BCC 500 HELP...

CONTROL-K IS THE SYSTEM INTERRUPT CHARACTER. TYPING THIS ANYTIME WILL STOP COMPUTING ACTIVITY AND RETURN YOU TO YOUR CURRENT SUBSYSTEM; IF TYPED 2 OR 3 TIMES IN QUICK SUCCESSION, IT WILL RETURN YOU TO THE SYSTEM COMMAND LANGUAGE INTERPRETER. (@)

YOU CAN STOP THIS PRINTOUT RIGHT NOW BY TYPING CONTROL-K. (IT WILL STOP ON ITS OWN IN ANOTHER PARAGRAPH OR SO.)

HELP WILL SOON TYPE A "?", INDICATING THAT IT IS WAITING FOR A COMMAND. ONLY TWO EXIST: "D" FOR DESCRIBE & "F" FOR FINISHED. FOR EXAMPLE, IF YOU WANT A LIST OF TOPICS WHICH HELP DISCUSSES, THE PAPER OR SCREEN WILL LOOK LIKE THIS:

```
     ?DESCRIBE TOPICS
```

THE "?" WILL HAVE BEEN TYPED BY HELP. WHEN YOU TYPE A "D", HELP RECOGNIZES IT AND TYPES THE REST OF THE WORD ("ESCRIBE"). THEN YOU TYPE THE TOPIC OF INTEREST... AND FINISH WITH A CARRIAGE RETURN.
HELP--IN THIS CASE--WILL THEN TYPE THE TOPICS IT CAN TALK ABOUT, OR WILL TYPE THAT TOPIC YOU HAVE JUST ASKED FOR.
THE "FINISHED" COMMAND LETS YOU ESCAPE FROM HELP WHEN YOU ARE PAU.
IT IS SUGGESTED THAT YOU DESCRIBE "HELP" FIRST.

GOOD LUCK.

@HELP

HELP IS THE SUBSYSTEM NOW RUNNING. AFTER PROVIDING THE USER WITH A COMPLETE LIST OF SYSTEM COMMANDS, HELP THEN DISPLAYS A "?" HERALD AND WAITS FOR THE USER TO GIVE THE COMMAND "DESCRIBE" OR THE COMMAND "FINISHED". "DESCRIBE" PRINTS OUT BRIEF INFORMATION ABOUT A NUMBER OF TOPICS ACCORDING TO THE LIST WHICH FOLLOWS. THE ENTIRE FILE OF HELP TOPICS WHICH CAN BE SEEN INDIVIDUALLY CAN BE PRINTED AT ONCE IF YOU WISH. IT´S 25 PAGES LONG! DO THE FOLLOWING:

GET OUT OF HELP AND BACK TO THE COMMAND LANGUAGE INTERPRETER. MAKE YOUR TERMINAL DISPLAY APPEAR AS FOLLOWS:

```
   @TYPE
   FILE: #2:HELP-TEXT.
```

TO GET A LIST OF AVAILABLE TOPICS, ASK HELP TO "DESCRIBE TOPICS".

```
@TOPICS


HELP IS AVAILABLE ON THE FOLLOWING TOPICS:

BCC 500 GENERAL INFO:
   EDITING-CHARACTERS        INTRODUCTION
   FILE-ORGANIZATION         SYSTEM-COMMANDS
   HISTORY

SUB-PROCESS COMMANDS:
   CONTINUE                  RESET
   RELEASE                   SAVE-CURRENT

940 PROCESS COMMANDS:
   DUMP-940                  RECOVER-940
   HELP-940

FILE RELATED COMMANDS:
   COPY                      FILE-MAINTENANCE
   DELETE-FILES              LIST-FILES
   EXAMINE-FILE              RENAME-FILE

LINK COMMANDS:
   ALLOW-LINKS               REFUSE-LINKS
   BREAK-LINKS               WHERE-IS
   LINK-TO                   WHO-IS-ON

OTHER COMMANDS:
   ?                         LINE
   DATE                      LOGOUT
   EMULATOR-VERSION          SCHEDULE
   HELP

SUBSYSTEMS:
   BASIC                     PFS
   CAL                       PRINT
   DECTYPE                   QED
   DDT                       QSPL
   FTC                       QRUN
   FOS                       RUNOFF
   LISP                      SNOBOL
   NARP                      TYPE
```

@INTRODUCTION

THIS SECTION IS A GENERAL INTRODUCTION AND GUIDE TO THE NEW USERS OF THE BCC 500 SYSTEM.

THE SYSTEM IS AN EXPERIMENTAL ONE (A ONE-OF-A-KIND), A PROTOTYPE OF A COMMERCIAL SYSTEM THAT WAS NEVER BUILT. (DESCRIBE HISTORY FOR DETAILS). WHILE THE HARDWARE IS RELATIVELY OLD (1968), THE SYSTEM DESIGN IS QUITE NEW; ASPECTS OF IT HAVE APPEARED IN THE 370 AND IN THE DEC SYSTEM-20, FOR EXAMPLE, ONLY RECENTLY. IT IS A HIGHLY ADVANCED SYSTEM (DISREGARDING ITS UNFINISHED, UNPOLISHED NATURE) UTILIZING FIVE SPECIALIZED PROCESSING UNITS EACH DEDICATED TO A PORTION OF THE SYSTEM'S OPERATION. THEY ARE:

CHIO: CHARACTER INPUT/OUTPUT-- RESPONSIBLE FOR COMMUNICATION AND TERMINAL MANAGEMENT.
AMC: AUXILIARY MEMORY CONTROL-- RESPONSIBLE FOR EFFICIENT CENTRAL MEMORY UTILIZATION AND FOR SWAPPING OF USER'S PAGES IN A VIRTUAL STORAGE ENVIORNMENT.
MSCH: MICRO SCHEDULER-- RESPONSIBLE FOR SCHEDULING USER'S REQUESTS ON THE TWO USER CPU'S.
(2)CPU: TWO USER'S CPUS. RUNS THE USER'S PROCESSES.

THE CHIO, AMC, AND MSCH PROCESSORS COLLECTIVELY RUN THE OPERATING SYSTEM. ON MOST OTHER SYSTEMS THE OS IS A VERY LARGE COLLECTION OF SOFTWARE WHICH RUNS ON THE CPU ALONG WITH THE USERS' PROGRAMS. THE CODE RUN BY THESE THREE PROCESSORS IS NOT SOFTWARE; RATHER, THEY EACH EXECUTE THEIR PART OF THE OS DIRECTLY FROM VERY FAST MICROSTORE (100 NANO-SECONDS PER INSTRUCTION). THIS GIVES THE 500 IMMENSE POWER AT EXECUTING ITS OPERATING SYSTEM AND PERMITS IT TO DO THINGS IN WAYS NO OTHER SYSTEM COULD ATTEMPT WITH ORDINARY SOFTWARE.

THE CPUS DO EXECUTE A LITTLE OPERATING SYSTEM CODE; THIS IS NECESSARY IN ORDER TO ALLOW EACH USER PROCESS TO COMMUNICATE WITH THE OS. THE CPUS PERMIT A NUMBER OF SYSTEM CALLS, IN WHICH THE CODE READS OR SETS BITS IN RESIDENT SYSTEM TABLES WHICH ARE BEING SCANNED BY THE DEDICATED PROCESSORS. THE "SOFTWARE PORTION" OF THE OS ALSO CONTAINS FREQUENTLY-NEEDED UTILITY ROUTINES WHICH CAN BE RUN BY SYSTEM CALL.

ANOTHER INTERESTING FEATURE OF THE 500 CPU IS THAT IT HAS TWO IDENTITIES! THAT IS, IT CAN EXECUTE THE INSTRUCTION SET AND ADDRESSING MODES OF TWO DIFFERENT MACHINES. ONE IS THE 500 CPU ITSELF (A MANUAL WHICH CAN BE OBTAINED FROM THE EE DEPT. DESCRIBES THIS MACHINE); THE OTHER IS THE USER INSTRUCTION SET OF THE OLD XDS 940 COMPUTER. A BIT IN THE STATE WORD OF THE PROCESS DETERMINES WHICH IDENTITY THE CPU ASSUMES WHEN IT IS ATTACHED TO A PROCESS FOR A QUANTUM OF TIME. THE 940-MODE (SO-CALLED) GIVES US THE ABILITY TO RUN WITHOUT CHANGE ANY EXISTING 940 SOFTWARE. TO DO THIS WE HAVE AN ASSOCIATED PIECE OF SYSTEM SOFTWARE CALLED THE "940 EMULATOR". THIS PROGRAM IS CALLED AUTOMATICALLY WHENEVER A 940 PROGRAM ATTEMPTS A 940 SYSTEM CALL. THE EMULATOR FIELDS THIS CALL, TRANSLATES IT INTO A 500 CALL, TRANS-

LATES THE RESULT BACK INTO THAT WHICH THE 940 WOULD GIVE UNDER THE CIRCUMSTANCES, AND RETURNS CONTROL TO THE 940 PROGRAM IN 940 MODE. OVER HALF OF THE EXISTING SOFTWARE ON THE 500 TODAY IS 940 SOFTWARE.

BECAUSE OF ITS UNFINISHED NATURE, THE SYSTEM´S CHARACTERISTICS SOMETIMES CONFUSE BEGINNING USERS. AN EXAMPLE OF THIS IS THE WAY IN WHICH FILES ARE TREATED. IN OTHER SYSTEMS (TSO AND HP), USER FILES ARE KEPT TRACK OF BY THE SYSTEM IN A DIRECTORY OF SOME KIND. THERE IS A LOT OF COMPLEXITY IN THESE SYSTEMS ASSOCIATED WITH MAINTAINING THESE DIRECTORIES CORRECTLY, ESPECIALLY IF THE SYSTEM CRASHES DURING A DIRECTORY UPDATE. THE 500 KEEPS TWO DIRECTORIES FOR EACH USER: ONE ON DISK (ALONG WITH THE LONG-TERM USER FILES) AND ANOTHER ON DRUM. ON LOGIN, THE DRUM DIRECTORY (THE INTERNAL SYSTEM DIRECTORY) WILL BE FOUND TO BE EMPTY. (IT IS EMPTIED AND ALL FILES DELETED WHEN THE USER LAST LOGGED OUT.) THE USER MUST THEN COPY OVER FILES FOUND ON DISK AND LOCATED BY HIS DISK DIRECTORY ONTO THE DRUM. AGAIN, UPON LOGOUT, THE USER MUST COPY BACK TO DISK ANY FILES HE WANTS SAVED. THIS IS ALL OUR DISK STORAGE UNIT DOES IN THE SYSTEM; IT IS NOT USED FOR SWAPPING USERS IN & OUT OF MEMORY OR FOR DIRECT FILE ACCESSES. IT IS USED EXACTLY LIKE A VERY LARGE, MAG TAPE CASSETTE UNIT!

HOW IS THIS COPYING BACK AND FORTH DONE? WE USE A SUBSYSTEM -- A BUILT-IN PROGRAM CALLED "PFS" (FOR PERMANENT FILE SYSTEM). PFS HAS A NUMBER OF COMMANDS WHICH CAN BE EXECUTED TO DO SUCH THINGS AS READ DISK FILES INTO THE SYSTEM, WRITE THEM BACK OUT, CONTROL VARIOUS KINDS OF ACCESS TO THEM, LIST OUT THE DISK FILE DIRECTORY ON THE TERMINAL, ETC. DESCRIBE PFS FOR DETAILS, OR SEE THE PFS MANUAL. WHILE THIS APPROACH MAY SEEM AT FIRST GLANCE SOMEWHAT CRUDE (IT IS), IT DOES HAVE SOME ADVANTAGES. FOR ONE, FILES MAY BE VERSIONED IN A MANNER DESIRABLE TO THE USER. MOSTLY, HOWEVER, THE ADVANTAGE IS THAT IT IS VIRTUALLY IMPOSSIBLE TO LOSE A FILE IN A SYSTEM CRASH. (THIS IS NOT TRUE OF THE HP-ACCESS OR OF TSO.)

@EDITING-CHARACTERS

THESE SPECIAL EDITING CHARACTERS ARE EFFECTIVE IN MANY SITUATIONS (COMMAND LANGUAGE INTERPRETER, QED, BASIC, ETC.) FOR CORRECTING IMMEDIATE TYPING ERRORS ON THE TERMINAL:

CONTROL-A: CHARACTER BACKSPACE
CONTROL-W: PREVIOUS WORD DELETE
CONTROL-Q: CURRENT LINE DELETE

@HISTORY

THE BCC 500 WAS TO BE THE PRINCIPAL RESULT OF A COMPUTER SYSTEM RESEARCH PROJECT AT THE UNIVERSITY OF CALIFORNIA, BERKELY IN THE ´60S. THIS PROJECT, FUNDED BY THE DOD, WAS TO DEVELOP MORE EFFECTIVE MEANS FOR INTERACTION WITH COMPUTERS (TIME-SHARING) AND MORE EFFICIENT CENTRALIZED COMPUTER SYSTEMS. WHEN MANUFACTURERS SHOWED RELUCTANCE TO USE THE IDEAS OF THIS GROUP, IT LEFT THE UNIVERSITY TO FORM BERKELEY COMPUTER CORPORATION (1969). IT WAS IN THIS ENVIRONMENT, WITH PRIVATE FUNDING, THAT THE BCC 500 WAS DESIGNED AND BUILT.

THE 500 WAS BUILT ONLY AS A DEMONSTRATION OF THE ACTUAL PRODUCT ANTICIPATED AND TO PROVE THE GROUP´S ABILITY TO BUILD BOTH A COMPANY (OF 150 PEOPLE) AND A LARGE-SCALE COMPUTER. THIS WAS DONE IN ONLY 18 MONTHS. HOWEVER, DUE IN PART TO THE TIMES, FURTHER CAPITAL WAS UNAVAILABLE AND BCC WAS FORCED TO DISBAND.

DR. LICHTENBERGER CAME TO HAWAII IN 1971 TO JOIN THE ALOHA SYSTEM AND, ARMED WITH GENEROUS FUNDING FROM THE ORIGINAL DOD SOURCE, BROUGHT THE FORMER COMPANY´S EQUIPMENT TO THE ELECTRICAL ENGINEERING DEPARTMENT. HERE IT WAS SOMEWHAT REBUILT AND BEGAN TO OPERATE ONCE AGAIN IN MARCH, 1973. WITH THE EXCEPTION OF TWO 2-WEEK PERIODS, IT HAS BEEN IN OPERATION AROUND THE CLOCK, SEVEN DAYS A WEEK SINCE THEN.

THE SYSTEM IS NOW OPERATED BY EE FOR ITS EDUCATIONAL AND SOME RESEARCH NEEDS. IT IS TENUOUSLY FUNDED, BUT ITS MINIMUM OPERATIONAL REQUIREMENTS ARE SMALL SINCE THE EQUIPMENT IS WHOLLY OWNED BY THE STATE OF HAWAII.

@FILE-ORGANIZATION

FILE NAMING AND COMMANDS

A FILE DESIGNATOR CONSISTS OF A NAME AND A TYPE WRITTEN AS

NAME:TYPE

THE NAME MAY HAVE WORDS CONCATENATED WITH "-", AS IN THE NAME:

WORD1-WORD2-WORD3

UP TO A TOTAL OF 16 CHARACTERS. THE WORDS MAY CONTAIN LETTERS AND DIGITS. FILES ARE GROUPED INTO DIRECTORIES ASSOCIATED WITH USERS. THE FILES OF A USER LOGGED IN AT A GIVEN TERMINAL ARE DESIGNATED BY THEIR NAME:TYPE. FILES IN OTHER DIRECTORIES MAY BE DESIGNATED BY PREFIXING TO THEIR NAME:TYPE EITHER A USER NUMBER OR NAME, AS:

```
#23:NAME:TYPE      OR
@GUEST:NAME:TYPE
```

WHEN A FILE IS CREATED (FIRST REFERRED TO), THE NAME AND TYPE MUST BE SURROUNDED BY DOUBLE QUOTES, AS:

```
*WRITE ON "NUTS:9SYM".
```

IF DOUBLE QUOTES ARE NOT REQUIRED, THE NAME MAY BE ABBREVIATED. ABBREVIATION ALLOWS ONE TO SUPPLY A SUBSTRING WHICH UNAMBIGUOUSLY SPECIFIES THE FILE. WHEN MORE THAN ONE WORD IS INVOLVED AND IF THE ABBREVIATION OF THE FIRST WORD DOES NOT COMPLETELY SPECIFY IT, THEN A NEW ATTEMPT IS MADE WITH A SECOND WORD. FOR EXAMPLE, "LIST-FILES" MIGHT BE ABBREVIATED AS "L-F", "LI-FI", OR MAYBE JUST "LIST". ALL, A PORTION, OR NONE OF THESE EXAMPLES MIGHT SUCCEED, DEPENDING ON THE CONTENTS OF THE DIRECTORY.

@SYSTEM-COMMANDS

WHEN THE COMPUTER TYPES AN "@" SIGN ON THE LEFT MARGIN OF YOUR TERMINAL, IT IS READY AND WAITING FOR A SYSTEM COMMAND TO BE TYPED IN. A COMMAND (A LISTING OF ALL OF THEM CAN BE OBTAINED FROM HELP) INSTRUCTS THE SYSTEM TO PROVIDE THE USER WITH ONE OF A NUMBER OF PRE-STORED SERVICES. WHEN A USER LOGS INTO THE SYSTEM, THE SESSION OR THE JOB STREAM CREATED BY THE USER IS CALLED A PROCESS. THUS, WHEN A USER TYPES IN A SYSTEM COMMAND AND INVOKES A SUBSYSTEM SUCH AS BASIC OR FORTRAN, THAT SUBSYSTEM RUNNING IS CALLED A SUBPROCESS. ALL SYSTEM COMMANDS ARE ACTUALLY FILE NAMES EXCEPT FOR SPECIAL COMMANDS. WHEN A COMMAND IS TYPED, WHAT HAPPENS IS THAT A SUBPROCESS OF THE USER'S PROCESS IS CREATED, THE SYSTEM COMMAND FILE IS ATTACHED TO THE SUBPROCESS (READ INTO USER'S WORKSPACE) AND THEN THE SUBPROCESS IS CALLED. THERE ARE 3 TYPES OF COMMANDS, PERMANENT, TEMPORARY AND SPECIAL.

PERMANENT COMMANDS CAUSE THE SYSTEM TO FIRST DESTROY THE "CURRENT" SUB-PROCESS (IF ANY) AND SET UP, ATTACH, AND CALL THE NEW SUB-PROCESS. WHEN THE SUB-PROCESS TERMINATES AND RETURNS TO THE SYSTEM, ITS MEMORY SPACE IS RETAINED, AND IT IS CONSIDERED AS THE CURRENT SUB-PROCESS UNTIL THE NEXT PERMANENT COMMAND IS TYPED. QED AND BASIC ARE EXAMPLES OF A PERMANENT COMMAND. SINCE THE MEMORY SPACE IS RETAINED AFTER A SUBPROCESS'S TERMINATION, IT IS POSSIBLE TO RETURN TO THE EXACT STATE OF THAT SUBPROCESS BEFORE IT WAS TERMINATED BY USING THE "CONTINUE" COMMAND. SEE "CONTINUE" DESCRIBED ELSEWHERE.

TEMPORARY COMMANDS DO NOT AFFECT OTHER SUB-PROCESSES. RATHER, THEY CAUSE THE SYSTEM TO CREATE, ATTACH, AND CALL A NEW SUB-PROCESS WHICH IS AUTOMATICALLY DESTROYED WHEN IT IS TERMINATED. PFS AND WHO-IS-ON ARE SOME EXAMPLES OF THIS TYPE.

SPECIAL COMMANDS ARE THOSE WHICH ARE NECESSARY TO MANIPULATE THE SUB-PROCESS STRUCTURE. THESE FOUR COMMANDS ARE THE ONLY COMMANDS WHICH ARE NOT FILE NAMES AND ARE BUILT INTO THE SYSTEM UTILITY. THEY ARE: "CONTINUE", "RELEASE", "RESET", AND "SAVE-CURRENT".

@X

SUB-PROCESS COMMANDS

@CONTINUE

CAUSES THE CURRENT SUB-PROCESS (PERMANENT COMMAND) TO RESUME EXECUTION. FOR INSTANCE IF YOU HAD JUST EXITED BASIC AND WANTED TO GET BACK IN, INSTEAD OF TYPING BASIC AND HAVING TO RELOAD YOUR PROGRAM, ALL YOU HAVE TO DO IS TO TYPE "CONTINUE" TO GET BACK TO THE EXACT STATE BEFORE YOU EXITED BASIC. THIS APPLIES TO ANY OTHER PERMANENT COMMAND.

@SAVE-CURRENT

CAUSES THE CURRENT SUB-PROCESS TO BE PLACED IN A SAVE STATUS (I.E. MADE NO LONGER "CURRENT") SO THAT SUBSEQUENT PERMANENT COMMANDS DO NOT CAUSE IT TO BE DESTROYED. UP TO 7 SUBPROCESSES CAN BE SAVED PROVIDED YOU DON´T RUN OUT OF MEMORY. THIS COMMAND RESPONDS WITH A SUB-PROCESS NUMBER TO BE USED WITH RELATED COMMANDS. YOU CAN USE THIS TO TEMPORARILY SAVE DIFFERENT VERSIONS OF A PERMANENT COMMAND SUCH AS QED WITH DIFFERENT TEXT IN THE BUFFER. THIS COMMAND IS NOT TO BE CONFUSED WITH SAVING YOUR FILES PERMANENTLY ON DISK. SEE "PFS" IF YOU WANT TO DO THIS.

@CONTINUE <SUB-PROCESS>

RESUMES EXECUTION OF THE DESIGNATED SUB-PROCESS WITHOUT DISTURBING OTHERS. THE SUB-PROCESS MAY BE DESIGNATED EITHER BY NUMBER OR BY THE COMMAND NAME IF IT IS UNAMBIGUOUS. FOR EXAMPLE, IF YOU HAD PREVIOUSLY SAVED A QED WHO´S NUMBER WAS 3, TYPE "CONTINUE 3" TO GET IT BACK.

@RELEASE <SUB-PROCESS>

DESTROYS THE DESIGNATED SUB-PROCESS

@RESET

DESTROYS ALL SUB-PROCESSES INCLUDING THE SAVED ONES.

@X

940 COMMANDS

THESE COMMANDS ARE USED ONLY IN CONJUNCTION WITH 940 PROGRAMS. THEY ARE:

@DUMP-940 <FILE-NAME>

THE 940 PROCESS--ITS MEMORY SPACE AND ITS STATE--IS COPIED TO A FILE ON WHICH IT MAY BE SAVED INDEFINITELY.

@RECOVER-940 <FILE NAME>

INVERSE OF ABOVE. A PREVIOUSLY DUMPED 940 PROCESS IS RELOADED AND IS MADE INTO THE CURRENT SUB-PROCESS.

@HELP-940

INITIATES A QUESTION-ANSWERING SUBSYSTEM DEALING WITH 940 SYSTEM CALLS.

@X

```
                         FILE RELATED COMMANDS:

@FILE-MAINTENANCE
     HAS A SUB-COMMAND STRUCTURE TO DO MANY USEFUL OPERATIONS WITH
     FILES.  YOU CAN ESCAPE FROM THE SUB-PROCESS WITH A "FINISH"
     COMMAND.  SEE FNS/W-4 FOR DETAILS.

@LIST-FILES <PARTIAL FILE NAME> <OPTION> <ORDER>
     LISTS ALL FILES DESIGNATED BY THE PARTIAL NAME IN THE
     DIRECTORY OF THE DESIGNATED USER.  IF NO PARTIAL NAME IS
     GIVEN, THEN ALL FILES ARE LISTED.  IF NO USER NAME IS
     DESIGNATED, LISTS THE DIRECTORY OF THE USER LOGGED IN AT THE
     GIVEN TERMINAL PLUS THE DIRECTORIES TO WHICH THE GIVEN DIR-
     ECTORY IS LINKED.
     OPTIONS: "ENTRY"  - ALL INFORMATION IN THE FILE DIRECTORY
              "BRIEF"  - TYPES ONLY THE FILE NAMES AND TYPE
              "LENGTH" - TYPES FILE NAME AND LENGTH OF FILE
     ORDER:   "ALPHABETIC" - LISTS FILES ALPHABETICALLY
              "ENTRY"      - LISTS FILES EXACTLY AS FOUND IN
                             THE DIRECTORY

@COPY-FILE <ORIGIN FILE> <DESTINATION FILE>
     OBVIOUS.  SEE FNS/W-4 FOR DETAILS.

@DELETE-FILE <PARTIAL FILE NAME>
     OBVIOUS.  USERS SHOULD DELETE ANY UNNEEDED FILES FROM
     THE SYSTEM TO SAVE ON DRUM SPACE. IF THE FILE NAME IS:
          "*:<TYPE>" - THEN ALL FILES OF THE <TYPE> ARE DELETED.
                       EXAMPLE: "*:9SYM", "*:9BIN", "*:9DMP".
          "*:*"      - THEN ALL FILES ARE DELETED.


@RENAME-FILE <OLD NAME> <NEW NAME>
     OBVIOUS.  THE NEW NAME DOES NOT REQUIRE DOUBLE QUOTES.

@EXAMINE-FILE <FILE NAME>
     USED TO LOOK AT FILE PARAMETERS.
@X
```

TERMINAL LINKING COMMANDS

A NUMBER OF COMMANDS ALLOW INSPECTION OF WHO IS ON THE SYSTEM AND WHERE SO THAT TERMINALS MAY BE CROSS-CONNECTED FOR COMMUNICATIONS PURPOSES. THEY ARE:

@WHO-IS-ON

TYPES A LIST OF USER NAMES CURRENTLY LOGGED INTO THE SYSTEM AND GIVES THEIR TERMINAL NUMBERS. TERMINAL NUMBERS CAN NORMALLY BE IGNORED UNLESS A GIVEN USER IS LOGGED IN AT MORE THAN ONE TERMINAL.

@WHERE-IS <USER>

GIVES THE TERMINAL LINE NUMBER OF THE DESIGNATED USER, IF HE IS LOGGED INTO THE SYSTEM.

@REFUSE-LINKS

INHIBITS LINKS FROM BEING ESTABLISHED BY ANYONE TO THE TERMINAL FROM WHICH IT IS EXECUTED. THE SYSTEM WILL NORMALLY ALLOW LINKING.

@ALLOW-LINKS

INVERSE OF ABOVE.

@LINK-TO <USER>

ESTABLISHES A LINK. EACH USER CAN SEE EACH OTHER´S ECHOES BUT THEIR INPUTS ARE KEPT APART. THE USER LINKED TO GETS A MESSAGE SO INFORMING HIM.

IN LINKING TO A USER, IT IS POSSIBLE NOT ONLY TO SEE HIS OUTPUT BUT ALSO TO CONDUCT CONVERSATION WITH HIM. TO DO THE LATTER, YOU MUST BE AT THE COMMAND LANGUAGE LEVEL (YOU WILL BE AFTER LINKING, EVEN THOUGH YOU TERMINAL WILL BE TYPING WITH ANY OF THE LINKED-TO USER´S OUTPUT). THEN TYPE A DOUBLE-QUOTE MARK, FOLLOWED BY ARBITRARY TEXT. THIS PUTS YOU INTO A CONVERSATIONAL MODE IN WHICH SUCCESSIVE CARRIAGE RETURNS SIMPLY GIVE A NEW LINE. TO GET OUT OF SO-CALLED QUOTE MODE, TYPE CONTROL-K.

@BREAK-LINKS

CLOSES THE LINK. EITHER PARTY MAY EXECUTE IT.

@X

MISCELLANEOUS COMMANDS

A FEW OTHER COMMANDS FALL IN NO SIMPLE CATEGORY. THEY ARE:

@?
SAME AS TYPING HELP. FIRES UP THIS SUBSYSTEM.

@DATE
TYPES OUT THE DATE AND TIME OF DAY. USEFUL FOR COMPUTER HACKERS THAT ARE ALWAYS LATE FOR APPOINTMENTS.

@EMULATOR-VERSION
TYPES THE VERSION OF THE 940 EMULATOR IN CURRENT USE.

@LINE
TYPES OUT THE LINE NUMBER TO WHICH THE PRESENT TERMINAL IS CONNECTED.

@LOGOUT
PERMITS THE USER TO LEAVE THE SYSTEM. AT THIS TIME, ALL FILES IN THE USERS SYSTEM DIRECTORY IS DELETED AND LOST FOREVER UNLESS SAVED ON THE DISK USING PFS. HE MAY LOGIN AGAIN BY TYPING CONTROL-K TO GET THE SYSTEM´S ATTENTION.

@SCHEDULE
TYPES OUT INFORMATION CONCERNING THE BCC 500 SYSTEM´S AVAILABILITY.

@X

SUBSYSTEMS

@BASIC

ACTIVATES THE BASIC LANGUAGE INTERPRETER. BASIC IS AN EASY, YET SOPHISTICATED LANGUAGE FOR ALL PURPOSES. PARTICULARLY STRONG IN MATRIX MANIPULATIONS AND FORMATTED INPUT OUTPUT, THIS EXTENDED BASIC FEATURES EXTENSIVE BUILT IN FUNCTIONS, FILE I/O, FUNCTION DEFINITIONS, CALCULATOR MODE AS WELL AS STANDARD BASIC. SEE THE BCC 500 BASIC MANUAL BY MARTHA CROSBY FOR DETAILS. MOST USED COMMANDS ARE:

```
RUN         EXECUTES YOUR PROGRAM
QUIT        EXITS BASIC AND RETURNS YOU TO EXECUTIVE
LOAD <FILE NAME>
            LOADS A BASIC LANGUAGE SYSTEM FILE ON THE DRUM
            INTO BASIC´S WORKSPACE
SAVE <FILENAME>
            INVERSE OF ABOVE. WRITES A SYMBOLIC FILE FROM
            BASIC ONTO THE DRUM. PROVIDES A MEANS OF
            SAVING YOUR PROGRAM TYPED IN THE BASIC SUBSYSTEM
            BY USING PFS.
LIST ([<STN 1> [-<STN 2>,<COMMA><STN2>...]])
            PRINTS YOUR PROGRAM IN YOUR WORKSPACE FOR YOUR
            INSPECTION.

            EXAMPLE:
              LIST                    LIST ENTIRE PROGRAM
              LIST 10                 LIST STATEMENT #10
              LIST 10-30              LIST STATEMENT NUMBERS
                                      10 THRU 30
              LIST 10,20,30           LIST STATEMENT NUMBERS
                                      10, 20 AND 30

DEL (ALL, [<STN 1> [-<STN2>,<COMMA><STN2>....]])
            DELETES STATEMENTS IN YOUR PROGRAM

            EXAMPLE:
              DEL ALL                  DELETE ENTIRE PROGRAM
              DEL 10                   DELETE STATEMENT #10
              DEL 10-30                DELETE STATEMENT NUMBERS
                                       10 THRU 30
              DEL 10,20,30             DELETE STATEMENT NUMBERS
                                       10, 20 AND 30
```

IN ADDITION, BCC 500 SYSTEM HAS LIBRARY OF BASIC PROGRAMS STORED IN PFS UNDER THE DIRECTORY CALLED @BCC:. VARIOUS GAMES SUCH AS FOOTBALL, GOLF, AND WUMPUS AS WELL AS STATISTICAL AND SIMULATIONS PROGRAMS RESIDE IN THIS DIRECTORY. WITH THE EXCEPTION OF FILES ENDING WITH :TCMD, ALL ARE BASIC. THE TCMD FILES ARE GAMES THAT ARE SYSTEM COMMANDS, STANDING FOR TEMPORARY COMMAND. TO EXECUTE THESE, SIMPLY READ THEM FROM PFS, AND THEN TYPE THEIR NAME (I.E. "LIMERICK") JUST LIKE A REGULAR SYSTEM COMMAND. A COMPLETE DESCRIPTION OF ALL THE PROGRAMS IN @BCC: IS WRITTEN ON THE FILE *INFO*, ALSO

STORED IN @BCC:.

EXAMPLE:

```
@PFS
&READ DISK FILE @BCC:WUMPUS;
1961 WORDS.
&FINISHED.
@BASIC
BASIC-5.17       1Ø-15-75
>LOAD WUMPUS
  (WUMPUS WILL BEGIN EXECUTION)
```

@CAL

ACTIVATES THE CAL SUBSYSTEM. SEE R-23 FOR DETAILS. CAL IS A CONVERSATIONAL ALGEBRAIC LANGUAGE BASED ON THE JOSS SYSTEM. IT IS DESIGNED TO FACILITATE THE COMPUTER SOLUTION OF SMALL AND MEDIUM-SIZED NUMERICAL PROBLEMS. SINCE ITS EXECUTION IS VERY SLOW, IT IS NOT SUITABLE FOR PROBLEMS REQUIRING LARGE AMOUNTS OF COMPUTATION. IT DOES PROVIDE NEARLY FOOL-PROOF OPERATION, COMPLETE ERROR-CHECKING AND A VERY POWERFUL, CONVENIENT LANGUAGE. COMMANDS ARE:

```
CONTROL-K                       EXIT CAL & INTERRUPT COMPUTATION
```

CAL PROGRAMS ARE COMPOSED OF STEPS. EACH STEP BEGINS WITH A STEP NUMBER OF THE FORM <INTEGER>.<INTEGER>. THE INTEGER PRECEDING THE DOT IS CALLED THE PART NUMBER. ALL STEPS WITH A GIVEN PART NUMBER BELONG TO THAT PART. THE ORDERING OF THE STEPS IN THE PROGRAM IS DETERMINED BY STEP NUMBERS AND NOT BY THE ORDER OF INPUT. IF SEVERAL STEPS WITH THE SAME NUMBER ARE INPUT, THE LAST ONE WILL BE KEPT.

INDIRECT (WITH STEP NUMBER) OR DIRECT (EXECUTE IMMMEDIATELY):

```
(SET) <V> = <E>  (SET MAY BE OMITTED)

TYPE <E1>, <E2>, ...              PRINT OUT
TYPE IN FORM <E>: <E1>, <E2>, ...
TYPE (STEP) <N.N>, <N.N>, ...
TYPE PART <E>
TYPE FORM <E>
TYPE "STRING"
TYPE ALL STEPS
TYPE ALL FORMS
TYPE ALL VALUES
TYPE ALL FUNCTIONS
TYPE ALL

DEMAND <V1>, <V2>, ...           CALL FOR TYPED INPUT
DEMAND IN FORM <E>: <V1>, <V2>, ...
```

```
OPEN <NAME> FOR INPUT  AS FILE <E>     FILE I/O
                OUTPUT
CLOSE <E>
WRITE ON <E>: <SAME OPTIONS AS FOR TYPE>
READ FROM <E>: <V1>, <V2>, ...
OUTPUT ON <E>: <E1>, <E2>, ...         FILE OF BINARY FLOATING POINT
INPUT FROM <E>: <V1>, <V2> ...         NUMBERS; 2 WORDS / NUMBER
CALL <FUNCTION> ON END OF FILE

PAGE                           FORMS SPACING
LINE

TO (STEP) <N.N>                EXECUTION CONTROL
TO PART <E>
DO (STEP) <N.N>
DO PART <E>


       DIRECT ONLY:
DEFINE <V>[<V>, ...] = <E>     DEFINE CAL FUNCTION
                     : <STATEMENT>

DUMP                           SAVE / RESTORE PROGRAM
  TO <FILE>.
LOAD
  FROM <FILE>.
GO                             CONTINUE EXECUTION AFTER A PAUSE
STEP                           EXECUTE NEXT STEP ONLY
CANCEL                         CANCEL RUN

EDIT (OR MODIFY) (STEP) <N.N>
EDIT FORM <E>
EDIT <V> (FUNCTION ONLY)

DELETE <V>
DELETE (STEP) <N.N>, <N.N>, ...
DELETE PART <E>
DELETE FORM <E>
DELETE ALL STEPS
DELETE ALL FORMS
DELETE ALL VALUES
DELETE ALL FUNCTIONS
DELETE ALL


       INDIRECT ONLY:
FORM <N>: <LINE FEED>          # IS TO BE REPLACED WITH A FLOATING
<STRING>=########/%%%%%%.%%%   POINT NUMBER, % WITH A FIXED POINT

RETURN <E>                     VALUE OF FUNCTION RECENTLY CALLED
PAUSE                          STOP EXECUTION (BREAKPOINT)
DONE                           STOPS A ´DO PART´
```

```
       SEVERAL MODIFIERS MAY BE ATTACHED TO ANY STEP:
IF <E>
UNLESS <E>
WHILE <E>
UNTIL <E>
WHERE <V1> = <E1> & <V2> = <E2>
FOR <V> = <E>
          <E> BY <E> TO    <E>  (´BY´ MAY BE OMITTED)
                     WHILE
                     UNTIL
          (AND ANY STRING OF THE ABOVE SEPARATED BY COMMAS)


       EXPRESSIONS (<E>) CONSIST OF OPERANDS & OPERATORS:
¬                              REPLACEMENT
AND OR                         SAME AS *, + EXCEPT FOR PRECEDENCE
NOT                            CHANGES Ø TO 1, ANYTHING NOT Ø TO Ø
= # < <= > >=
+ -
* / MOD
^                              EXPONENTIATION
PRECEDENCE IS INDICATED BY THE ORDER. PARENTHESES MAY BE USED FREELY.
IF <E> THEN <E> ELSE IF <E> THEN <E> ... ELSE <E>
´WHERE´ MAY BE APPENDED TO EXPRESSIONS.



       OPERANDS:
<VARIABLES>
<NUMBERS>
PI
<SPECIAL FUNCTIONS>
<ITERATIVE FUNCTIONS>
<CAL FUNCTIONS>


       VARIABLES (<V>) ARE SINGLE LETTERS FOLLOWED BY Ø OR 1 DIGITS,
WITH ANY NUMBER OF SUBSCRIPTS. E.G.:
       X1  S(1)  X(Y(Z5),W9^2)
NO RESTRICTIONS ON SUBSCRIPTS - TAKEN TO THE NEAREST INTEGER MOD 2^23.


       NUMBERS ARE WRITTEN WITH DECIMAL DIGITS, A DECIMAL POINT, & AN
EXPONENT FIELD INDICATED BY ´E´. E.G.:
       1Ø = 1E1 = .ØØ1E4 = 1ØØØØE-Ø3


       SPECIAL FUNCTIONS:
SIN, COS, TAN                  ARG IN RADIANS
ATAN <E,Y>                     RESULT IN RADIANS
LOG, LOG1Ø, EXP, SQRT, ABS
IP (INTEGER PART), FP (FRACTIONAL PART)
ARGUMENT NEED NOT BE ENCLOSED IN PARENTHESES UNLESS IT HAS AN OPERATOR
OF LOWER PRECEDENCE THAN *.
```

ITERATIVE FUNCTIONS:
SUM, PROD, MAX, MIN

CAL FUNCTIONS:
A CAL FUNCTION IS NAMED BY A VARIABLE, WHICH CANNOT THEN HAVE NUMERIC VALUES. IT IS CALLED THUS:

```
       F[A, 16.3*W(3)]
```

INPUT: YOU ARE EDITING THE STATEMENT LAST TYPED IN. CHARACTERS TYPED REPLACE OLD ONES. CONTROL CHARACTERS:

```
CONTROL-A      DELETE PRECEDING CHARACTER & PRINT ^
        W      DELETE PRECEDING WORD & PRINT \
        Q      DELETE PRECEDING LINE & PRINT ¬
        C      COPY A CHARACTER
        S      SKIP A CHARACTER & PRINT %
        Z<X>  COPY UP THROUGH CHARACTER <X>
        X<X>  COPY UP TO CHARACTER <X>
        R      RETYPE UNALIGNED
        T      RETYPE ALIGNED
        Y      COPY REST OF OLD LINE WITHOUT TYPING; START OVER
        D      COPY AND TYPE OUT REST OF OLD LINE; DONE
        F      COPY REST OF OLD LINE; DONE
        E      (1ST TIME) SWITCH TO INSERT CHARACTERS TYPED & PRINT <
        E      (2ND TIME) SWITCH BACK TO REPLACE CHARACTERS & PRINT >
        (CR)  THROW AWAY THE REST OF THE OLD LINE; DONE
```

@DECTYPE

THIS PROGRAM DOES THE EXACT SAME THING AS ´TYPE´ BUT IS SPECIALLY MODIFIED TO OUTPUT CORRECTLY ON THE DECWRITER II TERMINAL. SEE THE HELP ON ´TYPE´ FOR MORE INFORMATION.

@DDT

ACTIVATES THE INTERACTIVE XDS-940 MACHINE LANGUAGE DEBUGGING SUBSYSTEM. SEE R-11 FOR DETAILS. DDT HAS FACILITIES FOR SYMBOLIC REFERENCE TO AND TYPEOUT OF MEMORY CELLS AND CENTRAL REGISTERS. IT PERMITS THE USE OF LITERALS AS IN THE ASSEMBLER. IT CAN INSERT BREAKPOINTS INTO PROGRAMS, PERFORM A TRACE, AND SEARCH PROGRAMS FOR SPECIFIED WORDS AND EFFECTIVE ADDRESSES. THERE IS A CONDITIONAL BREAKING FACILITY. DDT CAN LOAD BOTH ABSOLUTE AND RELOCATABLE FILES IN THE FORMAT PRODUCED BY THE ASSEMBLER.

@FTC

ACTIVATES THE FORTRAN COMPILER. THIS SUBSYSTEM CONTAINS ITS OWN HELP SERVICE. IT IS A FORM OF FORTRAN II (A LITTLE FANCIER) AND IS DESCRIBED IN A FORTHCOMING MANUAL. THE COMPILER IS

ACCOMPANIED BY AN OPERATING SYSTEM (FOS) WHICH PERFORMS LOADING AND EXECUTION CONTROL.

@FOS

ACTIVATES THE FORTRAN II OPERATING SYSTEM. THIS SUBSYSTEM CONTAINS ITS OWN HELP SERVICE.

@LISP

A LISP MANUAL FOR THE 500 HAS ONLY RECENTLY BECOME AVAILABLE. (DON'T WORRY...THE LISP ITSELF IS OVER TEN YEARS OLD & THOROUGHLY DE-BUGGED.) WE HAVEN'T YET HAD TIME TO UPDATE THE HELP TEXT ON THIS SUBJECT.

@NARP

ACTIVATES NARP. SEE R-32 FOR DETAILS. NARP IS A ONE-PASS ASSEMBLER FOR THE XDS 940 WITH LITERAL, SUBPROGRAM, CONDITIONAL ASSEMBLY AND MACRO FACILITIES. THE SOURCE LANGUAGE FOR NARP, PRIMARILY A ONE-FOR-ONE REPRESENTATION OF MACHINE LANGUAGE WRITTEN IN SYMBOLIC FORM, IS VERY SIMILAR TO THAT FOR ARPAS, BUT THERE ARE NOTABLE EXCEPTIONS, MAKING IT NECESSARY TO DO A CERTAIN AMOUNT OF TRANSLITERATION TO CONVERT AN ARPAS PROGRAM TO NARP PROGRAM.

```
       MNEMONIC FOR INSTRUCTIONS:
COPY <SYMBOL>, ...              MNEMONIC FOR RCH

       DATA GENERATION:
DATA <EXPRESSION>, ...          GENERATE DATA
ASC <STRING>                    GENERATE TEXT (3 CHARACTERS/WORD)
TEXT <STRING>                   GENERATE TEXT (4 CHARACTERS/WORD)

       VALUE DECLARATION:
<SYMBOL> EQU <EXPRESSION>       EQUATE A SYMBOL TO A VALUE
<SYMBOL> EXT (<EXPRESSION>)     DEFINE A SYMBOL AS EXTERNAL
<SYMBOL> NARG                   NUMBER OF ARGUMENTS
<SYMBOL> NCHR (<CHARACTER STRING>)
                                NUMBER OF CHARACTERS
<SYMBOL> OPD <VALUE>(,<CLASS>(,<SHIFT KLUDGE>))
                                DEFINE AN OPCODE
<SYMBOL> POPD <VALUE>(,<CLASS>(,<SHIFT KLUDGE>))
                                DEFINE A PROGRAMMED OPERATOR

       ASSEMBLER CONTROL:
BES <EXPRESSION>                BLOCK ENDING SYMBOL
BSS <EXPRESSION>                BLOCK STARTING SYMBOL
END                             END OF ASSEMBLY
DEC                             INTERPRET INTEGERS AS DECIMAL
OCT                             INTERPRET INTEGERS AS OCTAL
FRGT <SYMBOL>, ...              DO NOT OUTPUT A SPECIFIC SYMBOL
<SYMBOL> IDENT                  IDENTIFICATION OF A PACKAGE
DELSYM                          DO NOT OUTPUT ANY SYMBOLS
```

```
RELORG <EXPRESSION>               ASSEMBLE RELATIVE WITH ABSOLUTE ORIGIN
RETREL                            RETURN TO RELOCATABLE ASSEMBLY
FREEZE                            PRESERVE SYMBOLS, OPCODES, AND MACROS

       OUTPUT AND LISTING CONTROL:
LIST (<SYMBOL>, ...)              SET LISTING CONTROLS
NOLIST (<SYMBOL>, ...)            RESET LISTING CONTROLS
PAGE (<EXPRESSION>)               BEGIN A NEW PAGE ON THE LISTING
REM <TEXT>                        TYPE OUT REMARK

       CONDITIONAL ASSEMBLY AND MACROS:
IF <EXPRESSION>                   BEGIN IF BODY
ELSF <EXPRESSION>                 ALTERNATIVE IF BODY
ELSE                              ALTERNATIVE IF BODY
ENDF                              END IF BODY
RPT <EXPRESSION>(,<INCREMENT LIST>)
                                  BEGIN REPEAT BODY
CRPT <EXPRESSION>(,<INCREMENT LIST>)
                                  BEGIN CONDITIONAL REPEAT BODY
ENDR                              END REPEAT BODY
<NAME> MACRO (<DUMMY>(,<GENERATED>,<EXPRESSION>))
                                  BEGIN MACRO BODY
LMACRO (SAME AS MACRO)            ALTERNATIVE TO MACRO
ENDM                              END MACRO BODY
```

@PFS

INITIATES THE PERMANENT FILE SUBSYSTEM WHICH ALLOWS THE USER TO STORE AND RETRIEVE FILES ON DISK OUTSIDE THE CONTROL OF THE OPERATING SYSTEM, USING THE DISK AS A CLASSICAL I/O DEVICE. THROUGH THIS SYSTEM THE USER CAN PROTECT HIS FILES FROM A SYSTEM MALFUNCTION. RETRIEVE ´@SOFTWARE:PFS-MANUAL´ FROM PFS FOR DETAILS BY FIRST BEING IN EXECUTIVE:

```
@PFS
&READ DISK FILE @SOFTWARE:PFS-MANUAL;
11842 WORDS.
&FINISHED.
@TYPE     (WE MUST USE TYPE SUBSYSTEM TO SEE THE TEXT)
FILE: PFS-MANUAL
```

THE MOST FREQUENTLY USED COMMANDS ARE:

```
&READ DISK FILE <NAME>;          COPY FILE FROM DISK INTO SYSTEM
&WRITE DISK FILE <NAME>;         COPY FILE FROM SYSTEM ONTO DISK
&LIST.                           LIST PFS DIRECTORY FOR USER
&CHANGE FILE NAME <OLD-NAME>.    OBVIOUS
 TO <NEW NAME>.
&DELETE FILE <FILE-NAME>.        OBVIOUS
&FINISH.                         EXIT SUBSYSTEM
```

@PRINT <FILE-NAME>,<SPACING>,<LINE-NUMBERING>

PRINTS SYMBOLIC FILES ON THE IOMEC LINE PRINTER LOCATED IN HOLMES 486. OPTIONS ARE:

```
<SPACING>            S     SINGLE (DEFAULT)
                     D     DOUBLE
                     A     NO PAGE BREAKS
<LINE-NUMBERING>     N     NO (DEFAULT)
                     Y     YES, NUMBER THE LINES
OMIT ALL OPERANDS    ENTER PRINT SUB-SYSTEM.  PRINT HAS A SUB-COMMAND
                     STRUCTURE TO CONTROL THE QUEUE OF SYMBOLIC FILES
                     FOR THE LINE PRINTER. YOU CAN ESCAPE FROM PRINT
                     WITH A "FINISH" COMMAND. TYPE "HELP" ONCE YOU´RE
                     IN THE PRINT SUBSYSTEM FOR DETAILS.
```

```
@QED
       ACTIVATES THE SYMBOLIC TEXT EDITING SUBSYSTEM. SEE R-15 FOR DE-
       TAILS. QED HAS FACILITIES FOR INSERTING, DELETING AND CHANGING
       LINES OF TEXT STORED WITHIN IT. THE TEXT MAY HAVE BEEN TYPED INTO
       QED OR IT MAY HAVE BEEN COPIED IN FROM AN EXISTING SYMBOLIC FILE.
       BEFORE QED IS TERMINATED, IT MUST BE TOLD TO 'WRITE' ITS TEXT OUT
       ON SOME FILE. IF THAT FILE IS THE SAME ONE USED INITIALLY TO LOAD
       QED, THEN THE FILE WILL BE CHANGED. IT IS IMPORTANT TO UNDERSTAND
       THAT QED WORKS ON A COPY OF A FILE RATHER THAN ON THE FILE
       ITSELF. QED CONTAINS A LINE EDIT FEATURE, A SYMBOLIC SEARCH FEA-
       TURE, AUTOMATIC TABS WHICH MAY BE SET BY THE USER AND 36 STRING
       BUFFERS. A SUBSTITUTE COMMAND PERMITS ALL OCCURENCES OF A SPECI-
       FIED STRING OF CHARACTERS TO BE REPLACED WITH ANOTHER.
       GENERAL RULES FOR USE ARE:

CONTROL-K IS THE ESCAPE CHARACTER.
FIRST COMMAND MUST BE A 'READ' OR AN 'APPEND' TO LOAD QED'S TEXT COPY.
WHEN TEXT HAS BEEN ENTERED INTO QED, THE FOLLOWING COMMANDS MAY BE USED:

       INPUT/OUTPUT COMMANDS, <ADDR> DEFAULTS TO LINE IN PARENTHESES
(SEE BELOW FOR <ADDR> OPTIONS):
*<ADDR>READ FROM <FILE>.        READ FILE & PLACE IN QED'S TEXT COPY
                                BELOW (EXISTING TEXT) ADDRESS
*<ADDR1>,<ADDR2>WRITE ON <FILE>.  WRITE (ENTIRE) LINE(S) OF TEXT ON FILE
*<ADDR>APPEND.                  PLACE FOLLOWING TERMINAL INPUT BELOW
                                (EXISTING TEXT) ADDRESS
*<ADDR>INSERT.                  INSERT FOLLOWING TERMINAL INPUT BEFORE
                                (CURRENT) LINE(S)
*<ADDR1>,<ADDR2>CHANGE.         REPLACE (CURRENT) LINE(S) WITH THE
                                FOLLOWING TERMINAL INPUT TEXT
       FOLLOWING COMMANDS ARE VALID DURING APPEND, INSERT, & CHANGE:
       CONTROL-D                STOP QED FROM PLACING TERMINAL INPUT
                                INTO TEXT (STOP APPEND, INSERT, & CHANGE)
       CONTROL-A (TYPES ^)      BACKSPACE ONE CHARACTER
       CONTROL-W (TYPES \)      BACKSPACE ONE WORD
       CONTROL-Q (TYPES ¬)      BACKSPACE ONE LINE
       (CR)                     START NEW LINE
*<ADDR1>,<ADDR2>DELETE.         DELETE (CURRENT) LINE(S)
*<ADDR1>,<ADDR2>PRINT.          PRINT (CURRENT) LINE(S), PAGE FORMATTED
*<ADDR>,<ADDR>/                 PRINT (CURRENT) LINE(S)
*(LF)                           PRINT NEXT LINE
*^                              PRINT PREVIOUS LINE
*FINISHED.                      FINISH & EXIT QED SUBSYSTEM

       TEXT LINE ADDRESSING - <ADDR>:
<NUMBER>                        FIRST LINE OF TEXT IS NUMBERED 1
   .                            CURRENT LINE OF TEXT
   $                            LAST LINE OF TEXT
   @                            ENTIRE COPY OF TEXT = 1,$
[<TEXT>]                        ARBITRARY TEXT (SEQUENCE OF CHARACTERS)
:<LABEL>:                       LABEL (FIRST TEXT OF LINE, FOLLOWED BY
                                CHARACTER OTHER THAN LETTER OR DIGIT)
<ADDR> IS ANY LEGAL ADDRESS FROM THE ABOVE ADDRESSING COMMANDS;
       <ADDR1>,<ADDR2>          ADDR1 THROUGH ADDR2
       <ADDR1>+/-<ADDR2>        START AT <ADDR1> & SEARCH FOR NEXT(+)
                               OR PREVIOUS(-) <ADDR2>; SPACE MAY BE
                               TYPED IN PLACE OF ´+´; SEARCH WILL WRAP-
                               AROUND

      COMMANDS TO MODIFY LINES, <ADDR> DEFAULTS TO ():
*<ADDR1>,<ADDR2>EDIT.          PRINT & ALLOW EDIT OF (CURRENT) LINE(S)
*<ADDR1>,<ADDR2>MODIFY.        EDIT (CURRENT) LINE(S)

       CONTROL CHARACTERS FOR TEXT ´EDIT´ & ´MODIFY´.  QED KEEPS A
COPY OF THE LINE BEING EDITED UNTIL THE EDIT IS TERMINATED.
BACKSPACE ONE CHARACTER        CONTROL-A (TYPES ^)
          ONE WORD             CONTROL-W (TYPES \)
          ONE LINE             CONTROL-Q (TYPES ¬)
          ONE CHARACTER,       CONTROL-N (TYPES ^)
            RESTORATIVE
TERMINATE LINE OF EDIT/MODIFY  (CR) / CONTROL-M
COPY UP TO CHARACTER <X>       CONTROL-O <X>
     THROUGH CHARACTER <X>     CONTROL-Z <X>
     REST OF LINE & TERMINATE  CONTROL-D
     REST OF LINE (NOT TYPED)  CONTROL-F
       & TERMINATE
     ONE CHARACTER             CONTROL-C
     TO TAB STOP               CONTROL-U
     TO END OF LINE            CONTROL-H
SKIP ONE CHARACTER             CONTROL-S
     UP TO CHARACTER <X>       CONTROL-P <X>
     THROUGH CHARACTER <X>     CONTROL-X <X>
MODE CHANGE - INSERT/REPLACE   CONTROL-E (TYPES ´<´ FOR INSERT,
     (NORMAL MODE IS REPLACE)             ´>´ FOR REPLACE)
CONCATENATE-RE-EDIT            CONTROL-Y
TAKE CHARACTER <X> LITERALLY   CONTROL-V <X>
RETYPE ALIGNED                 CONTROL-T
RETYPE UNALIGNED               CONTROL-R

*<ADDR>,<ADDR>SUBSTITUTE(<OPTIONS>)/<NEW-TEXT>/FOR/<OLD-TEXT>/
                               SUBSTITUTES <NEW-TEXT> FOR <OLD-TEXT> IN
                               (CURRENT) LINE(S)
       SUBSTITUTE OPTIONS:
       DEFAULT      MAKE ALL SUBSTITUTIONS
       :L           PRINT LINE AFTER SUBSTITUTIONS
       :W           PRINT LINE BEFORE SUBST., ´S´ CAUSES SUBST., NEW
                    OPTION CAUSES OPTION CHANGE, ANY OTHER CHARACTER
                    CAUSES NO SUBST., & THEN CONTINUE
       :V           :W & :L COMBINED
       :<NUMBER>    MAXIMUM NUMBER OF SUBSTITUTIONS TO MAKE
       :G           NOT PRINT AFTER SUBSTITUTION

       OTHER USEFUL COMMANDS:
*<ADDR>=                       GIVE LINE NUMBER
*<ADDR>¬                       GIVE SYMBOLIC ADDRESS
*TABS.                         FOLLOWING INPUT IS <13 DECIMAL NUMBERS
                               FOR TAB POSITIONS
CONTROL-I                      TAB
```

QED HAS 36 BUFFERS, NAMED Ø-9 & A-Z, FOR HOLDING USER´S STRINGS OF CHARACTERS. QED USES Ø & 1 FOR SEARCHES, SUBSTITUTES, & CONTROL-L.

```
BUFFER CONTROL COMMANDS:
*<ADDR1>,<ADDR2>LOAD #<BUFFER>.  LOAD BUFFER WITH (CURRENT) LINE(S)
*<ADDR1>,<ADDR2>GET #<BUFFER>.  LOAD BUFFER WITH (CURRENT) LINE(S) &
                               DELETE FROM TEXT
*JAM INTO #<BUFFER>.           LOAD FOLLOWING TERMINAL INPUT INTO BUFFER
                               [CONTROL-D TERMINATES]
*BUFFER #<BUFFER>.             PRINT CONTENTS OF BUFFER
*KILL #<BUFFER>.               CLEAR BUFFER
       SPECIAL BUFFER COMMANDS:
       CONTROL-B <BUFFER>      INSERT BUFFER TEXT INTO QED TEXT
       CONTROL-L               LOAD BUFFER 1 WITH COPY OF TEXT IMMED-
                               IATELY FOLLOWING UNTIL ANOTHER CONTROL-L
```

@QRUN

ACTIVATES THE QSPL SUBSYSTEM WHICH RUNS QSPL PROGRAMS. SEE R-28 FOR DETAILS. THE QSPL LANGUAGE IS INTENDED TO BE A SUITABLE VEHICLE FOR PROGRAMS WHICH WOULD OTHERWISE BE WRITTEN IN XDS-94Ø MACHINE LANGUAGE FOR REASONS OF EFFICIENCY OR FLEXIBILITY. IT IS PART OF A SYSTEM WHICH INCLUDES A COMPILER CALLED QSPL AND THIS RUNTIME WHICH IMPLEMENTS THE INPUT-OUTPUT AND STRING-HANDLING FEATURES OF THE LANGUAGE AS WELL AS A FAIRLY ELABORATE STORAGE ALLOCATOR.

CONTROL-K IS THE ESCAPE CHARACTER.

QRUN IS ACTUALLY A MODIFIED VERSION OF THE 94Ø DDT; IT HAS ADDED KNOWLEDGE ABOUT CERTAIN QSPL CONSTRUCTS SUCH AS THE "CALL" POPCODE DEFINITION, AND SEVERAL RUNTIME SYMBOL LOCATIONS. (QRUN SETS SEVERAL POP DEFINITIONS FOR THE USER.) IN ADDITION, IT CONTAINS (PRE-LOADED AT 34ØØØB) THE CODE WHICH IMPLEMENTS THE QSPL RUNTIME. QRUN SHOULD BE USED FOR DEBUGGING OF QSPL PROGRAMS WHICH USE THE RUNTIME FEATURES, OTHERWISE 94Ø DDT WILL BE ADEQUATE (THE ONLY POP WHICH WILL BE REQUIRED IS FOR "CALL" WHICH CAN BE IMPLEMENTED BY A "BRU* Ø" AT LOCATION 1ØØB).

@QSPL

ACTIVATES THE QSPL SUBSYSTEM WHICH COMPILES QSPL PROGRAMS. SEE R-28 FOR DETAILS. THE QSPL LANGUAGE IS INTENDED TO BE A SUITABLE VEHICLE FOR PROGRAMS WHICH WOULD OTHERWISE BE WRITTEN IN XDS-94Ø MACHINE LANGUAGE FOR REASONS OF EFFICIENCY OR FLEXIBILITY. IT IS PART OF A SYSTEM WHICH INCLUDES THIS COMPILER CAPABLE OF PRODUCING REASONABLY EFFICIENT OBJECT CODE AND A RUNTIME CALLED QRUN.

CONTROL-K IS THE ESCAPE CHARACTER.

A PROGRAM IS COMPILED WITH THE FOLLOWING ACTIONS:

```
12/18/69                        DATE OF LAST QSPL ASSEMBLY
SOURCE FILE: <FILE-NAME> <TERMINATING-CHARACTER>
                                <TERMINATING-CHARACTER> IS EITHER "."
                                - ASSUME DEFAULT FILE FOR BINARY AND
                                LISTING, OR "," OR ";" - DEMAND FURTHER
                                FILE NAMES
BINARY FILE: <FILE-NAME> <TERMINATING-CHARACTER>  DEFAULT IS "NOTHING"
LISTING FILE: <FILE-NAME>.      DEFAULT IS "NOTHING"
```

THE LISTING FILE CAN BE USED AS INPUT TO THE NARP ASSEMBLER AND IS THUS AVAILABLE FOR MODIFICATION BY THE USER.

@RUNOFF

INITIATES A TEXT PROCESSING SYSTEM WHICH PERMITS NEATLY FORMATTED DOCUMENTS TO BE PRODUCED BY COMPUTER. IT FEATURES AUTOMATIC PAGE NUMBERING, HYPHENATION, RIGHT MARGIN JUSTIFICATION, UNDERLINING, CENTERING UPON SPECIFIC COMMANDS. A MUST FOR STUDENTS OR FACULTY WHO MUST WRITE A TERMPAPER, THESIS OR EVEN A BOOK. FOR DETAILS SEE THE RUNOFF MANUAL STORED IN PFS UNDER @SOFTWARE:. (17 PAGES!) IT MAY BE RETRIEVED AS FOLLOWS:

```
@PFS
&READ DISK FILE @SOFTWARE:RUNOFF-MANUAL;
1262Ø WORDS.
&FINISHED.
@TYPE
FILE: RUNOFF-MANUAL
```

@SNOBOL

ACTIVATES THE XDS-940 SNOBOL4 SYBSYSTEM. SEE R-34 FOR DETAILS. THE XDS-940 SNOBOL4 SYSTEM WILL ACCEPT PROGRAMS WRITTEN IN A LANGUAGE WHICH IS COMPATIBLE WITH A SUBSET OF BELL LABS´ NOV., 1967, VERSION OF SNOBOL4. SNOBOL4 PERMITS PROGRAMS TO BE CREATED, RUN AND DEBUGGED INTERACTIVELY. THE DATA OBJECT IS A STRING OF CHARACATERS. THE LANGUAGE PERMITS BUILDING UP LONGER STRINGS FROM SHORTER STRINGS THROUGH CONCATENATIONS. THROUGH PATTERN MATCHING, STRINGS CAN HAVE THEIR CONTENTS TESTED AND HAVE THE MATCHED SUBSTRINGS ASSIGNED TO STRING VARIABLES. OTHER FEATURES ARE ARITHMETIC ON INTEGER STRINGS, BUILT-IN FUNCTIONS, AND PROGRAMMER DEFINED FUNCTIONS WHICH MAY HAVE LOCAL VARIABLES AND CAN BE RECURSIVE TO ARBITRARY DEPTH. I/O FROM FILES AND TELETYPE IS PROVIDED.

THE SNOBOL SYSTEM IS DIVIDED INTO 2 PARTS: EDITOR-COMPILER AND RUNTIME. EDITOR-COMPILER IS USED TO WRITE, MODIFY, AND COMPILE SOURCE STATEMENTS. RUNTIME IS RESPONSIBLE FOR EXECUTION OF STATEMENTS.

EDITOR-COMPILER COMMANDS:
AS EACH SOURCE STATEMENT IS READ OR TYPED, IT IS COMPILED. EDITING IS DONE IMMEDIATELY. EDITOR COMMANDS AND ADDRESSING ARE LIKE QED.

RUNTIME COMMANDS:

```
$<ADDR>GO.                        BEGIN EXECUTION OF PROGRAM AT (DEFAULT -
                                  FIRST LINE) ADDRESS (CONFIRM WITH ´.´)
$<ADDR1>,<ADDR2>BREAK.            SET BREAKPOINTS ON ALL STATEMENTS IN THE
                                  INTERVAL ADDRESSED
$<ADDR1>,<ADDR2>KILL.             RELEASE ALL BREAKPOINTS IN THE INTERVAL
                                  ADDRESSED
$<ADDR1>,<ADDR2>LIST.             PRINT ALL BREAKPOINTS THE INTERVAL
                                  ADDRESSED
$PROCEED.                         CONTINUE EXECUTION AFTER BREAKPOINT
CONTROL-K  (ONCE)                 DURING EXECUTION CAUSE A BREAK AT THE
                                  START OF THE NEXT STATEMENT; AT OTHER
                                  TIMES, EXIT SUBSYSTEM
CONTROL-K  (TWICE)                DURING EXECUTION, RETURN IMMEDIATELY
                                  TO EDITOR
$(SPACE)<UNLABELED-STATEMENT>.    EXECUTE SNOBOL STATMENT IMMEDIATELY
```

APPLICABLE QED COMMANDS (SEE QED FOR MORE INFORMATION):

/
=
¬
APPEND / INSERT
CHANGE / DELETE
EDIT / MODIFY
READ FROM / WRITE ON
FINISHED
SUBSTITUTE
TABS AND (CR)

EVERY INPUT OF A LINE CAN BE TREATED AS AN EDIT OF THE PREVIOUS LINE TYPED OR DELETED. THUS, THE EDIT COMMANDS CAN BE USED TO COPY AND MODIFY THE PREVIOUS LINE TO CREATE THE NEW LINE. CONTROL-D IS A TERMINATOR FOR INPUT ONLY WHEN NO CHARACTERS ARE IN THE INPUT LINE.

@TYPE

PROVIDES A FACILITY FOR TYPING SYMBOLIC FILES AT THE TERMINAL, AS OPPOSED TO PRINTING THEM ON THE LINE PRINTER. QED PROVIDES A SIMILAR FACILITY, EXCEPT THAT A RUNOFF-PRODUCED FILE CANNOT BE HANDLED BY QED BECAUSE QED DOES NOT RECOGNIZE LOWER CASE LETTERS. TYPE EXISTS FOR THIS PURPOSE; AND IF THE TERMINAL IS CAPABLE OF UPPER/LOWER CASE, THEN BOTH CASES WILL BE HANDLED. OTHERWISE THE TEXT WILL COME OUT IN UPPER CASE ONLY. TYPE EXPECTS A FILE NAME, BUT IT ASKS FOR THE NAME EXPLICITLY, RATHER THAN TAKING THE NAME AS A COMMAND ARGUMENT.