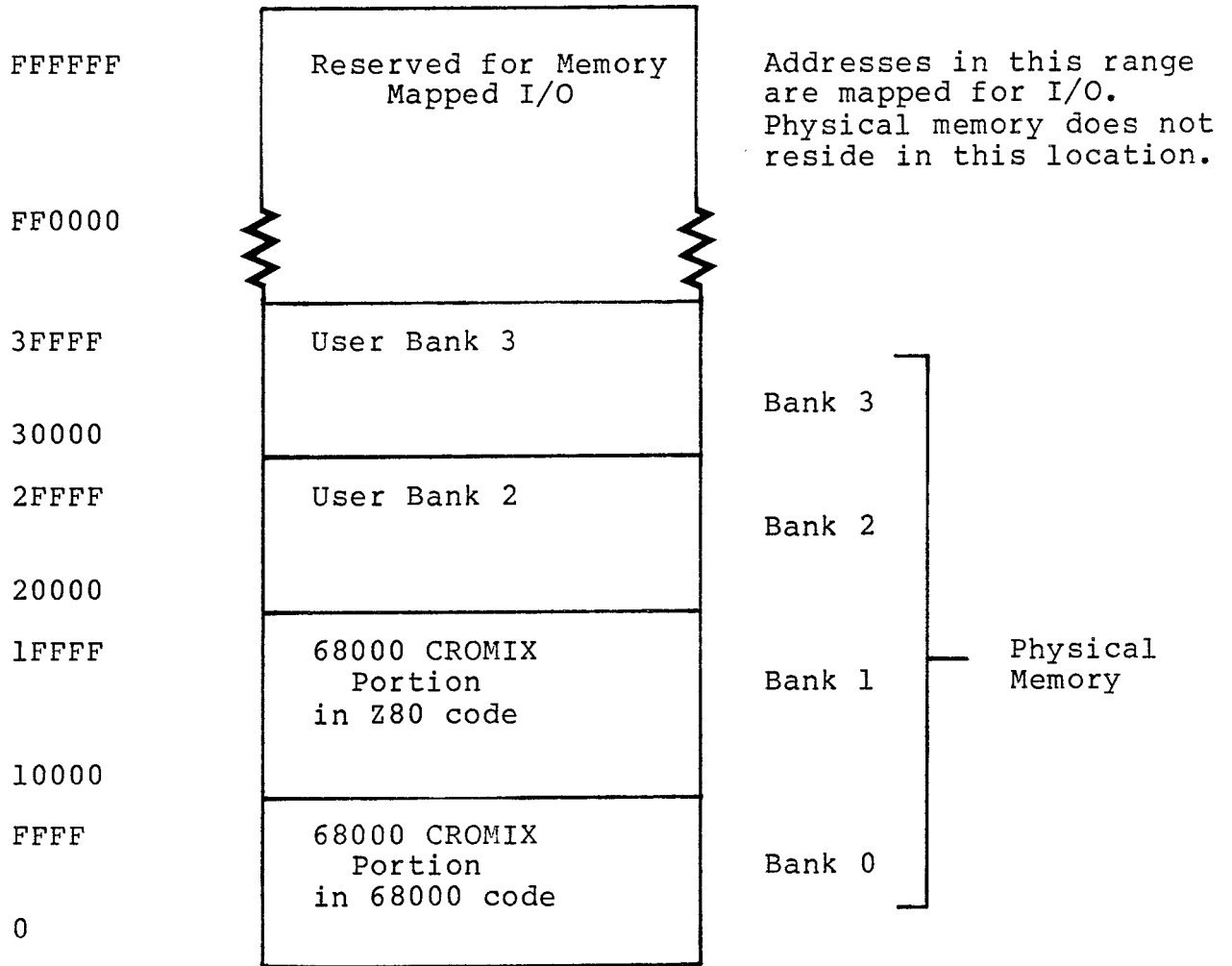
The 68000 Cromix Operating System has been adapted from the Z80 Cromix Operating System, so some sections are written in Z80 code and are executed by the Z80. Remaining sections are written in 68000 code and are executed by the 68000.

68000 Cromix memory is divided into two parts, **user memory** and **system memory.** The system memory is further divided into three parts: 64K bytes which contain the portion of Cromix software written in 68000 code, 64K bytes which contain the Z80 portion of the software, and the upper 64K bytes which provide memory mapped I/O for the 68000. User memory is divided into 64K byte sections one for each user.

Memory allocation for Z80 jobs is the same as with Z80 Cromix software. Jobs for the 68000, however, may be as large as available memory will allow. The number of available shell buffers, which depend on the options selected in generating the operating system, may be less than the number available with the Z80 Cromix software.

The 68000 generates 24 bit addresses and can directly address any portion of memory. The Z80 outputs 16 bit addresses, which comprise the low order 16 bits of 24 bit memory addresses. The upper eight bits of this address are contained in an address latch on the DPU board. The contents of this latch are set to the high order eight bits of the last op-code fetched by the 68000. This is the op-code that initiates Z80 processing.

The following is a graphic representation of 68000 Cromix memory usage:

```
                    ┌─────────────────────┐
FFFFFF              │  Reserved for Memory │        Addresses in this range
                    │     Mapped I/O       │        are mapped for I/O.
                    │                      │        Physical memory does not
FF0000              ╳                      ╳        reside in this location.
                    ├─────────────────────┤
3FFFF               │   User Bank 3        │                          ┐
                    │                      │         Bank 3           │
30000               │                      │                          │
                    ├─────────────────────┤                          │
2FFFF               │   User Bank 2        │                          │
                    │                      │         Bank 2           │
20000               │                      │                          ├── Physical
                    ├─────────────────────┤                          │   Memory
1FFFF               │   68000 CROMIX       │                          │
                    │      Portion         │         Bank 1           │
                    │   in Z80 code        │                          │
10000               │                      │                          │
                    ├─────────────────────┤                          │
FFFF                │   68000 CROMIX       │                          │
                    │      Portion         │         Bank 0           │
                    │   in 68000 code      │                          ┘
0                   │                      │
                    └─────────────────────┘
```

68000 Cromix System Calls

The same set of system calls available under the Z80 Cromix Operating System are available under the 68000 Cromix Operating System, with the exception of nine calls which remain to be implemented in the next version. System calls used by the 68000 are identical in function to corresponding Z80 system calls. The **Cromix Instruction Manual** describes the function of each call. Since these calls are being implemented on the 68000, however, their use of 68000 registers needs to be defined.

Upon encountering an error, the system call sets the carry flag and deposits an error code in register D0. The following table lists each 68000 system call, its call number, its calling parameters, and its return parameters. An asterisk following the name of a call indicates that the call has not yet been implemented.

System Call	System Number	Calling Parameters	Return Parameters
_alarm	43H	D3 = number of seconds	
_caccess	27H	D1 = channel D2 = access mask	
_cchstat *	23H	D1 = channel D2 = status type D3 = new value A1 -> buffer	
_chdup	0AH	D1= existing channel	D2 = duplicate channel
_chkdev	07H	D2 = type of device D3 = major device # D4 = minor device #	
_clink	25H	D1 = channel A1 -> new pathname	
_close	0BH	D1 = channel	
_create	08H	D2 = access mode D3 = exclusive mode A0 -> pathname	D1 = channel
_cstat *	21H	D1 = channel D2 = desired info A1 -> buffer	D3 = Return value
_delete	06H	A0 -> pathname	
_error	1CH	D0 = error number D1 = channel Other registers as returned by the call that generated the error	
_exchg	0CH	D1 = channel D2 = channel	

System Call	System Number	Calling Parameters	Return Parameters
_exec *	4CH	A0 -> pathname A1 -> argument list	
_exit	46H	D3= termination status	
_faccess	26H	D2 = access mask A0 -> pathname	
_fchstat *	22H	D2 = status type D3 = new value A0 -> pathname A1 -> buffer	
_fexec *	4BH	D1 = signal mask D2 = signal values A0 -> pathname A1 -> argument list	
_flink	24H	A0 -> old pathname A1 -> new pathname	
_fshell *	48H	D1 = signal mask D2 = signal values A0 -> pathname A1 -> argument list	
_fstat *	20H	D2 = desired info A0 -> pathname A1 -> buffer	D3 = return value
_getdate	30H		D0 = day of the week D1 = year D2 = month D3 = day of the month
_getdir	02H	A0 -> buffer	
_getgroup	36H	D2 = id type	D3 = group id
_getmode	12H	D1 = channel D2 = mode type	D3 = return value
_getpos	10H	D1 = channel	D3 = file position
_getprior	38H		D3 = priority number

9

System Call	System Number	Calling Parameters	Return Parameters
_getproc	3AH	D3 = process id	
_gettime	32H		D1 = hour D2 = minute D3 = second
_getuser	34H	D2 = id type	D3 = user id
_indirect	51H	D0 = call number Registers are used according to call number	Registers are used according to call number.
_kill	41H	D2 = signal type D3 = process id	
_lock	3EH	D2 = lock type D3 = lock length A0 -> lock sequence	
_makdev	00H	D2 = type of device D3 = major device # D4 = minor device # A0 -> pathname	
_makdir	01H	A0 -> pathname	
_mount	04H	D1 = type of access A0 -> dummy pathname A1 -> device pathname	
_mult	MULU D1,D0		
_open	09H	D2 = access mode D3 = exclusive mode A0 -> pathname	D1 = channel
_pause	44H		
_pipe	0EH		D1 = input channel D2 = output channel
_printf	1BH	D1 = channel A0 -> control string arguments on stack	

System Call	System Number	Calling Parameters	Return Parameters
_rdbyte	16H	D1 = channel	D0 = byte
_rdline	18H	D1 = channel D3 = maximum bytes A0 -> buffer	D3 = bytes read
_rdseq	14H	D1 = channel D3 = maximum bytes A0 -> buffer	D3 = bytes read
_setdate	31H	D1 = year D2 = month D3 = day of the month	
_setdir	03H	A0 -> pathname	
_setgroup	37H	D1 = type to change D2 = new id type D3 = new group id	
_setmode	13H	D1 = channel D2 = mode type D3 = new value D4 = mask	old value
_setpos	11H	D1 = channel D2 = mode D3 = file pointer	
_setprior	39H	D3 = priority number	
_settime	33H	D1 = hour D2 = minute D3 = second	
_setuser	35H	D1= type to change D2= new id type D3 = new user id	
_shell *	49H	A1 -> argument list	
_signal *	40H	D2 = type of signal A0 -> execution addr.	
_trunc	0DH	D1 = channel	

System Call	System Number	Calling Parameters	Return Parameters
_unlock	3FH	D2 = lock type D3 = lock length A0 -> lock sequence	
_unmount	05H	D2 = eject flag A0 -> device pathname	
_update	52H		
_version	55H		D3 = version number
_wait	45H	D2 = conditional flag D3 = process id	D1 = process term. D2 = system term. D3 = child pid
_wrbyte	17H	D0 = byte D1 = channel	
_wrline	19H	D1 = channel A0 -> buffer	D3 = bytes written
_wrseq	15H	D1 = channel D3 = byte count A0 -> buffer	D3 = bytes written

TEMPORARY RESTRICTIONS ON THIS VERSION

The following capabilities are not supported by this version
of the 68000 Cromix Operating System, but will be added in
succeeding versions:

Color Graphics   The addressing of color graphics memory
boards currently assumes horizontal memory
organization and must be reconfigured.

Default   This Z80 Cromix command setting default
parameters for automatic login is not
implemented in this version.

KSAM   The KSAM package has not yet been adapted
to run under the 68000 Cromix Operating
System.

Shell Buffers   Currently, there are a limited number of
shell buffers available, the actual number
depending on how the operating system is
generated. Succeeding versions will have
more shell buffers.

System calls   The following system calls are not
implemented in this version:

Exec
Fexec
Shell
Fshell
Cstat
Fstat
Cchstat
Fchstat
Signal

VERSION NUMBER SUMMARY

Files in /	Version	
cromix.iop.sys	20.05	-new-
cromix.sys	20.05	-new-

Files in /bin	Version
access.bin	00.06
backup.bin	00.08
blink.bin	00.13
boot.bin	00.02
cdoscopy.bin	00.15

```
chowner.bin      00.06
cmpasc.bin       00.05
compare.bin      00.07
copy.bin         00.10
cptree.bin       00.07
day.bin          01.02
dcheck.bin       00.12
deltree.bin      00.03
dump.bin         00.10
ecc.bin          00.02      -new-
echo.bin         00.05
ed.bin           01.35
find.bin         00.07
free.bin         00.09
group.bin        00.01
h.bin            00.04
help.bin         00.04
icheck.bin       00.15
idump.bin        00.06
init.com         02.76      -new-
input.bin        01.00
l.bin            00.11
mail.bin         00.06
makdev.bin       00.07
makfs.bin        00.13
maklink.bin      00.04
match.bin        00.03
mode.bin         01.12
mount.bin        00.13
move.bin         00.09
msg.bin          00.08
ncheck.bin       00.09
passwd.bin       00.09
patch.bin        00.03
priv.bin         00.07
restore.bin      00.05
root.bin         00.02
screen.bin       01.35
sim.bin          00.34
sort.bin         00.06
spool.bin        00.12
tee.bin          01.02
testinp.bin      01.01
time.bin         00.07
unmount.bin      00.11
usage.bin        00.06
version.bin      00.09
wboot.bin        00.09
who.bin          00.06
```

Files in /dev	Version	
iop/cromix.iop	11.11	
iop/ioprun.bin	03.00	

Files in /etc	Version	
fdboot	00.09	
login.bin	00.02	
sfdboot	00.09	

Files in /gen	Version	
crogen68.bin	00.00	-new-
default.bin	00.01	

Cromemco Software Update Service Note 68000 Cromix-2

Date: __October 20, 1982__

Product: __CRO-DL and CRO-DS__

Release: ____2____

Date production of this version began: __Oct. 11, 1982 on 8"__
                                         __Oct. 11, 1982 on 5"__

First serial number with this version: _____2-10000 on 8"__
                                          _____2-10000 on 5"__

SUMMARY

Version 20.09 of the 68000 Cromix™ Operating System is now available. Several problems found in the initial release have been fixed. Please refer to the 68000 Cromix Suds Note-1 (part number 023-9548) for a full description of the D-series Cromix Operating System.

CORRECTIONS

1. When **.bin** files generated from 68000 source code were loaded, the Cromix Operating System did not check to see if sufficient memory was available to hold the module. If the **.bin** file was too large, it would overrun the top of user memory, causing vectored interrupts. Also random characters would be sent to the terminal.

    The current version now correctly loads modules of any size. If a program is too big to fit in available memory, the message **All memory in use.** appears on the screen.

    The amount of memory a 68000 executable module requires may be determined by giving the command:

    dump filename.bin

    The four bytes at location 08h, 09h 0Ah, and 0Bh of the dump display indicate the amount of memory in hexadecimal needed for this module. Location 08h is the high order byte, and location 0Bh is the low order byte.

023-9553

1

2. In cases where more than one module was running in a 64K block of memory, the scheduler was not initializing the pointer at locations 6 and 7 of that block as the modules were time sliced in and out. Correct initialization occurred when programs were first loaded and run. However, when already resident modules began execution during their second time slice, the scheduler would not update locations 6 and 7 for them. The values would be correct only for the most recently loaded module. If this module was not executing and the running module used the value at 6 and 7, a vector interrupt error could occur, hanging the system.

    This has been fixed, and any mix of RB and non-RB modules may be loaded and run, up to the limits of memory or until 18 processes are running concurrently.

3. Terminals on quadarts would not allow a user to log in on a name that was password protected. When given either the correct or an incorrect password, **Name or Password incorrect.** was displayed. This problem is corrected in the current version.

4. If 68000 programs were doing terminal I/O and CONTROL-S was entered followed by CONTROL-C, a normal abort might not occur. In some cases, random characters were sent to the screen before the system prompt was displayed. In cases where a FORTRAN-77 or Pascal program was executing, a runtime error was often generated before the programmed aborted.

    This occurred when CONTROL-C was entered while a system call was executing. This has been fixed, and programs now exit properly when CONTROL-C is entered.

5. For terminals using the TU-ART drivers, if Mode WRAP was set and the Mode WIDTH specification was exceeded before entering a RETURN, when the RETURN was entered, the system would lock up. Terminals on Quadarts did not exhibit this problem.

    The current version corrects this error.

6. Though much of the 68000 Cromix Operating System uses routines from Cromix version 11.11, **iolib.rel** in the **/gen** directory of the 68000 Cromix Operating System is not at all compatible with the similarly named module in Cromix version 11.11. The Crogen68 utility must use the **iolib.rel** shipped on the 68000 Cromix version 20.09 system diskette.

NEW SOFTWARE

INIT version 2.77

The motor-on delay for floppies has been changed to 2.4
seconds, for increased reliability. When used under the
Cromix Operating System to initialize a hard disk in CDOS
format, Version 2.76 would place the directory label
incorrectly. This problem has been fixed. However, Init
does not write the CDOS bitmap back to the hard disk. This
means that the Stat utility must be run under CDOS to correct
the bitmap on the hard disk **before** any other programs are
run.

HARDWARE CONSIDERATIONS

When a 68000 system is turned on, the error light on the MCU
is on. This does not necessarily mean that a memory error
has occurred. To turn the light off, type the command

        ecc on or
        logerr nn &

The light remains off unless a memory error occurs or the
system is turned off and on again.

Normally if error checking has been enabled and the light is
on, a memory error has occurred. Type the command

        ecc -e

to display a description of the memory error on the terminal.
This command also turns off the MCU error light.

This light may be on for two different reasons: the user who
sees the MCU light on may be seeing a computer with a memory
error or a computer that has just been turned on.

One way to turn off the error light automatically upon
booting is to edit the **startup.cmd** file to include the
command:

        ecc on >* /dev/null

By the time the Login message is displayed on the console,
the MCU light should be off, even if the system was just

3

powered up. The redirection of console output to the null
device prevents the message **Error correcting memory is turned
on** from being displayed within the sequence of boot messages.

With error checking enabled, the computer user should always
suspect that a memory error has occurred if this light is on.

A small disadvantage to having ECC enabled is that an extra
wait state is required for each memory reference. This slows
program execution slightly. If this is undesirable, disable
error checking by typing the command

    ecc off

VERSION NUMBER SUMMARY

Files in /	Version	
cromix.iop.sys	20.09	-new-
cromix.sys	20.09	-new-

Files in /bin	Version	
access.bin	00.06	
backup.bin	00.08	
blink.bin	00.13	
boot.bin	00.02	
cdoscopy.bin	00.15	
chowner.bin	00.06	
cmpasc.bin	00.05	
compare.bin	00.07	
copy.bin	00.10	
cptree.bin	00.07	
day.bin	01.02	
dcheck.bin	00.12	
deltree.bin	00.03	
dump.bin	00.10	
ecc.bin	00.02	
echo.bin	00.05	
ed.bin	01.35	
find.bin	00.07	
free.bin	00.09	
group.bin	00.01	
h.bin	00.04	
help.bin	00.04	
icheck.bin	00.15	
idump.bin	00.06	
init.com	02.77	-new-
input.bin	01.00	
l.bin	00.11	

```
        mail.bin              00.06
        makdev.bin            00.07
        makfs.bin             00.13
        maklink.bin           00.04
        match.bin             00.03
        mode.bin             01.12
        mount.bin            00.13
        move.bin             00.09
        msg.bin              00.08
        ncheck.bin           00.09
        passwd.bin           00.09
        patch.bin            00.03
        priv.bin             00.07
        restore.bin          00.05
        root.bin             00.02
        screen.bin           01.35
        sim.bin              00.34
        sort.bin             00.06
        spool.bin            00.12
        tee.bin              01.02
        testinp.bin          01.01
        time.bin             00.07
        unmount.bin          00.11
        usage.bin            00.06
        version.bin          00.09
        wboot.bin            00.09
        who.bin              00.06
```

Files in /dev	Version
iop/cromix.iop	11.11
iop/ioprun.bin	03.00

Files in /etc	Version
fdboot	00.09
login.bin	00.02
sfdboot	00.09

Files in /gen	Version
crogen68.bin	00.00
default.bin	00.01

Date: March 7, 1983

Product: CRO-DL and CRO-DS

Release: 3

Date production of this version began: March 7, 1983 on 8"

First serial number with this version: 3-10000   on 8"
                                             3-10000   on 5"

SUMMARY

68000 Cromix Operating System, version 20.14, is now available and updates the previous release, version 20.09. This version of the 68000 Cromix Operating System contains modified hard disk drivers capable of supporting Cromemco's new 5-1/4" 20-megabyte hard disk. This version also allows users to provide their own character drivers created with the **Cromix Driver Package**, Cromemco model CXDR. System calls have been added to complete the list of Z80 Cromix system calls available under the 68000 Cromix Operating System. New signals have been added corresponding to the 68000 exception vectors. In addition, some bugs in the operating system and utilities have been fixed.

ENHANCEMENTS

New Hard Disk Capacity

The hard disk drivers can now support Cromemco's new **HD20** 5-1/4" 20-megabyte hard disk drive. The alternate tracks on this disk are positioned on cylinders 270-275 (decimal), rather than on the innermost cylinders. Only 32 of these 36 tracks are used as alternate tracks.

User-Supplied Character Drivers

The Crogen68 utility now has a **-u** option to allow inclusion of user-supplied character drivers in the operating system. The character drivers may be created with a new software product, the Cromix Driver Package. If the **-u** option is used, Crogen68 will prompt for names of the driver entry points and for names of files that contain the drivers.

υ23-9561

Enhanced Serial Line Printer Driver

The serial line printer driver has added capabilities to
support Cromemco's CLQ letter-quality serial printer, a
printer compatible with Cromemco's C-10 Personal Computer.
The busy line from the CLQ printer should be connected to the
data-input line (pin 2) of the Tu-art serial port.

The following table lists the device definitions of printers
whose protocols are supported by the Tu-art serial line
printer driver:

Device Name	Board	Base Port (hex)	DEVICE NUMBER Major : Minor		
			XON/XOFF	CLQ	ETX/ACK
slpt1	16fdc	00h	7 : 0	7 : 64	7 : 128
slpt2	TU-ART A	20h	7 : 2	7 : 66	7 : 130
slpt3	TU-ART B	50h	7 : 5	7 : 69	7 : 133
slpt4	TU-ART A	60h	7 : 6	7 : 70	7 : 134
slpt5	TU-ART B	70h	7 : 7	7 : 71	7 : 135
slpt6	TU-ART A	80h	7 : 8	7 : 72	7 : 136
slpt7	TU-ART B	90h	7 : 9	7 : 73	7 : 137
slpt8	TU-ART A	A0h	7 : 10	7 : 74	7 : 138
slpt9	TU-ART B	B0h	7 : 11	7 : 75	7 : 139

System Calls

System calls now available that were not implemented in the
original release of the 68000 Cromix Operating System are:

- **.exec**
- **.fexec**
- **.shell**
- **.fshell**
- **.cstat**
- **.fstat**
- **.cchstat**
- **.fchstat**
- **.signal**

The **.ksam** system call is the only system call available on
the Z80 but not the 68000 version of the Cromix Operating
System. Z80 programs which use the .ksam call will work
correctly, however 68000 programs cannot use the .ksam call.

Signals

Twenty-six new signals numbered 9 through 34 have been added which correspond to the 68000 exception vectors. Signals 9-24 correspond to the sixteen 68000 Trap instruction vectors, vector numbers 32-47. Trap 0 is the jsys call for executing Cromix system calls. Trap 3 is used by the 68000 debugger, debug68.bin. Signals 25-34 correspond to 68000 exception vectors 2-11 for bus error, address error, illegal instruction, zero divide, CHK instruction, TRAPV instruction, privilege violation, trace, line 1010 emulator, and line 1111 emulator. Signals 25-34 are sent to the process by the operating system when the errors occur.

Signals 0-7 except signal 3 can be caught, ignored or can kill a process, as under Z80 Cromix. The new signals can be caught or can kill a process, but cannot be ignored by a process. Signal 9, the signal for the jsys call, differs in the following manner: if signal 9 is set to kill a process, receipt of the signal will cause execution of the system call which triggered the signal 9.

A process may set its own routines for system calls by catching signal 9 through the use of the .signal system call. When a trap 0 is executed, the content of the top of the stack is the address of the word which contains the system call number. The process should check the value of that word to determine which system call is being executed. In addition, the process should increment the content of the top of the stack so it will properly be the return address.

If a process supplies its own signal routines and a signal is caught, subsequent signals of the same type received by the process will have the following effect:

a. Signals 0-7 will be ignored by the process.
b. Signal 9 will cause execution of the system call.
c. Signals 10-34 will kill the process.

The .signal system call must be executed again in order for the process to catch another signal of the same kind.

Vector Interrupts

Vector interrupts now result in more descriptive error messages.

CHANGES TO THE OPERATING SYSTEM

.Ksam System Call

Previously, the .ksam system call did not work for programs
that ran on the Z80 processor. This has been fixed.

IOP Driver

Previously a bug in **cromix.iop** occurred in certain
circumstances when using a modem. If a user hung up the
phone before logging off, and SIGHUPall was reset, the IOP
could fail to respond. This bug has been fixed.

CHANGES TO THE UTILITIES

Ddump Utility

The Ddump utility has been restored to the Cromix Operating
System package.

Help Utility

The help files have been updated.

Init Utility

Init now writes a 2-byte number in the hard disk type
specifier structure. This number is the cylinder position of
the start of the alternate tracks on the disk. Init's
procedure for alternate tracks changed because of the new
location of the alternate tracks of the Cromemco HD-20 drive.

Mount Utility

Mount now returns an error which can be used by the If Shell
command if a mount is unsuccessful.

Screen Editor

Previously, when only a RETURN was typed after entering the
Read or Write command, the command read the directory file or
attempted to write to the directory file. This bug has been
fixed.

You can no longer use the Screen editor program to edit files
with .sav or .ovr filename extensions.

Sim Utility

Previously, system call 6, Direct Console I/O, always
returned a 0 in the A-register if input was from a redirected
file. This has been changed to return the ASCII value read
in the A-register, indicating a character had been read.

Previously, system calls 11h, 12h and 9Ch, which are Search
Directory, Find Next Directory Entry, and List Directory,
respectively, opened the directory file each time the calls
were executed and did not close the directory file when
finished. If these system calls were executed enough times,
the Cromix error message "too many channel numbers for this
process" was generated. This bug has been fixed by not
opening the directory file if it is already open.

Previously, programs running under Sim were not dependably
receiving input from redirected input. This bug has been
fixed.

USING THE TU-ART UNDER THE CROMIX OPERATING SYSTEM

If a Tu-art board is connected to the system, the following
conditions will cause the Cromix Operating System to slow
drastically:

1.   A serial port of the Tu-art board is not connected to a
     terminal, and

2.   The **tty** entry in the **/etc/ttys** file of the terminal
     assigned to the port has a 1 in the first column.

A Tu-art serial port that is not connected to a terminal can
constantly send interrupts as though a character were ready
for input. If there is a 1 in the first column of a **tty**
entry in the **/etc/ttys** file for the terminal assigned to that
port, the Cromix Operating System will service the
interrupts, thus causing a slowdown in the system. This
problem is not produced by Tu-art serial ports that are
connected to either powered-up or powered-down terminals.

A 1 in the first column of the **tty** entry in the **/etc/ttys** file for an unconnected Tu-art serial port should be changed to a 0 using the Screen editor program. Then the operating system should be reset to reconfigure the Tu-art board.

A **tty** device listed in the **/etc/ttys** file is assigned to a terminal connected at a port on the Tu-art board as determined by the switch settings on the Tu-art board. More information on terminal configuration is described in Appendix A and Appendix C of the Cromix Disk Operating System Instruction Manual, part number 023-4022.

VERSION NUMBER SUMMARY

		Version
Files in /		
	cromix.iop.sys	20.14
	cromix.sys	20.14
Files in /bin		
RB	access.bin	00.06
	backup.bin	00.08
	blink.bin	00.13
	boot.bin	00.02
	cdoscopy.bin	00.15
RB	chowner.bin	00.06
	cmpasc.bin	00.05
RB	compare.bin	00.07
RB	copy.bin	00.10
	cptree.bin	00.07
RB	day.bin	01.02
	dcheck.bin	00.12
	ddump.bin	02.02
RB	deltree.bin	00.03
RB	dump.bin	00.10
RB	ecc.bin	00.02
RB	echo.bin	00.05
	ed.bin	01.40
RB	find.bin	00.07
	free.bin	00.09
RB	group.bin	00.01
RB	h.bin	00.04
RB	help.bin	00.04
	icheck.bin	00.15
	idump.bin	00.06
	init.com	02.82
RB	input.bin	01.00
RB	l.bin	00.11
	mail.bin	00.06
RB	makdev.bin	00.07
RB	makfs.bin	00.13
RB	maklink.bin	00.04
RB	match.bin	00.03
RB	mode.bin	01.12
RB	mount.bin	00.14
RB	move.bin	00.09
RB	msg.bin	00.08
RB	ncheck.bin	00.09
	passwd.bin	00.09
	patch.bin	00.03
RB	priv.bin	00.07
	restore.bin	00.05
RB	root.bin	00.02
	screen.bin	01.40

	sim.bin	02.55
	sort.bin	00.06
RB	spool.bin	00.12
RB	tee.bin	01.02
RB	testinp.bin	01.01
RB	time.bin	00.07
RB	unmount.bin	00.11
RB	usage.bin	00.06
RB	version.bin	00.09
RB	wboot.bin	00.09
RB	who.bin	00.06

Files in /dev/iop

RB	cromix.iop	11.16
RB	ioprun.bin	03.00

Files in /etc

	fdboot	00.09
	group	
	iopstartup.iop.cmd	
RB	login.bin	00.02
	motd	
	mtab	
	passwd	
	sfdboot	00.09
	startup.cmd	
	startup.msg	
	ttys	
	ttys.iop	
	warning	

Files in /gen

RB	crogen68.bin	00.02
	crogen68.dat	
	crolib68.rel	
	cromov.rel	
RB	default.bin	00.01
	iolib68.rel	

SOFTWARE NOTICE

CROMIX SIMULATOR AND VERSIONS 11.16 & 20.14

Many CP/M programs will not run correctly under the CDOS Simulator version 2.55 which is included in the current versions of Cromix (11.16) and Cromix-D (20.14).

Users who need CP/M compatibility should continue to use the Simulator from Cromix versions 11.11 or 20.09.

Cromemco programs like **init**, however, require the new version of the Simulator. Therefore, at times, it may be necessary to switch back and forth between the old and the new version of the Simulator. To simplify this process, use the following procedure to set up the old and the new versions of the Simulator for easy switching when necessary.

1. With Cromix 11.16 or 20.14 booted and the new Simulator version 2.55 loaded, as a priviledged user copy that simulator into a file called **newsim.bin** with the following command:

copy -vf /bin/sim.bin /bin/newsim.bin

2. Mount the older release of Cromix (version 11.11 or 20.09) and copy that version of the Simulator (0.34) into a bin file named /bin/oldsim.bin.

copy -vf /a/bin/sim.bin /bin/oldsim.bin

where /a is the mounted disk.

3. Now, using the Screen Editor, create two command files to facilitate switching between the versions. Programs which require the Simulator will be looking for the program /bin/sim.bin. Use the following procedure:

 # screen /cmd/sim34.cmd
maklink -f /bin/oldsim.bin /bin/sim.bin

This file **sim34.cmd** will now automatically make a link between the old simulator and the file **sim.bin**. Likewise, create a file to make a link between the new simulator and **sim.bin**.

 # screen /cmd/sim255.cmd
maklink -f /bin/newsim.bin /bin/sim.bin

The name of the command files to make the link between the simulators must be different from the names of the **bin** files which store them; i.e. the command files cannot be named **newsim.cmd** or **oldsim.cmd** because Cromix will search the directories for a **bin** file first, before looking in the **/cmd** directory.

Now when using a CP/M program, use the command **sim34** to set the old version of the Simulator in sim.bin. Later, when using a Cromemco program, like **init** which requires the new simulator, use the command **sim255** to set up that version of the Simulator.

This notice takes the place of the Software Notice dated May 25, 1983.

Customer Service
June 27, 1983

Cromemco Software Update Service Note 68000 Cromix-4

Date: ___November 4, 1983___

Product: ___CRO-DL and CRO-DS___

Release: ___4___

Date production of this version began: ___Nov. 4, 1983 on 8"___
                                           ___Nov. 4, 1983 on 5"___

First serial number with this version: ___4-10000 on 8"___
                                            ___4-10000 on 5"___

SUMMARY

Version 20.52 of the 68000 Cromix® Operating System is now available. This version corrects occasional interrupt handling and memory allocation problems encountered with previous releases. In order to provide a much larger number of available shell buffers and process table entries, **shell.bin** now resides in a separate bank of memory. This results in 68000 Cromix requiring 192 Kbytes of memory instead of 128 Kbytes. For details refer to the discussion of the Shell program. A number of enhancements have also been made to several utility programs, and a few new utility programs have been added.

The CRO-DS software package is now shipped on three 5.25-inch diskettes instead of two.

Versions of the 68000 Cromix Operating System later than 20.14 cannot be warm booted from versions 20.14 and earlier.

This version of 68000 Cromix must be COLD BOOTED from the supplied floppy disks.

CORRECTIONS

Interrupt Handling

This release corrects an occasional interrupt handling problem encountered in previous versions of the 68000 Cromix Operating System. The problem occurred when a Z80 or 68000 program made a system call that had to be serviced by the other processor. If, while executing this system call, the processor received an interrupt, it would save the signal with the processor's state and return control to the calling processor. A second interrupt, occurring while the 68000 was installing or checking the signal address. caused the state of the processor to be lost. The most common symptom of this

023-9576

problem was that executing command files would occasionally
terminate abruptly without any diagnostic message. Other
cases of incorrect system behavior might also have been
attributable to this problem.

Program Abort

Aborting the loading of 68000 programs under previous
versions of the 68000 Cromix Operating System resulted in the
loss of one shell buffer. Additionally, attempting to
unmount the disk the aborted program resided on resulted in
the message that the device was busy. This release of the
software correctly aborts the loading of 68000 programs,
without side-effects.

Memory Allocation

After allocating memory for a very large 68000 program (1 to
2 megabytes), previous versions of the 68000 Cromix Operating
System incorrectly allocated memory for processes created
while the large program was loaded. The memory allocated to
such processes could partially overlap the memory in use by
the original program. This release corrects this problem.

68000 programs that required an exact multiple of 64 Kbytes
were incorrectly allocated an extra 64-Kbyte block of memory.
This additional block would be inaccessible to the program,
but could not be allocated to other programs. This problem
is now corrected.

SLEEP System Call

In previous versions of the 68000 Cromix Operating System.
the sleep system call returned an incorrect value when
interrupted by a signal. It has been corrected to return the
number of seconds remaining when the signal was received.

User Signal

Previous versions of the Screen editor program and the shell
handled the user signal incorrectly. Specifically, if the
user called a shell from Screen (%) and set SIGenable, it was
then possible to kill Screen by sending it the user signal.
Aside from potentially destroying the data being edited, this
also made the shell a parentless process.

Blink Utility

Versions of Blink on previous releases would not allow the specification of a program address less than 100h. The current version of Blink no longer imposes this restriction.

Makfs Utility

In the previous version of **makfs.bin.** problems would arise if the **-i** option was used to create a non-default number of inodes and the number of inodes specified was not a multiple of 4 (4 inodes are contained in a block). Makfs now rounds the number of inodes specified by the user **down** to the nearest multiple of 4.

ENHANCEMENTS

RAM Disk Driver

Cromix 20.52 includes the RAM disk driver. The RAM disk is an area of memory that is set aside during system initialization which can then be used as a fast pseudo disk. A file structure should be built on it using the Makfs utility. The RAM disk can then be mounted and unmounted. It does not survive shutdown or boot. The RAM disk driver uses a simple checksum to ensure integrity. If a checksum error is reported on it, the contents of the RAM disk are very likely destroyed. Use the program

        fixrd device, device, ...

to recompute the checksums (otherwise, the checksum error will persist). (It will behave as if a hard disk had a bad track.) Later you should re-execute the Makfs utility on the RAM disk.

There are four devices supplied: **rd0**, **rd1**, **rd2**, and **rd3**. If your **/dev** directory does not already contain device files for RAM disks, you must create them using the Makdev utility (e.g., makdev rd0 B 5 0). Note that the major device number must be 5. Consult the **/dev** directory on your release diskette for examples.

68000 Cromix must be generated properly to use RAM disks. Crogen68 now includes a question which asks if you want to use RAM disks. If your answer is YES, you will be asked how many 64K segments of memory you are prepared to allocate for each of them.

One possible use of the RAM disk is to mount it to a file called **/ram**, and then copy frequently used programs and command files from the **/bin** and **/cmd** directories onto it, thus allowing for very fast program execution. The shell has been modified to look into the **/ram** directory (if it exists) before looking into the **/bin** and **/cmd** directories. The following is an example of a command file which creates such a RAM disk:

```
% This command procedure will set up a RAM Disk of size 128K.
% It may be called by /etc/startup.cmd upon each system boot.
echo; echo "Creating /ram directory ..."

create /ram >* /dev/null
makfs -i 64 rd0
if -err exit
mount rd0 /ram

d /bin
copy access.bin blink.bin cdoscopy.bin chowner.bin /ram
copy cmpasc.bin compare.bin copy.bin cptree.bin dump.bin /ram
copy echo.bin free.bin input.bin l.bin ls.bin mail.bin /ram
copy match.bin mode.bin mount.bin move.bin root.bin /ram
copy screen.bin sim.bin spool.bin testinp.bin time.bin /ram
copy unmount.bin usage.bin version.bin who.bin /ram

d /cmd
copy shutdown.cmd /ram

d /ram
chowner bin *.bin *.cmd /ram
chowner system cptree.bin mail.bin sim.bin version.bin
maklink screen.bin ed.bin
access rewa.e.e *.bin
access rewa shutdown.cmd

echo "/ram directory created"
```

Ccall Utility

```
   utility:  CCALL
   purpose:  This program allows users to call up another
             Cromix system (or possibly a non-Cromix system)
             using the Cromemco MDM-1200 (or possibly
             another modem).

user access:  all users

   summary:  ccall [-q] [-d devname] [-b baud]
                   [-n phonenumber] [-w]

 arguments:  none

   options:  -q   quiet (default is verbose)
             -d   qtty device name (default is /dev/modem)
             -b   baudrate (default is 1200 baud)
             -n   phone number (dashes may be used in number)
             -w   wait for manual dial and connection
```

Description

Ccall may operate at either 300 or 1200 baud. It must use an
asynchronous modem such as the Cromemco MDM-1200, a Bell
212A, or a Bell 103 type. The modem used must be compatible
with the modem on the remote system. The modem can be
connected to any serial port on the Quadart using a 12-wire
cable constructed for this purpose.

The qtty driver is used to connect ccall with the IOP/Quadart
and modem. The Cromix system being used must include the
IOP/Quadart drivers (see Crogen) and a device file for the
qtty should be set up in the **/dev** directory using Makdev.
The corresponding entry in the **/etc/ttys** file should have a 0
in the first column.

If a Cromemco MDM-1200 modem is being used, a telephone
number can be added onto the command line to perform
automatic dialing. Otherwise the call must be manually
originated using standard modem procedures.

Example

```
% ccall -b 300 -n 555-1212
```

This command line sets the transmission rate at 300 baud and
uses device qtty1 with a connected MDM-1200 that auto-dials
the number 555-1212.

Ccall interprets lines beginning with '~' as special escape sequences:

~.   Terminate Ccall.

~~   Send the line '~...'. To type a line beginning with '~' to the remote computer you must first type '~' twice followed by a carriage return; then you can type the rest of the line.

~< filename   Send the contents of a file to the remote system, as though typed at the terminal.

~> filename   Divert all characters received from the remote system into the specified file. '>' may be followed by '>' to append the output to the file. Either '>' may be followed by ':' to specify that the output be sent only to the file. If ':' is omitted, output is written both to the file and to the standard output. If '>>' is not used then the file must not exist prior to issuing this command. Typing '~>' will terminate the diversion. (Full syntax: ~>[>][:] filename)

~sh   Invoke an interactive shell on the local system.

~sh cmd   Run the command on the local system (via shell -c).

~put [-f] file-list
                Copy the specified files from the local system to the current directory on the remote system. If a filename already exists at the destination the force option (-f) must be used if you want the existing file to be overwritten, otherwise an error will be reported and the file will not be copied. **put** uses the Sfile and Rfile utilities to perform error-free block transfers. Rfile must exist on the remote system.

~take [-f] file-list
                Copy the specified files from the remote system to the current directory on the local system. If a filename already exists at the destination the force option (-f) must be used if you want the existing file to be overwritten, otherwise an error will be reported and the file will not be copied. **take** uses the Sfile Rfile utilities to perform error-free block transfers. Sfile must exist on the remote system.

~#   Send a one-second break signal to the remote
                       system.

~h   Print a summary of these commands.

Messages Returned by Ccall

Now waiting for call to complete ...

Ccall is waiting for a connection to be made with another
modem over the phone lines. This call can either be dialed
manually using Bell equipment or the MDM-1200 can dial the
number and establish the connection automatically.

No answer -- Call aborted

If a connection is not established within 120 seconds Ccall
exits back to the Cromix shell.

Options

The -q option keeps Ccall from being too verbose.

The -d option specifies which qtty device is to be the
transmitter

The -b option sets the baud rate of the transmitting device.

The -n option specifies a phone number to be used by the
MDM-1200 modem.

The -w option specifies that Ccall is to wait for the
connection to be manually originated.

Notes

If the -w option is used without a phone number then Ccall
waits for the user to establish a connection manually using
standard modem procedures. If the -n option and the -w
option are not used Ccall attempts to establish an
interactive dialog with the MDM-1200. The user can then use
all of the features of the MDM-1200 to establish a connection
with a remote system. (see MDM-1200 manual)

Rfile utility

```
        utility:    RFILE
        purpose:    This program allows binary files to be received
                    from users over the phone lines with error free
                    results.

   user access:    all users

        summary:    rfile [-q] [-f] [-d device-name] [-b baudrate]
                          dst-directory

      arguments:    destination directory pathname (must already
                    exist)

        options:    -q  quiet (default is verbose)
                    -f  force
                    -d  qtty device name (default is stdin/stdout)
                    -b  baudrate (default is current baudrate)
```

Description

The Rfile utility allows binary disk files to be received by
a user on one Cromix system from a user on another Cromix
system that is using the Sfile utility to transmit files.

Rfile may operate at either 300 or 1200 baud. It must use an
asynchronous modem such as the Cromemco MDM-1200, a Bell
212A, or a Bell 103 type. The modem used must be compatible
with the modem the Sfile utility is using to transmit the
data. The modem can be connected to any serial port on the
Quadart using a 12-wire cable constructed for this purpose.

The qtty driver is used to connect Rfile with the IOP/Quadart
and modem. The Cromix system being used must include the
IOP/Quadart drivers (see Crogen) and a device file for the
qtty should be set up in the **/dev** directory using Makdev.
The corresponding entry in the /etc/ttys file should have a 0
in the first column.

Example

```
    % rfile -d qtty2 -b 300 recvtemp
```

This command line sets the reception rate at 300 baud and
stores all of the data received from qtty2 into the existing
directory named recvtemp.

8

Messages Returned by Rfile

Waiting for phone call --

Rfile is in an idle state waiting for a connection from the Sfile utility. Rfile will remain in this mode indefinitely if Sfile fails to make a valid connection.

Receiving <filename> from <devname> into <directory-name>

A valid connection to the Sfile utility has been established and data is being transferred from the qtty device to the specified diskfile in the specified directory.

Connection lost -- ABORT RFILE

The line to Sfile was disconnected prematurely.

ABORT RFILE

A CONTROL-C character was received from the user at the keyboard; Rfile does an orderly exit back to the Cromix shell.

Options

The **-q** option specifies a different set of Rfile console messages. (Used by Ccall and when Rfile is running on a remote machine.)

The **-f** option causes Rfile to overwrite any existing file with the same pathname as the file sent by Sfile. If this option is not specified and another file exists with the destination pathname, an error is reported.

The **-d** option specifies which qtty device is to be the receiver.

The **-b** option sets the baud rate of the receiving device.

Notes

Rfile is used in conjunction with Sfile. Refer to the Sfile utility for additional information.

When used without an argument, Rfile displays a summary of the command line syntax.

Sfile Utility

```
     utility:  SFILE
     purpose:  This  program  allows  binary  files  to  be  sent
               between  users  over  the  phone  lines  with  error
               free  results.

user access:   all  users

     summary:  sfile  [-q]  [-d devname]  [-b baud]
                      [-n phonenumber]  [-l login-name]
                      [-p password]  file-list

   arguments:  one  or  more  filenames

     options:  -q  quiet  (default  is  verbose)
               -d  qtty  device  name  (default  is  stdout)
               -b  baudrate  (default  is  current  baudrate)
               -n  phone  number  (dashes  may  be  used)
               -l  login  name
               -p  password
```

Description

The Sfile utility allows binary disk files to be transmitted
from a user on one Cromix system to a user on another Cromix
system using the Rfile utility to receive files.

Sfile may operate at either 300 or 1200 baud.   It must use an
asynchronous modem such as the Cromemco MDM-1200, a Bell
212A, or a Bell 103 type.   The modem used must be compatible
with the modem the Rfile utility is using to receive the
data.   The modem can be connected to any serial port on the
Quadart using a 12-wire cable constructed for this purpose.

The qtty driver is used to connect Sfile with the IOP/Quadart
and modem.   The Cromix system being used must include the
IOP/Quadart drivers (see Crogen) and a device file for the
qtty should be set up in the **/dev** directory using Makdev.
The corresponding entry in the **/etc/ttys** file should have a 0
in the first column.

If a Cromemco MDM-1200 modem is being used a telephone number
can be added onto the command line to perform automatic
dialing.   Otherwise the call must be manually originated
using standard modem procedures.

Example

```
     % sfile -d qtty1 -b 300 -n 555-1212 letter.txt
```

This command line sets the transmission rate at 300 baud and transmits the file **letter.txt** using device qtty1 with a connected MDM-1200 that auto-dials the number 555-1212.

Messages Returned by Sfile

Now waiting for call to complete ...

Sfile is waiting for a connection to be made with another modem over the phone lines. This call can either be dialed manually using Bell equipment or the MDM-1200 can dial the number and establish the connection automatically.

No answer -- Call aborted

If a connection is not established within 60 seconds Sfile exits back to the Cromix shell.

Transmitting <filename> to <devname>

When a valid connection is established with the Rfile utility at the other end the specified file is then transmitted through the specified qtty device.

Rfile not responding -- Sfile aborted

If a connection is established with another modem Sfile determines if the Rfile utility is ready at that end. If Rfile is not running at the other end or if Rfile is running at an incompatible baud rate Sfile disconnects the line and exits back to the Cromix shell.

Connection lost -- ABORT SFILE

The line was disconnected in the middle of a transmission; Sfile then exits to the Cromix shell.

ABORT SFILE

A CONTROL-C character was received from the user at the keyboard; Sfile does an orderly exit back to the Cromix shell.

Options

The **-q** option specifies a different set of Sfile console messages (used by CCall and when Sfile is running on a remote system).

The **-d** option specifies which qtty device is to be the transmitter.

The **-b** option sets the baud rate of the transmitting device.

The **-n** option specifies a phone number to be used by the MDM-1200 modem.

The **-l** option specifies a login name to be used on the remote Cromix system.

The **-p** option specifies a password to be used on the remote Cromix system.

Notes

Sfile is used in conjunction with Rfile. Refer to the Rfile utility for additional information.

When used without an argument, Sfile displays a summary of the command line syntax.

Make Utility

Note: Due to space limitations, **make.bin** has been added to the **/bin** directory on the <u>second</u> CRO-DS release disk.

purpose: This utility automates the construction of executable programs from separate modules.

user access: all users

summary: make [-vdft] filename[.mak]

arguments: instruction file

options: -v verbose
-d debug
-f force
-t touch

Description

Most complex programs are constructed from a number of separate modules. Make provides the means for automatically executing the necessary steps (i.e., compilations, assemblies, linkages) to construct a finished program. It also provides selective execution of just those steps necessitated by the modification of any of the constituent modules.

Since the actions of Make are predicated on date and time, it is very important to keep the system time and date reasonably accurate.

Make takes its instructions from a text file, which must have the suffix **.mak**. The **.mak** file consists of two kinds of lines:

**conditional
statements** - those starting in column one with a colon somewhere in the line.

commands - those starting with at least one white space character (blank or TAB).

Make scans each conditional statement line. If any of the files following the colon have been modified later than any of the files before the colon, the commands on the command lines following the conditional statement are executed.

The following is an example based on the Cromemco 68000
FORTRAN-77 environment:

```
    prog1.obj:      prog1.for progdata.for
        fortran prog
        code prog1.i prog1.obj
    prog2.obj:      prog2.for progdata.for
        fortran prog2
        code prog2.i prog2.obj
    prog.bin:       prog1.obj prog2.obj /usr/lib/ftnlib.obj
        crolinker -oprog prog1 prog2 /usr/lib/ftnlib
```

The above lines of instructions tell the utility which parts
of the program construction process should be executed if any
of the constituent files have been modified.

Multiple files on either side of the conditional colon are
permitted. They must, however, be separated by at least one
white space. Thus:

```
    prog2.obj:      prog2.for progdata.for
```

is equivalent to:

```
    prog2.obj:      prog2.for
    prog2.obj:      progdata.for
```

and

```
    foo.obj:      foo.for /usr/lib/ftnlib.obj
    bar.obj:      foo.for /usr/lib/ftnlib.obj
```

is equivalent to:

```
    foo.obj bar.obj:      foo.for /usr/lib/ftnlib.obj
```

If the files to the right of the colon have not been modified
more recently than those to the left of the colon, none of
the commands are executed and the Make utility scans for the
next conditional statement.

Options

The **-v** option displays the program's progress.

The **-d** option displays the times that Make used for its conditional comparisons.

The **-f** option causes all actions to be taken regardless of modify times.

The **-t** option updates the modified times for the dependent files, thus artificially updating all of the files to the current time. This may be used if one or more of the dependent files has been "touched", but not actually changed.

Default Utility

The Default program, used to configure the default parameters for automatic login, has been revised to work with the 68000 Cromix Operating System.

Flush Utility

Flush is a new system-maintenance program that flushes (i.e., writes to disk) system I/O buffers every specified number of seconds. Normally, for efficiency, the Cromix Operating System will not write the contents of a buffer to disk until it has been completely filled. If you conclude an edit session and leave your system running overnight, portions of your file may reside in system buffers overnight (not written to disk). If the system is later abruptly powered down without running shutdown, or if there is a power failure, this information will be lost. As a safeguard, it is desirable to periodically flush the I/O buffers using the Flush utility.

The file **flush.bin** resides in the **/bin** directory. It is called by:

        flush number &

where **number** is the number of seconds between two consecutive flushes. Currently the **startup.cmd** file is configured to invoke flush, as a background process with an interval of 30 seconds, every time the system is booted.

Hard Disk Drivers

The hard disk drivers have been improved to compensate for timing variations in the Model 5007 five-inch hard disk. In the past. timing differences between the writing and reading of data could result in CRC errors. The new drivers greatly reduce the probability of data loss due to timing differences.

Hdboot

With the impending release of RDOS 3.08, Cromix systems will be able to boot directly from their hard disk. The file **hdboot**, contained in the **/etc** directory, is included in this release to provide users who upgrade to RDOS 3.08 with this capability. As with **fdboot** and **sfdboot**, the information contained in **hdboot** can be written to the system tracks using **wboot**. For example, the command "wboot hd0" will write the boot information onto hard disk unit 0.

List Utility

The file **ls.bin** has been added to the **/bin** directory. This is an enhanced version of the standard List utility. When called without options, **ls** displays filenames only, arranged in a multi-column format. Since file length and link count information is not displayed, the display is produced much faster than that of the standard List utility. The options to List are unchanged. Specifying any option (or just typing **ls -**) causes the program to act in the same manner as the standard List utility.

Sim Utility

The CDOS simulator, **sim.bin**, has been upgraded. It not only executes faster, but is also more flexible than the previous version. As in previous versions, **sim.bin** will, by default, map the CDOS directories B-H to the Cromix directories **/b-/h**. With this version, however, the user can permanently redefine the default mapping or, when calling **sim.bin** explicitly, specify an alternate mapping. The use of **sim.bin** is as described in the Cromix Instruction Manual, except for the added capability to specify different mappings when calling **sim.bin** explicitly.

The permanent mapping of CDOS directories to Cromix directories is defined by the contents of a table contained in **sim.bin**, starting at byte location 509h. This table contains a 16-byte entry for each CDOS directory. This entry is the pathname of the Cromix directory that the particular CDOS directory maps to. The Dump utility may be used to

display the contents of this table. The entry for CDOS
directory A starts at location 509h, the entry for CDOS
directory B starts at location 519h, and so on.

If you want to permanently change the mapping of one or more
CDOS directories, you can use the Patch utility to change the
contents of this table. The entry for drive A refers to the
current directory and should not be changed. When changing
an entry:

1.   Each pathname must have a '/' at the end.

2.   Each string must be terminated with a null.

3.   The pathname, including '/', cannot exceed 15 characters
      in length.

The mapping of CDOS directories to Cromix directories may
more conveniently be defined when calling **sim.bin** explicitly.
For instance:

     sim -c /usr/lib pathname arguments

will refer all references to the CDOS C directory to the
Cromix **/usr/lib** directory.   Thus, one or more CDOS
directories may be remapped to specific Cromix directories by
typing the letter of the CDOS directory as a flag and then
typing the Cromix directory as the next argument. Note that
the pathname of the executable program above must be a
complete pathname (including extension). The shell's search
mechanism will apply only to the name **sim.**

Version Utility

The **version** program has been extended to check the
consistency of a file. In general, if the following string
of bytes is found anywhere within a file:

     0FDH 0EDH 0FDH 0EDH

the bytes immediately following are assumed to contain CRC
information, version number, release number, and program
name. When called with the **-c** option, **version** will compute a
CRC value for that file.   In the two bytes immediately
following the FDEDFDED pattern, will be written the version
of the **version** program actually writing the CRC, followed in
the next two bytes by the CRC value itself.   When called
without **-c**, **version** reports good or bad file consistency by
comparing a newly calculated CRC with the recorded CRC. As.

before, version information about the file is also displayed. Thus, **version** can be run on the **/bin** directory for example, to check the correctness and completeness of all executable programs.

To be more specific, consider the following portion of a file:

```
db      0FDH,0EDH,0FDH,0EDH
db      0,0,0,0
db      version, revision
db      'program name\n'
```

When called with the **-c** option, the version/revision number of the **version** program itself will be written to the first two bytes following the FDEDFDED pattern. The CRC will then be written to the following two bytes.

Retype Mode

The "RETYpe" mode (command line editor) has been included in the Shell program, version 01.07. It can be turned on or off using the Mode program (affects mode3, bit6).

```
mode -retype
```

or

```
mode retype
```

Note that the "RETYpe" feature can be used only on a 3102 terminal or equivalent, e.g., C-10.

"RETYpe" works as follows. If CONTROL-R is typed as the first character on a new command line, the previous command line is redisplayed. If the first character typed is something else, the previous line is lost. Whatever line the display shows may be edited. The editing functions will work on a current command line (one not retrieved using CONTROL-R) as well. The following keys are effective:

```
DELETE key          delete the character to the left of the
                    cursor
LEFT ARROW          move cursor left
(CONTROL-H)
RIGHT ARROW         move cursor right
(CONTROL-L)
CONTROL-I           turn on Insert mode
ESCAPE              turn off Insert mode
```

If Insert mode is on, the new characters are inserted in the line at the cursor position. If Insert mode is off, the new characters replace the existing ones. Line editing starts with Insert mode off.

The line that is sent consists of whatever the screen shows at the time RETURN is pressed. Note that moving the cursor left does not erase characters.

MOVE CCR Instruction

The 68000 and 68010 processors differ in handling the instructions that move the status register:

```
+--------+-----------------------------+------------------------+
|        |      MOVE SR,<ea>           |     MOVE CCR,<ea>      |
+--------+-----------------------------+------------------------+
| 68000  | normal instruction          | illegal instruction    |
| 68010  | privileged instruction      | normal instruction     |
+--------+-----------------------------+------------------------+
```

Under these circumstances, it is normally impossible to write user programs that will run on both processors without modification.

Version 20.52 of the Cromix Operating System emulates the MOVE CCR instruction on the 68000 processor so that programs which use it can run on either processor. The emulation is completely transparent to the user. The system behaves as if the 68000 had the MOVE CCR instruction in its instruction set.

The MOVE SR instruction is likewise emulated by Cromix on the 68010 processor, artificially allowing the nonprivileged user to execute the instruction. Since it may not be possible to continue with this emulation scheme in the future, users are strongly encouraged to write their code as though meant for execution on the 68010 processor (using the MOVE CCR instruction).

Shell Program

Cromix version 20.52 has the Shell program located in a user
bank, not in the system bank (one copy of the shell is shared
by all users). The result is that Cromix, together with
Shell, takes 192 Kbytes instead of 128 Kbytes of memory. On
the other hand, there is a much greater number of shell
buffers available than previously. To implement this
solution, a block device driver that can access all the 68000
memory has been included in the system.

The present version of Shell is 01.07. For the new Shell to
be able to run, the memory driver (3,0) must be included in
cromix.sys, the Shell program must be in the **/etc** directory,
and the device **amem** must be in the **/dev** directory. The file
/etc/shell.bin must be owned by the system. and the access
privilege specifier string for this file must be
r--- ---- ----.

/dev/amem

The **/dev** directory should include the device **amem**. This
device driver allows all 68000 memory to be examined. It may
be created with the command

makdev /dev/amem b 3 0

For this command to be executed, the (3, 0) memory driver
must be in the system. Access privileges should be rewa..
and ownership should be bin.

/gen/iolib68.rel

This is a new version of the library, which includes the
(3,0) block device driver. This driver allows the authorized
user to read and write at any position in the 68000 memory.
The Pstat function in the shell uses this driver to inspect
the process tables. Inclusion of this driver into Cromix is
guaranteed by the new version of Crogen68.

Other changes to **iolib68.rel** include corrections to the qtty
driver and a rewritten typ printer driver, described
elsewhere.

/gen/crolib68.rel

The size of this version of **crolib68.rel** has been reduced as
the Shell program has been removed from the system. Also the
larger available space for the system tables is now utilized
differently.

/gen/crogen68.dat

This file contains the code for the 68000 processor, mainly
the system call interface. The interface from the 68000
system calls to the Z80 system call handler has been
thoroughly inspected. A number of minor errors have been
found and corrected. Also, all system calls now modify only
registers that are explicitly defined to render the results.
In case of an error, the carry flag is set, D0 contains the
error number, and all other registers are unchanged. Also
the jsys _boot system call has been added.

To use the _boot system call, the user should kill all other
processes, read the new operating system into memory, and
then:


```
lea       address,A0
lea       stack,A1
move.l    size,D1
jsys      #_boot
```


Here **address** is the address of the first byte of the new
operating system, **stack** is the initial stack pointer for the
new operating system (if it is 68000 code, otherwise
meaningless), **size** is the size of the operating system in
bytes (if it is Z80 code, just passed to the operating system
as if 68000 code).

After verifying the user is a privileged user, the system
turns off interrupts and ecc. If the first byte of the new
operating system is 0xC3 (beginning of Z80 jmp instruction),
the current operating system believes the new operating
system starts with Z80 code, more specifically, that it is a
cromix.sys file. The **size** bytes (limited to 0xFA00 bytes)
are moved to address 0x100. The system believes the
communications between Z80 processor and the 68000 processor
(over 0xFC00 - 0xFFFF area) are still operable and uses them
to execute **jmp 100h** in Z80 in bank 0. From this point, the
new operating system in Z80 should take over. The way Cromix
operates now, the Z80 code believes that the 68000 processor
was either reset or was executing a Cromix version 20.16 or
later before so that the top of memory communications still
work to the extent that the 68000 processor waits at the
expected address.

If the first byte of the new operating system is not 0xC3, the current operating system believes the new operating system is 68000 code. It then loads:

       AO   address of new code (user AO register)
       SP   address supplied by the user (user A1 register)
       D1   value supplied by the user (usually size of code)
       D2   major and minor device number of the root device of
               the current operating system

After **jmp (AO)**, the new operating system is on its own. At this time, interrupts and ecc are disabled, and the new operating system is in supervisor mode.

Crogen68 Program

This is a new version of Crogen68. It differs from the previous version in that it automatically includes the memory driver (3,0) and asks the user if the RAM disk driver (5,0) or the SMD driver (4,0) (currently not supplied) should also be included. Crogen68 no longer prompts for a simple YES/NO response regarding the inclusion of hard disk and typewriter printer drivers. Crogen68 now expects a response indicating the number of these devices in the range: 0-2 for the typewriter printer and 0-3 for hard disks. The size of **cromix.sys** can be reduced slightly by including fewer of these devices.

TYP - Cromix Typewriter Printer Driver

The Cromix Typewriter Printer Driver has been enhanced. The following is a list of new modes, status conditions and special character sequences.

Additional Modes:

LMargin   Left printing margin in 0.1 inch steps from print position 1. Defines where printing will begin after a carriage return. The printer may be backspaced to the left of the left margin setting. The printer will clip at print position 1.

Width   Right printing margin in 0.1 inch steps from print position 1. Printer will clip at right margin if -WRAParound mode is set or wrap at right margin if WRAParound is set.

CWidth   Character width. Default spacing between
                    characters in 1/120 of an inch.

LHeight   Line height. Default spacing between lines in
                    1/48 of an inch.

Printer Status Conditions Returned By Mode:

(-)CHECK   Printer is (not) in a check condition. A
                    check condition requires the printer to
                    be reset before use.

(-)PAPER IN   Paper is (not) loaded into the printer.

(-)RIBBON IN   Ribbon is (not) loaded into the printer.

(-)ON LINE   Printer is (not) on-line and selected.

Single Special Characters Recognized By the Driver:

(CONTROL-A) 01h   Toggle Underline

(CONTROL-B) 02h   Line space back 1/2 current line setting
                    (begin superscript)

(CONTROL-C) 03h   Line space forward 1/2 current line
                    setting (begin subscript)

(CONTROL-D) 04h   Force immediate printing of any
                    characters and spacing commands in
                    printer buffer

(CONTROL-E) 05h   Enter dot plot mode

(CONTROL-F) 06h   Toggle boldface

(CONTROL-Y) 19h   Special mode prefix character (see below)

(Delete) 7Fh   reset printer

80h-BFh   Set temporary CWidth = char-80h in 1/120
                    of an inch increments

C0h-FFh   Variable Width Space = char-C0h in 1/120
                    of an inch increments

23

Two Character Special Sequences Recognized by the Driver:

19h, 80h-BFh	Set temporary LHeight = char2-80h in 1/48 of an inch increments
19h, 20h	32 spaces
19h, 08h	32 backspaces
19h, 02h	Line space back 16 times current line setting
19h, 03h	Line space forward 16 times current line setting

Three Character Special Sequences Recognized by the Driver:

19h, C0h-FFh, C0h-FFh   Reserved

For further information about single character commands refer to:

Cromemco 3355A Printer Driver Instruction Manual, part number 023-4005.

CHANGES

Modeequ.z80

The section at the end of this file, which equated the old names for mode functions to the new names, has been removed. Be sure to refer to all mode functions by their new names.

Mode.bin

Mode.bin has been modified to reflect the changes in the new Typ driver.

Added Files

The following file has been added to the CRO-DL (8-inch) and CRO-DS (5.25-inch) diskettes:

1. The Cdosfix utility has been added to the /bin directory. The Cdosfix program can be used on a Cromix system to remove the padding characters (CONTROL-Z) from the ends of CDOS files after they are transferred from CDOS to Cromix diskettes. See the file **/help/cdosfix.hlp** for documentation.

Cromemco Software Update Service Note 68000 Cromix-5

Date: <u>December 2, 1983</u>

Product:<u> CRO-DL and CRO-DS</u>

Release: <u>   5   </u>

Date production of this version began: <u>Nov. 30, 1983 on 8"</u>
<u>Nov. 30, 1983 on 5"</u>

First serial number with this version: <u>   5-10000   on 8"</u>
<u>   5-10000   on 5"</u>

SUMMARY

Version 20.56 of the 68000 Cromix® Operating System is now available. This version corrects several problems encountered with previous releases. Enhancements have also been made to several utility programs.

ENHANCEMENTS

Cptree Utility

The Cptree utility now preserves not only the user and access mode of copied files, but also all the dates (creation, modify, access, and dumped). Copies of files made with the Cptree utility are identical to the original in every respect. Because the new copy of the file keeps the original modify date, copy operations with the **-t** (time) option will behave more naturally.

Copy Utility

The Copy utility behaves the same way as Cptree. The copied file has the same user, access mode, and dates as the original.

Move Utility

The Move utility (when copying from a source file to a destination file) behaves the same way as Cptree. The destination file has the same user, access mode, and dates as the source file.

Sim Utility

System call 28h is now implemented in Sim.

023-9590

CORRECTIONS

Shell Utility

In the previous release, very long argument lists would sometimes cause the Shell (or even the Cromix Operating System) to crash. Shell also sometimes mixed up command lines. These problems have been corrected.

.Getdir System Call

A bug in the .getdir (02h) system call has been corrected.

.Error System Call

The .error system call (1Ch) now handles CONTROL characters correctly.

Printer Devices

Previously, discarded printer drivers could cause the system to crash. This problem has been corrected.

Tee Program

In the previous release, the Tee program did not erase old files. This problem has been corrected.

VERSION NUMBER SUMMARY

Files in /
cromix.iop.sys	20.56	-new-
cromix.sys	20.56	-new-

Files in /bin
access.bin	00.06	
backup.bin	00.08	
blink.bin	00.14	-new-
boot.bin	00.02	
ccall.bin	00.07	-new-
cdoscopy.bin	00.15	
cdosfix.bin	00.01	-new-
chowner.bin	00.06	
cmpasc.bin	00.05	
compare.bin	00.07	
copy.bin	00.11	-new-
cptree.bin	00.08	-new-

2

```
             day.bin           01.02
             dcheck.bin        00.12
             ddump.bin         02.02
             deltree.bin       00.03
             dump.bin          00.10
             ecc.bin           00.02
             echo.bin          00.05
             ed.bin            01.45        -new-
             find.bin          00.07
             fixrd.bin         00.01        -new-
             flush.bin         00.01        -new-
             free.bin          00.09
             group.bin         00.01
             h.bin             00.04
             help.bin          00.04
             icheck.bin        00.15
             idump.bin         00.06
             init.com          02.82        -new-
             input.bin         01.00
             l.bin             00.11
             ls.bin            00.01        -new-
             mail.bin          02.02        -new-
             makdev.bin        00.07
             make.bin          00.04        -new-
             makfs.bin         00.13
             maklink.bin       00.04
             match.bin         00.03
             mode.bin          01.15        -new-
             mount.bin         00.14
             move.bin          00.10        -new-
             msg.bin           00.08
             ncheck.bin        00.09
             passwd.bin        00.09
             patch.bin         00.03
             priv.bin          00.07
             restore.bin       00.05
             rfile.bin         00.07        -new-
             root.bin          00.02
             screen.bin        01.45        -new-
             sfile.bin         00.07        -new-
             sim.bin           02.65        -new-
             sort.bin          00.06
             spool.bin         00.12
             tee.bin           01.03        -new-
             testinp.bin       01.01
             time.bin          00.07
             unmount.bin       00.11
             usage.bin         00.06
             version.bin       00.10        -new-
             wboot.bin         00.09
             who.bin           00.06
```

Files in /dev/iop
 cromix.iop   11.21   -new-
 ioprun.bin   03.00

Files in /etc
 fdboot   00.11   -new-
 hdboot   00.05   -new-
 login.bin   00.02
 sfdboot   00.11   -new-
 shell.bin   01.10   -new-

Files in /gen
 crogen68.bin   00.08   -new-
 crogen68.dat   -new-
 crolib68.rel   -new-
 cromov.rel
 default.bin   00.02   -new-
 iolib68.rel   -new-

Cromemco Software Update Service Note 68000 Cromix-6

Date: February 21, 1984

Product: CRO-DL and CRO-DS

Release: 6

Date production of this version began: Feb. 21, 1984 on 8"
Feb. 21, 1984 on 5"

First serial number with this version: 6-10000 on 8"
6-10000 on 5"

SUMMARY

Version 20.61 of the 68000 Cromix® Operating System is now available.

This SUDS note describes the following: 1) changes to the Cromix release-disk configuration (how the operating system is supplied on disk); 2) changes to the Cromix Operating System and utilities; and 3) the new Octart driver, which is now supplied with the Cromix system.

CROMIX RELEASE-DISK CONFIGURATION

CRO-DL (8") and CRO-DS (5-1/4") Disks

The Cromix release disks are no longer supplied with a version of the operating system to run with IOP/Quadart terminals (**cromix.iop.sys**). IOP/Quadart users who wish to boot their systems from the new release disks can use this procedure:

1. Attach a terminal to the 16FDC or 64FDC.

2. Update to primary storage media (for CRO-DS users, this is a new procedure, described in the next section.

3. Boot onto primary storage media (hard disk).

4. Run Crogen68 to generate a new, properly configured Cromix Operating System.

5. Reboot, with the system hardware in its original set-up.

The **/dev/iop** directory has been deleted from all Cromix release disks. The files from this directory have been renamed and moved to the **/etc** directory.

023-9593

File Name in /dev/iop	File Name in /etc
ioprun.bin crcmix.iop	ioload.bin quadart.iop

A new file, **octart.iop**, has been added to the **/etc** directory to support Octart system configurations.

CRO-DS (5-1/4") Disks

Release disk 2 (the second 5-1/4" disk) can no longer be used to boot the system. As a result, the procedure for updating the hard disk (or other primary storage media) has changed.

The Update command file on disk 2 has been removed. Disk 1 now contains <u>two</u> Update command files -- **update1.cmd** and **update2.cmd**. Update1 transfers the contents of disk 1 to the hard disk -- Update2 transfers the contents of disk 2 to the hard disk.

To transfer the contents of release disk 1 to the hard disk, boot the system from disk 1. Then log in as a privileged user, and enter the command:

update1 [drive]

where **drive** is the <u>destination</u> drive (for example, **hd0**).

After transferring the files from disk 1, you will then use Update2 to transfer the files from release disk 2.

First, boot onto the updated disk, and insert disk 2 into a disk drive. Then, as a privileged user, enter the command:

update2 [drive]

where **drive** is the <u>source</u> drive (**sfda**, **sfdb**, **sfdc**, or **sfdd**).

Note: As with previous versions of Update, the argument to Update1 is the <u>destination</u> drive. For Update2, it is the <u>source</u> drive (the floppy disk drive containing release disk 2).

CROMIX OPERATING SYSTEM AND UTILITIES

Cromix Operating System

During startup, Cromix now checks the files **iolib68.rel** and **shell.bin** for version compatibility. Cromix will not boot if either of these is not the appropriate version.

Cromix has been upgraded to recognize version 01.01 of **iolib68.rel**.

Cromix no longer begins executing the **startup.cmd** file before the **iostartup.cmd** file has finished executing.

A process waiting on pipe read can now be aborted, thus avoiding interlock problems.

CDOS Simulator

Mode handling in Sim (the CDOS simulator) has been reorganized to correct problems encountered with certain programs (for example, WordStar⊕).

Cromix System Calls

The Cchstat and Fchstat system calls now correctly change the times associated with a file.

When using the Getmode and Setmode system calls in 68000 Cromix, results larger than one byte are now handled properly.

C-Net

The command file that starts the network (**net.cmd**) must be changed to reflect the fact that file **ioprun.bin** has been renamed **ioload.bin** and moved to the **/etc** directory.

Cptree Utility

Device files are now displayed properly when copied by the Cptree utility.

\

WordStar is a registered trademark of Micropro International Corporation.

\

Screen Program-Entry Editor

The Screen Program-Entry Editor now sets the Cromix error-return flag correctly.

Iolib68

The file **iolib68.rel** now includes an internal version number. This version number is used by the Cromix Operating System -- it cannot be displayed by the Version utility.

KNOWN PROBLEMS

If you are using a typewriter printer, reloading paper (after running out) does not restart the printer. To resume printing, enter the following command:

    % mode device-name

THE OCTART DRIVER

Background

The Octart board reduces the overhead associated with character processing by utilizing distributed processing techniques. This reduces the burden on the central processing unit (Cromemco DPU board), which, in turn, can increase processor throughput.

Connecting Terminals with the Octart

Terminals may be connected to a Cromemco computer running under the Cromix Operating System by using Octart boards. This section covers hardware installation of the Octart boards.

Hardware Set-up

The Cromix Operating System will accommodate up to four Octarts. Eight terminals can be connected to each Octart, for a theoretical total of 32 terminals.

Octarts and IOP/Quadarts can be installed in the same machine. (One Octart is equivalent to one IOP and two Quadarts.)

To simplify installation, each Octart has been assigned a
number (Octart 1 through 4). These numbers are used to refer
to the corresponding Octart for each qtty terminal.

Octart Switch Settings

Switch 7 (address selection switch) on the Octart should
be set as follows:

IOP Number	Base Address	Terminals Supported
Octart(1)	CEh	qtty1 - qtty8
Octart(2)	BEh	qtty17 - qtty24
Octart(3)	AEh	qtty33 - qtty40
Octart(4)	9Eh	qtty49 - qtty56

Note: If you're using IOP/Quadarts and Octarts in the same
system, the base addresses of the IOP/Quadarts and
Octarts must be distinct.

Refer to the section "Device Definitions" for major and minor
device numbers.

Octart Priority

Each Octart must be connected in the priority interrupt
chain. It is suggested that the Octarts be connected after
the 16FDC/64FDC and before the PRI (16FDC or 64FDC priority
out connected to priority in on Octart, Octart priority out
connected to priority in on PRI).

Software

/etc/quadart.iop   Formerly **cromix.iop**. This program is
                              loaded into the IOP/Quadart I/O
                              processor. It is responsible for
                              controlling up to four Quadarts (16
                              channels).

/etc/octart.iop   This program is loaded into the Octart
                              I/O processor. It contains the drivers
                              for Octart's eight serial channels.
                              It is responsible for communication
                              between Cromix and those channels.

/etc/ioload.bin	This program loads the software (octart.iop or quadart.iop) into the appropriate I/O processor.
/etc/oct_reset.bin	This program resets the Octart, preparing it for reloading.

Examples

```
/etc/ioload octart.iop io1
/etc/ioload quadart.iop io2
```

The first command loads the **octart.iop** program into the Octart I/O processor addressed at port CEh. The second command loads the **quadart.iop** program into the IOP/Quadart I/O processor addressed at port BEh.

Device Definitions

Device Name	Octart	Base Port	Device Number Major : Minor
qtty1	Octart(1)	CEh	2:0
qtty2	Octart(1)	CEh	2:1
qtty3	Octart(1)	CEh	2:2
qtty4	Octart(1)	CEh	2:3
qtty5	Octart(1)	CEh	2:4
qtt61	Octart(1)	CEh	2:5
qtty7	Octart(1)	CEh	2:6
qtty8	Octart(1)	CEh	2:7
qtty17	Octart(2)	BEh	2:16
qtty18	Octart(2)	BEh	2:17
qtty19	Octart(2)	BEh	2:18
qtty20	Octart(2)	BEh	2:19
qtty21	Octart(2)	BEh	2:20
qtty22	Octart(2)	BEh	2:21
qtty23	Octart(2)	BEh	2:22
qtty24	Octart(2)	BEh	2:23
qtty33	Octart(3)	AEh	2:32
qtty34	Octart(3)	AEh	2:33
qtty35	Octart(3)	AEh	2:34
qtty36	Octart(3)	AEh	2:35
qtty37	Octart(3)	AEh	2:36
qtty38	Octart(3)	AEh	2:37
qtty39	Octart(3)	AEh	2:38
qtty40	Octart(3)	AEh	2:39
qtty49	Octart(4)	9Eh	2:48

qtty50	Octart(4)	9Eh	2:49
qtty51	Octart(4)	9Eh	2:50
qtty52	Octart(4)	9Eh	2:51
qtty53	Octart(4)	9Eh	2:52
qtty54	Octart(4)	9Eh	2:53
qtty55	Octart(4)	9Eh	2:54
qtty56	Octart(4)	9Eh	2:55

VERSION SUMMARY

CRO-DL (8") Disks

Files in /
cromix.sys	20.61	-new-

Files in /bin
access.bin	00.06	
backup.bin	00.08	
blink.bin	00.14	
boot.bin	00.02	
ccall.bin	00.07	
cdoscopy.bin	00.15	
cdosfix.bin	00.01	
chowner.bin	00.06	
cmpasc.bin	00.05	
compare.bin	00.07	
copy.bin	00.11	
cptree.bin	00.09	-new-
day.bin	01.02	
dcheck.bin	00.12	
ddump.bin	02.02	
deltree.bin	00.03	
dump.bin	00.10	
ecc.bin	00.02	
echo.bin	00.05	
ed.bin	01.46	-new-
find.bin	00.07	
fixrd.bin	00.01	
flush.bin	00.01	
free.bin	00.09	
group.bin	00.01	
h.bin	00.04	
help.bin	00.04	
icheck.bin	00.15	
idump.bin	00.06	
init.com	02.82	
input.bin	01.00	
l.bin	00.11	
ls.bin	00.01	
mail.bin	02.02	
makdev.bin	00.07	
make.bin	00.04	-new-

```
        makfs.bin          00.14
        maklink.bin        00.04
        match.bin          00.03
        mode.bin           01.15
        mount.bin          00.14
        move.bin           00.10
        msg.bin            00.08
        ncheck.bin         00.09
        passwd.bin         00.09
        patch.bin          00.03
        priv.bin           00.07
        restore.bin        00.05
        rfile.bin          00.07
        root.bin           00.02
        screen.bin         01.46         -new-
        sfile.bin          00.07
        sim.bin            02.67         -new-
        sort.bin           00.06
        spool.bin          00.12
        tee.bin            01.03
        testinp.bin        01.01
        time.bin           00.07
        unmount.bin        00.11
        usage.bin          00.06
        version.bin        00.10
        wboot.bin          00.09
        who.bin            00.06
```

Files in /etc
```
        fdboot             00.11
        hdboot             00.05
        ioload.bin         03.01         -new-
        login.bin          00.02
        octart.iop         11.18         -new-
        oct_reset.bin      01.00         -new-
        quadart.iop        11.21
        sfdboot            00.11
        shell.bin          01.10
```

Files in /gen
```
        crogen68.bin       00.09         -new-
        crogen68.dat                     -new-
        crolib68.rel                     -new-
        cromov.rel
        default.bin        00.02         -new-
        iolib68.rel                      -new-
```

CRO-DS (5-1/4") Disks

Disk 1:

Files in /
```
        cromix.sys         20.61         -new-
```

Files in /bin

access.bin	00.06	
boot.bin	00.02	
cdoscopy.bin	00.15	
chowner.bin	00.06	
cmpasc.bin	00.05	
compare.bin	00.07	
copy.bin	00.11	
cptree.bin	00.09	-new-
day.bin	01.02	
dcheck.bin	00.12	
deltree.bin	00.03	
dump.bin	00.10	
ecc.bin	00.02	
echo.bin	00.05	
ed.bin	01.46	-new-
find.bin	00.07	
fixrd.bin	00.01	
flush.bin	00.01	
free.bin	00.09	
group.bin	00.01	
h.bin	00.04	
help.bin	00.04	
icheck.bin	00.15	
idump.bin	00.06	
init.com	02.82	
input.bin	01.00	
l.bin	00.11	
ls.bin	00.01	
mail.bin	02.02	
makdev.bin	00.07	
makfs.bin	00.14	
maklink.bin	00.04	
match.bin	00.03	
mode.bin	01.15	
mount.bin	00.14	
move.bin	00.10	
msg.bin	00.08	
ncheck.bin	00.09	
passwd.bin	00.09	
patch.bin	00.03	
priv.bin	00.07	
root.bin	00.02	
screen.bin	01.46	-new-
sim.bin	02.67	-new-
sort.bin	00.06	
spool.bin	00.12	
tee.bin	01.03	
testinp.bin	01.01	
time.bin	00.07	
unmount.bin	00.11	
usage.bin	00.06	
version.bin	00.10	
wboot.bin	00.09	

9

who.bin   00.06

Files in /etc
    etupdboot   00.11
    hdupboot   00.05
    ioload.bin   03.01   -new-
    login.bin   00.02
    octart.iop   11.18   -new-
    octart_reset.bin   01.00   -new-
    quadart.iop   11.21   -new-
    sfdboot   00.11
    shell.bin   01.10

Files in /gen
    crogen68.bin   000.09   -new-
    crogen68.dat   -new-
    crolib68.rel   -new-
    cromov.rel
    default.bin   00.02
    iolib68.rel   -new-

On Disk 2:

Files in /biv
    backup.bin   00.08
    hlink.bin   00.14
    ooall.bin   00.07
    dosfix.bin   00.01
    ddump.bin   02.02
    make.bin   00.04
    restore.bin   00.05
    rfile.bin   00.07
    sfile.bin   00.07

CROMIX-PLUS-C-4

Date: November 30, 1987

Product: CRO-PLUS-C

Release: 9

Date production of this version began: Dec. 11, 1987

First serial number with this version: 9-10000

SUMMARY

Release 9 (version 164) of the Cromix-Plus Operating System for the XXU is now available. Version 164 is functionally identical to version 162 except that:

1. The 64FDX (16FDC, 64FDC) controller board is not required.

2. Installation can be done directly from SCSI tape (without floppy disk)

3. The /dev directory has been reorganized

4. Handling of /dev/console has been changed

5. Raw terminal drivers have been changed

023-9691
12-10-87
Rev. A

925 1578

USING CROMIX-PLUS WITHOUT THE 64FDX (FLOPPY DISKS)

If the system is running without the 64FDX board, it must contain:

- XDOS version 2.03 or higher
- An Octart with at least one terminal connected
- An ESDC board to support the SCSI tape drive
- A SCSI tape drive

Cromix-Plus version 164 may be distributed on floppy disks as in the past, or on SCSI tape. To be loaded from SCSI tape, a new XDOS ROM (release 2.03 or higher) must be inserted into the XXU.

The Cromix-Plus Release Tape

The Cromix-Plus SCSI release tape contains three tape files:

File 1 (Xfs format)	Contains a cromix.sys file that was specially generated to contain a RAM disk from which Cromix can run at a basic level.
File 2 (Ftar format)	Contains the entire Cromix distribution file system.
File 3 (Ftar format)	Contains the Kermit distribution.

Installation Procedure From SCSI Tape

Cromix release 164 requires at least one MByte of system memory. To use the SCSI tape to update the system, the system must contain at least two MBytes of memory.

Upon system reset, XDOS (2.03 or higher) will begin communicating with either a 64FDX terminal or an Octart terminal. The XDOS CON command can be used to define the default console. Either the 64FDX or any Octart channel can be designated. The baud rate may also be specified. If the setting is changed, reset the system to allow the new setting to become effective. If the CON setting specifies automatic boot, use the ESC key to prevent booting. To execute the CON command, at the XDOS prompt (;) type:

          ;con

Answer the prompts appropriately. Remember that decimal values must be terminated with ".".

Insert the SCSI tape into the drive and execute the XDOS command:

;lxsc0 (load and execute the first file from the SCSI tape on channel 0)

After the tape is read, Cromix-Plus will automatically log-in as system on the XDOS console. Whenever XDOS executes a program (or boots a device) it will pass its definition of the console to the new program (eg. cromix.sys). Cromix will continue to use the same terminal (and baud rate) for the raw terminal. If it is not running on a write protected device it will also create the file /dev/console as that device.

Perform any maintenance necessary as described in section 2.3 of the *68020 Cromix-Plus Administrator's Guide*.

Execute the **update1** command:

> system[1] **update1** <**hard disk partition**>

When rooted on the hard disk, execute the **update2** command (specifying the SCSI device). The **update2** command will reposition the tape to *tape file 2* and will execute the ftar command to update the rest of the system. The system will be automatically rebooted.

> system[2] **update2** <**tapes/stp1**> (Device may vary)

Proceed with the customization of the system. Note that the RAW entry in the **/gen/sysdef** file should NOT have an argument. Also note that /dev/console has been created automatically.

Note that the "console" entry in the /etc/ttys file is the only file enabled to run at level one. Do not disable it. The device /dev/console is always the existing console used by XDOS. This terminal should stay enabled even if the CON command is used to change the system console.

A terminal which is physically the same terminal as the "console" can safely be enabled. **Process_one** will be aware that the terminal is already enabled and will ignore the second entry.

Pay special attention to the downloading of Octarts. If XDOS uses an Octart channel for the console, the appropriate Octart firmware (/etc/oct.iop) will be downloaded by the RAW Octart driver. If the 64FDX console is used, Octart firmware will NOT be downloaded by the RAW driver.

The same Octart may be downloaded again with the same firmware by the /etc/iostartup.cmd command file if desired. This might be useful if the console is to be switched back and forth between the 64FDX and an Octart console. Note however, that downloading an Octart resets the baud rates on ALL its channels to 9600. If this is the correct setting, no harm is done. If a different baud rate is selected, the baud rate in the /etc/ttys file for the terminal named "console" must be changed from "n" to "a" (requires CR's to be pressed), or to a fixed value (inconvenient if the console is moved).

THE /DEV DIRECTORY

Another of the changes from version 162 to 164 is the reorganization of the /dev directory. In older versions, the /dev directory tended to grow quite large in one dimension. This has become inefficient and cumbersome.

Version 164 is distributed with only a few device files in the /dev directory. Many classes of devices are now organized into sub-directories:

/dev/disks	contains all disk devices
/dev/terminals	contains all terminal devices
/dev/printers	contains all printer devices
/dev/tapes	contains all tape devices
/dev/z80	contains all Z80 devices

After the **update1** and **update2** procedures are completed, the system administrator will most likely wish to make links from the device sub-directories to the /dev directory, possibly changing the name. For example:

system[1] **maklink /dev/printers/lpt1 /dev/prt**

The names of devices in the sub-directories should NOT be deleted or changed. Additional names can be added if required.

No changes to the /dev directory should be made until the **update1** and **update2** procedures are completed. This means that until updating is complete, all devices must be referenced using full names, for example:

system[1] **check disks/esd0**
system[2] **makfs disks/std2**
system[3] **update1 disks/esd2**
system[4] **update2 tapes/stp1**

After the update procedure is completed, make all the needed links. Do not make links for unused devices. Suit the linked names to taste, for example:

system[1] **maklink /dev/disks/esd0 /dev/disk1**
system[2] **maklink /dev/disks/esd1 /dev/disk2**
system[3] **maklink /dev/disks/std0 /dev/disk3**
system[4] **maklink /dev/disks/esd31 /dev/all_esd**

Similarly, for terminals:

system[1] **maklink /dev/terminals/otty1 /dev/tty1**
system[2] **maklink /dev/terminals/otty4 /dev/tty2**
system[3] **maklink /dev/terminals/mtty1 /dev/tty3**
system[4] **maklink /dev/terminals/tty1 /dev/tty4**

As distributed, the /dev/terminals sub-directory contains devices such as:

otty1   Octart terminals using the otty driver and the **oct.iop** firmware

qtty1   IOP/Quadart terminals using the qtty driver and the **quadart.iop** firmware

Note that the /etc/ttys file contains only the names tty1..tty4 in addition to the name "console". If different names are chosen for terminals, do not forget to include them in /etc/ttys file. If any lines are added or deleted from /etc/ttys, reboot the system before executing:

> system[1] **kill -1 1**

> OR

> system[1] **init <number>**

KERMIT

After updating and customization is complete, the Kermit software package can be installed from the SCSI tape. Execute the commands:

> system[1] **makd /usr/pkg**
> system[2] **makd /usr/pkg/kermit**
> system[3] **d /usr/pkg/kermit**
> system[4] **mode tapes/stp1 load**
> system[5] **mode tapes/stp1 file 3**
> system[6] **ftar -vx -b 400 /dev/tapes/stp1**
> system[7] **mode tapes/stp1 unload**
> system[8] **maklink -vf kermit.bin /usr/bin**

CURRENT DOCUMENTATION FOR THIS PRODUCT:

023-5012   CROMIX-PLUS INTRO REV. F
023-5013   CROMIX-PLUS USERS REV. F
023-5014   CROMIX-PLUS PROGRAMMERS REV. F
023-5044   68020 CROMIX-PLUS ADMINISTRATOR'S GUIDE REV. D
023-9238   KERMIT NOTICE REV. A
023-9691   CROMIX-PLUS-C-4 REV. A *

* Included in Version 164 SUDS package.

KERMIT NOTICE

A copy of the public domain communications software **Kermit** (both source and object) has been included with this version of Cromix-Plus. This software is provided by Cromemco at *no additional charge* as a service to our customers. The port of this software to Cromix-Plus was done by Adrian Pickering and Igor Uherkovich (see **read.me** for addresses).

The software is provided "as is" and includes the C language source code. Since this software is in the public domain, and was not developed by Cromemco, it is not possible for Cromemco to provide technical support for this product.

Documentation is provided in the file **read.me** and in various Help files in the **kermit** directory.

We hope that you will find this software useful.

INSTALLATION

To install **Kermit**, mount the floppy disk and execute the **install** command file.

```
system[1] mount fda /drive (or sfda)
system[2] d /drive
system[3] install
system[4] d /
system[5] unmount fda
```