digital INTER~~OFFICE~~ GALACTIC MEMORANDUM

TO: OS/8 V3 Test Sites

DATE: 12/31/73

FROM: S. Rabinowitz/R. Lary

DEPT: PDP-8 Development

EXT: 2130/3338 LOC: 12-3

SUBJ: Contents of V3 Field Test tapes

The tape marked "SYSTEM" is a bootstrappable TC08 system. The tape marked "BINARIES" contains TDINIT as its bootstrap. TDINIT has changed from V2 but its messages should be self-explanatory. It is suggested that the first thing you do after bootstrapping the system is get directories of both tapes (type

.DIR DTA0:,DTA1: followed by -L if you have a line printer)

and then get a listing of the file BUILD.HL which gives a command summary of the new BUILD (The new commands are enough of a superset of the old commands that you will be able to operate it in almost total ignorance of the new features - but read the file anyway.)

For those of you who do not have the OS/8 F4 system, the following patch to CCL.SV will invoke FORT.SV in the .COMPILE and .EXECUTE commands when the file extension is .FT :

```
.GET SYS CCL
.ODT
13414/6631 6625
13431/6555 6551
↑C
.SAVE SYS CCL
```

Please use the tapes as much as you can. We have been using them in-house for several weeks and they seem quite solid, but we do not exersise all of OS/8 in system development work. Please write us upon receipt of the tapes and give us a phone number so that we can call you semi-periodically to collect bugs and complaints.

We have purposely left in several minor, non-destructive bugs to test your perspicacity. You will be graded on how many you find. A score of 65% is passing, neatness counts, and cribbing will be severely dealt with.

digital

Dear Field Test Site:

Enclosed is a copy of the new OS/8 version 3 (pre-release) operating system. This software is the result of a continuing development effort which we hope will culminate in the release of OS/8 V3 early next spring.

We are distributing preliminary system tapes to selected, experienced users such as yourselves. We encourage you to experiment with the new software, evaluate its performance in your data processing environment, and notify us of any shortcomings or areas for improvement that may occur to you. In this manner, your expertise will benefit all OS/8 users and assist us in the final stages of development and evaluation.

We will forward complete documentation for this operating system as soon as it becomes available. Meanwhile, you will find that although OS/8 version 3 includes many new features, it is essentially a proper superset of the version 2 system. All operations which execute under OS/8 version 2 should execute under the enclosed software. Many of the new features are fairly obvious, and may prove useful to you even in the absence of full documentation. If unpredictable behavior should occur, kindly contact us. As always, sufficient information to duplicate the problem is very helpful and greatly appreciated.

We expect to send you some preliminary documentation within the week. Temporarily, you can list the .HL files which gives summaries of most of the features.

When OS/8 version 3 is released to the field, your suggestions will be incorporated into the software along with modifications proposed at other test sites. Thus, you may expect the final system to differ from the enclosed version in many respects. Subsequent maintenance of the new software will be greatly facilitated if you could keep very close track of these tapes, or any copies you may care to make. This will prevent other users who may have access to your system from confusing the pre-release software with the final, supported version available later.

DIGITAL EQUIPMENT CORPORATION, 146 MAIN STREET, MAYNARD, MASSACHUSETTS 01754
(617)897-5111 TWX 710 347-0212 TELEX: 94 8457

1/8/74

Intergalactic Branch

Dear field test site:

Enclosed you will find some documentation concerning new features of OS/8 version 3. Please keep in mind that this is temporary documentation originally meant for in-house use only, and although we believe it is all up-to-date, it is not in the same form as documentation which will eventually be published in the new OS/8 handbook. Unfortunately, this handbook will not be ready for several months, so the enclosed memos will have to do.

The final system is scheduled to be submitted to the library in about two weeks, so it is imperative that if you find any bugs, they should be reported to us immediately. You can send bugs to us in writing, or for faster service, try to phone us. Bugs should be reported to the following people. (Try to get through to the highest person on the list, if he's accessible:)

Shawn Spilman	X2073
Stanley Rabinowitz	X2130
Richard Lary	X3338
Dennis Pavlock	X3871
Herb Jacobs	X2130
Mary Tamir	X4068
Marty Hurley	X4058

After hours, ask for any of the above at X2876

Once V3 is formally released by Digital through the Software Distribution Center, bugs should no longer be sent to any of these people, but should be submitted via the usual SPR service. At that time, the above list of phone numbers should be destroyed (fire is best).

Please be sure to note the version number of the monitor, CCL, and the CUSP when reporting bugs. Errors in the documentation will also be appreciated.

We hope that you will find time to pound on all features of the new system, especially those obscure little-used options. Some bugs have already been discovered in-house and patches produced. Many of these got into the field test tapes which were mailed out. These are enclosed as well as some more recent patches. Be sure to put these patches into your CUSPs if they are not already there.

Sincerely yours

Stanley Rabinowitz
official scavenger, P?S

DIGITAL EQUIPMENT CORPORATION 146 MAIN STREET, MAYNARD, MASSACHUSETTS 01754
(617)897-5111 TWX 710 347-0212 TELEX 94-8457

(these are probably in field test save images)

digital

Field Release Patches

Updates monitor V3E to V3F. Fixes bug re monitor error messages:

This patch should be made (using EPIC or otherwise) to absolute record 56 on the system device and/or relative record 5Ø of any system head file:

rel loc 53/7Ø1Ø←711Ø

Change to absolute block 7 on system device, or relative block 1 of system head file:

rel loc 31/3Ø5←3Ø6

Patch to update CCL edit N to become edit O. Allows altmode to be passed to SRCCOM, fixes bug re processor switches, fixes special mode spooling, and allows CCL switch after an = option:

```
1262Ø/4ØØØ←4ØØ1
1352Ø/5676←5751
13551/?←4276
14431/1422←5343
14543/?←1267;764Ø;5233;1422;5232
17446/3433←7ØØØ
13722/5272←5335
13735/?←4737
13736/?←5272
13737/?←4275
Ø6537/1651←1751
```

Patch to CCL edit O, updating it to edit P. Allows recursive U cmds.

```
12Ø27/62Ø1←5332
12132/?←724Ø;3736;62Ø1;523Ø;2422
6537/1751←2Ø51
```

Patch to CREF release version 9, updating it to V9A:

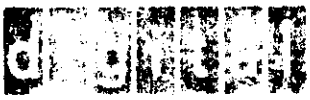
```
2477/53Ø3←72ØØ
24Ø6/114Ø←1364          Fixes close bug re CREFLS.TM .
2564/?←3Ø1
```

Patch to BUILD V3, updating it to V3A. Allows alter to work, fixes bug re P command.

```
213/6Ø32←62Ø1
2231/5221←5365
2365/?←222Ø;5221
737/4ØØØ←1ØØ
```

if you can't read this line, here it is again:
2365/?←222Ø;5221

DIGITAL EQUIPMENT CORPORATION, 146 MAIN STREET, MAYNARD, MASSACHUSETTS 01754
(617)897-5111 TWX: 710 347-0212 TELEX: 94-8457



OS/8 Field Release Patches

these are probably not in the distributed save images

Optional patch to CCL (any edit) to allow it to use FORT and LOADER instead of F4 and LOAD. (The way to change these as described in the memos doesn't work yet)

13414/6631←6625
13431/6555←6551

Patch to update BUILD V3A to V3B. Allows BUILD to work on 8K machines:

352/7ØØØ←74ØØ
737/1ØØ←2ØØ

Patch to BCOMP to allow CCL .EXECUTE command to work with BASIC:

7Ø17/3Ø46←5126
7126/?←745Ø;5732;3Ø46;522Ø;46Ø5 (version # if any not updated)

Patch to CCL edit P updating it to edit Q. Allows .EXEC to chain to BCOMP correctly for BASIC files:

6571/Ø2Ø1←Ø2Ø3
6572/2311←1715
6573/Ø3ØØ←2ØØØ
6537/2Ø51←2151

Patch to OS/8 V3F updating it to V3G. Disables waiting for TSF on .run under BATCH:

Fixes to record 11 on system device:

rel loc 32/5Ø35←53Ø3
rel loc 33/5Ø46←53ØØ
rel loc 1ØØ/?←37Ø2;5Ø46;1722;37Ø2;5Ø35

Fixes to record 7 on SYS:

rel loc 31/3Ø6←3Ø7

Patch to BATCH updating V4 to V4A. Eliminates strango bug:

63Ø6/765Ø←535Ø
635Ø/?←765Ø;5764;5312
17Ø1/37ØØ←Ø1ØØ

OS/8 Field release problems

Some systems are being built wrong and incorrectly getting a bad software core size. If the core command claims you have more core than you really have, or if you have 8K and FOTP doesn't work, then you want this fix:

Examine relative location 177 in record Ø on SYS:. If it is not Ø, then change it to Ø. Be sure to check this location after you run BUILD since we haven't found out yet what's making it non-Ø, but it's most likely BUILD.

If the core image of CCL on your tape is not edit P, then you should reassemble the source (CCL.PA) to create edit N. viz:

```
.r PAL8
*CCL←CCL
.R ABSLDR
*CCL$
.SAVE SYS CCL;12ØØ1=2ØØ3
.R CCL
```

Then apply all the patches to update to edit Q.

---

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: August 15, 1973

FROM: S. Rabinowitz

DEPT: Small Systems

EXT : 2130

SUBJ: OS/8 BOOT

BOOT is used to make it convenient to bootstrap from one system to another and to bootstrap from one device to another by typing commands on the keyboard.

BOOT can run conveniently from OS/8, and COS and can also run from any other monitor system (e.g., CAPS-8).

To run BOOT from COS, see DEC-08-OCOSA-D-D, chapter 9.

To run BOOT from any version of OS/8, type

.R BOOT/DV

or

.RUN DEV:BOOT/DV

where DV is a two character mnemonic which must immediately follow a slash.

To run BOOT from a V3 OS/8 device with CCL enabled, type

.BOOT/DV

In this case, BOOT.SV must be on the system device.

You may also type

.R BOOT

and the system responds with a slash, at which time you respond with the DV mnemonic.

If an illegal mnemonic is typed, the system types "NO" and prints a slash to let you try again. In this case, you can type rubout to erase a line and try again.

If you type a legal mnemonic but your configuration doesn't include the corresponding device (or it's not ready), the boot strap may hang.

The legal mnemonics are as follows:

Mnemonic	Device	System or Comments
CA	TA8E cassette	CAPS-8
DK	Any disk (RF08, DF32, RK8E, RK8)	OS/8, COS-300
DL	LINCtape	DIAL-V2, DIAL-MS
DM	RF08 or DF32	Disk Monitor
DT	Any tape (TC08, TD8E, LINCtape)	OS/8 COS-300
LT	LINCtape	OS/8, COS-300
PT	PT8E Paper tape	Loads BINLDR into field 0
RE	RK8E disk	OS/8, COS-300
RF	RF08, DF32 disks	OS/8, COS-300
RK	RK8 disk	OS/8, COS-300
TD	TD8E DECtape	OS/8, COS-300
TY	TC08 DECtape unit 4	Typeset Bootstrap
VE		types BOOT's version number
TC	TC08 DECtape	OS/8, COS-300, Disk monitor, DEC library system, and others
ZE		Zeroes core (field 0)

If the device mnemonic is followed by a period, the program will load the correct bootstrap into core and then halt. Hitting continue branches to the bootstrap.

be

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: August 22, 1973

FROM: S. Rabinowitz

DEPT: Small Systems

EXT : 2130

SUBJ: V3 CREF

The following new features have been added to V3 CREF by M. Hurley:

1. /M invokes Herb's kludge which causes CREF to chain to itself after crefing all symbols up to LG. This causes CREF to run much slower but allows twice as many symbols.

   Note: M stands for 'mammoth'.

2. /R has been changed to /Q. That is, /Q means the input is from SABR.

3. /R now means the input is a RALF file.

4. /U means no listing and no symbol table.

5. CREF now prints its version number at the very end of the CREF listing.

In addition, several bugs were fixed.

If you know of any more bugs in CREF, please report them to me or Marty.

fp

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: August 28, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT : 2130

SUBJ: V3 CREF (Continued)

6. /E means don't delete CREFLS.TM, otherwise CREFLS.TM is automatically deleted after completion.

7. The output file now has .LS as the default extension.

fp

080-100-007-02

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3

DATE: November 27, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT : 2130 LOC: 12-3

SUBJ: OS/8 CREF

Against her will, we coerced Marty to add the following changes to CREF:

With /M, there is now only one listing file produced. Forget about stuff about CREFM2.LS.

CREF output on non-file-structured devices now starts at block 0.

Note: With /M, CREF must be called SYS:CREF.SV.

CREF now uses software core-size.

080-100-007-03

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: December 24, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: More on CREF

This memo obsoletes the previous memo, 080-100-007-02 but does not obsolete the two before that one.

/9 is used internally by CREF and should not be typed by a user.

/M causes entire CREF to go to the output file.
Forget anything that was ever said about CREFM2.TM .

Note that CREF V9 has a bug in that if it is chained to by something other than PAL8, it may produce a CLOSE ERROR when trying to delete a non-existent CREFLS.TM.
To get around this, merely specify the /E switch.

080-100-008-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: August 22, 1973

FROM: S. Rabinowitz

DEPT: Small Systems

EXT : 2130

SUBJ: V3 TECO

The following new features have been added to TECO by R. Lary:

1. The command FS is now synonymous with the ↑R command.

2. The command FN has been added. It is the same as the FN command on the PDP-10. FN is similar to FS except that it performs a non-stop search; i.e., it performs a P whenever the search fails on the current page.

S:N::FS:FN

3. If the character ↑S is typed immediately as the very first character of a new command line, then the entire previous command line (even if it was in error) is saved as a text string in Q-register Z.

4. ↑U typed at any time deletes the current line of the command string being typed. The TECO command by the same name is still available in the two-character ↑U form.

5. Internal changes have been made to facilitate chaining to TECO. Upon being chained to, TECO expects a legal command to be in locations 176ØØ-17645 inclusive (padded with altmodes or nulls if necessary); one 7-bit ASCII character per location. It first executes this as a command string.

Features I would still like to see added:

A. An alternate starting address for the SUPER-TECO patch; i.e., the one that causes TECO to ignore ↑Z's.

B. Some form of the EG command.

fp

080-100-008-01

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: October 26, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: Final Fixes to TECO

The final version of TECO, for OS/8 release 3 will contain the following features in addition to those given in PDM #080-100-008-00:

6) The EG command (exit and go) causes TECO to perform an EC followed by a return to the OS/8 keyboard monitor with a simulated

```
.EX
```

CCL command. This CCL command will execute whether or not CCL is enabled.

7) The ER command now allows a device specification with no filename even if the device is file-structured. For example:

```
ERDTAØ:$
```

It will treat the device as if it were non-file structured.

8) If TECO is chained to, and if a ?16 error occurs before TECO prints its asterisk, then TECO returns control to the OS/8 keyboard monitor.

fp

080-100-008-02

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: December 24, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: Correction about EG

This paragraph revises item 6 of the previous edition of this memorandum. All other items remain current.

6) The EG command (exit and go) causes TECO to perform an EC followed by a return to the OS/8 keyboard monitor at which time CCL is automatically invoked and CCL then executes whichever one of the following four commands had been executed most recently:

a) .EXECUTE

b) .LOAD

c) .COMPILE

d) .PAL

080-100-009-00

INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: August 28, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT : 2130

SUBJ: OS/8 Standard Extensions

Many OS/8 programs (both system programs and commonly used user programs) which read or create files have default (or assumed) extensions associated with the file. These should be documented and put in the next user's manual just for standardization reasons. I am not familiar with all such extensions; if you know of any others please add them to this list.

Extension	Mneumonic	Use	Used by (as default input)	Used by (as default output)
BK	Backup	Backup ASCII file		TECO
BA	BASIC	Basic source	BASIC	
BN	Binary	PAL8 binary file	ABSLDR BUILD BITMAP	PAL8
DA	Data	Data file		
DI	Directory	Directory listing		
DC	Document.	Documentation file		
FT	Fortran	Fortran source file	FORT	
HL	Help	Help file	HELP	
LS	Listing	Listing file		FORT PAL8 ....
BI	Batch Input	Batch input file	BATCH	
PA	PAL	PAL8 source file	PAL8	
RL	Relocat-able	Binary relocatable file	LOAD	SABR
SV	Save	Core image Save file		SAVE
TE	Teco	Teco Macro file	MUNG	
TM	Temporary	Temporary file	CREF	PAL8
TX	Text	Text file		
CM	Cassette Master	Cassette Master file	LIBCOP	CREATE

MA	Macro.	Macro Source
RB	Relocatable Binary	RB Source
AS	ASCII Source	Dibol Source
DB	Dibol Binary	Dibol Binary
DF	FGBG Binary	FGBG Binary
DM	Multi terminal	DIBOL Binary
SY	System	System Head

fp

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: August 31, 1973

FROM: S. Rabinowitz

DEPT: Small Systems

EXT: 2130 LOC: 12-3

SUBJ: LPTSPL

Warren Brackesbusch has written a program called LPTSPL which is an OS/8 utility which prints source files on the line printer preceded by large block letters giving the name of the file. It calls the command decoder in special mode so you can use *'s and ?'s in your file specifications. CCL chains to this program with the print command. Thus

```
.PRINT  *.LS
```

prints all your listing files on LPT: with headers.

I am including this program on the in-house OS/8 V3 distribution tape because I believe some people in-house may find it useful. I'm not sure whether we want to distribute it on a non-supported basis or just put it in DECUS.

Options:

/N	Narrow	Assumes LPT has narrow (80 column) paper
/B	No Block Letters	Headers but no block letters
/H	No Headers	Block letters but no headers

fp

080-100-011-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: September 4, 1973

FROM: S. Rabinowitz

DEPT: Small Systems

EXT: 2130 LOC: 12-3

SUBJ: PIPlØ

Although PIPlØ will probably go into DECUS instead of on the OS/8 system tape, I am including it on the in-house distribution tape because of its usefulness.

The new version [by R. Lary] has all known bugs fixed and works on TD8E controllers as well as TCØ8's.

Whenever a DECtape drive is specified, the program examines the tape mounted to determine if it is a PDP-8 or PDP-1Ø DECtape and handles the files accordingly.

Switches:

/L	List directory (only valid if PDP-1Ø input)
/F	Fast directory (only valid if PDP-1Ø input)
/Z	Zero directory before transfer (only valid if PDP-1Ø output)
/D	Delete old output file before transfer
/B	Binary mode transfer (8 bits per 36 bits)
/I	Image mode transfer (3 12-bits per 36 bits)
/P	Preserve Line numbers (default is to delete them)

fp

080-100-012-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: September 5, 1973

FROM: S. Rabinowitz

DEPT: Small Systems

EXT: 2130 LOC: 12-3

SUBJ: V3 PIP

I have made the following changes to V3 PIP:

1) Fixed problem with one-page write.

2) In /S or /Z mode, the = option is now taken modulo 100 (octal). This prevents bombs when = nnnn gets too large.

   Any number other than Ø which is congruent to Ø modulo 1ØØ means write no additional information words in the directory. Although we don't recommend this (since it gets rid of dates) it will allow users with lots of files to get more of 'em on their device.

3) The message 'FREE BLOCKS' now reads 'Ø FREE BLOCKS' when the device is full.

4) Devices may now be filled up to their capacity. Previously one block was always wasted.

5) A parity ↑C is now accepted.

6) ↑C doesn't write on system device.

7) Ending a command with altmode returns to OS/8 after PIP operation completed.

8) PIP no longer halts on /L if TTY: is not found. It now acts as a NOP.

I regret to report that I couldn't get the dates to line up.

Future fixes:

1) The /O option (meaning OK) when used on /S or /Z command will prevent the message ARE YOU SURE? and ZERO SYS? from coming out.

fp

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: October 8, 1973

FROM: S. Rabinowitz

DEPT: Small Systems

EXT: 2130 LOC: 12-3

SUBJ: Final Fixes to PIP

This is a continuation of memo 080-100-012-00.

9. The /O option (standing for OK) is now in. When used in conjunction with /S or /Z, it will prevent the message ARE YOU SURE? and ZERO SYS? from coming out. It assumes the user really wants this command and prints YES on the teletype to remind him that he's performing a potentially dangerous operation. This message prints even if BATCH is running.

10. The first time PIP calls the command decoder, if you include the /V option in your command line, PIP will print its version number on the teletype. The current version number is V1.

11. If an = option is included with an image mode transfer, then the low order 12-bits of this = option specifies the desired length to close the output file with. The output file is given this length except in the following two cases:

    a) More data was written than this size. In this case, the output file is given its correct size.

    b) The size specified is greater than the empty space available. In this case, the data is transferred but the file is not closed. Instead the error message:

    MONITOR ERROR 4

    is printed and control returns to the keyboard monitor. Data in the file following the EMPTY is <u>not</u> destroyed.

12. Using the /I option, the output file is always opened even if there were no input files. This feature, combined with the previous feature, allows you to create a named file out of an empty.

Application: Suppose you accidentally deleted your 23 block file IMPORT.PA. You can recover it with the command:

```
.R PIP
*IMPORT.PA[23]</I = 27$
```

Note that $23_{10} = 27_8$

fp

080-100-012-02

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3

DATE: November 27, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT : 2130 LOC: 12-3

SUBJ: PIP Finally

Last minute fixes:

If /O is used, 'YES' does not print.

↑C now goes to 7600 not 7605. This fixes bug re ↑C with NO INTERRUPTIONS.

If /Z and /I are used together, any = option specified applies only to /Z.

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: August 30, 1973

FROM: S. Rabinowitz

DEPT: Small Systems

EXT: 2130

SUBJ: CCL

CCL stands for concise command language. The OS/8 version of CCL has been patterned after the PDP-1Ø's. In many cases, CCL only makes it easier for the operator to enter certain commands, e.g., the command

```
.R DIRECT
*LPT:<DTA3:
```

can be effected by the CCL command

```
.DIR DTA3:/L
```

However in some cases, CCL commands perform operations which could not be performed otherwise, e.g.,

```
.CORE
```

Theory of operation:

When a command is typed to the OS/8 monitor, if it doesn't recognize the command keyword and if CCL is disabled, then the OS/8 monitor echoes the command (up to 6 characters) followed by a question mark. If, however, CCL is enabled, and an unknown command is typed, the OS/8 monitor loads in the CCL overlay (from block 67 on SYS: into core locations 4ØØ-777) and branches to it (at location 6ØØ). This overlay then looks up the command in the CCL command table (locations 4ØØ-577). If the command is not found, the monitor is rebootstrapped (after echoing a ? following the illegal command). The user's core image remains intact. If the CCL overlay determines that the command keyword is a legal one, it chains to the file SYS:CCL.SV for further processing. This file should be the first file on the system device (for DECtape systems) for maximal efficiency. Depending on the command, this program may then chain to other files on the system device. While this procedure may seem complicated, it is almost unnoticeable when running off of a disk. Although it could run off DECtape, it is not advisable because of the extra overhead involved. For this reason, you can enable or disable the CCL feature.

Enabling CCL: CCL comes disabled on the system to be distributed from the library. It will be disabled when you build the system from paper tape. To enable CCL, you type

```
.R CCL
```

Disabling CCL: To disable CCL, you type

```
.CCL
```

If CCL was already disabled, the system will reply with

```
CCL?
```

The OS/8 monitor still detects and responds to the usual keyboard monitor commands:

```
.ASSIGN
.DATE
.DEASSIGN
.GET
.ODT
.R
.RUN
.SAVE
.START
```

Only the first two characters of each command are significant and other letters or digits may be added. The command keyword is terminated by a space (or other delimeter).

If CCL is enabled, two exceptions are made to the above statement:

1) The DEASSIGN statement may not start with the letters DEL since this is taken as the DELETE statement.

2) If the DATE command is immediately followed by a carriage return, different action is performed than the usual OS/8 DATE command which is followed by an argument.

CCL Commands

CCL commands consist of letters only. The full command keyword need not be typed (for example DEL is a legal abbreviation for DELETE). For each command, certain characters are required. The remaining characters

are optional in the sense that if additional letters are typed, these must be them. Thus, DELEXE is not a legal command. The command keyword is terminated by a non-letter. Then, succeeding letters and digits are scanned past until the first non-alphanumeric character. If this is a space, multiple spaces are then scanned over. We have now reached what is called the argument portion of the command. Not all commands require arguments. For example,

```
.DELETEABC      FOO
```

is a legal command and FOO is the argument.

The current CCL commands are given below in alphabetical order. The required part of the command is underlined.

BOOT
CCL
COMPILE
COPY
CORE
CREATE
CREF
DATE
DELETE
DEA
DIRECT
EDIT
EXECUTE
HELP
LIST
LOAD
MAKE
MAP
MUNG
PAL
PRINT
PUNCH
RENAME
RES
SQUISH

SUBMIT

TECO

TYPE

VER

ZERO

The exact number of characters which are significant for a given command may change in future implementations.
These commands are explained below. Some of them may not be implemented yet in the CCL version 3G.

Feasibility of Adding New CCL Commands

It is very important that it be easy for the user to add his favorite command to the system. This has been kept in mind when designing the system. The commands are table driven and easy to add to. A separate document will be produced explaining the internal workings of CCL and telling how to add new commands. One of the strong selling points of CCL is that new commands are easily implemented. For this reason it is crucial that the source and/or listing of CCL.PA be made available to the user of OS/8 as cheaply as possible.

There will be room in the CCL command keyword table for the user to add at least ten additional 3-letter commands.

Of course, if the user does add his own CCL commands, we will no longer support his system.

CCL and the Command Decoder

Many CCL commands allow file specifications on the same line as the command keyword. They then chain to a cusp which expects the command decoder tables to be already set up. This is accomplished by having CCL.SV contain its own command decoder. This command decoder is completely compatible with the usual OS/8 command decoder but is not as limited in space restrictions. For this reason, CCL.SV's command decoder has extensions and capabilities far beyond those of mortal men.

One such extension is this: An alternative method of specifying an 8-character filename (6 chars of name and 2 of extension separated by a dot) is by a 16 digit sequence of octal digits

preceded by a number sign (#) which represents the internal packed sixbit representation for the filename. For example, the specification ABCD.EF could be replaced by #Ø1Ø2Ø3Ø4ØØØØØ5Ø6. To use such a notation, all 16 digits must be given. This notation is somewhat compatible to the PDP-1Ø's. This notation is useful for referencing files whose names are not easily accepted by the usual OS/8 command decoder, for example, a filename which contains an embedded space.

Explanation of CCL Commands

BOOT This command chains to SYS: BOOT.SV. Any option which may appear on the .R BOOT command may occur on the .BOOT command. For example

```
.BOOT/RF
```

bootstraps onto your RFØ8 disk.

CCL If CCL is enabled, typing this command disables CCL on the monitor residing on the system device. To re-enable CCL, type .R CCL.

COMPILE file specification

This command chains to one of the OS/8 compilers (or assemblers) depending on the extension of the first input file.

Extension	Compiler
.PA	PAL8
.SB	SABR
.FT	F4 if present, otherwise FORT
.MA	MACRO
.RA	RALF
.BA	BASIC
.LI	LISP
.SN	SNOBOL

If no extension is explicitly given, each is tried in order until one is found and the appropriate compiler is invoked.

If no argument is given to the COMPILE statement, the system remembers the previous argument used either in the last COMPILE, LOAD, or EXECUTE command.

If no output filename is specified, the name of the first input file is substituted.

COPY file-specification

This command chains to SYS: FOTP.SV including the /L option among any specified by the operator. Before chaining, it prints the message

```
FILES COPIED:
```

To understand this command fully, the reader should consult the FOTP user's document. For example, to copy all save files from DSK: to DTA3: you type

```
.COPY  DTA3:< * .SV
```

CORE This command types out on the Teletype® how much core is on the computer upon which the system is being run.

CREATE file-specification

This command chains to SYS: EDIT.SV. The file specification must consist of a single file only. No back arrow may be present. The file is assumed to be an output file. For example

```
.CREATE   FOO.FT    is equivalent to

.R EDIT
*FOO.FT<
```

If no argument is given, the argument used in the last CREATE or EDIT statement is remembered.

CREF file-specification

This command chains to PAL8 Including the /C option which will cause it to chain to CREF.SV. Of course, if no listing file is specified, the listing will be sent to LPT: Example:

(R) Teletype is a registered trademark of Teletype Corp.

```
CREF  BARF
```

produces a CREF listing of the file BARF.PA on the line printer.

DATE If an argument is given, this command is treated as the standard keyboard monitor DATE command. If no argument is given, this command types the current date (including day of week) on the Teletype or types NONE if none were specified. If the file SYS:DATE.SV exists, it is chained to. This could allow implementing messages of the day.

DELETE file-specification

This command chains to SYS: FOTP.SV including the /L and /D options. First it prints FILES DELETED: .

DEA This is exactly the same as OS/8's DEASSIGN command. If CCL is enabled, CCL performs this function instead of the monitor.

DIRECT file-specification

This command chains to SYS: DIRECT.SV. If no output file or device is specified, CCL looks at the options specified. If /L is specified, CCL forces LPT: as the output device. If /S is specified, CCL forces TV: as the output device. /L overrides /S.

Example: .DIR FOO.*/L

is equivalent to

```
.R DIRECT
*LPT:<FOO.*$
```

EDIT file-specification

This command simply chains to SYS: EDIT.SV. If no argument is given, CCL remembers the argument used on the last CREATE or EDIT command.

EXECUTE file-specification

This command is similar to the COMPILE command with the addition that options are passed to the chained program causing the binary produced to be loaded and executed. The source is always recompiled.

HELP file-specification

This command causes CCL to chain to SYS: PIP.SV. The extension .HL is used as the default input extension. If no output device is specified, TTY: is forced. If no input file is specified, CCL.HL is assumed. This command may be used to type a specified help file on the Teletype.

LIST file-specification

This command chains to SYS: FOTP.SV. If no output device is specified, LPT: is forced.

LOAD file-specification

This command chains to the appropriate loader depending on the extension of the first input file.

Extension	Loader
.BN	ABSLDR
.RL	LOADER
.RB	?

If no extension is given, a search is made. If the option /G is specified, execution will start after loading.

If no argument is given, CCL remembers the argument on the last COMPILE or EXECUTE command.

MAKE file-specification

A single file must be specified. This command chains to SYS: TECO.SV and opens the specified file for output. Example:

```
.MAKE  DTA1:FOO.TX
```

is equivalent to

```
.R TECO
*EWDTA1:FOO.TX$$
```

MAP file-specification

This command chains to SYS: BITMAP.SV. If no output device is specified, LPT: is assumed. Example:

```
.MAP  FOO,BARF
```

is equivalent to

```
.R BITMAP
*LPT:<FOO.BN,BARF.BN$
```

MUNG filename, text

This is the most general and fantastic command in the entire repertoire of all of DEC's software. It chains to SYS: TECO.SV which then reads the first page of the file specified into Q-register Y. The contents of this file are assumed to be a TECO macro. If no extension is specified, then .TE is assumed. If no extension is desired, type the dot after the name with no extension. Then all the text between the comma and the end of the line are entered into the TECO text buffer. This text is presumed to be an argument to the Macro. If no text is desired, no comma is necessary. With the text pointer at the end of the buffer, the macro in Q-register Y is then executed. The text may specify source files to be munged among other things. MUNG is a recursive acronym for Mung until no good. Example:

```
.MUNG SYS:CLEAN,FOO,3
```

is equivalent to

```
.R TECO
*ERSYS:CLEAN.TE$YHXYHKIFOO,3$MY$$
```

There is a restriction on the length of the text argument. If it is too long, CCL will print an error message.

PAL file-specification

This chains to SYS.PAL8.SV. This command is similar to COMPILE except PAL8 is always chained to.

PRINT file-specification

This command chains to SYS: LPTSPL.SV.

PUNCH file-specification

This command chains to SYS: PIP.SV. If no output device is specified, PTP: is forced.

RENAME file-specification

Not yet implemented. It should chain to FOTP.SV with the /R option.

RES This command chains to SYS: RESORC.SV. The options /L and /S force LPT: and TV: as default output devices.

SQUISH file-specification

This command chains to SYS: PIP.SV including the /S option. If no output device is specified, the output device is forced equal to the input device.

Example 1: .SQUISH DTA1:<DTAØ:

Example 2: .SQUISH SYS:

is equivalent to .R PIP
*SYS:<SYS:/S$

SUBMIT file-specification

This command chains to SYS:BATCH.SV.

TECO file-specification

This command chains to SYS: TECO.SV. If no argument is specified, the argument used on the last TECO or MAKE command is remembered. If an argument is given, one input file must be specified and at most one output file. TECO then opens the input file for reading and creates the output file. If no output file is specified, then TECO does an edit backup on the specified file.

Examples:

.TECO FILE.BA

is equivalent to

.R TECO
*EBFILE.BA$Y$$ and

```
.TECO  FOO2.PA<LTA2:FOO.PA
```

is equivalent to

```
.R TECO
*EWFOO2.PA$ERLTA2:FOO.PA$Y$$
```

The first page of the input file is read into the text buffer before control is returned to the user.

TYPE file-specification

This command chains to SYS: FOTP.SV. If no output device is specified, TTY: is forced.

VER This command prints the version number of both the OS/8 monitor and of CCL.

ZERO file-specification

This command chains to SYS: PIP.SV including the /Z option. Precisely one device must be specified and no backarrow may be present. The device is forced to be an output device and is zeroed. For example:

```
.ZERO  DTA7:
```

is equivalent to

```
.R PIP
*DTA7:<$
```

Additional Comments

1. CCL passes the altmode bit to most programs (mainly PIP and FOTP) so that after the command is executed, control returns to OS/8 rather than the program chained to.

2. Arguments remembered by CCL will be forgotten when the date is changed. These remembrances are saved in monitor block 65. Commands which require remembered arguments will produce SYNTAX ERROR messages if no argument was ever remembered.

3. The .R CCL command recreates the CCL overlay block in the monitor. The command keyword table includes the CCL edit level which must be changed whenever the table changes. Much version number checking is performed by CCL to prevent accidents from occurring when old versions and new versions of CCL and the monitor are accidentally mixed.

ERROR Messages:

CCL 3x & MONITOR INCOMPATIBLE

To remedy, try typing .R CCL

I/O ERROR ON SYS:

TOO MANY FILES

ILLEGAL SYNTAX

file NOT FOUND

device DOES NOT EXIST

COMMAND NOT YET IMPLEMENTED

BAD MONITOR

The monitor predates the version of CCL in use.

NO CCL!

CCL.SV not found on SYS:

% CAN'T REMEMBER

This is a warning message only which means the argument was too long to remember or an I/O error occurred.

fp

S. Rabinowitz

Ø8Ø-1ØØ-Ø13-Ø1

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: 1/2/74

FROM: S. Rabinowitz

DEPT: Small Systems

EXT: 2130 LOC: 12-3

SUBJ: CONCISE CHANGES TO CCL

This memo includes additions and corrections to the previous memo.

Indirect Command Files

A construct of the form

ⓐDEV:FILE.EX

may appear anywhere within the argument portion of a CCL command. If DEV: is omitted, DSK: is assumed. If .EX is omitted, .CM is assumed.

What this does is replace the construct in the command line by the contents of the specified file. Carriage returns and line feeds in the file are ignored; but nulls are not (they signify end-of-line). For example, if the file FOO.CM contains the characters F.PA then the command

.EXEC BARⓐFOO,FILE.PA

is equivalent to

.EXEC BARF.PA,FILE.PA

Command files may not exceed 1 block in length.

The message

COMMAND LINE OVERFLOW

is given if a command line grows to be more than 512 characters long.

If in any CCL command, a filename is given with no extension but with a dot, then no default extensions are tried.

Thus

.PAL FILE.

will only assemble the file called FILE but the command

.PAL FILE

might instead assemble FILE.PA if it is found.

COMPILE

Delete the extensions .MA, .LI, .SN from old list.

The CCL switch -LS is used to generate a listing. The listing file goes to SYS: if no device was specified. Its filename is the same as the name of the file on which the -LS is given but with a .LS extension.

This holds also for the EXEC, and PAL commands.

If no binary is wanted, type -NB at the end of the command line.

The binary file will be given the same name as the first source file unless the CCL switch -NB is given in which case, the name is taken from the first filename after the -NB switch. The -NB switch means don't create a binary file yet.

The -MP switch is used to get a map.

The .FT extension goes to F4 unless FORT was present on SYS: when CCL was enabled in which case, .FT goes to FORT. Same for LOAD and LOADER.

If you must use an unusual extension, you can type a processor switch as follows:

.EX FILE.Ø6 -FT

means use the FORTRAN compiler on the file FILE.Ø6.

CCL switches are of the form hyphen followed by 1 or 2 letters or digits. In addition to the ones above we have

-L	Send output to LPT:
-T	Send output to TTY:
-P	Send output to PTR:
-S	Send output to TV:

These all force the stated device to be the first output device, except for commands such as -EXEC and .COMP which force the device to be the second output device.

RENAME is implemented. So are new commands

.SKIP
.BACKSPACE
.REWIND
.UNLOAD
.EOF

which chain to SYS:CAMP.SV

There are 3 commands, UA, UB, UC.

If typed with an argument, they merely remember the argument. If typed wihout an argument, they recall the last argument and execute it as a CCL command.

jg

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: September 12, 1973

FROM: S. Rabinowitz

DEPT: Small Systems

EXT: 2130 LOC: 12-3

SUBJ: OS/8 V3 Memos (Ø8Ø-1ØØ- series)

I am writing this memo to try out my new number 2 pencil. The excuse I use is that people who joined the V3 list late may refer to this memo and find out what useful memos they may be missing. This list includes all retrievable memos regarding OS/8 V3 as of the date at the top of this memo. They may be retrieved through the usual channels. Memos prior to -ØØ6 in the Ø8Ø-1ØØ series relate to PS/8 and are obsolete.

-ØØ6-ØØ	OS/8 BOOT
-ØØ7-ØØ	V3 CREF
-Ø1	"
-ØØ8-ØØ	V3 TECO
-ØØ9-ØØ	OS/8 Standard Extensions
-Ø1Ø-ØØ	LPTSPL
-Ø11-ØØ	PIP1Ø
-Ø12-ØØ	V3 PIP
-Ø13-ØØ	CCL
-Ø14-ØØ	OS/8 V3 Memos

fp

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: October 26, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: OS/8 V3 Memos (080-100- series)

My new number 2 pencil (first tested 9/12/73) has now undergone a month of exhaustive field testing*. Although preliminary, these tests seem to yield satisfactory results. Its use in this memo marks a major milestone in the testing procedure.

As you will recall from the previous memo, only retrievable OS/8 V3 memos are listed here-in and memos prior to -006 in the 080-100- series relate to PS/8 and are obsolete.

-006-00	OS/8 BOOT
-007-00	V3 CREF
-01	V3 CREF
-008-00	V3 TECO
-01	Final fixes to TECO
-009-00	OS/8 Standard Extensions
-010-00	LPTSPL
-011-00	PIP-1Ø
-012-00	V3 PIP
-01	Final fixes to PIP
-013-00	CCL
-014-01	OS/8 V3 Memos
-015-00	Fixes to TD8E Non-System Handlers
-016-00	V3 Fixes to TTY Handler
-017-00	RESORC
-018-00	Preliminary Project Plan for OS/8 V3
-019-00	New RK8 System Handler
-01	Final Changes to RK8 System Handler
-020-00	Fixes to Card Reader Handler
-01	CDR Handler
-021-00	New LPT Handler

---

* Note from the typist: And your secretary has gone through a month of exhaustive typing.

-022-00	RFØ8 Non-System Handler
-023-00	VTECO
-024-00	Version Numbers for Handlers
-025-00	FOTP Remarks
-026-00	Final Changes to EDIT
-027-00	BAT Handler
-028-00	Fixes to L645 Handler
-029-00	DIRECT
-030-00	VR12 Handler
-031-00	New KL8E Teletype Handler
-032-00	TM8E Magtape Handler
-033-00	PAL8 V8 Notes

Final testing will include a heat test and shock vibration test.

To obtain any memo relating to OS/8 V3, please contact Lynn Clark at extension 4953.

fp

080-100-015-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: September 17, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: Fixes to TD8E Non-System Handlers

All four TD8E Non-System Handlers have been fixed as follows:

1) Parity ↑C is now accepted.

2) The ↑C check no longer advances the low speed reader.

In addition, TD8EC and TD8ED have been fixed as follows:

3) A bug was fixed in their initialization table so that these handlers now work. Previously they would change random locations in the user's program and not work at all.

Bug 3 appears to have gotten out into release 2 only because the handlers were never tested! Unfortunately, this unexcusable practice could happen again in V3 unless someone tells me where we have an 8/E with 8 TD8E DECtape drives.

All these new handlers can be identified by the fact that they are version A.

fp

080-100-016-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: September 14, 1973

FROM: S. Rabinowitz

DEPT: Small Systems

EXT : 2130

SUBJ: V3 Fixes to TTY Handler

The TTY handler know as AS33 has been fixed for V3 as follows:

1. A parity ↑Z is recognized on output.

2. ↑Z is recognized on output even if it is not followed by zeroes. This fixes the bug of sometimes getting extra characters on TTY output after the end of the file.

These fixes correspond to version A of the handler.

Remark: The TTY handler is currently set up so that typing ↑O stops output for that buffer only. Output resumes with the next buffer. Some people might find the following patch useful:

Change relative location 175 from 6∅32 to a 7∅∅∅.

This patch causes ↑O to terminate TTY output forever throughout multiple buffer dumps until either the program ends (dismissing the handler from core) or until a character other than ↑O is typed which will cause output to resume.

This patch works for version A and almost all previous versions.

fp

080-100-017-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: September 14, 1973

FROM: S. Rabinowitz

DEPT: Small Systems

EXT : 2130

SUBJ: RESORC

RESORC is a new OS/8 utility which I have written for OS/8 V3. RESORC is short for RESOURCES. It is used to allow you to find out what handlers you presently have configured on a given system (or system head) and has an option to tell you further information about these handlers.

Example:

```
.RES)
SYS,DSK,DTAØ,DTA1,TTY,LPT,CDR,PTR
```

This example shows RESORC being invoked through the CCL command RES. RESORC then prints out the system names for all the handlers on your system device.

Although RESORC is most conveniently called via CCL, the remainder of this memo will give illustrations showing RESORC invoked the long way, e.g.,

```
.R RESORC
*LPT:←DTA3:
```

This example shows the list of handlers on the system tape currently mounted on DTA3: being sent to the line printer.

Detailed Explanation:

RESORC takes an optional output specification and from Ø to 9 input files (separated by commas).

The output file is where the RESORC listing is sent. If no output device is explicitly specified, TTY: is assumed. (This may be changed to LPT: or TV: by the /L and /S options in CCL.) If no filename extension is given, .LS is assumed. If no output filename is given, RE is assumed. The output device must not be read-only.

The input specifications may be of two types:

1) A device name only

   In this case, the device must be file-structured, and is presumed to have a valid OS/8 directory and monitor on it. RESORC then looks at the handlers that are built into the system on this device. These handlers are not available to the user unless he bootstraps into this device.

2) A device and filename

   In this case, the file must be what is known as a system-head file. (Such files are created by the /Y option in PIP and are copies of the system portions of devices). If no filename extension is specified, the extension .SY is tried first (by default). RESORC then looks at the handlers in the system saved on this file. System-head files are 5Ø (decimal) blocks long.

NOTE

To directly access these handlers, the user would have to recreate the system from whence they came. He must know in advance (or find out via RESORC) what device it is a system-head for. He could then recreate this configuration by the commands:

```
.R PIP
*DEV:<FILE.SY/Y$
.R BOOT/DV
```

Important: If DEV:=SYS:, you must still rebootstrap in order to adjust your in-core tables.

A special case occurs when no input files are given or when the first input file is null, e.g.,

```
.R RESORC
*LPT:<$
```

or

```
.R RESORC
*DSK:FILE<,DTA3:
```

In such a case, the device referred to is SYS: (not DSK:), and wherever applicable, the in-core tables are examined rather than the default tables on SYS:. This mode is typically used to find the user-defined names for the in-core system.

Regular Mode:

If no options are specified, RESORC runs in regular mode, and merely prints the system device names for the handlers which exist on the system in question. If RESORC cannot figure out the ASCII device name for one of the devices, it will print the internal octal representation of the device name enclosed in parentheses. Example:

```
SYS,DSK,DTA2,DTAØ,DTA1,(4667),TTY,PTR2
```

The first two devices are always SYS and DSK respectively. The devices are separated by commas and listed in order of their internal device numbers.

Full Mode:

The /F option may be specified in order to get the full mode printout. Example:

```
.R RESORC
*/F
128 FREE BLOCKS
NAME  TYPE  USER
SYS    RK
DSK    RK    IN
DTAØ   TC8   Ø
TTY    TTY
LPT2   LPTR  LPT
```

In the first column are the system device names which are the same as the ones you get in regular mode.

In the second column is the physical type of the corresponding handler. Every different type of device is assigned a unique number by OS/8 which is used in the device control word table. This is what is referred to here. Note that physically different devices which are similar have the same type (for example, all line printers, whether LPØ8, LS8E or L645 have the type number 4). The currently assigned device type numbers and the names associated with these types by RESORC are given below:

Internal Type Code	RESORC Type Name	Explanation
Ø	TTY	Teletype
1	PTR	paper tape reader
2	PTP	paper tape punch
3	CR8E	card reader
4	LPTR	line printer
5	RK	non-fixed head disk
6	RFØ8	1 platter RFØ8 disk
7	RFØ8	2 platter RFØ8 disk
1Ø	RFØ8	3 platter RFØ8 disk
11	RFØ8	4 platter RFØ8 disk
12	DF32	1 platter DF32 disk
13	DF32	2 platter DF32 disk
14	DF32	3 platter DF32 disk
15	DF32	4 platter DF32 disk
16	TCØ8	TCØ8 DECtape
17	LINC	LINCtape
2Ø	TM8E	Magtape
21	TD8E	TD8E DECtape
27	TA8E	Cassette
3Ø	VR12	VR12 Scope

Type codes 22-26 and 31-37 are reserved for future use by DEC. Codes 4Ø-57 should be reserved for use by users.

Column 3 gives the currently assigned name given to the device with the monitor ASSIGN command. If RESORC cannot reconstruct the name from the internal octal, it will print the octal enclosed in parentheses.

Remark: Because of the scheme used by OS/8 to pack 4-character device names into one 12-bit word, devices may not have unique names. That is, two different names may have the same octal representation and hence both refer to the same device. For example,

TTY2 and LTA2

both refer to the device whose internal octal representation is 56Ø6. There is no guarantee that RESORC will pick the name preferred by the user.

Preceding the table of device names, RESORC prints the number of free blocks on the device (as found by examining the directory of the device.) This information is not given in the case of system head files, since it is not applicable.

Expanded Mode:

The /E mode will be explained in a future memo, because if this memo gets any longer, the time required to get it typed will get disproportionately out of hand.

fp

080-100-017-01

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: December 6, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT : 2130 LOC 12-3

SUBJ: Concluding Remarks on RESORC

This memo describes the /E option in RESORC.

E stands for extended listing.

Using this option, one gets more detailed information concerning the handlers configured into a system. With the /E option, a table is produced with the following headings: #, NAME, TYPE, MODE SIZ, BLK, KIND, U, V, ENT and USER. These are explained below.

Internal device number for the handler. If a number is missing, then there is no handler for this internal number.

NAME Permanent device name for the handler, if decipherable by RESORC. Otherwise, internal coding for this name. Same as with /F option.

TYPE Type of device. Same as with /F option. Determined entirely by internal device type as follows:

Device Type	Corresponding Name
00	TTY
01	PTR
02	PTP
03	CDR
04	LPTR
05	RK8
06	RF08
07	RF08
10	RF08
11	RF08
12	DF32
13	DF32
14	DF32
15	DF32
16	TC08
17	LINC
20	TM8E
21	TD8E
22	BAT
23	RK8E
27	TA8E
30	VR12

MODE Consist of from 1 to 3 letters with the following meanings

R The handler may be used for reading
W The handler may be used for writing
F The handler handles file-structured devices

SIZ The size of the device in blocks (decimal). Only applicable for file-structured devices.

BLK The block on the system device in which this handler resides. If this number is followed by a +, this indicates that the handler is 2-pages long. If this entry is SYS, this means that the handler is permanently resident in core in page 07600.

KIND This entry tries to differentiate the specific handler better than the TYPE column. There may be several devices all of the same type (for example, DEC has several lineprinters, and their handlers all have device type 04). There may be several different handlers for the same device. If there is only one handler for the type specified, this entry may be blank. This entry has no meaning for user-written handlers. Explanation of common kinds of handlers.

Type	Kind	Description	How Identified
TTY	AS33	1-page handler	by no. of pages
TTY	KL8E	2-page handler	by no. of pages
PTR	KS33	low-speed reader	by IOT codes
PTR	PT8E	high-speed reader	by IOT codes
PTP	KS33	low-speed reader	by IOT codes
PTP	RT8E	high-speed punch	by IOT codes
CR8E	026	DEC-026 card codes	by table codes
CR8E	029	DEC-029 card codes	by table codes
LPTR	LP08	Old LP08 handler	location dependent
LPTR	LS8E	Old LS8E handler	location dependent
LPTR	LPSV	LP08/LS8E/LV8E handler	location dependent
LPTR	LV8E	LPSV altered for LV8E	location dependent
LPTR	L645	annalex line printer	location dependent

U Unit. Certain handlers can handle many units of a given device. This specifies the particular unit number.

V Version number (letter) of handler. No entry means the handler predates OS/8 version 3. Version numbers are of the form A-Z. The sixbit of the ASCII representation of the version letter of a handler resides in the handlers entry print location.

ENT The relative entry point of the handler.

USER Same as for /F option. Current user name for the handler (only applicable for the in-core system).

In addition to the handler information, the /E option also tells the user the following information:

If device has directory (as opposed to system head file):

No. of files in directory
No. of blocks used
No. of segments used
No. of free blocks
No. of empties
No. of additional information words (if not 1)

Number of free device numbers.
Number of free slots.
Version number of monitor if device is system device.

080-100-017-02

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: December 24, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: Additional documentation for RESORC

The switch /F (standing for full mode) has been removed and replaced by /L . The /F switch now means fast mode and is also the default.

To summarize:

/F Fast Gives 1 line printout (default)

/L Regular Gives 3 column printout

/E Extended Gives enormous printout

Note on RK8E's: One can have up to 4 physical drives attached to an RK8E disk controller under OS/8. These drives are numbered 0, 1, 2, 3. OS/8 treats the disk cartridge in each drive as two logical units. The lower half is called the A unit and the upper half is called the B unit. Thus drive 2 consists of two logical units called A2 and B2.

Since there is room for only one character in the U column of an extended RESORC printout, RESORC will (arbitrarily) number the logical units from 0 to 7. the correspondence between these numbers, logical units, and physical devices is given below:

U	logical unit	Physical device
0	A0	0
1	B0	0
2	A1	1
3	B1	1
4	A2	2
5	B2	2
6	A3	3
7	B3	3

Furtermore, RESORC unit numbers may be incorrect when running OS/8 under SRT-8. This bug will be fixed in a later release of RESORC.

Please add device type 24 with name NULL to the table in the previous memo.

080-100-019-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: September 17, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT : 2130 LOC: 12-3

SUBJ: New RK8 System Handler

The RK8 system handler has been rewritten from scratch (mostly mimicing the COS 3ØØ handler). It is now a lot shorter and cleaner.

In addition, I have added a second entry point so now the non-system handler, RKA1: is coresident with the system handler.

This does not obsolete the old non-system handler (which I incidentally fixed so that it can now correctly perform a full page read or write).

fp

digital

080-100-019-01

INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: October 22, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: Final (hopefully) Changes to RK8 System Device

The new RK8 system handler can run off any platter. After bootstrapping into any unit, the secondary bootstrap modifies the resident handler in core to refer to this unit. Thus, SYS: might refer to drives Ø, 1, 2, or 3.

It would be unwise to assign the user name (or permanent name) RKAØ: to be the same as SYS: since SYS: is not always unit Ø. The coresident entry point at 7621 always refers to unit 1 however, and may be given the name RKA1: .

This memo does not obsolete 080-100-019-00.

fp

080-100-020-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: September 17, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: Fixes to Card Reader Handler

The new card reader handler (now version A) includes the patch given in the January 1973 SIS bulletin which fixes the bug converning reading cards with an odd number of columns.

fp

080-100-020-01

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: October 22, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: CDR Handler

This memo is a continuation of 080-100-020-00.

The card reader handler now clears the card done flag when it's through. This is version B.

fp

080-100-021-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: September 17, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT : 2130 LOC: 12-3

SUBJ: New LPT Handler

Thanks to Richard Lary we now have a new line printer handler. It has the generic name LPLS but will still be referred to with LPT as its device name.

It is a one-page handler which handles both the LPØ8 and LS8E style lineprinters.

It handles tabs and line overflow. It ignores nulls and prints escape as a dollar sign. Of course, it handles ↑Z and ↑C.

It does not simulate vertical tabs and does not treat ↑N specially.

This handler obsoletes the two previous ones, LPØ8 and LS8E.

This handler has two useful locations which may be patched (using ALTER in BUILD).

(i) Relative location Ø specifies the width of your line printer. Set it to the one's complement of the width desired. It comes initially set to 7573 (octal) which corresponds to a 132 (decimal) column printer. Thus, you would change it to 7657 (octal) for an 8Ø column printer.

(ii) Relative location 1 specifies whether you want carriage returns to be inhibited. It is initially Ø. Set it to -15 (octal) to inhibit echoing of carriage returns for the sake of improving speed if your line printer already does a carriage return automatically upon seeing a line feed.

fp

080-100-022-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: September 17, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: RFØ8 Non-System Handler

I have modified the RFØ8 system handler to give us an RFØ8 non-system handler. Although not of extreme use, there have been many times when I wished for one. It is called RF: in the copy of BUILD to be put on the in-house distribution tape.

Also, there is a DF32 non-system handler called DF:.

Both handlers require the user to specify how many platters are on his disk if he plans to use these handlers with either the /Z or the /S option in PIP. There is currently no way to do this with the existing BUILD but there will be a way in the future.

As supplied, both handlers assume you have 4 platters.

fp

080-100-023-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: September 17, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: VTECO

The following patch to TECO causes it to take input and output from a teletype device other than the standard system one with device codes 3 and 4.

xx denotes the device code of the input device

yy denotes the device code of the output device

This patch is normally used when you have a VTØ5 as a second teletype. The starred items need only be used if the VTØ5 is high speed.

```
*Ø152/ØØ25   117Ø;764Ø5551;2Ø1Ø;5153;5551
 Ø23Ø/6Ø32   6xx2
 Ø461/6Ø31   6xx1
 Ø463/6Ø34   6xx4
 Ø473/6Ø41   6yy1
 22Ø2/6Ø31   6xx1
 22Ø4/6Ø34   6xx4
 2211/6Ø41   6yy1
 2214/6Ø46   6yy6
*2215/72ØØ   4151
 3143/6Ø31   6xx1
 3146/6Ø36   6xx6
 5211/6Ø46   6yy6
```

This patch only works on TECO V3Ø4 and the result should be called VTECO.

The corresponding patch for version 2 of TECO can be retrieved as memo IF-22 of 1Ø/17/72.

fp

080-100-024-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: September 18, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT : 2130 LOC: 12-3

SUBJ: Version numbers for handlers

I am implementing the following scheme for assigning version numbers to OS/8 handlers.

Version numbers consist of either a space or a letter from A to Z. (Actually, these should be called version letters) The version number is represented internally by the code:

Ø	means	space
1	means	A
2	means	B
.		.
.		.
.		.
32	means	Z

The version number for a handler appears in the handler and is located at the first location which is less than or equal to the entry point of the handler, and which contains a number in the range Ø - 32, and is closest to the handler entry point.

All existing V2 handlers have the version number space.

Handler version numbers can be determined using the /E mode of RESORC.

fp

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: September 24, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: FOTP Remarks (by popular demand)

This document is a technical reference manual and is not meant to be a user's guide. Persons only interested in using FOTP should read the examples in the back and burn the intervening material.

FOTP stands for File-Oriented-Transfer Program. It is used to perform a sequence of image-mode transfers, file deletions, and file nomenclature revisions.

It can be run from the OS/8 system device by typing the command

.R FOTP

It responds with an asterisk (and calls the OS/8 command decoder in special mode). At this time you specify one output specification and from one to five input specifications. Switches may also be specified.

Input specifications consist of a device, a filename, and a filename-extension. Each of these is optional except that no extension is permitted if the name is omitted. Each name consists of either

a) from 1 to 6 letters, digits, or question-marks

or

b) a single asterisk

Each extension consists of either

a) 1 or 2 letters, digits, or question-marks

or

b) a single asterisk

The filename and extension are separated by a period which is optional if no extension is given.

Each device consists of from 1 to 4 letters or digits followed by a colon.

Examples of legal input specifications:

```
DSK:
SYS:A
LTA3:FOOBAR
DTA7:A.BC
FILE
FILE3.06
4
32:A.*
NAME?.TX
N?ME.
??????.A?
*
*.BN
FOO:*.??
?A?B?C.?D
```

Input specifications are separated from each other by commas.

Illegal input specifications:

```
A.B.C
A:B:C
A?*.B
.AB
FOO:A.*B
A?B:C
*:FOO
```

The output file specification has the same general form as the input specification except that question-marks are not currently legal.

If no output device is specified but a filename is given, then DSK: is assumed.

For each input specification given, if no device is explicitly given, then the device associated with the previous specification is assumed. If no device is explicitly given for the first specification, then DSK: is assumed.

Thus, the following command strings are equivalent

```
A<DTA3:B                          DSK:A<DTA3:B
A.*<SYS:B.*,C.*,D.*               DSK;A.*<SYS:B.*,SYS:C.*,SYS:D.*
FOO:*.*<B.*,DTAØ:C.*,,SYS:*.BN    FOO:*.*<DSK:B.*,DTAØ:C.*,DTAØ:,SYS:*.BN
```

With the above understanding, we will assume in the following discussion that the device is always given.

Only file-structured devices are legal for input devices. They must not be write-only.

Any device which is not read-only is legal as the output device.

Definition of Device Group

If the /U option is given, then a device group is a single input specification.

If the /U option is not given, then a device group is a maximal sequence of consecutive input specifications all of which specify the same logical device.

Example: The command line

```
SYS: < A,B, LTA2:C,D, LTA2:E
```

contains two device groups. These are

```
DSK:A,DSK:B
```

and

```
LTA2:C, LTA2:D, LTA2:E
```

Example 2: The command line

```
A < SYS:B,C/U
```

contains two device groups. These are

```
SYS:B        and      SYS:C
```

Example 3:

```
.ASSIGN DSK FOO
.R FOTP
*A<B,FOO:C,SYS:D,E,DSK:F
```

contains the three device groups:

1) DSK:B,DSK:C
2) SYS:D,SYS:C
3) DSK:F

assuming that DSK: and SYS: do not refer to the same physical device.

Operation of FOTP:

I. The output device directory is read into core. If there is no output device or if it is non-filestructured, then operation proceeds as if it had an empty directory. However, if there was no output device and the /D option was specified, then FOTP assumes the output device is the same as the first input device.

II. For each input device group (in order), the following operations are performed:

1. The directory of the input device is read into core. A distinct copy is kept in core even if the input and output devices are the same.

2. The input file specifications are examined. If one is missing, then *.* is assumed except if /D is specified in which case the null filename is assumed.

3. A set of filenames is made up of all files on the input device which have the form specified by any one of the input specifications for this device group. If the /V option is specified, then this set consists instead of all files on the input device which are not of any of the forms specified by the input specifications.

Furthermore, whether or not /V is specified, if /C is specified, only files which have the same date as today's date may enter this set. If the /O option is specified, only files with other than today's date may join this set. An undated file does not have today's date unless no date was specified to OS/8 today.

Each letter, digit, or null in the input specification can match only input filenames with the same character in the corresponding position.

If the input specification contains no *'s or ?'s, then it can match only a file by that name. If no extension is given, it can only match a file with no extension.

If the filename is a *, then this input specification matches any file which has an extension matched by its extension.

If the extension is a *, then this input specification matches any file which has a name matched by its name.

If the filename (or extension) contains a ? as the kth character, then this input specification matches any file which is matched by the other characters with any arbitrary character in position k. For purposes of this test, all filenames (both real ones and input-specification filenames other than *) are presumed to consist of exactly 6 characters, padded with nulls on the right. Similarly, extensions are presumed padded with nulls to 2 characters. Thus if the symbol Δ denotes a null, then the input specification, AB??, is the same as AB??ΔΔ and could match a real file by the name ABCDΔΔ.ΔΔ or by the name ADCΔΔ.ΔΔ but could not match one by the name ABCDEΔ.ΔΔ.

Examples:

Specification	What it Matches
ABC.DE	Only ABC.DE
A.*	Any file with name A and any extension (including no extension)
*.BC	Any file with extension BC.
.	Any file
*.	Any file with null extension
*	Same as *.
AB?CDE.FG	Any file whose 1st, 2nd, 4th, 5th, 6th, 7th, and 8th characters are A,B,C,D,E,F,G, respectively but whose 3rd character is arbitrary
???.PA	All files whose names consist of from 1 to 3 characters and which have the extension PA.
ABC.P?	Any file with the name ABC and extension beginning with a P.

????P.*	Any file with a 5-character name ending in P and any extension.
A.??	Same as A.*
A.?	Any file with name A and either a null extension or a one-character extension.

Each ? must match exactly one character (which may be a null). Each * must match an entire field (either a filename or a filename extension).

A specification may not include embedded *'s, thus A*B is an illegal specification and has no meaning.

Even if a device group consists of more than one specification, no filename will appear more than once in the list made up by FOTP at this stage of the process. For example, in the specification.

A.* , ??.PA , A?.?A

the file A.PA will appear once in FOTP's list even though it matches all three of the forms given.

A file formed now may of course be matched later on during another device group.

More examples:

Specification	What it Matches
A.B/V	All files other than A.B
A.*/V	All files with name other than A
A.* , B.*	All files with name A or B
A.* , B.*/V	All files other than files with name A or B
*.?/V	All files with 2-character extensions

./V	No files
. , A.B/V	No files
A,B,C/V	All files other than A,B, or C
P?????.*,*.P?/V	All files which don't either start with P or have an extension starting with P.

4. After this set is made up, FOTP examines these files, one at a time, in the order of their appearance in the input device directory.

   If the set is empty, FOTP prints the warning message

   NO FILES OF THE FORM specification

   on the Teletype, once for each specification among those included for this device group. If /V were specified, it is so mentioned. Similarly if /C or /O is mentioned.

   Note: If a device group contains multiple specifications, then one of these specifications could have no matches without the user being told. If he desires notification, he should use the /U option to force separate device groups. This is ugly.

   The filenames in this set are now put into an ordered list, in the order of their appearance in the input device directory.

   For each file in this input list, FOTP now forms a corresponding output filename using the input file name and the output specification as follows:

   1) For each letter, digit, or null in the output specification, that character appears in that position in the output filename.

   2) [Not implemented in OS/8 FOTP]

      For each ? in the output specification, the corresponding character in the input file name appears in that position in the output filename.

   3) For each * in the output specification, the corresponding field (name or extension) in the input file name appears in that field in the output filename.

The action of ?'s is described here only because they are on the PDP-10 and so the reader might be interested in their use when appearing in an output specification. FOTP does not support this feature and will not in this release.

Examples:

Input filename	Output Specification	Corresponding Output Filename
ABC.DE	FGHI.J	FGHI.J
ABC.DE	FGHI.*	FGHI.DE
ABC.DE	*.J	ABC.J
ABC.DE	*.*	ABC.DE
ABC.DE	F?HI.J	FBHI.J
ABCDEF.GH	P?S?T.*	PBSDT.GH
ABCD.XY	*.?K	ABCD.XK

If an output device is given but no output specification, then FOTP assumes *.* as the output specification.

If no output device or filename is given, then FOTP assumes null as the output specification and consequently all the constructed output filenames are null.

We are now left with an ordered list of pairs of filenames, an output filename and an input filename. There remain no *'s or ?'s in any names. Input filenames are not duplicated, but output filenames may be.

FOTP is now ready to go into action.

5. FOTP now performs the following steps, in order, for each filename pair, proceeding in their order of appearance in the list just constructed (i.e., from low to high block numbers on the input device):

Definition: The relevant filename of a filename pair is the input filename unless the /D option is specified in which case the relevant filename is the output filename.

a) If the /Q option is specified, FOTP prints the relevant filename on the Teletype and waits for the user to respond. If the user types a "Y", then FOTP proceeds to the next step. If the operator types an "N" or in fact, any character other than "Y" or "↑C", then FOTP skips the following steps and proceeds to the next filename pair.

b) If the /L option is specified, FOTP now prints the relevant filename on the Teletype. This step is skipped if /Q is also specified.

c) FOTP now gets the block number where the input file starts from the input directory. If the output device is the same as the input device, FOTP then looks up this same file on the output directory to see if the starting block numbers agree. (Although both directories are for the same device, they may not be identical at this point because output operations are performed only on the output directory in core and corresponding changes are not made on the in-core input directory. This artifice is necessary.) If the file's location is the same in both directories then FOTP proceeds to step d). If the file is not found in the output directory (e.g., it got deleted in an earlier step) or if the file's location has changed (e.g., it got moved in an earlier step) then FOTP ignores this input file and proceeds back to step a) going on to the next filename pair. This step is necessary but not sufficient to prevent anomalous results. Let the interested reader contemplate the specification A.*<*.B and consider the file DSK:A.B.

d) If the /R option was not specified, then FOTP proceeds to step e).

   If the /R was specified, then FOTP performs the following actions:

   (i) The input filename is looked up in the output device directory. If it is not there, FOTP proceeds to step (iv). No error message is given.

   (ii) The output filename is looked up in the output device directory. If it is not there, fine, well and good. However, if it is found there, FOTP generates the error message

   ALREADY EXISTS - output-filename

   and then proceeds back to step a), going on to the next filename-pair.

(iii) Without performing any data transfer, the name of the output file on the output device is changed (in-place) to become the name specified by the input filename. The date and any additional information words are not affected, unless the /T option is specified, in which case the file is given today's date.

(iv) FOTP now proceeds back to step a), going on to the next filename pair.

e) If the /N option is not specified, and if the output device is not the same as the Input device, then FOTP now looks on the output file directory (in core) to see if a file with name the same as the current output file name exists. If one exists, it is deleted from the output directory (in core). If none exists, it is not an error. The directory physically on the output device is not yet touched.

f) If the /D option is not specified, then FOTP proceeds with the image transfer. First, a tentative output file is opened on the output directory in core. It has the same name as the current output name. It is located in place of the first empty which is the best fit for the file, i.e., has the smallest size which is not smaller than the size of the input file. If there is no empty large enough, FOTP prints

NO ROOM SKIPPING input-filename

and proceeds back to step a) going on to the next filename pair. The new tentative output file entry is given the same additional information words as those in the input file entry with the following exceptions:

1. If the /T option is specified, then the first additional information word (if present) is set to today's date.

2) If the output directory has fewer additional information words than the input directory, the extra words are not copied.

3) If the output directory has more additional information words than are in the input directory, the extra words are set to zeroes.

Then the input file is copied to the output file in image mode (block for block). If an I/O error occurs while either reading or writing, FOTP prints:

ERROR ON INPUT DEVICE, SKIPPING input-filename

or

ERROR ON OUTPUT DEVICE, SKIPPING output-filename

and proceeds back to step a) going on to the next filename pair. The tentative file in the output directory is not closed. When the image transfer is completed, the active tentative file is changed to a permanent file. (i.e., the file is closed). The output directory is not yet written out onto the physical output device.

6. After the last filename pair has been processed, FOTP proceeds back to step 1 going on to the next input device group.

III. When all device groups have been finished, FOTP then writes the output directory from core onto the physical output device. An I/O error at this time is disastrous to the user. No operation is performed if the output device is non-file-structured.

IV. If the command line ended with a carriage return, FOTP now recalls the command decoder and allows another operation to be specified. If the command line had ended with an altmode, FOTP returns instead to the OS/8 keyboard monitor by branching to location 7605.

Mnemonic Meanings of the Switch Options:

/C	Current Date	Match only input files with the current (today's) date.
/D	Don't Transfer	Don't perform any I/O transfers, i.e., at most perform only deletions. Note: /D is not an abbreviation for DELETE although it usually performs this option. If no transfer occurs, no post-deletion occurs. Predeletion might still occur unless the /N option is also given.

/L Produce Log	Produce a written record on the Teletype of the relevant filenames which occurred during the operation. Note that the device is not specified nor is the second file of the filename pair.
/N No pre-delete	Don't delete output filenames before I/O transfers. If I/O transfer proceeds, when the file is closed, any existing ones of the same name will automatically be deleted.
/O Other than current date	Match only input filenames which have a date which is not the same as what OS/8 thinks is today's date.
/Q QueryMode	Query the user about each relevant filename to find out whether or not he wants the specified operation to occur for that file.
/R Rename	Change the output filename(as specified by the input filename) without performing any transfer.
/T Today's Date	Give each output file today's date instead of the date on the corresponding input file.
/U Ugly Mode	Treat each input specification as if it belonged to a separate device group. This is wasteful because it will cause a device's directory to be read multiple times, once for each input specification, instead of just once. However, it will cause matching files to be found in the same order as specified by the input specifications.
/V inVert sense of matches	Match only input files which do not have the form specified by the input specifications.

Operation of ↑C:

Because of safety reasons, ↑C works slightly differently in FOTP than in other OS/8 cusps. If ↑C is typed on the keyboard, then FOTP continues operation until it reaches the end of step II part 5 as previously described. At this point, if FOTP had performed any output on the physical output device, then it proceeds to step III as if all device groups had been processed. This assures that the files on the output device will agree with the directory on the output device. If on the other hand, no output had physically been performed on the output device, (for example, the user did delete only, or only got a log)then FOTP proceeds to step IV.

If ↑C had been typed, step IV always returns to OS/8 via location 7600.

Note that it would be suicidal for the user to hit halt while FOTP was running and manually branch to location 7600.

Examples of FOTP command lines:

Transfer the file A.B from disk to DECtape:

```
*DTAØ:<A.B
```

Transfer the files A,B,C,D and E from SYS: to DTA3:

```
* DTA3:<SYS:A,B,C,D,E
```

Transfer all FORTRAN source files from one DECtape to another, producing a log of those copied:

```
* DTA2:<DTA5:*.FT/L
```

List all FORTRAN and BASIC files on the lineprinter in order of appearance on DSK:

```
* LPT:<*.FT,*.BA
```

List all FORTRAN and BASIC files on the lineprinter listing all FORTRAN files before all BASIC files:

```
* LPT:<*.FT,*.BA/U
```

Copy all files other than .SV and .BN files from DTA3: to DSK:, then copy all files other than those whose name begins with a K from DTA2: to DSK:. Log all files copied:

```
* DSK:<DTA3:*.SV,*.BN,DTA2:K?????.*/V/L
```

Copy the file A.B from DSK: to DTA1: changing its name to C.D. Give the new file today's date:

```
* DTA1:C.D<A.B/T
```

Copy all files from LTA2: which have the extension .PA to SYS: changing the extension to .PL allocating storage on SYS: without doing pre-deletions:

```
* SYS:*.PL<LTA2:*.PA/N
```

Make a copy of all text files on the disk, but change the third character of the name to an "X" (not implemented):

```
* ??X???.*<*.TX
```

Find all files on RKA2: with name FOO and any extension but which have today's date, and copy them to SYS: Changing the name to BARF yet keeping the extension:

```
*SYS:BARF.*<RKA2:FOO.*/C
```

Delete all disk files (other than one's with today's date) which either have the extension .LS, .TM, or .BK or have a name whose name starts with TMP:

```
* DSK:<*.LS,*.TM,*.BK,TMP???.*/D/O
```

Delete the file FOO.BN from LTA7:

```
* LTA7:<LTA7:FOO.BN/D
```

This is the same as the command

```
* LTA7:*.*<LTA7:FOO.BN/D
```

Note that although it may seem strange at first, files to be deleted appear as the input files. Nevertheless only output files are deleted. This is because, in the above examples, the output specification is *.* so that the input file names really also become output file names.

This might become clearer by a more complicated example.

Delete the files A,B, and C on DTA2: but only if they already exist on DSK:, log those files that get deleted.

```
* DTA2:<A,B,C/L/D
```

If at least one of the files A,B, or C exists on DSK: then you will not get the NO FILES OF THE FORM message, whether or not A,B, or C appear on DTA2:. If the file A appears on DTA2:, it will only be deleted if the file A appears on DSK:. The file B might occur on DSK: but not on DTA2: in which case no deletion of B occurs and no error message is given. If the input device on a delete is not the same as the output device, then the input device may be write-locked.

Copy all files from DSK: to SYS:

```
*SYS:<DSK:
```

This is the same as

```
*SYS:*.*<DSK:
```

which is the same as

```
*SYS:<DSK:*.*
```

For your own protection, *.* is not assumed as an input specification if /D is given. Thus

```
*SYS:<SYS:/D
```

deletes no files and is not the same as

```
*SYS:<SYS:*.*/D
```

which deletes all files.

Note: the command

```
*SYS:A,B,C/D
```

is the same as

```
*SYS:*.*<SYS:A,B,C/D
```

because the /D option causes the output device (if none given) to be the same as the first input device.

Some more complicated examples:

Delete all .BN files for which there is a corresponding .PA file:

```
**.BN<*.PA/D
```

Delete all .LS files on DTA3: for which there is a file on RKAØ: with the same name but an extension of either .PA, .RA, or no extension:

```
*DTA3:*.LS RKAØ:*.PA,*.RA,*/D
```

Delete all files on the disk for which there are already copies on one of the 4 DECtape drives:

```
*DSK:<DTAØ:*.*,DTA1:*.*,DTA2:*.*,DTA3:*.*/D
```

Produce a log of all files on TDA1: which happen to have the name FOO and an extension which is the same as an extension of some file on SYS: which has a one or two-character filename beginning with a T. Do not perform any transfers or deletions:

```
*TDA1:FOO.*<SYS:T?.*/N/D/L
```

Change the name of the file DSK:FILE.PA to FILE2.PA.

```
*FILE2.PA<FILE.PA/R
```

Rename all files on DTA6: with .PA extensions to have .PB extensions instead.

```
*DTA6:*.PB<DTA6:*.PA/R
```

Change the extension from .RL to .OL of all files on DTA1: with extension .RL which correspond to files on DSK: with the same name and today's date:

```
*DTA1:*.OL<*.RL/C/R
```

Interaction between FOTP and CCL:

All CCL commands which invoke FOTP,pass FOTP both the /L option and the altmode bit.

LIST If no output device is specified, LPT: is forced.

TYPE If no output device is specified, TTY: is forced.

COPY CCL first prints FILES COPIED:

DELETE CCL first Prints FILES DELETED:
The /D option is included.

RENAME CCL first prints FILES RENAMED:
The /R option is included.

FOTP Error Messages

ILLEGAL ?	? is not allowed in output specification
ILLEGAL *	Embedded *'s are not permitted
SYSTEM ERROR-CLOSING FILE	"Man are you in trouble" - H.J. 1973. If you are brave, hit continue after computer halts.
ALREADY EXISTS-output-filename	You cannot rename a file to a name which already exists on the output device.
ERROR READING INPUT DIRECTORY ERROR READING OUTPUT DIRECTORY ERROR WRITING OUTPUT DIRECTORY	Self explanatory
NO ROOM, SKIPPING input-filename	There was no room on the output device to perform the transfer. Pre-deletion may have already occurred.
ERROR ON INPUT DEVICE, SKIPPING input-filename ERROR ON OUTPUT DEVICE, SKIPPING output-filename	Clear
NO FILES OF THE FORM specification	No files of these forms were found on the current input device group
BAD INPUT DIRECTORY BAD OUTPUT DIRECTORY	Obvious

Remarks:

1. FOTP can transfer files greater than 256 blocks long.
2. FOTP's starting address is 146ØØ.
3. Output sizes specified with the square bracket notation are ignored.
4. Typing ↑P at most any time causes FOTP to stop what it's doing at the next safe spot and return to the command decoder for a new input command.

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: December 24, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: Remarks Re Remarkable and Remarked-about FOTP

/W Print FOTP's version number on teleprinter

/F Failsafe! This option is used to backup large numbers of files in one big swoop even if they won't all fit onto a single DECtape (or other device).

If this option is not given, then if a file doesn't fit on a device, FOTP prints a warning message (described elsewhere) and skips the file. It may have been a big file and so perhaps other files will still fit on the device in question.

If /F is specified, however, then if a given file won't fit on the output device, FOTP will fix up the directory of that output device and then print a message asking you to dismount this device (if you can) and mount another volume. It will use this next volume to continue the transfer operation. When you have mounted the new volume, type any character and FOTP will continue. You can not specify a different device at this time. Be sure the device you mount has a good OS/8 directory.

In the previous memo, add /O to the third example on page 14. Note that by the previous memo, I mean 080-100-025-00 inasmuch as memo 080-100-025-01 was censored by upper management and has not been distributed. It is, however, not obsolete.

↑P causes FOTP to abort it's current operation. It first of course fixes up the output device directory.

Under no circumstances should the user ever hit halt while FOTP is running and branch to 7600 or 7605. Disastrous results will happen. If you must terminate FOTP unexpectedly (for example, if an earthquake causes a power failure), then you must manually rebootstrap onto your OS/8 device.

↑C causes FOTP to abort as explained in the earlier memo except if the /Q option had been specified, in which case, even though no data had been written to the output device (for example with /D/Q), FOTP will update the output directory. (In all other cases it will print OUTPUT DIRECTORY PRESERVED). This feature allows you to get tired of the querying, and return to OS/8 with all the actions up to that point having been done.

080-100-026-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: October 22, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: Final Changes to EDIT

I have made the following changes to OS/8 EDIT:

1) It is now legal to chain to EDIT. In fact, the CCL commands CREATE and EDIT both chain to EDIT.

2) The ?5 error message now goes away because of item 1.

3) I added the # command. Typing # in response to EDIT's #, and then typing carriage return, causes EDIT to type its version number (currently V7). Unfortunately, the V command was already in use by EDIT; that's why I chose "#".

I could not duplicate any digit-swallowing or buffer-full search problems.

fp

080-100-027-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: October 22, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: BAT Handler

Mark Rosenthal has written us a new BAT handler. It is used from a BATCH job to read from the BATCH stream.

Characteristics:

It is a one-page read-only non-file structured device. It uses internal device code 22.

It gives a fatal error if used at a time that BATCH is not running.

When used, it read characters from the BATCH stream (ignoring line feeds, and always creating a line feed after a carriage return).

Whenever it sees a line which begins with a dollar sign, it pads the buffer with ↑Z and nulls, and takes the end-of-file return.

fp

080-100-006-01

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: December 24, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: OS/8 BOOT (Last licks)

BOOT may only be used to bootstrap into unit 0 of a device. It cannot be used, for example, to boot onto RKA1.

080-100-027-01

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: December 24, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: BAT can get stuck

Note the fact that when BAT: encounters a line beginning with a $, it causes an EOF-return to the program which called it. It does not then pass up the $ but in fact, gets stuck there, so that if the user program should call BAT: again, it will get an immediate end-of-file.

This is good for parameter files to PAL8 which consist of only equates; however it is bad for other input to PAL8. This is because PAL8 tries to read the source multiple times, and on the 2nd and 3rd passes, it is going to get immediate EOF's.

080-100-028-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: October 22, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: Fixes to L645 Handler

I have fixed the annalex line printer handler so that it now recognizes a parity ↑C. Also, it will recognize ↑C from the keyboard even if the lineprinter is off, off-line, jammed, etc.

This version is version A.

fp

080-100-029-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: October 26, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: DIRECT

Herb Jacobs has written a new OS/8 CUSP known as DIRECT. It is used to produce directory listings à la PDP-1Ø and is better than PIP. It supports the wild card construction.

When DIRECT is run, it calls the OS/8 command decoder. You then specify an output specification and from one to five input specifications. If no output specification is given, TTY: is assumed. This output specification specifies where the directory listing is to be produced. It may be produced into a file. If a filename is given but no extension, the extension .DI is assumed. The output filename may contain no *'s or ?'s.

The input specification has the same meaning as in FOTP. See the FOTP memo for details of what *'s and ?'s mean. If an input device is given with no filename, then *.* is assumed. For each device group, in order, DIRECT produces a directory listing of only those files on the device which match the input specification. This listing is preceded by today's date and followed by the number of free blocks on the device. The length of each file (in decimal blocks) and the file's date are given. A warning message is given on the teletype if there are no files (on a given device group) of the form specified.

Options:

~~/A List no. of free blocks only~~
/B Include starting block numbers (in octal)
/C List only files with current date
/E Include empties in the directory listing
/F Fast mode. Omit dates and file lengths
/M List empties only
/O List only files with other than today's date
/U Ugly mode. Treat each input specification separately
/V List files not of the form specified
/W ~~List additional information words (in octal)~~ Version number
/L Usual mode
/I additional information words

fp

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: December 24, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: More DIRECT Directions

Delete the /A and /W explanations from the previous memo.

/I Print additional information words (after the first) in octal.

/L usual mode

/W type DIRECT's version number on the teletype

=n Use n columns when making the directory listing.
It is up to the user to make sure that he doesn't specify an n which causes the directory listing to exceed the page size on the ouput media.

=4 is nice on a 132-column lineprinter.

.DI is the default output extension.

/R List the remainder of the files after the first one found. This option causes DIRECT to find the first file (physically) which matches all other specifications given and then give a directory listing of all files above and including this one (and still considering the /C and /O options). Actually, when /R is encountered, as soon as a file matches, the current specification is changed to *.* and all other switches set as they were. This should actually include the additional hack that if a /V switch is specified, it is temporarily removed until the next device group comes around. Right now (V2), if /V is also specified, you get a directory listing of precisely the first file which matched the current specification. (Actually that's useful too...)

No warning message is given by DIRECT if there are no files of a given form on the specified device. This should be apparent upon the examination you will give to the directory listing produced.

080-100-030-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: October 26, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: VR 12 Handler

We have written a VR12 scope handler for OS/8 (running on a PDP-12). It is an improved version of (and therefore obsoletes) our former program, DECUS 12-81. We are now supporting it and it will be included as a standard handler in V3 BUILD. CCL supports it too.

Acknowledgement

I wish to thank the following organizations for their unquestionable help:

P?S
I?S
WCFMPG
FOBM

In particular, I would like to thank the following bums*:

Mario DeNobili
Richard Lary
Hank Maurer
Lenny Elekman
Jack Burness
Rød Dorman
Herb Jacobs
Bob Bean

* A bum is a person who has bummed at least 5 locations out of a tightly coded PDP-8 page of code.

Operation

When characters are sent to the scope using the handler, they are displayed on the VR12 scope (on both channels). Whenever the scope gets full, the handler stops reading more characters from the buffer, and just sits there displaying characters on the scope. It also does this upon encountering a ↑Z in the text, or upon encountering the end of the user's buffer. It displays what is known as a scope page. The user can advance to the next scope page by hitting any character on the teletype (not C).

The screen is full whenever the end of the buffer is reached or whenever the number of lines displayed becomes equal to the maximum number specified by the user.

Control does not return to the calling program until after a character has been struck at a point when the handler is displaying the last scope page of a particular buffer load.

The user sets the number of lines he wishes to see in a single scope page via the switch register (right switches). The switch register is set to the negative of the number of lines to be displayed in a scope page. Depending on the length of the lines being displayed, fewer lines should be allowed so as to prevent flicker. By 'line' here we refer to a physical line across the scope face. When text reaches the right margin of the scope face, it is continued on the next physical line (after a slight indentation).

Whenever a line feed or form feed character is encountered, succeeding text goes on the next physical line. Carriage return characters have no effect on the display.

The group name for this handler is VR12. It has a single entry point known as TV:. It is a two-page write only non-file-structured handler. Its internal device code is 3Ø.

Typing ↑C on the keyboard while the handler is in operation causes control to return to OS/8 via location 76ØØ.

Restrictions:

1) The buffer specified in the call to this handler must start at an even address.

2) The handler does not preserve the status of the keyboard flag (it always clears it) or of the LINC-mode special function register (always sets small character size).

3) The handler does not give an error if the user tries to read data using it.

4) If the buffer associated with the handler call happens to overlap beta register 1 of the LINC segment containing the handler, two characters of every buffer load may display badly. The user should not halt the CPU while the handler is in operation and expect his core image to be intact since the handler temporarily destroys beta register 1 but restores it upon leaving.

5) The handler does not handle tabs, vertical tabs, or rubouts correctly.

fp

ØØ8Ø-1ØØ-Ø31-ØØ

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: October 31, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Software Engineering

EXT: 2130 LOC: 12-3

SUBJ: New KL8E Teletype Handler

We have done it again! After managing to write a VR12 handler in two pages, we have now succeeded in cramming the most remarkable features into a 2-page Teletype handler. Although we will still support and distribute the old handler (for user programs which don't support 2-page handlers), I imagine that everyone will soon be using the new handler.

The new handler has the group name KL8E with entry point TTY. It is currently up to version C.

This handler may be used only to read or write ASCII files. The results are unpredictable with non-ASCII files. Data may be 7-bit, 8-bit, any parity, etc. because it only looks at the low-order 7-bits.

Some of the less common features (like support for lower-case characters) have been conditionalized out. There are about 3Ø locations free in which the user may conditionalize in any features he wants which fit. We must give him the source if he is to use non-standard features.

Below is a list of its features. Those that are conditional are marked by an asterisk.

On input:

1. It reads a line at a time. Whenever the operator types CR, it enters CR, LF into the buffer, it echoes CR, LF; and then it pads the remainder of the buffer with nulls and returns to the calling program. The characters get put into the buffer, one character per word. Thus every third character is a null as far as OS/8 is concerned.

2. RUBOUT rubs out the previous character. It echoes as either a back slash (\) or as the character rubbed out, depending on assembly parameters. RUBOUT at the beginning of the line acts as ↑U.

3. CTRL/U echoes as ↑U and erases the current line, allowing the user to retype it. (It also echoes CR, LF.) The buffer pointer is reset to the beginning of the buffer.

4. CTRL/Z echoes as ↑Z (followed by CR, LF) and signals end-of-input. The ↑Z enters the buffer and the remainder of the buffer is padded with nulls. The error return is taken with a positive AC (soft error).

5. Nulls are ignored.

*6. The altmode characters (octal 175 and 176) are converted to escapes (octal 33).

*7. Lower-case characters typed may be automatically converted to upper case.

8. CTRL/C echoes as ↑C and returns control to the keyboard monitor via location 76ØØ.

On output: (either normal output or when echoing input)

1. CTRL/C on keyboard echoes as ↑C and returns control to the keyboard monitor via location 76ØØ.

2. CTRL/O on keyboard stops further echoing. All echoing ceases (through possibly many buffer loads) until either the handler is reloaded into core or the user types a character other than ↑O on the keyboard. Not operative during input.

*3. ↑S causes the handler to stop sending to the terminal. No characters are lost and outputting resumes when a ↑Q is typed. ↑S and ↑Q do not echo. These characters are operative only upon output. On input, they are treated like any other input characters.

4. Nulls are ignored.

*5. Lower case characters may be optionally printed as upper case and flagged with an apostrophe.

*6. Tabs may be handled in one of three manners:

(a) output as actual tabs,

(b) output as actual tab followed by padding of two rubouts,

(c) output as the correct number of spaces to bring the text to the start of the next tab stop.

7. Whenever the output line reaches the end of the physical line (length set at assembly time), the handler automatically performs a carriage-return line-feed.

*8. The escape character (octal 33) prints as a dollar sign.

*9. The handler may be set to delay about 16 ms after typing any character (specified at assembly time), for example, line feed.

*10. Control characters are printed as their corresponding letter preceded by an up-arrow. Thus CTRL/K prints as ↑K.

080-100-032-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: November 1, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Soft. Engineering

EXT: 2130 LOC: 12-3

SUBJ: TM8-E HANDLER

The TM8-E handler has the same calling sequence as any other OS/8 handler, namely

```
JMS I (ENTRYPT)
ARG1                    /function word
ARG2                    /buffer address
ARG3                    /block number
<error return>
<normal return>
```

There are 8 entry points, symbolically MTAØ through MTA7, one for each unit. The handler has the internal device code of 2Ø, and it is a 2-page handler. It is non-file structured and can be used to read or write. It has version letter A. It reads and writes either 7- or 9-channel magtape with odd parity at 8ØØ bpi. (On 9-channel tapes it uses core-dump mode.) Each PDP-8 12-bit core location is written out on the tape in 2 frames, each containing 6-bits. Each PDP-8 page becomes a 256-frame record on the magtape.

Keyboard Interaction:

Typing ↑C on the keyboard while the handler is in operation causes the handler to abort and return to the OS/8 keyboard monitor via location 76ØØ. Such action is ill-advised since it leaves the magtape without an end-of-file indicator.

Special Handling for Block Ø:

As currently set up, if the handler is called to read or write block Ø, it will first perform a rewind. This feature can be patched out if desired by altering relative location 1 from a Ø to a 1. This altered handler, is said to be operating in file-mode. The original handler is said to be operating in single-file mode.

It would be possible to supply a patch which would cause the tape to perform a backspace record function instead for block Ø (for example to backspace 1 of 2 consecutive file marks at end-of-file data). If anyone thinks this is of use, let me know.

Special Handling for CLOSE:

A close operation is signalled to the handler by calling it with a function word which has a page count of Ø (bits 1-5) and which has bits 9-11 all zeroes. This is how the USR CLOSE operation calls the handler (OS/8 V3 only). This causes the handler to write two successive file marks on the tape. (Would we ever want to patch this so it writes one EOF?) Two successive EOF's is the software indication of end-of-data (EOD).

Restrictions:

In single-file mode, the user may not have more than 4Ø95 blocks because on trying to write the 4Ø96th block, the handler will think it's writing block Ø and perform a rewind. This restriction does not apply when using the handler in file-mode; but beware, some cusps, such as PIP, are suspected to behave strangely on block 4Ø96 of non-file-structured devices. Such phenomena has been hinted at in the past but skeptics have denied these allusions because of lack of evidence. Please be assured that if any definite problems are found, they will be fixed before V3 is released.

General Operation:

When the handler is used in its normal mode, single-file mode, magtapes may consist of exactly one file. It starts at the beginning of the tape and consists of consecutive records until and end-of-file mark (EOF) is reached. In this sense, a magtape is similar to one big paper tape. This is the same way that OS/8 currently treat cassettes.

Since the capacity of magtapes is so big, we have made provision for storing multiple files per tape. In such a structure, several files may exist on one magtape. They are unlabeled and are separated from each other by a single file mark. The last one is followed by two file marks. Each 'file' looks like a paper tape, that is, it is referenced in a non-file structured manner. The user must first alter his magtape handler to work in file mode. Then he must position the magtape to exactly the correct spot where he wants the read or write operation to commence. He may do this with a program of his own, using the auxiliary capabilities of the magtape handler (described below), or he may use the positioner program, CAMP, which is described in another memo. To read a file, you should first position yourself to just before the first data record of that file. To write file #1, you must rewind the tape (i.e., be at BOT). To write file #n, (n>1) you must position yourself after the (n-1)st file mark on the tape. Previous file n and all files past it then become unreadable (non-existent).

Device-Dependent Capabilities:

The TM8-E handler has several auxiliary features which may be invoked by a user program which calls the handler in a device-dependent manner. These features all rely on the contents of bits 9-11 of the function word (argument 1 of the handler call) and some require argument 3 in addition.

These features are brought to life whenever the handler is called with a page count of ∅ (bits 1-5 of the function word). Let us call bits 9-11 of the function word the special function register (SFR) for short and also we refer to bit ∅ of the function word as the direction bit. If the page count is not ∅, the contents of the SFR are ignored.

If the page count is ∅, then the SFR together with the direction bit (and possibly argument 3) determine what special function to perform, as follows:

SFR	Operation
∅	CLOSE. Write two EOF's.
1	Rewind.
2.	Space forward/reverse files. The direction to space is determined by the direction bit (∅ means space forward, 1 means space reverse). The negative of the number of file marks to space past is given by argument 3 (-1 means space past one file mark; ∅ means 4∅96 file marks). In reverse mode, the tape is left positioned at the end of a file; an error is given if BOT is encountered. In forward mode, the tape is left positioned at the beginning of a file. If EOD is reached, the handler automatically performs a backspace record to leave you between the two file marks; no error is given.
3.	Special read/write function. The direction bit (as usual) determines read or write (∅ means read, 1 means write). The specified I/O operation is performed between the user's buffer (start is specified by argument 2) and the very next magtape record. Only one record is transferred and the user's buffer must be large enough to contain it. The record length is specified by the negative of argument 3 (in words). ∅ means a record length of 4∅96.
4.	Rewind the unit and put drive off-line.
5.	Write a single EOF.
6.	Space forward/reverse records. The direction to space is determined by the direction bit (∅ means space forward, 1 means space reverse). The negative (two's complement) of the number of records to space over is given by argument 3 of the handler call. (-1 means space past one record, ∅ means 4∅96 records.) The error return is taken if either a file mark or BOT is encountered. In such cases, you would be left positioned at the beginning of a file.
7.	Unused. Reserved for future use. If specified, it currently acts as a NO-OP.

In each case, the unit affected is determined by the handler entry point.

Other Common Operations:

(a) To backspace n files, use special code 2 to pass over n+1 file marks backwards, then use special code 6 to advance (forwards) over one record (EOF) ignoring the EOF error.

(b) To advance to EOD, first perform a backspace of one record (or perform a rewind to play safe) then use special code 2 to advance over 4ø96 files in the forward direction (argument 3=ø).

Error Conditions:

On a hard error, the handler takes the error return (with a negative AC) and the AC contains the contents of the main status register, as follows:

Bit on	Meaning
ø	Error flag
1	Tape rewinding
2	BOT
3	Select error
4	Parity error (vertical, longitudinal, or CRC)
5	EOF
6	Record length incorrect
7	Data request late
8	EOT
9	File protect
1ø	Read compare error
11	Illegal function

Not all of these can occur. Refer to the engineering spec for more details.

The only soft error is EOF error (either on a read or on a space record command). In such a case, the error return is taken with a 1øø in the AC.

NOTE

To use this handler under OS/8 V2, you must make the USR patch given in the March 1973 SIS bulletin, namely,

```
.R EPIC
*SYS:</1$
R,15
O,2ø1
527ø/5266
E
```

ø8ø-1øø-ø33-øø

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: October 30, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Software Engineering

EXT: 2130 LOC: 12-3

SUBJ: PAL8 V8 Notes

Good news! The long lost source of PAL8 version 8 has been found. This is the version we have been using in-house. Our thanks go to Bob Hassinger.

This memo lists the major differences of V8 over V7. We still have to decide if we want to remove any of these features and also if we want to add any more (other than those bugs and features already noted in the V3 bug book).

Command decoder line:

1. If output file #1 is omitted, and if no /G or /L option is specified, then pass 2 is skipped.

2. If output file #3 is specified, then it has the default extension of .TM. If /C is included, then this file and output file #2 are passed along to CREF to be used as its listing file with PAL8's file #3 as CREF's input file.

Switch options:

3. /B Byte shift option.
   This option makes the operator ! a 6-bit left shift instead of an inclusive OR. (A!B equals A↑1øø+B).

4. /C Chain to SYS: CREF.SV after assembly. The second output file specified is the output file passed to CREF. The third output file is where PAL8 generates its output. If no third output file is given, DSK:CREFLS.TM is used. /C supersedes /L and /G.

5. /E Enable error messages if a link is generated. The LG error message would be generated as well as the link being flagged.

6. /O Disable automatic origining to 2ØØ after a field pseudo-op. The origin remains what it was before the FIELD pseudo-op.

7. /F Disable extra zero fill in TEXT pseudo-op. If the text in a TEXT pseudo-op contains an even number of characters, no word of zeroes will be added to the end.

Operation:

8. The alternate starting address of 56ØØ has been removed. (It used to ask questions concerning the features covered by the above switches.)

9. PAL8 now saves part of itself in block 4Ø on SYS: . This is an overlay used by it at some later time.

10. At end, it prints out number of errors and number of links generated (on Teletype if no listing file given).

Pseudo-ops:

11. EJECT If this pseudo-op is followed by a string of characters, the first 4Ø (or until a comment is reached) are used as the new header beginning with the next page. This feature was not in V7 even though the system reference manual describes it.

12. XLIST If followed by an expression, then the operation is as follows. If the value of the expression is Ø, then enable listings. If non-zero, disable listing.

Error Messages:

13. CF Chain to CREF error.

14. LG Link generated error.

Current switch conflicts between PAL8 and CREF, ABSLDR, and BITMAP:

/S PAL8 uses to mean no symbol table.
ABSLDR uses to mean allow multipe binaries per file no problem.

/E PAL8 uses to mean enable LG error messages.
CREF used to mean don't delete .TM file when through. One of these usages has to change.

I'm not sure of BITMAP conflicts since R.L. hasn't given me the revised spec yet. Even though we may not chain to BITMAP for this release, we should keep all switches separate for possible future expansion.

080-100-034-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: October 30, 1973

FROM: S. Spilman

DEPT: Small Systems Softw. Eng.

EXT: 2073 LOC: 12-3

SUBJ: Core Image Files under OS/8 BASIC V3

Core image files have been implemented under OS/8 BASIC V3. To create a core image file from a BASIC language source, type:

```
.R BCOMP
*DEV:PROG.BA/K
```

where PROG.BA is the source. The K switch indicates that a core image file is to be created. The following additional switch/option designations apply:

/N If the resulting core image will never be executed on a TD8E system. This saves 400 words of memory but reduces configuration independence.

/B If a copy of the run-time system should be loaded into the core image. This increases the size of the core image file by 10-50% (exactly 15 blocks) but eliminates the need for a file access to read in BRTS at run-time.

=n Where n is an octal digit in the range 1≤n≤m and m is the highest memory field available on the host machine. N will be used as the highest memory field on the target machine. This may reduce configuration independence, since the resulting core image will not load correctly on a machine with fewer than n+1 memory banks. If n is not specified, n= 1 is assumed. If n is specified larger than m, n=m is assumed.

In the absence of error conditions, the compiler post-processor (BLOAD) will exit to OS/8. At this time, type:

```
.SA DEV PROG.EX
```

to create an executable core image. Additional arguments to the save command must not be specified. The core image is executed by typing:

```
.R PROG
```

or

```
.RUN DEV PROG
```

as appropriate. Any errors flagged during compilation will prevent creation of an executable core image file.

In contrast to normal BASIC program execution, which requires a minimum of 6 file access operations, core image file execution requires no more than two file accesses; one to read the core image file and one to read BRTS if /B was not specified. Compiler/loader overhead is also eliminated, so that the reduction in execution time should be very significant, especially on DECtape systems.

Execution is terminated via the monitor, except in the case of BASIC CHAIN statements which exit to the compiler; at present it is not possible to chain to a BASIC core image file. I hope to implement this feature shortly.

Suggestions, modifications, enhancements, etc. will be very welcome during the next week or so.

fp

080-100-035-00

TO: OS/8 V3 LIST
N. Dombrowik
R. Gafford
B. Lavin
T. Mitchell
K. Richer
J. Willis

DATE: October 31, 1973

FROM: Judi Hall

DEPT: Software Documentation

EXT: 2436

SUBJ: OS/8 HANDBOOK DOCUMENTATION PLAN

I. OS/8 HANDBOOK

The OS/8 HANDBOOK will contain all the standard user documentation for OS/8 that currently exists in several manuals. These manuals will be updated for OS/8 Version 3 and any errors will be corrected. Necessary changes will be made to make the style consistent with the OS/8 SYSTEM REFERENCE MANUAL.

Section V of this plan contains a detailed outline of the material to be included in the handbook. The existing material to be included in the handbook consists of the following:

OS/8 SYSTEM REFERENCE MANUAL (formerly Chapter 9 of INTRODUCTION TO PROGRAMMING)
BASIC USERS MANUAL
BATCH USERS MANUAL
BITMAP USERS MANUAL
8K FORTRAN AND SABR ASSEMBLER
EPIC USERS MANUAL
FLAP USERS MANUAL
FORTRAN IV USERS MANUAL
LARGE FUNCTIONS FOR OS/8 BASIC
PAL8 ASSEMBLER
PIP8 USERS MANUAL
SRCCOM USERS MANUAL
TECO USERS MANUAL

In addition, new material will be added to include all features of OS/8 Version 3. This material includes changes and additions to existing programs and the following new programs:

BOOT
CCL (Concise Command Language)
DIRECT
FOTP
PIP10
RESORC

The handbook will written, edited, and proofed by Judi Hall and Bob Emma. The handbook will be divided into four main sections: OS/8 Fundamentals, Utility Programs, Assemblers, and Languages. OS/8

DOCUMENTATION SCHEDULE

PROJECT	Edited Draft*	Progmr Review		Typeset First Galley		Typeset Final Galley		Paste-up*	To Software Evaluation	Printing	
		IN	OUT	IN	OUT	IN	OUT			IN	OUT
INTRO & CHAP. 1 (Hall)	11/5	11/5	11/8	11/8	11/15	11/16	11/30	12/6	12/7	1/14	2/30
CHAP. 2						11/27	12/6	12/14	12/7		
BATCH	11/9	11/9	11/14	11/14	11/21						
BITMAP	10/24	10/24	11/1	11/5	11/12						
CREF	10/31	10/31	11/5	11/5	11/12						
EPIC (Emma)	11/9	11/9	11/14	11/14	11/21						
FLAP	11/9	11/9	11/14	11/14	11/21						
GRCCOM	11/9	11/9	11/14	11/14	11/21						
HECO	10/24	10/24	11/1	11/5	11/21						
IBOOT	11/5	11/5	11/7	11/7	11/21						
JCL	11/8	11/8	11/12	11/13	11/21						
KDIRECT	11/19	11/19	11/21	11/21	11/26						
LOTP (Hall)	11/13	11/13	11/16	11/16	11/21						
MIPC	10/31	11/1	11/6	11/6	11/12						
NIP10	10/31	11/1	11/6	11/6	11/12						
OSORC	11/13	11/13	11/16	11/19	11/26						
CHAP. 3 PAL8 (Emma)	11/29	11/30	12/4	12/4	12/12	12/13	12/21	12/28	12/7		
CHAP. 4 SABR (Emma)	12/10	12/10	12/13	12/14	12/21	12/28	1/4	1/10	12/7		
CHAP. 5 BASIC (Emma)	11/19	11/20	11/27	11/27	12/10	12/17	12/21	1/2	12/7		

*Completion dates.

DOCUMENTATION SCHEDULE

PROJECT	Edited Draft*	Progmr Review		Typeset First Galley		Typeset Final Galley		Paste-up*	To Software Evaluation	Printing	
		IN	OUT	IN	OUT	IN	OUT			IN	OUT
CHAP. 6 (Hall) FORTRAN II	11/19	11/20	11/27	11/27	12/7	12/12	12/17	12/21	12/7	1/14	2/30
CHAP. 7 (Hall) FORTRAN IV	11/27	11/27	12/3	12/4	12/10	12/13	12/18	1/3	12/7		
APPENDICES (Hall & Emma)	12/14	12/14	12/19	12/20	1/2	1/2	1/10	1/14	NA		
INDEX & TABLE OF CONTENTS (Emma & Hall)	NA	NA		1/7	1/10	11/1	11/13	11/14	NA		

*Completion dates.

fundamentals will include all the standard programs necessary to run a minimum OS/8 system. All programs other than assemblers and languages will be included in the Utility Programs section.

The handbook will be approximately 800 pages long; 40,000 copies will be printed. The handbook will be available from the Software Distribution Center the week of February 30, 1974.

II. OS/8 SOFTWARE SUPPORT MANUAL

The existing OS/8 SOFTWARE SUPPORT MANUAL will be updated for Version 3 and expanded to include internal descriptions that are now in user manuals. The software support manual will be written by development programmers and edited and proofed by the writers.

III. TIME SCHEDULE

The attached is an estimated schedule of the time required to complete the OS/8 HANDBOOK. This schedule reflects the time necessary to make technical corrections and format changes. It does not allocate time to perform major rewrites of any section, i.e., the schedule assumes that most material is already written in near-final form.

The writer responsible for each section is indicated in parentheses. All questions and comments concerning the handbook should be directed to Judi Hall.

IV. DOCUMENTATION PLAN REVIEW

A review of this documentation plan will be held on November 5 at 1:00 p.m. In Larry Portner's conference room, 12-2. All comments concerning this documentation plan should be directed to Judi Hall, 12-3, by November 5 or discussed at the review meeting.

V. OS/8 HANDBOOK OUTLINE

The following is an outline of the OS/8 HANDBOOK. Page approximations shown are for the section as a whole.

CHAPTER TWO UTILITY PROGRAMS

080-100-036-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: November 7, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: POSITIONER PROGRAM (CAMP)

The program, CAMP (Cassette and Magtape Positioner), is used to position cassettes, magtapes, and certain other devices.

It can be run from the OS/8 system device by the command

```
.R CAMP
```

and it responds with a number sign (#) to indicate that it is ready to accept a command. At this point, you enter your command terminated by either a carriage-return or an altmode (escape). You have all the usual OS/8 editing capabilities (LF, Rubout, ↑U). The command is then executed and control returns to CAMP which then prints another # to accept another command (unless the previous command line ended with an altmode, in which case control now returns to the OS/8 keyboard monitor).

Command lines begin with a keyword. This consists of a sequence of two or more letters or digits, of which only the first two characters are significant. The permissible keywords are:

BACKSPACE
EOF
HELP
REWIND
SKIP
UNLOAD
VERSION

Some of the keywords take arguments in which case the keyword is followed by one or more spaces and then the arguments. These are described below.

Whenever a device (dev:) is mentioned, we refer to a legal OS/8 device handler. This handler must be configured into your system. You cannot, for example, position magtapes unless you have a magtape handler.

In the case of cassettes and magtapes, the handlers are used to perform the positioning. Thus, the V2 cassette handler is unacceptable and will produce unexpected results if used with CAMP. Similarly, user-coded handlers for these devices will only work if their calling conventions concerning the device-dependent bits in the function word agree with the standards established by the DEC V3 handlers.

Commands

BACKSPACE dev: nnnn [RECORDS / FILES]

dev: may be either a cassette handler or a magtape handler

nnnn is an unsigned decimal number representing the number of records (or files) to backspace. It may be optionally preceded and followed by spaces. If omitted, nnnn=1 is assumed. This number is followed by a keyword beginning with either an R or an F. If omitted, F is assumed.

R Backspace the number of records specified. If nnnn=0, nnnn=1 is assumed. If a file mark is read before the proper number of records have been backspaced over, the warning message

% CAN'T - AT BOF

is typed and the program then does a forward record to leave the device positioned at the beginning of the file (just before a data record).

F Backspace the number of files specified. This does not count the current file. That is, the command to backspace 3 files skips backwards over 4 file marks and then skips forward 1 record to get over the file mark and leave you positioned at the beginning of a file. This command always leaves you at the beginning of a file (just before a data record). If nnnn=0, this command backspaces to the beginning of the current file.

EOF dev:

dev: may be either a cassette or a magtape.

This command writes a single file mark (file gap).

HELP This command types a short help-frame on the Teletype which reminds the user of the syntax of CAMP commands.

REWIND dev:

If dev: is either a cassette, magtape, TC08 DECtape or LINCtape, then this command issues a rewind command to the appropriate controller and then returns to CAMP while the device rewinds.

If dev: is any other OS/8 file-structured handler, the program uses it to read block 0 of the device.

UNLOAD dev:

If dev: is a magtape, this command issues a 'rewind and turn off-line' command to the magtape controller and then returns to CAMP while the magtape rewinds. To re-use, the magtape controller must be manually put back on-line.

If dev: is a TC08 or TD8E DECtape or LINCtape, this command issues a rewind command to the appropriate controller and then selects a different unit. Control returns to CAMP while the tape unloads off its reel. This drive then becomes unusable until either an I/O PRESET pulse is generated or some other legal command is issued to the controller.

VERSION This command types CAMP's version number on the Teletype.

SKIP dev: { nnnn [RECORDS] / EOD [FILES] }

dev: may be either a cassette or magtape.

nnnn is an unsigned decimal number representing the number of files (or records) to skip over. It may be optionally preceded and followed by spaces. If neither nnn or EOD is specified, nnnn=1 is assumed. EOD (or any keyword beginning with E) may replace the nnnn. If nnnn is specified, it may be followed by a keyword beginning with either an R or an F. If omitted, F is assumed.

R Skip (advance) over the number of records specified. If nnnn=0, nnnn=1 is assumed. If a file mark is read before the proper number of records have been skipped, the warning message.

% CAN'T - AT EOF

is typed and the program then does a backspace record to leave the device positioned at the end of the file (just after the last data record but before the file mark). This command is not implemented for cassettes.

F Skip the number of files specified. This includes the current file. Thus nnnn=1 means skip to the start of the next file. If nnnn>0, this command leaves you at the beginning of a file (just after a file mark but before any data records). If nnnn=0, this command advances to the end of the current file (before a file

mark but after all data records). If the tape is not initially at EOD and if EOD is encountered before the specified number of files have been skipped, then the warning message,

% CAN'T - AT EOD

is typed and the tape is then positioned at EOD. In the case of magtape, EOD is a point between two file marks. In the case of cassettes, EOD is a point in the middle of a double size file gap. If a device is at or past EOD, this command produces meaningless results.

EOD Advance to EOD. If the device is already past EOD, it must first be rewound before this command is issued.

Remarks:

1. If dev: is not followed by any arguments, then the colon is optional.

2. ↑C may be typed at any time to return to the OS/8 keyboard monitor.

3. ↑O may be typed to surpress teletype output until the next #.

4. nnnn must be in the range 0 - 4095.

Examples:

```
REWIND  DTA0:
REWIND  MTA4
UNLOAD  FOO:
EOF  CSA3:
SKIP  MTA0:2 RECORDS
SK  CSA7:  6 F
BACKSPACE  MTA2
BAK  MTA4:  17  RECS
SKP  MTA1:E
```

If CCL is enabled, any CAMP command (other than HELP and VERSION) may be issued at monitor level and CAMP will be chained to with the typed command terminated by an altmode.

Error messages:

? SYNTAX ERROR

? NUMBER TOO BIG /nnnn>4095

? CAN'T - DEVICE DOESN'T EXIST

? CAN'T FOR THIS DEVICE

Operation doesn't make sense for device specified, e.g., SKIP LTA2:3 or REWIND LPT:

? CAN'T I/O ERROR

This is followed by a brief explanation of the error.

? CAN'T - DEVICE IS READ-ONLY

? CAN'T - DEVICE IS WRITE-ONLY

% OPERATION NOT YET IMPLEMENTED

? CAN'T - AT BOT

% CAN'T - AT EOF

% CAN'T - AT BOF

% CAN'T - AT EOD

080-100-037-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: November 14, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT : 2130 LOC: 12-3

SUBJ: Changes to Command Decoder

This memo documents the principal changes made to the V3 command decoder.

1. Lots of bugs were fixed. These had mostly to do with syntax checking; for example A:B:C now gives an error. *'s are now handled correctly in special mode. The altmode bit is not affected by a 15-bit equals option. And so on.

2. Added support of "?" in special mode.

3. Allowed special mode to work under BATCH.

4. Removed DCC code.

5. ↑U, runbout past begin of line, and LF now all reprint "*" on the next line.

6. A version # was added which resides in location Ø (currently = 3). This number is not readily available to the user.

7. Output device handlers are not automatically loaded.

fp

080-100-038-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: November 14, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT : 2130 LOC: 12-3

SUBJ: Changes to ODT

1. All DCC code has been removed.

2. The core size routine has been replaced by the more trustworthy OS/8 standard core size routine.

3. Support for software settable core size has been included.

4. An internal version number has been added at source origin 12ØØ. This number is not readily available to the user. It is currently 3.

5. During a breakpoint, if the breakpoint location is modified, this modified value will not be used after a "GO" command.

fp

080-100-039-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: November 8, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT : 2130 LOC: 12-3

SUBJ: Core Size Routines under V3

All OS/8 V3 cusps must use the standard core size determining routine which is given in the old PS/8 Software Support Manual.

Furthermore, before calling this routine, they must examine bits 6-8 of location Ø7777. If this field is non-zero, then its contents give the field number of the highest core bank available to OS/8.

This location may be set by a user if he wishes to protect high core from OS/8.

fp

080-100-040-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: November 16, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT : 2130 LOC: 12-3

SUBJ: OS/8 RUNOFF

Shawn Spilman has written us a RUNOFF program similar to the PDP-1Ø's.

THIS MEMO DESCRIBES THE PRELIMINARY VERSION OF OS/8 RUNOFF AND INDICATES WHAT PROGRESS HAS BEEN MADE IN ADAPTING PDP-10 RUNOFF TO EXECUTE UNDER OS/8. IT ALSO LISTS THE DIFFERENCES BETWEEN THE TWO VERSIONS OF RUNOFF, MOST OF WHICH SHOULD DISAPPEAR OVER THE NEXT FEW MONTHS, AND DESCRIBES A FEW ADDITIONAL FEATURES THAT ARE AVAILABLE WITH THE OS/8 VERSION.

OS/8 RUNOFF IS CALLED FROM THE KEYBOARD MONITOR BY TYPING:

R RUNOFF

(IN RESPONSE TO THE DOT GENERATED BY THE KEYBOARD MONITOR), RUNOFF THEN LOADS THE OS/8 COMMAND DECODER TO ACCEPT ONE FILE/OPTION COMMAND THAT DESIGNATES ONE INPUT FILE AND ONE OUTPUT FILE. THE INPUT FILE IS ASSUMED TO BE AN ASCII FILE IN RUNOFF INPUT FORMAT. THE DEFAULT EXTENSION FOR INPUT FILES IS ".RO". THERE IS NO DEFAULT INPUT FILE NAME. THE OUTPUT FILE WILL BE A FORMATTED ASCII FILE THAT MAY BE ROUTED TO ANY SUPPORTED DEVICE. THE DEFAULT OUTPUT DEVICE IS LPT:. THERE IS NO DEFAULT OUTPUT FILE EXTENSION.

A RUNOFF COMMAND LINE BEGINS WITH A DOT AT THE LEFT MARGIN. THE DOT IS FOLLOWED BY ONE OF THE COMMANDS LISTED IN TABLE 1, DELIMITED BY ONE OR MORE SPACES, AND ANY NUMBER OF ARGUMENTS MAY FOLLOW THE COMMAND. ARGUMENTS ARE DECIMAL NUMBERS (MODULO 4096) SEPARATED BY COMMAS AND OPTIONAL SPACES. A CARRIAGE RETURN TERMINATES THE COMMAND LINE.

TABLE 1: OS/8 (PRELIMINARY) RUNOFF COMMANDS

COMMAND	FUNCTION
BREAK	TERMINATES THE CURRENT LINE WITHOUT JUSTIFICATION AND RESUMES PROCESSING AT THE BEGINNING OF THE NEXT LINE.
BLANK N	EXECUTES A BREAK AND INSERTS N BLANK LINES, THEN RESUMES PROCESSING AT THE BEGINNING OF THE NEXT LINE. DEFAULT N=1.
SKIP N	EXECUTES A BREAK AND INSERTS S TIMES N BLANK LINES, THEN RESUMES PROCESSING AT THE BEGINNING OF THE NEXT LINE. DEFAULT N=1.

PARAGRAPH N	EXECUTES A BLANK AND INDENTS THE NEXT LINE N SPACES. DEFAULT N=5.
INDENT N	EXECUTES A BREAK AND INDENTS THE NEXT LINE N SPACES. MOST OFTEN USED IN TABULAR MODE WITH N A NEGATIVE NUMBER. NO DEFAULT N.
SPACING N	SETS LINE SPACING TO N (WHERE N=1 IMPLIES SINGLE SPACING, N=2 IMPLIES DOUBLE, ETC.). DEFAULT N = ORIGINAL SETTING = 1.
LEFT MARGIN N	SETS LEFT MARGIN TO N. IF N IS NEGATIVE, N=0 IS USED. THE ORIGINAL SETTING IS 10. THERE IS NO DEFAULT N.
RIGHT MARGIN N	SETS RIGHT MARGIN TO N. IF N IS TOO LARGE, N=120 IS USED. THE ORIGINAL SETTING IS 70. THERE IS NO DEFAULT N.
PAGE N	EXECUTES A BREAK, OUTPUTS A FORM FEED, PRINTS THE CURRENT HEADING, AND RESUMES PROCESSING. IF N IS SPECIFIED, THE CURRENT PAGE NUMBER IS SET TO N. THERE IS NO DEFAULT N.
CENTER N	EXECUTES A BREAK AND CENTERS THE NEXT N LINES BETWEEN THE CURRENT MARGINS. OTHER CENTERING OPTIONS WILL BE IMPLEMENTED LATER. IF N IS NOT SPECIFIED, N=1 IS ASSUMED.
CENTRE	SAME AS CENTER. INCLUDED FOR THE BENEFIT OF OUR FRIENDS ACROSS THE WATER.
TAB STOPS N1,N2...	SETS THE TAB STOPS TO N1, N2, ETC. UP TO A MAXIMUM OF 7 TAB STOPS. IF NO ARGUMENTS ARE SPECIFIED, NO TAB STOPS ARE SET.
FILL	ENABLES FILL MODE. THIS CAUSES EACH LINE TO BE FILLED WITH AS MANY WORDS AS WILL FIT BETWEEN THE CURRENT MARGINS.
JUSTIFY	ENABLES JUSTIFY MODE. THIS CAUSES EVERY LINE NOT PRECEEDING A BREAK COMMAND TO BE RIGHT JUSTIFIED.
NOFILL	DISABLED FILL MODE. EVERY LINE WILL BE TERMINATED BY AN INPUT CARRIAGE RETURN AND JUSTIFIED IN THE USUAL MANNER.
NOJUSTIFY	DISABLES JUSTIFY MODE. EVERY LINE WILL BE FILLED IN THE USUAL MANNER, BUT A RAGGED RIGHT MARGIN WILL BE USED.
NUMBER N	ENABLES PAGE NUMBERING BEGINNING WITH PAGE N AS THE CURRENT PAGE. IF N IS NOT SPECIFIED, PAGE NUMBERS CONTINUE IN SEQUENCE (SEE BELOW).

NONUMBER — DISABLES PAGE NUMBERING. EACH PAGE ADVANCE CONTINUES TO INCREMENT THE INTERNAL PAGE COUNTER, SO THAT NUMBERING MAY BE RESUMED LATER.

PAPERSIZE N,M — SETS THE PAPER WIDTH TO N CHARACTERS AND THE PAPER LENGTH TO M LINES. THESE PARAMETERS WILL BE USED TO IMPLEMENT OPTIONAL CENTERING FUNCTIONS LATER.

TITLE (TEXT) — LOADS THE REMAINDER OF THE COMMAND LINE INTO THE TITLE BUFFER FOR USE IN PRINTING HEADINGS. IF THE TITLE BUFFER IS EMPTY, ONLY A PAGE NUMBER IS PRINTED (AS ON PAGE 1 OF THIS DEMO).

THE FOLLOWING PDP-10 COMMANDS ARE NOT IMPLEMENTED UNDER OS/8 RUNOFF AT THIS TIME:

FIGURE
TEST PAGE
SUBTITLE
FOOTNOTE
INDEX
PRINT INDEX

THE FIRST TWO WILL BE IMPLEMENTED SHORTLY. THE REMAININD FOUR COMMANDS ARE NOT VERY USEFUL IN PRACTICE; I HAVE LEFT ROOM FOR THE NECESSARY CODE BUT I DO NOT PLAN TO IMPLEMENT THESE COMMANDS UNTIL MUCH LATER.

THE PRESENT VERSION OF OS/8 RUNOFF PROCESSES THE FULL ASCII SET BUT DOES NOT EXECUTE CASE CONVERSION. THE CASE CONVERSION ROUTINES HAVE BEENWRITTEN AND WILL BE ADDED AS SOON AS A LOWER-CASE OUTPUT DEVICE IS AVAILABLE FOR DEBUGGING.

CORE ECONOMY WAS AN IMPORTANT FACTOR IN THIS PROJECT TO DATE. THE PRESENT VERSION OF RUNOFF OCCUPIES LESS THAN 2K IN FIELD 1, INCLUDING ERROR HANDLERS. THE REMAINDER OF FIELD 1 IS AVAILABLE FOR EXPANSION, INCLUDING PLENTY OF SPACE ON PAGE ZERO. ALL OF FIELD 0 IS USED FOR DEVICE HANDLERS AND BUFFERS.

fp

080-100-041-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: November 27, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT : 2130 LOC: 12-3

SUBJ: Preliminary Work on BUILD

A preliminary version of BUILD will shortly go on the in-house distribution tape in order to support the new monitor. (The old BUILD had a bug and will no longer run under V3). Here are the things I've currently done to BUILD:

1. Added VE version number command
2. Fixed problem with Z's and 9's
3. Fixed length of DF32
4. Fixed RK8 bootstrap problem
5. On LOAD, null extension first searches for .BN
6. CR to $ gives no errors
7. Fixed bug with building ROM sys
8. Rubouts to BOL give $ as will as ↑U and LF
9. LOAD assumes default device is DSK:
10. No dots are printed if no extension
11. Revamped ↑O stuff
12. Allowed parity ↑C
13. Added EXAMINE command
14. Fixed bug reaccessing USR tables incorrectly
15. 'OVERFLOW' error message changed to 'BAD ORIGIN'
16. Fixed several bugs involving parsing of names
17. Increased number of allowable entry PTS per handler to 16
18. Names in PRINT now line up
19. Cleaned up code concerning move routine, number checker, and symbol print
20. Allow multiple inserts, deletes, replaces per line
21. Allow specifying number of platters on an insert
22. Added general support of 2-page system handlers
23. UNLOAD command automatically deletes any active handlers
24. Added DSK command
25. Added DCB command
26. Added QL (quick list) command
27. Use extra core if available

080-100-042-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: November 27, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT : 2130 LOC: 12-3

SUBJ: Remarks on Version Numbers

All users who wish to report bugs must do so on the proper form. They must include the following information:

a) Type of machine they're running on (PDP-8/L, PDP-12, etc.)

b) Amount of core

c) Version number of monitor, program under question and all related programs. Also what patches has user installed.

To answer (c), he must know how to find version number of all cusps.

Here is how for some of them:

KBM	Type .VER to KBM
CCL	Type .VER to KBM
FOTP	Type /W to FOTP's CD
DIRECT	?
EDIT	Type # to EDIT
PIP	Type /V initially to PIP's CD
TECO	Type ↑V= to TECO
PAL8	Look on heading of listing
CREF	Look at end of listing
SRCCOM	Look at top of output
BITMAP	Look at top of output
EPIC	?
ABSLDR	Internal only
ODT	Internal only
CD	Internal only
CAMP	Type VE to CAMP
BUILD	Type VE to BUILD
CCL overlay	comes out on error message if you use wrong version of CCL
PIP10	?
RUNOFF	?
SABR	?
FORT	?
RALF	?
BASIC	?
RESORC	Type /V to RESORC's CD

080-100-044-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: November 27, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT : 2130 LOC: 12-3

SUBJ: PAL8 Additions

Richie has optimized PAL8 V8 and added the following features:

1. /W means don't remember literals when origining off a page.
2. /J means don't list lines containing code conditionalized out.
3. RELOC command now in (ask R.L. for details).
4. 8/E mnemonics in
5. DF32, TC08 mnemonics out
6. Heading is larger
7. Origin back to begin of page bug fixed cleanly
8. forward = stuff works now
9. "XLIST" never lists
10. End-of-file treated as "$"
11. Form feed at end bug fixed
12. ↑O and ↑C now work on ERRORS DETECTED message
13. Above message is no longer an overlay
14. I and FIXMRI interaction better
15. FIXMRI syntax errors now caught
16. Fixed bug re PAL8BN.TM a block short

080-100-045-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: November 27, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT : 2130 LOC: 12-3

SUBJ: OS/8 BATCH

Remarks:

a) BATCH now accepts input files from handlers coresident with SYS.

b) Long dialog mode has gone away completely. May it rest in peace.

c) /N need not be specified if no lineprinter is physically present.

d) BATCH uses software core size.

e) A single $ all alone on a line is now ignored. This is used to terminate a BAT: stream.

080-100-046-00

digital INTEROFFICE MEMORANDUM

TO: Jack O'Connor
OS/8 V3 List

DATE: November 21, 1973

FROM: Mark Rosenthal

DEPT: Small Systems Softw. Eng.

EXT : 3372 LOC: 12-3

SUBJ: Mark Sense Batch

Mark sense batch will translate mark sense cards and chain to BATCH passing the pre-processed job stream. There are two kinds of mark sense cards. Each has 4 distinct fields. The first is column 1. This contains batch control key words. The next field includes columns 7 and 8 and the top two rows of columns 2-6 which contain language keywords (either Fortran or Basic). Rows 0-9 of columns 2-6 contain the line number. Finally columns 9-40 contain the text. There may be only one keyword marked on a card (either batch control or language). If a language keyword is marked, the line number may also be marked. If neither a keyword nor line number is marked, column 9 appears at the beginning of the line. If an error is made in marking either the line number or text, mark rows 7, 8, and 9 of the erroneous column and it will be ignored. All spaces on the card are kept except trailing spaces. A column in the text field with rows 8 and 9 marked is a continuation column. All columns on the current card past a continuation column are ignored. The line continues on the next card.

The batch control cards are:

```
$BAS
$DATA
$DECK
$END
$EOD
$FOR
$JOB
$LOAD
$MSG
```

$JOB, $END, $MSG are standard control cards for the program BATCH.

In the following, the symbol "n" will represent an OS-8 file name in standard form:

```
DEV:NAME.EX
```

where DEV: and .EX are optional.

If the card can specify options, the option name is preceded by a "/". If more than one file name is specified, the names are separated by a ",". In some cases, an option will specify an associated file name. If it does, the option name precedes the file name and is separated from it by a "=". If the card is followed by an input stream, this is terminated by the next batch control card.

$DECK n

This card copies the cards which follow it into an OS-8 file with the name "n". It also lists the cards.

Options:

/BAS - the cards which follow are Basic cards (default)

/FOR - the cards which follow are Fortran cards

/NOLIST - do not list the cards

$BAS (n)

This card prepares the cards which follow for execution as a Basic program. A name is optional and if specified, input comes from that OS8 file rather than cards. $BAS also lists the program. The program is executed when $DATA is encountered.

Options:

/NOLIST - Do not list the Basic program.

$DATA following $BAS

This card executes the Basic program which was prepared by the $BAS card. Data cards, if any, follow this card.

FORTRAN IV

$FOR n

This card will compile a FORTRAN program from the cards which follow, and produce a relocatable binary file with the name "n". The program is listed.

Options:

/SRC=s - If specified, input comes from the OS-8 file "s" rather than cards.

/NOLIST - Do not list the program.

/LIST=l - Send the listing to OS-8 file "l".

$LOAD n_1, n_2, n_3, ...

This card loads the relocatible modules n_1, n_2,... generated by FORTRAN and prepares them for execution with $DATA. If a name is preceded by "L=" a new level is opened. If it is preceded by "O=" a new overlay is opened. For further information see OS-8 FORTRAN IV Users Manual - Chapter 4.

Options:

/IMAGE=i - the loader image file produced will be named "i".

/LIST(=l) - the loader map will be listed, or put into file "l" if specified.

/LIB=li - use file "li" as the library of system subroutines.

$DATA

following a $LOAD

Run the program just loaded. Data follows this card. The $DATA card may contain file specifications as follows:

Each file specification is preceded by a "/".

The first character is a digit from 0-9 specifying a FORTRAN logical unit number. This may be followed by a C to reverse normal carriage control processing. See OS-8 FORTRAN IV User's Manual - Chapter 5. It also may be followed by a N to indicate that the file does not already exist. The next character must be "=". Finally there is either an OS-8 filename or a digit from 0-9 which is a FORTRAN logical unit number. The name or number is assigned to the first number for the duration of program execution.

E.g.,

```
$DATA /5=FILE.FI/6C=3
```

The program will access the OS-8 file "FILE.FI" through unit number 5.

The program will access the line printer without carriage control through unit 6. It is normally accessed through unit 3.

If $DATA follows a $FOR card, it also does a $LOAD, and and any of the $LOAD options are legal.

FORTRAN II

The $FOR, $LOAD, $DATA cards are identical to those for FORTRAN IV with the following exceptions.

$FOR

Option:

/MAP - Produce a map of the program on the listing device.

$LOAD

There are no overlays or levels.

$DATA - there is no assignment of logical units

$EOD - End of data

080-100-047-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: December 3, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: What you've always wanted to know about the new BUILD but were afraid to ask

BUILD can run under ØS/8 or standalone. It is run standalone when building an OS/8 system from scratch (i.e., from paper tapes or cassettes).

BUILD contains OS/8 handlers (both system handlers and non-system handlers) which the user might wish to include in his new OS/8 system. Each such handler has a unique group name and has one or more permanent device names. These permanent device names correspond to entry points in the handler. Some handlers (such as the TM8E magtape handler) have multiple entry points (such as MTAØ, MTA1,...MTA7) corresponding to different physical units on one controller. Such entry point names are said to be coresident with each other. If two coresident handlers are configured into a given system, then whenever one is loaded into core, the other is automatically brought into core at the same time.

When BUILD is ready to accept a command, it types a dollar sign ($). At this point, you have the usual OS/8 editing capabilities, i.e., rubout, CTRL/U, and line feed, until the input line is terminated by typing either carriage return or altmode.

BUILD Commands

Only the first two characters in the keyword of a BUILD command are significant.

$PRINT

This command prints out on the Teletype the list of all the handlers currently in BUILD which are available for possible insertion into the user's system.

Example of PRINT output:

```
RFØ8:    SYS
RK8E:   *SYS      RKAØ      *RKBØ
KL8E:   *TTY
PT8E:    PTR     *PTP
LPSV:    LPT
```

This shows 5 groups of handlers. The group names are shown first in each line followed by a colon. Following each are the list of permanent names (entry points) available with each group.

If one of the permanent names in a group is "SYS", then this handler is a system handler. This means, that it will always be resident in page Ø76ØØ of any system it is configured into.

All systems must contain precisely one system handler. Some system handlers have other handlers coresident with them.

Starred permanent names indicate those which are currently active, i.e., are chosen to be inserted in the user's new system. The example shows that the user has chosen the entry point RK8E:SYS to be his system device. This is the RK8E system handler. The handler RK8E:RKBØ is also marked as being active. It is coresident with his proposed new system device. Note that we refer to a permanent name together with its group name because there may be 2 or more groups which happen to have the same entry point name. For example, the groups LPSV (The LPØ8/LS8E/LV8E line printer handler) and L645 (The annalex line printer handler) both have entry points with the name LPT. However, when there is no confusion, we may only refer to the entry point name. Two entry points with the same name should never be made active at the same time.
Handler entry points can be made active by the INSERT command; they can be deactivated by the DELETE command. See below.

After typing the list of available handlers, the PRINT command might also type out some additional information. If the user has specified his DSK device with the DSK command, it will be printed, example:

```
DSK = RK8E:RKBØ
```

If the user has specified a core restriction for his new system with the CORE command, it will be printed, example:

```
CORE = 2
```

This example shows that the user requests field 2 to be the highest core field available to the OS/8 system.

The output from PRINT can be aborted at any time by typing CTRL/O on the keyboard.

$QLIST

QLIST stands for Quick List. It is the same as the PRINT command except that only active permanent names are typed. No * is shown and group names are not printed except in the case of the system device.

Example:

```
PTR DTA3 RKØ8:SYS  LPT2  DTA4
```

$INSERT

This command allows you to make particular entry points active. No error results if it is already active. Although there is a limit to the number of devices which may be active in a given system, you may go over this limit temporarily as long as you deactivate handlers later so that you are under the limit by the type you type the BOOT command.

Basic form:

```
INSERT  groupname:permanentname
```

Examples:

```
INSERT KL8E:TTY
INSERT TCØ8:SYS
```

For compatibility with earlier versions of BUILD, a comma is acceptable instead of the colon.

Other forms:

If no permanent name is specified (and no :), then the first permanent name in the group is assumed. Example:

```
INSERT LPSV
```

Several handler names all belonging to the same group can be inserted at the same time by separating their names by commas. Example:

```
INSERT  TC:DTAØ, DTA3, DTA7
```

Several permanent names which are all 4-characters long and which differ only in the last character, can be simultaneously inserted with the hyphen construction provided the last characters from a sequence of consecutive ASCII characters.

Example:

```
INSERT  TC:DTA2-5
```

is equivalent to

```
INSERT TC:DTA2, DTA3, DTA4, DTA5
```

and

```
INSERT RKØ5: DSKA-C
```

is equivalent to

```
INSERT RK05:DSKA, DSKB, DSKC
```

If the permanent name specified is not part of the groupname specified, or if the groupname doesn't exist, the error message

```
name NOT FOUND
```

will be printed.

Any permanent name being inserted can be followed by a construction of the form = n where n is an octal number between 1 and 7. This represents the number of platters of that device which are available (only applies to certain disks). If no such option is specified, =1 is assumed. Example:

```
INSERT  RF:RF=2
```

inserts a non-system RFØ8 handler into a system which has a 2-platter RFØ8 disk.

The maximum number of platters available to various devices is given below:

Device	Maximum number of Platters
RK8	1
RK8E	1
RFØ8	4
DF32	4
TCØ8	1
TD8E	1
LINCtape	1

If more platters are requested than is possible, the error message

?PLAT

is printed.

$SYS

The SYS command is identical to the INSERT command except it is used only to insert handlers which are either system handlers or coresident with system handlers. If another type of handler is so inserted, the warning message:

?SYS

is typed and no further handlers in that command line will be inserted.

Thus, the commands

SYS RK8

and

INSERT RK8:SYS

are equivalent. Similarly, the commands

SYS RF∅8=3 and INSERT RF∅8:SYS=3

are equivalent. The SYS command is included only for compatibility with earlier versions of BUILD.

$DELETE or $DEACTIVATE

The DELETE command is used to deactivate handlers that are already active,

Basic Form:

DELETE activename

Example:

DELETE LPT

The name must already be active, otherwise the error message

```
name NOT FOUND
```

will be printed. More than one permanent active name can be deleted at one time by separating them by commas, and/or by specifying a range via the hyphen construction. Example:

```
DELETE TTY, LPT, DTA3-5, PTP
```

$REPLACE

This command combines the capabilities of the DELETE and INSERT commands into one command. It is followed by a legal argument to the DELETE command followed by an equal sign followed by a legal argument to the INSERT command.

Example:

```
REPLACE  LPT=L645:LPT2
```

is equivalent to the two commands

```
DELETE LPT
INSERT L645:LPT2
```

Another example:

```
REPLACE LTA3-5, LTA7=TC:DTA3-5, DTA7
```

$DSK

The DSK command is used to specify which handler is to be used as 'DSK'. It has two forms:

```
DSK = groupname:permanentname
DSK = activename
```

In the second case, the name given must be an active permanent name, and the group name of this entry point is then determined.

The groupname and permanent name are then saved for future reference by the BOOT command. The permanent name specified in the DSK command need not be around now, as long as it exists by the time the BOOT command is given. The command

```
DSK=
```

or

```
DSK
```

causes BUILD to forget about any name for DSK. When a BOOT occurs, if no DSK was specified, then DSK=SYS is assumed.

$LOAD

The LOAD command is used to load a new handler into BUILD. This can be one supplied by DEC or one written by the user. Consult the software support manual for how to write a device handler for OS/8 and how to put it in BUILD format. This handler is input into BUILD as a binary file or image.

If BUILD is being run standalone, then the command is of the form

```
LOAD activename:
```

where activename is a permanent name in BUILD's current tables which has been made active. It must be a handler for a non-file-structured device.

Example:

```
LOAD PTR:
```

This command would cause BUILD to load a new handler from a binary paper tape using the PTR handler already in BUILD.

If BUILD is being run from OS/8, then the command has the form:

```
LOAD  dev:filename
```

where dev is the name of an OS/8 device currently configured in the user's system and has no relationship to handlers of the same name in BUILD's tables. If no dev: is specified, then DSK: is assumed. If dev is non-file-structured, then filename may be omitted. The filename has the form name.extension where if no extension is given the dot may be omitted. The filename is looked up on the given device and it is presumed to be a binary file of the new handler to be loaded. If no extension is specified, BUILD looks for a file by the same name but with the extension .BN first.

Examples:

```
LOAD   PTR:
LOAD   SYS:FILE
LOAD   DTA3:HANDLR.$3
LOAD   GT8E
```

Several files to be loaded may be specified on one line by separating them by commas. Each specification must give a device or else DSK: is assumed. Each file contains a separate handler to be loaded. Example:

```
LOAD  DTA3:FILE1, DTA5: FILE2
```

$UNLOAD

UNLOAD is used to unload handlers or permanent names that are not wanted. This makes more room in BUILD for more badly wanted handlers.

To unload an entire group of handlers, type

```
UNLOAD groupname
```

This unloads the handler and all descriptors associated with it. Any active permanent names in this group are automatically deactivated.

To remove just a particular permanent name from BUILD's tables, type

```
UNLOAD groupname:permanentname
```

This does not unload the handler, just the entry point name. It frees up space in the descriptor table but not in the handler table.

Several permanent names can be removed at once. Example:

```
UNLOAD  DT:DTA4, DTA6
```

The hyphen construction can not be used with the UNLOAD command.

It is more advantageous to unload an entire group rather than each entry point.

$ALTER

ALTER is used to modify a location within a handler. General form:

```
ALTER groupname, loc=newvalue
```

where loc is an octal number in the range ∯-177 for a 1-page handler or in the range ∯-377 for a 2-page handler. The contents of this relative location in the handler is replaced by the new value specified.

Example:

```
ALTER KL8E,23=74∯2
```

If no = newvalue is given, then BUILD first prints the old value of the specified location followed by a slash. At this point you can type the new value or just carriage return to not change the value.

Example:

$ALTER TA8A, 144

3266/ 3267

where the underlined items are typed by BUILD.

$EXAMINE

EXAMINE is like ALTER, but it only lets you examine a location. Form:

EXAMINE groupname, loc

$DCB

This command allows modification of the DCB word associated with a permanent name (See Software Support Manual for information on DCB word.) The DCB word is the first word after the permanent name in a description (from the handler header information words). This command works like ALTER. The command

DCB activename

causes BUILD to print the current DCB word for the permanent name specified (which must be active). Then a slash is typed and you type the new value followed by a carriage return (or just carriage return o keep old value).

Example:

$DCB DTA4=616Ø

changes the DCB of DTA4 so that this handler becomes a read-only device.

This could also have been typed as

$DCB DTA4

416Ø/616Ø

$CTL

This command is exactly like the DCB command except it allows modification of the control word which is the word after the DCB word in the handler header block. Example:

```
CTL LTA3=24
```

changes the entry point of the LTA3 handler to be relative location 24.

$CORE

This command has the form

```
CORE  n
```

where n is an octal number in the range Ø-7. This command is used to specify the highest core bank which is to be made available to the OS/8 system being built. If n is Ø or omitted, or if this command is not used, the system built will allow all of core to be used. If n specifies more core than is physically available (ignoring current software core restrictions), the error message

```
?CORE
```

results.

Example:

```
CORE  3
```

specifies that the resulting system should not use more than 24K.

$NAME

This command is the same as in previous versions of BUILD, namely

```
NAME  activename=newname
```

which changes the active permanent name to the new name specified.

$VERSION

This command prints BUILD's version number on the Teletype.

$BOOT

This command builds the new system.

BUILD error messages:

name NOT FOUND

xxxxxx?

NO ROOM

?SYS

(a) permanent name in SYS command was not a system handler or coresident with one.

(b) a BOOT was issued when two system handlers were active or an active handler which must be coresident with a SYS handler didn't have system handler active.

?BAD LOAD

The binary handler you attempted to load didn't have the correct format, possibly due to an input error.

?HANDLERS

Tried to BOOT when you had more than 15 (decimal) active handlers including SYS and DSK.

SYS ERR

An I/O error occurred with a system handler. The computer will halt. Hit continue to retry or start all over again from scratch. Do not assume core contains a valid system.

BAD ORIGIN

I/O ERR

?DSK

The device specified as DSK is not file-structured.

?CORE

(a) The CORE command specified more core than is physically available.

(b) The BOOT command was issued with a two page system handler active but only 8K is currently available to OS/8. Two page system handlers require at least 12K.

?SYNTAX

?BAD INPUT

?BAD ARG

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: December 25, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: What you never wanted to know about the new BUILD and never cared to ask

Please add the following information to your notes on BUILD:

You must always have a system device specified before you type BOOT when reconfiguring. This is true even if you are reconfiguring your current system and not changing devices. You might be changing the system handler (or bootstrap) for this device. If the new device is the same as the current device, then BUILD won't bother to copy the sytem blocks over (but it will copy record 0). Sameness of the two devices is determined by looking at the devices' internal type (in DCB word) so that extra tape motion may occur when building a ROM system from a 12K system.
If no entry point by the name of SYS is active, BUILD will type

SYS NOT FOUND

when you try to do a BOOT. You should never change the name of any entry point called SYS nor should you rename any other entry point to SYS.

If the devices are the same, then the message WRITE ZERO DIRECT? wont be printed, in which case BUILD assumes you want the directory preserved.

Add the following error message to your list:

?NAME This indicates a syntax error wherein a name was expected but not found.

Note that the program, CONFIG, has been completely obsoleted, since all its functions can now be performed by BUILD. Specifically, never use CONFIG.06 with the new monitor since it had bugs in it. It is possible to use CONFIG.07 in an emergency, which is available from me (both the source and the emergency). Note also, that V2 BUILD had a bug in it, having to do with how it located certain internal monitor tables, and thus it may not be used to copy a V3 system. The new BUILD can copy earlier systems.

Building a new monitor (in-house) from the binaries, using BUILD:

Run build, insert handlers, and specify a new SYS (which must not be the same as the current system device), then type

$BUILD

at which point BUILD will respond with

LOAD OS8:

At this point (unlike the case wherein you're building from scratch), you type in the file specification for the binary of OS/8 using the handler name already configured into your current system (this has nothing to do with any active handlers in the BUILD core image). Forexample DTA3:OS8.BN .
This file specification is similar to the one used by the LOAD command. If no device is specified, DSK: is assumed. The default extension is .BN if none is specified.

BUILD will now read in this binary and start creating the new system using the new internal SYS handler specified. When it is done, it will type

LOAD CD:

at which point you must not type carriage return; instead type the file specification for the location of CD.BN . Carriage return will use the previous specification (which might be OK if you're reading in from paper tape).

When this is done, BUILD will print a $. You must eventually do a BOOT in order to complete the process. If you had previously done a BUILD, the BOOT command will not copy over the current system. Nor will it ask you if you want to zero the directory, it will assume you don't want to, and the new tape will wind up with ABSLDR.SV as the only entry in its directory. The BUILD command has code which resides in the handler space. You must then use either a virgin copy of BUILD or one without too many handlers. If the BUILD code has been overlaid by handlers, when you type the BUILD command, you will get the error message

?NO ROOM .

Also, the BUILD code goes away after a BOOT. So under no circumstances, should you take a copy of BUILD which has been resaved after a BOOT and type BUILD. This will be disaster.

080-100-049-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: December 11, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT : 2130 LOC 12-3

SUBJ: Creating a BUILD Binary Tape

This document is for in-house use only and explains how the BUILD binary paper tape gets created.

1. Assemble BUILD.

```
.PAL BUILD<BUILD
```

This gives you a binary of BUILD but with no handlers.

2. Save it as a core-image.

```
.R ABSLDR
*BUILD$
.SAVE SYS BUILD Ø-7577, 1ØØØØ-17577; 2ØØ=Ø
```

It is important that all of core be saved so that the core control block is set up well. As long as the RUN command is used later (instead of the R command) the CCB will remain set. This alleviates extra typing.

3. Assemble all the handlers.

This can be accomplished by using the BATCH job, HASM. First copy all the handler sources to DSK,

```
.COPY DSK:<DTAØ
```

then

```
.SUBMIT HASM
```

4. Load the desired handlers into BUILD.

This is a bit tricky because you must get all the system handlers into BUILD and their bootstraps eat up tremendous core used by the descriptor table.

Solution: Unload unnecessary entry points, thus freeing room in the descriptor table. Once the system is build, the user can unload those system handlers he doesn't need, thus making lots of room to reload all the missing handlers and entry points.

Second restriction: Once only code for this process is location inside the handler space at location 166ØØ.

Solution: Don't load more than 25 (decimal) pages worth of handlers.

A typical example is shown below. This may be changed. Handlers may be loaded in any order. Handler sizes are shown in parentheses on the right (in pages).

```
.RUN SYS BUILD


$LOAD TCØ8NS                                          (1)
UN TC:DTA3,DTA4,DTA5,DTA6,DTA7
$LOAD LINCNS                                          (1)
$UN LNC: LTA3,LTA4,LTA5,LTA6,LTA7                     (1)
$UN RKØ5:RKA1,RKA2,RKA3,RKB1,RKB2,RKB3
$LOAD RKØ8NS                                          (1)
$UN RKØ1:RKA1,RKA2,RKA3


$LOAD PT8E                                            (1)
$LOAD KL8E                                            (2)
$LOAD LSPT                                            (1)
$LOAD CSA                                             (2)
$LOAD DF32SY                                          (1)
$LOAD RFØ8SY                                          (1)
$LOAD LPSV                                            (1)
$LOAD L645                                            (1)
$LOAD ROMMSY                                          (1)
$LOAD RK8ESY                                          (1)
$LOAD RKØ8SY                                          (1)
$↑C
.SAVE SYS BUILD
```

Note that entry points should be deleted as quickly as possible, and system handlers should be loaded near the end.

In the above example, handlers which were not loaded include VR12, ASR33, RF08NS, DF32NS, TD8ENS.

Note that TCØ8, TD8E, and LINC system handlers need not be included since these will never be built from paper tape.

5. Punch the paper tape of this core image.

   This can only be done with a special binary punch routine incorporated into BUILD, on a machine with a high speed punch.

```
.GET SYS BUILD
.START 166ØØ
```

BINPUN will now punch some leader on the high speed punch (which should have been turned on) followed by an origin setting to 2ØØ. Then it will type a * at which point it is ready for commands. Each command consists of those locations to be punched, in the form: field number (Ø-7), a comma, Lower core limit, a hyphen, and upper core limit. Terminate line with carriage return. You have full editing capabilities over this line. When you are all through, type a carriage return all by itself and BINPUN will punch special finishing data (some required origins and data into Ø76ØØ, and a self-starting origin at end) then will punch some trailer. Punch out the required parts of core as follows:

```
*Ø, Ø-5377                  /BUILD
*Ø, 64ØØ-7577               /descriptor table
*1, 4ØØ-5777                /handlers
*1, 72Ø0-7577               /more BUILD
*<CR>
```

After punching trailer, BINPUN will halt. Branch manually to either 7605 to restart OS/8 or 2ØØ to start-up BUILD.

080-100-050-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: December 17, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: DIRECT

Here is the new DIRECT help file.

```
        DIRECT.SV

*       WILD NAME OR EXTENSION
?       WILD CHARACTER


/B      INCLUDE STARTING BLOCK NUMBERS (OCTAL)
/C      LIST ONLY FILES WITH CURRENT DATE
/E      INCLUDE EMPTIES
/F      FAST MODE
/I      PRINT ADDITIONAL INFO WORDS
/L      USUAL MODE
/M      LIST EMPTIES ONLY
/O      LIST ONLY FILES WITH OTHER THAN TODAY'S DATE
/R      LIST REMAINDER OF FILES AFTER FIRST ONE (BUT USE /C,/O)
/U      TREAT EACH INPUT SPECIFICATION SEPARATELY
/V      LIST FILES NOT OF FORM SPECIFIED
/W      GIVE VERSION NUMBER
/N      USE N COLUMNS

DEFAULT INPUT SPECIFICATION:    *.*
DEFAULT OUTPUT DEVICE:          TTY:
DEFAULT OUTPUT FILE EXTENSION:  .DI
```

fp

080-100-51-00

INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: December 24, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: New disk bootstraps

The OS/8 V3 RK8 and RK8E system handlers now can run with SYS: being any unit number.

To create such a system, mount your OS/8 system cartridge in the desired unit. Then key into core the special bootstrap given below. (If you want unit 0, then the old bootstrap still works). Upon executing the bootstrap, the desired system will be created.

RK8: loc contents

26	7604
27	6732
30	6733
31	5031

Load address 26. Put the unit number (0-3) into bits 9 and 10 of the console switch register (if you have one). Set all other switches to 0. Hit clear, continue (or start).

RK8E:loc contents

25	7604
26	6746
27	6743
30	7604
31	5031

Load address 25. Put the unit number (0-3) of the unit you want to bootstrap into into bits 9 and 10 of the console switch register. Hit clear, continue.

Notes:

1. Be sure to put the unit into bits 9 and 10. You may have the tendency to put it into bits 10 and 11.

2. Some people assign SYS: an alternate name when they configure their system. For example, you might create a system where SYS: has the alternate name RKA0. Note that if you were to bootstrap onto unit 2 of such a system, both SYS: and RKA0 would then refer to unit 2.

080-100-052-00

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: December 24, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: Patching your LV8E handler.

The following patches are available for the LPSV lineprinter handler:

rel. loc 0: set to -printer width-1 (i.e. set to -121_8 for an 80 column lineprinter)

rel loc 1:	set to 4	for the LV8E lineprinter
	set to 14	for the LP08 and LS8E printers
rel loc 2:	set to -40	to convert lower case to upper case
	set to 0	if your printer can print lower case

digital INTEROFFICE MEMORANDUM

TO: OS/8 V3 List

DATE: December 25, 1973

FROM: S. Rabinowitz

DEPT: Small Systems Softw. Eng.

EXT: 2130 LOC: 12-3

SUBJ: New TDINIT

I have made drastic changes in the TDINIT program. This memo describes these changes.

TDINIT is used to create a TD8E (either ROM or 12K) system tape from a TC08 system tape. Unlike the V2 TDINIT, it will work on any TC08 system tape and not just the one supplied from the library. If the user has the files

TDINIT.SV
TDROM .SY
TD12K .SY

and if he has his original tape #2 from the library (the OS/8 binaries tape), or a copy, or has created the special bootstrap record 0 on a tape of his own, then he can use these files to convert any system DECtape to a TD8E tape.

Procedure: Place the tape with the special record 0 on drive 0. This same tape must have the files TDINIT.SV, TDROM.SY, and TD12K.SY on it, in that order and starting at the beginning of the tape (i.e. TDINIT.SV must be at block 7). Bootstrap into this tape. This can be accomplished by keying in the TD8E bootstrap, or by using the diode ROM bootstrap switch if available.

At this point, the special record zero is read into core and reads the beginning of TDINIT.SV into core (it skips the core control block and the first page and reads in the next 3 pages). TDINIT.SV then reads in the rest of itself. If this last read gets an error, the program will halt at location 346. Hit continue to retry (or restart manually at location 202).

If everything is successful, TDINIT prints

TD8E INITIALIZER PROGRAM VERSION 4

on the teletype. It then looks around to see whether you have a ROM in field 7. If you do, it prints

12K SYSTEM

If not, it checks to see if you have 12K or more of core, if you do, it types

12K SYSTEM

If you don't, it is an error and it prints

NEED ROM OR 12K

and halts. It is now ready to configure the appropriate system. If you have a ROM but wish to create a 12K system, or you wish to fake out TDINIT, you can hit halt and set location 17 as follows:

0 if you want a 12K system
1 if you want a ROM system

and then manually branch to location 214.

Back to normal situation:
TDINIT then prints

MOUNT A CERTIFIED DECTAPE ON UNIT 1 WRITE-ENABLED
ALWAYS KEEP ORIGINAL SYSTEM DECTAPES WRITE-LOCKED
STRIKE A CHARACTER TO CONTINUE

Do as it says and strike any character to continue, at which point it will copy the system from the appropriate system head file on unit 0 onto the tape on unit 1. When this is through, it will type

DISMOUNT TAPE #2 FROM UNIT 0 AND SAVE IT
MOUNT ORIGINAL SYSTEM TAPE #1 ON UNIT 0
PREPARE TO COPY FILES OVER
STRIKE A CHARACTER TO CONTINUE

It is now ready to copy the files from the system tape over to the new tape being created. Strike any character (except CTRL/Z to continue).

If you want to perform non-standard special processing, type CTRL/Z at this time, and TDINIT will print

TYPE 1 TO COPY FILES FROM UNIT 0 TO UNIT 1
TYPE 2 TO ZERO THE DIRECTORY OF UNIT 1
TYPE 3 TO LEAVE THE DIRECTORY OF UNIT 1 ALONE
STRIKE A CHARACTER TO CONTINUE

Reply with either a 1,2, or 3 (which will not echo) to pick the option you desire. Typing any other character will cause the request message to repeat.

Depending upon whether you typed 1, 2, or 3, you will receive one of the following confirmatory messages in response, respectively:

COPYING FILES FROM UNIT 0 TO UNIT 1
ZEROING THE DIRECTORY ON TAPE UNIT 1
DIRECTORY ON UNIT 1 PRESERVED

If you didn't respond with CTRL/Z above, you merely get the message

COPYING FILES FROM UNIT 0 TO UNIT 1

at which point it copies the files and updates the directory.
(In the other cases it zeroes the directory or leaves it alone).

Then it types:

REMOVE AND SAVE TAPE ON UNIT 0
TAKE NEW TAPE (ON UNIT 1) WHICH WAS JUST CREATED
AND PLACE IT ON UNIT 0
IT IS YOUR NEW OS/8 SYSTEM TAPE
STRIKE A CHARACTER TO CONTINUE

Dial the unit number of the newly created tape to 0 and
strike a character for the program to bootstrap into this tape.
(The tape should be write-enabled and on unit 0).
The OS/8 monitor should start up and print a dot.

Other messages:

If the system head files are bad or in the wrong place, the
program will print

NOT ORIGINAL OS/8 SYSTEM TAPE #2
MOUNT CORRECT TAPE ON UNIT 0
STRIKE A CHARACTER TO CONTINUE

If any I/O errors occur on the DECtape, TDINIT will not retry
3 times, but instead type

FATAL IO ERROR
TYPE A TO ABORT AND START OVER AGAIN
TYPE ANY OTHER CHARACTER TO RETRY THIS I/O OPERATION
STRIKE A CHARACTER TO CONTINUE

Once TDINIT has correctly started up, it may be restarted
from location 200.

Alternate procedure: For people who don't want to copy the
entire tape, they may type CTRL/Z to the key message and then
zero the directory of the new tape. When they bootstrap onto
it, it will have a good system but no files. They may then
run the cusps from the other drive, for example, by typing

.RUN DTA1:PIP
*SYS:<DTA1:/S$

Creating the special record 0 (in-house use only):
Run the program TDINIT directly and it will print

READY TO CREATE BLOCK 0 OF UNIT 1
STRIKE A CHARACTER TO CONTINUE

It will print OK and halt when it has been successful.
Hit continue to rebootstrap.