

```

EEEEEEEEEE XX XX AAAAAA MM MM PPPPPPPP LL EEEEEEEEE SSSSSSSS
EEEEEEEEEE XX XX AAAAAA MM MM PPPPPPPP LL EEEEEEEEE SSSSSSSS
EEEEEEEEEE XX XX AAAAAA MM MM PPPPPPPP LL EEEEEEEEE SSSSSSSS
EE XX XX AA AA MMMM MMMM PP PP LL EE SS
EE XX XX AA AA MMMM MMMM PP PP LL EE SS
EE XX XX AA AA MMMM MMMM PP PP LL EE SS
EE XX XX AA AA MM MM MM PP PP LL EE SS
EE XX XX AA AA MM MM MM PP PP LL EE SS
EEEEEEEE XX XX AA AA MM MM PPPPPPPP LL EEEEEEEEE SSSSSSS
EEEEEEEE XX XX AA AA MM MM PPPPPPPP LL EEEEEEEEE SSSSSSS
EEEEEEEE XX XX AA AA MM MM PPPPPPPP LL EEEEEEEEE SSSSSSS
EE XX XX AAAAAAAAAA MM MM PP LL EE SS
EE XX XX AAAAAAAAAA MM MM PP LL EE SS
EE XX XX AAAAAAAAAA MM MM PP LL EE SS
EE XX XX AA AA MM MM PP LL EE SS
EE XX XX AA AA MM MM PP LL EE SS
EE XX XX AA AA MM MM PP LL EE SS
EEEEEEEEEE XX XX AA AA MM MM PP LLLLLLLLLL EEEEEEEEE SSSSSSSS
EEEEEEEEEE XX XX AA AA MM MM PP LLLLLLLLLL EEEEEEEEE SSSSSSSS
EEEEEEEEEE XX XX AA AA MM MM PP LLLLLLLLLL EEEEEEEEE SSSSSSSS

```

```

DDDDDDDD  RRRRRRRR  MM      MM  AAAAAA  SSSSSSSS  TTTTTTTTTT
DDDDDDDD  RRRRRRRR  MM      MM  AAAAAA  SSSSSSSS  TTTTTTTTTT
DD      DD  RR      RR  MMMM  MMMM  AA      AA  SS      TT
DD      DD  RR      RR  MMMM  MMMM  AA      AA  SS      TT
DD      DD  RR      RR  MM  MM  MM  AA      AA  SS      TT
DD      DD  RR      RR  MM  MM  MM  AA      AA  SS      TT
DD      DD  RRRRRRRR  MM      MM  AA      AA  SSSSSS  TT
DD      DD  RRRRRRRR  MM      MM  AA      AA  SSSSSS  TT
DD      DD  RR  RR  MM      MM  AAAAAAAAAA  SS      TT
DD      DD  RR  RR  MM      MM  AAAAAAAAAA  SS      TT
DD      DD  RR      RR  MM      MM  AA      AA  SS      TT
DD      DD  RR      RR  MM      MM  AA      AA  SS      TT
DDDDDDDD  RR      RR  MM      MM  AA      AA  SSSSSSSS  TT
DDDDDDDD  RR      RR  MM      MM  AA      AA  SSSSSSSS  TT

```

```

MM      MM  AAAAAA  RRRRRRRR
MM      MM  AAAAAA  RRRRRRRR
MMMM  MMMM  AA      AA  RR      RR
MMMM  MMMM  AA      AA  RR      RR
MM  MM  MM  AA      AA  RR      RR
MM  MM  MM  AA      AA  RRRRRRRR
MM      MM  AA      AA  RRRRRRRR
MM      MM  AAAAAAAAAA  RR  RR
MM      MM  AAAAAAAAAA  RR  RR
MM      MM  AA      AA  RR      RR
MM      MM  AA      AA  RR      RR
MM      MM  AA      AA  RR      RR
MM      MM  AA      AA  RR      RR

```

```

++
FU
SI
SI
Bo
OY
C
I
I
O
I
C
S

```


.TITLE DRMSTR - DRCOPY MASTER MACRO ROUTINES
.IDENT 'V04-000'

```

*****
*
* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
* ALL RIGHTS RESERVED.
*
* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
* TRANSFERRED.
*
* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
* CORPORATION.
*
* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*
*****

```

```

++
: FACILITY:      DRCOPY
:
: ABSTRACT:
:   These routines are a portion of the Master part of DRCOPY
:
: ENVIRONMENT:  User mode
:
: AUTHOR:  STEVE BECKHARDT,      CREATION DATE:  17-July-1979
:
: MODIFIED BY:
:
--

```

DRSI

: Mc
: \$C

COMM

: Cc

OPEN

.SBTTL DECLARATIONS

INCLUDE FILES:

MACROS:

EQUATED SYMBOLS:

OWN STORAGE:

.PSECT MASDATA, LONG

MASFAB: \$FAB FAC = <BIO,GET,PUT> ; Master FAB

MASRAB: \$RAB FAB = MASFAB,- ; Master RAB
ROP = <BIO,ASY>

.PSECT MASCODE

.SBTTL OPEN_FILE - Open a file for PUT

++
FUNCTIONAL DESCRIPTION:

This routine opens an existing file for a PUT
and copies the file attributes into a buffer.

CALLING SEQUENCE:

CALLS/G OPEN_FILE

INPUT PARAMETERS:

4(AP) Address of descriptor for filename buffer
8(AP) Address of size of filename
12(AP) Address of buffer to store file attributes
16(AP) Address of location to store return status

OUTPUT PARAMETERS:

16(AP) Address of location to store return status

COMPLETION CODES:

Those returned by \$OPEN or \$CONNECT

-.
.ENTRY OPEN_FILE, ^M<R2,R3,R4>

```
MOVAB  MASFAB,R2          ; Put address of FAB in R2
MOVL   4(AP),R0           ; Descriptor to filename
MOVL   4(R0),FAB$L_FNA(R2) ; Store filename address
MOVB   @8(AP),FAB$B_FNS(R2) ; Store filename size

$OPEN  FAB = (R2)         ; Open the file
BLBC   R0,90$            ; Error
$CONNECT RAB = MASRAB
BLBC   R0,90$
```

; Copy file attributes into buffer

```
MOVL   12(AP),R4          ; Address of buffer
MOVL   FAB$L_ALQ(R2),(R4)+ ; Allocation quantity
MOVL   FAB$L_FOP(R2),(R4)+ ; File process options
MOVL   FAB$L_MRN(R2),(R4)+ ; Maximum record number
MOVW   FAB$W_DEQ(R2),(R4)+ ; Default extension quantity
MOVW   FAB$W_BLS(R2),(R4)+ ; Block size
MOVW   FAB$W_MRS(R2),(R4)+ ; Maximum record size
MOVB   FAB$B_BKS(R2),(R4)+ ; Bucket size
MOVB   FAB$B_FSZ(R2),(R4)+ ; Fixed control area size
MOVB   FAB$B_ORG(R2),(R4)+ ; Organization
MOVB   FAB$B_RAT(R2),(R4)+ ; Record attributes
MOVB   FAB$B_RFM(R2),(R4)+ ; Record format
```

```
90$:  MOVL   R0,@16(AP)    ; Store status
```


RET

++
FU

..
CJ

..
IM

..
OJ

..
CJ

..
SI

..
--

..
CJ

..
INI

..
I

.SBTTL CREATE_FILE - Create a file for GET

++
FUNCTIONAL DESCRIPTION:

This routine creates a file using the attributes passed to it.

CALLING SEQUENCE:

CALLS/G CREATE_FILE

INPUT PARAMETERS:

4(AP) Address of descriptor for filename buffer
8(AP) Address of size of filename
12(AP) Address of buffer to get file attributes
16(AP) Address of location to store return status

OUTPUT PARAMETERS:

16(AP) Address of location to store return status

COMPLETION CODES:

Those returned by \$CREATE or \$CONNECT

-.
.ENTRY CREATE_FILE,^M<R2,R3,R4>

; Copy file attributes into FAB

MOVAB	MASFAB,R2	; Address of FAB in R2
MOVL	4(AP),R0	; Get address of filename desc.
MOVL	4(R0),FAB\$L_FNA(R2)	; Store filename address
MOVB	@8(AP),FAB\$B_FNS(R2)	; Store filename size
MOVL	12(AP),R4	; File attr. buffer addr. in R4
MOVL	(R4)+,FAB\$L_ALQ(R2)	; Allocation quantity
MOVL	(R4)+,FAB\$L_FOP(R2)	; File process options
MOVL	(R4)+,FAB\$L_MRN(R2)	; Maximum record number
MOVW	(R4)+,FAB\$W_DEQ(R2)	; Default extension quantity
MOVW	(R4)+,FAB\$W_BLS(R2)	; Block size
MOVW	(R4)+,FAB\$W_MRS(R2)	; Maximum record size
MOVB	(R4)+,FAB\$B_BKS(R2)	; Bucket size
MOVB	(R4)+,FAB\$B_FSZ(R2)	; Fixed control area size
MOVB	(R4)+,FAB\$B_ORG(R2)	; Organization
MOVB	(R4)+,FAB\$B_RAT(R2)	; Record attributes
MOVB	(R4)+,FAB\$B_RFM(R2)	; Record format

; Create the file

\$CREATE FAB = (R2)
BLBC R0,90\$; Error
\$CONNECT RAB = MASRAB

DRMAST.MAR;1

16-SEP-1984 17:04:11.^H23⁷ Page 6

; Store return status

908: MOVL R0,@16(AP)
 RET

DRS

++
F

0

++
G

.SBTTL START_RMS - Start a RMS Read or Write

++
FUNCTIONAL DESCRIPTION:

This routine starts a RMS read or write operation

CALLING SEQUENCE:

CALLS/G START_RMS

INPUT PARAMETERS:

4(AP) Address of buffer
8(AP) Address of size of transfer
12(AP) Address of flag:
1 = Write
2 = Read

OUTPUT PARAMETERS:

None

--
.ENTRY START_RMS,^M<>

MOVAL MASRAB,R0 ; Address of RAB
CMPB @12(AP),#1 ; Read or write
BEQL 10\$; Write

; Do a READ

MOVL 4(AP),RAB\$L_UBF(R0) ; Buffer address
MOVW @8(AP),RAB\$W_USZ(R0) ; Size

\$READ RAB = (R0),-
ERR = MRMS_AST,-
SUC = MRMS_AST

RET

10\$: ; Do a Write

MOVL 4(AP),RAB\$L_RBF(R0) ; Buffer address
MOVW @8(AP),RAB\$W_RSZ(R0) ; Buffer size

\$WRITE RAB = (R0),-
ERR = MRMS_AST,-
SUC = MRMS_AST

RET

.SBTTL CLOSE_FILE - Close the file

++
FUNCTIONAL DESCRIPTION:

This routine closes the file used by the Master

CALLING SEQUENCE:

CALLS/G CLOSE_FILE

INPUT PARAMETERS:

4(AP) Address of location to store status

OUTPUT PARAMETERS:

4(AP) Address of location to store status
--

.ENTRY CLOSE_FILE,^M<>

\$CLOSE FAB = MASFAB
MOVL R0,@4(AP)
RET

.END

The image displays a grid of 100 small terminal window screenshots, arranged in a 10x10 grid. Each window shows a different VAX/VMS command and its output. The windows are arranged in a 10x10 grid. Many windows have titles like 'LPMULT B32', 'DRMAST MAR', 'ADDRIVER MAR', 'TORIVER MAR', 'USSTEST MAR', 'GBLSECURF MAR', 'USSDISP MAR', 'DOD_ERAPAT MAR', 'LBRMAC MAR', 'XADDRIVER MAR', 'LABLOCIN MAR', 'DRSLU MAR', 'DTE_DF03 MAR', 'SECRET MAR', 'WORKO LIS', and 'EXAMPLES'. Each window contains text-based data, including system status, command prompts, and various system messages.