

TABLE OF CONTENTS

PREFACE	V
<u>OPERATION OF THE TIME SHARING SYSTEM.</u>	
OPERATING THE TSS MCP	I- 2
SYSDISK/MAKER	I- 5
KEYBOARD INPUT MESSAGES	I- 6
THE OPTIONS	I- 15
PROGRAMMING LANGUAGES	I- 17
COBOL	I- 19
FORTRAN	I- 22
TSPOL	I- 26
FILE SECURITY	I- 27
USER/CANDE.	I- 28
USERS/CANDE	I- 31
THE LOG	I- 33
MESSAGES APPEARING IN THE LOG	I- 36
<u>PROGRAMMING INFORMATION</u>	
THE MCP	II- 2
SEGMENTED SAVE AREAS.	II- 3
MCP TABLES.	II- 5
UV ARRAY.	II- 6
TABLES USED IN SWAPPING	II- 8
TABLES USED BY LINE MAINTENANCE	II- 10
THE FORK QUEUE; ITS STRUCTURE AND OPERATION	II- 17
THE BED; SLEEPING AND WAKING.	II- 20
ASSIGNING JOBS TO CORE.	II- 23

SWAPPING.	II- 24
SWAPPING STATUSES	II- 29
THE SWAPPING PROCEDURES	II- 31
EXPANDING AN AREA	II- 34
LINE MAINTENANCE.	II- 35
LINE MAINTENANCE PROCEDURES.	II- 40
SYSTEM/DISK, AND HALT/LOAD RECOVERY	II- 52
MOVING THE FENCE.	II- 54
OPERATION OF CANDE.	II- 55
THE WORK FILE	II- 66
THE NEGATIVE COMMUNICATES	II- 69
DESCREANCIES BETWEEN THE INITIAL RELEASE AND THIS MANUAL	A- 1

PREFACE

THIS DOCUMENT IS DIVIDED INTO TWO PARTS. PART I, OPERATING THE B5500 TIME SHARING SYSTEM, DESCRIBES THE DIFFERENCES BETWEEN THE TIME SHARING SYSTEM AND THE STANDARD SYSTEM. THE READER IS ASSUMED TO BE FAMILIAR WITH OPERATION OF THE STANDARD B5500 SYSTEM AS DESCRIBED IN THE OPERATIONS MANUAL. PART II, PROGRAMMING INFORMATION, CONTAINS DETAILED DESCRIPTIONS OF THE PROCEDURES AND PROGRAMS IN THE TIME SHARING SYSTEM WHICH ARE NOT A PART OF THE STANDARD SYSTEM. SINCE IT IS INTENDED PRIMARILY FOR THE SYSTEMS PROGRAMMER, THE READER IS EXPECTED TO BE FAMILIAR WITH THE CODING OF THE STANDARD MCP.

PART I

OPERATION OF THE TIME SHARING SYSTEM

OPERATING THE TSS MCP

THE OPERATION OF THE TSS MCP IS QUITE SIMILAR TO THAT OF THE STANDARD MCP. THE CHANGES THAT HAVE BEEN MADE TO ALLOW TIME SHARING ARE MOSTLY ADDITIONS. FOR INSTANCE, MOST OF THE KEYBOARD INPUT MESSAGES HAVE BEEN RETAINED AND SEVERAL NEW ONES HAVE BEEN ADDED. THUS, ANYONE FAMILIAR WITH THE OPERATION OF THE STANDARD MCP SHOULD HAVE NO TROUBLE OPERATING THE TSS MCP.

IN ORDER TO USE REMOTE TERMINAL, THE DCTU MUST BE IN REMOTE AND THE FILE SYSTEM/DISK, CONTAINING INFORMATION ABOUT THE HARDWARE SET UP OF EACH LINE, MUST BE PRESENT. IF EITHER OF THESE CONDITIONS IS NOT MET AFTER A HALT/LOAD OR WHEN SETTING THE REMOTE OPTION (SEE "OPTIONS"), THE REMOTE OPTION IS RESET AND AN APPROPRIATE MESSAGE, EITHER

-*NO SYSTEM DISK

OR

-*DTC NOT READY

IS PRINTED AT THE SPD, THE FILE SYSTEM DISK CAN BE CREATED BY RUNNING THE PROGRAM SYSDISK/MAKER.

AS ON THE STANDARD SYSTEM, THE INTRINSICS FILE, INTRNSC/DISK, SHOULD ALSO BE ON DISK WHEN A HALT/LOAD IS DONE. TO COMPILE THE INTRINSICS, THE FOLLOWING \$ CARD SHOULD BE USED:

\$ SET TIMESHARING = TRUE

IF THIS OPTION IS NOT SET, OR IS SET FALSE, THE RESULTING INTRINSICS WILL BE THOSE FOR THE STANDARD SYSTEM.

AFTER THE HALT/LOAD HAS BEEN PERFORMED, IN ORDER TO DO ANY TIME-SHARING THE PROGRAM CANDE/TSHARER MUST BE RUNNING. THIS PROGRAM IS INITIATED BY TYPING THE CE MESSAGE. WHEN CE IS TYPED, THE OBJECT VERSIONS OF THE CANDE FILES MUST BE ON DISK. THESE FILES ARE:

<u>SOURCE</u>	<u>OBJECT</u>
SOURCE/CANDE	CANDE/TSHARER
MERGSC/CANDE	MERGE/CANDE
LOADSC/CANDE	LOAD/CANDE
APNDSC/CANDE	APPEND/CANDE
RSEQSC/CANDE	RESEQ/CANDE
DLETSC/CANDE	DELETE/CANDE
COPYSC/CANDE	COPY/CANDE
LISTSC/CANDE	LIST/CANDE
GARDSC/CANDE	GUARD/DISK
FILESC/CANDE	GETFIL/CANDE
PAPRSC/CANDE	PAPER/CANDE

IN ADDITION, THE FILE

MESSAGE/CANDE

WHICH CONTAINS THE ERROR MESSAGES MUST ALSO BE PRESENT ON DISK. ALL CANDE PROGRAMS ARE COMPILED IN TSPOL.

THE REMAINING PROGRAM IN THE CANDE LIBRARY IS USERSC/CANDE WHICH

COMPILES INTO USER/CANDE. THIS PROGRAM CREATES AND UPDATES THE FILE
USERS/CANDE WHICH CONTAINS THE LIST OF QUALIFIED USERS AND IS DESCRIBED
ELSEWHERE.

THE CORE FACTOR APPLIES ONLY TO THE AREA BELOW THE FENCE. IT SHOULD
BE SET TO 1 AND THEN LEFT ALONE.

RUNNING BATCH JOBS

JOB'S MAY BE ENTERED AND RUN VIA THE CARD READER IN THE USUAL WAY.
WHEN AN EXECUTE CARD IS USED, JOBS ARE RUN ABOVE THE FENCE AND ARE
THEREFORE SUBJECT SWAPPING WITH OTHER JOBS. AS A RESULT, CARE MUST
BE TAKEN TO ENSURE THAT SUCH JOBS DO NOT SERIOUSLY DEGRADE THE RESPONSE
TO THE USER. JOBS CAN BE RUN BELOW THE FENCE BY USING A RUN CARD.
HOWEVER, EXCEPT FOR A FEW SPECIAL CASES SUCH AS LDCNTRL/DISK (WHICH
IS BEST STARTED BY AN LDDK MESSAGE ANYWAY), RUNNING JOBS BELOW THE
FENCE CAUSES A PLETHORA OF NO MEM-S AND SERIOUSLY HAMPERS THE MCP AND
CANDE.

SYSDISK/MAKER

THE PROGRAM SYSDISK/MAKER IS USED TO CREATE THE FILE SYSTEM/DISK. THE INFORMATION NEEDED FOR EACH LINE IS READ IN FREE FIELD FORMAT FROM THE CARD FILE PARAM. THE PARAMETERS WHICH ARE READ ARE:

TU TERMINAL UNIT NUMBER

BUF BUFFER NUMBER

PNGPNG PING-PING FLAG. ENTER A 1 IF THIS BUFFER IS A PING-PING BUFFER AND ZERO OTHERWISE.

GMFLAG GROUP MARK FLAG. MUST BE 0 FOR A 980 ADAPTER.

BUFSIZE BUFFER SIZE FOR DATA-COM READS. MUST BE 28, 56, OR 112.

ADTYP ADAPTER TYPE. ENTER A 1 FOR A 980.

TERMTYP REMOTE TERMINAL TYPE. MUST BE A 1 FOR MODEL 33/35 TELETYPE.

XLATE TRANSLATE FLAG. MUST BE 0 FOR A 980 ADAPTER.

DIRECT DIRECT CONNECTION FLAG. ENTER A 1 IF THIS LINE IS DIRECTLY CONNECTED AND A 0 IF IT IS A DIAL-UP.

THE INPUT MUST BE ARRANGED IN ASCENDING ORDER OF TERMINAL UNIT AND BUFFER NUMBERS, WITH BUFFER NUMBER THE FASTER MOVING.

SYSDISK/MAKER SIMPLY READS THE DATA FOR ONE LINE INTO AN ARRAY AND THEN WRITES THAT DATA ON DISK, CONTINUING UNTIL IT REACHES THE END OF THE FILE. CURRENTLY IT DOES NOT PERFORM ANY ERROR CHECKING. HOWEVER, A MORE SOPHISTICATED VERSION WILL BE PROVIDED IN THE NEAR FUTURE.

KEYBOARD INPUT MESSAGES

THE KEYBOARD INPUT MESSAGES WHICH ARE THE SAME IN BOTH THE STANDARD MCP AND THE TIME SHARING MCP ARE:

AX	EI	LS	PI	RY	TR
CC	ES	MX	PR	SF	TS
CD	EX	OF	PS	SO	UL
CI	FM	OK	QT	ST	WD
CT	FR	OL	RD	SV	WM
DB	IL	OU	RM	TF	WY
DD	LC	PB	RN	TI	XI
DP	LD	PD	RO	TL	XS
DT	LF	PG	RW	TO	XT
ED	LR				

THE MESSAGES IN THE STANDARD SYSTEM WHICH ARE NOT INCLUDED IN THE TIME SHARING SYSTEM ARE:

BO	LI	QV	SM	WA	WR
HM	LO	RR	TC	WP	ZZ
HR	PT	RS			

THE DS, IN AND QT MESSAGES ARE INCLUDED WITH RESTRICTIONS.

DS CANNOT BE USED IN THE FORM DS<PROGRAM SPECIFIER>. ONLY
<MIX INDEX>DS IS ALLOWED.

IN CAN BE USED ONLY WITH JOBS RUNNING BELOW THE FENCE.

QT CAN BE USED ONLY WITH JOBS RUNNING BELOW THE FENCE.

THE MESSAGES WHICH HAVE BEEN MODIFIED FOR TIME SHARING ARE BS, CL, LN, SC, SS, US AND WU AND THE MESSAGES WHICH HAVE BEEN ADDED ARE CE, CX, MF, MU, AND RS. THESE MESSAGES ARE DESCRIBED BELOW.

THE BS MESSAGE

THE BS MESSAGE IS USED TO DESIGNATE A TELETYPE AS A SPO, OR TO RESTORE OUTPUT TO THE REAL SPO AFTER IT HAS BEEN TURNED OFF BY A US MESSAGE. THE FORMATS OF THE BS MESSAGE ARE

BS SPO

BS <LOGICAL LINE NUMBER>

ONLY ONE TELETYPE CAN BE USED AS A SPO AT A TIME. THE BREAK KEY ON THE ALTERNATE SPO IS EQUIVALENT TO THE INPUT REQUEST KEY ON THE REAL SPO. THE REAL SPO MAY RUN CONCURRENTLY WITH THE ALTERNATE OR MAY BE UN-SPO-ED.

THE CE MESSAGE

THE CE MESSAGE IS USED TO INITIATE CANDE/TSHARER. ITS FORMAT IS

CE

THE CL MESSAGE

THE CL MESSAGE CAN BE USED TO CLEAR A LINE OR TO CLEAR A PERIPHERAL UNIT. WHEN A UNIT IS CLEARED, THE JOB WHICH IS USING THAT LINE, IF ANY, IS DS-ED. WHEN A LINE IS CLEARED, THE LINE IS DISCONNECTED AND

IF THE USER WAS RUNNING A JOB, IT IS DS=ED. THE FORMATS ARE:

CL <UNIT MNEMONIC>

CL <LOGICAL LINE NUMBER>

THE CX MESSAGE

THE CX MESSAGE IS USED TO SEND MESSAGES TO CANDE. THE MESSAGES ARE TREATED LIKE INPUT FROM ANY OTHER LINE AND MUST THEREFORE CONFORM TO THE RULES FOR THE COMMAND AND EDIT LANGUAGE. THE FORMAT IS

CX <MESSAGE>

THE LN MESSAGE

THE LN MESSAGES CAUSES THE NAME OF THE FILE LOG/DISK TO BE CHANGED AND ALSO CREATES A NEW FILE LOG/DISK, ITS FORMAT IS:

LN

THE LNDK MESSAGE

AFTER AN LNDK MESSAGE, A DISK CHARGES MESSAGE IS ENTERED IN THE LOG FOR EACH FILE ON DISK AND THE CREATION DATE AND THE TIME FOR EACH FILE IS UPDATED. THE FORMAT IS

LNDK

THE MF MESSAGE

THE MF MESSAGE, WHICH IS USED TO CHANGE THE LOCATION OF THE FENCE,
HAS THE FORMAT

MF <FENCE LOCATION>

WHERE <FENCE LOCATION> IS AN INTEGER BETWEEN 8184 AND 28644. THE
LOCATION OF THE FENCE WILL NOT BE CHANGED UNTIL THE NEXT HALT/LOAD,
AT WHICH TIME IT WILL BE PLACED AT THE SPECIFIED LOCATION.

THE MU MESSAGE

THE MU MESSAGE IS USED TO SPECIFY THE MAXIMUM NUMBER OF USERS ALLOWED
ON THE SYSTEM AT ONE TIME. IF A USER ATTEMPTS TO DIAL-IN WHEN THE
NUMBER OF USERS IS GREATER THAN OR EQUAL TO THE MAXIMUM, HE IS
DISCONNECTED. USERS ALREADY LOGGED ON ARE NOT AFFECTED BY AN MU MESSAGE.
THE FORMATS ARE:

MU

MU <NUMBER OF USERS>

IF A NUMBER OF USERS IS NOT GIVEN, THE NUMBER OF LINES PHYSICALLY
ATTACHED TO THE SYSTEM IS ASSUMED.

THE RS MESSAGE

THE RS MESSAGE IS IDENTICAL TO THE SS MESSAGE EXCEPT THAT IT BYPASSES
CANDE AND CAUSES THE MESSAGE TO THE USER TO BE OUTPUT WITHOUT REGARD
FOR WHAT THE USER IS DOING. IT SHOULD BE USED ONLY WHEN THE MESSAGE
NEEDS TO BE SENT IMMEDIATELY. ITS FORMATS ARE

RS <LOGICAL LINE NUMBER> <MESSAGE>

RS <USER CODE> <MESSAGE>

RS ALL <MESSAGE>

THE RESTRICTIONS CONCERNING THE SEPARATION OF THE LINE NUMBER OR USER CODE FROM THE MESSAGE ARE THE SAME AS FOR THE SS MESSAGE.

THE SC MESSAGE

THE SC MESSAGE IS USED TO FIND OUT WHICH UNITS ARE DESIGNATED AS SPO-
S. THE FORMAT IS

SC

THE SM MESSAGE

DEPENDING ON ITS FORMAT, THE SM MESSAGE IS USED TO OBTAIN A SUMMARY OF THE ACTIVITIES ON THE SYSTEM, ON A GIVEN LINE, OR FOR A GIVEN JOB. IF THE FORMAT IS

SM

A SYSTEM SUMMARY WILL BE PROVIDED AS FOLLOWS:

- 1) FOR EACH USER ON THE SYSTEM, ONE OF THE FOLLOWING TWO MESSAGES WILL BE TYPED DEPENDING ON WHETHER OR NOT THE USER IS CONNECTED TO A NORMAL STATE PROGRAM.

<USER CODE> ON LINE <LOGICAL LINE NUMBER>

<USER CODE> ON LINE <LOGICAL LINE NUMBER> USING<JOB SPECIFIER>

PST=<PROCESSOR TIME> IN <ELAPSED TIME>

2) FOR EVERY JOB IN THE MIX WHICH IS NOT CONNECTED TO A REMOTE USER, THE FOLLOWING IS TYPED.

<USER CODE> USING <JOB SPECIFIER> PST=<PROCESSOR TIME>
IN <ELAPSED TIME>

IF THERE ARE NO USERS AND NO JOBS, THE ANSWER IS

NOTHING

IF THE FORMAT IS

SM<LOGICAL LINE NUMBER>

A SUMMARY OF THE ACTIVITIES ON THAT LINE WILL BE GIVEN AS DESCRIBED IN (1).

IF THE FORMAT IS

<MIX>SM

THE SUMMARY WILL BE ONLY FOR THE JOB WITH THE SPECIFIED MIX INDEX.

THE SS MESSAGE

THE SS MESSAGE IS USED TO SEND A MESSAGE TO A REMOTE STATION OR STATIONS. IF CANDE/TSHARER IS RUNNING, IT CONTROLS THE SENDING OF THE MESSAGE TO THE TERMINAL AND, IF NECESSARY, DELAYS IT UNTIL THE MESSAGE CAN BE SENT WITHOUT BEING MIXED WITH OTHER OUTPUT FOR THAT LINE. IF CANDE IS NOT RUNNING, THE MESSAGE IS OUTPUT IMMEDIATELY.

THE SS MESSAGE HAS THREE FORMATS.

SS <LOGICAL LINE NUMBER><MESSAGE>

SS <USER CODE><MESSAGE>

SS ALL <MESSAGE>

IN THE FIRST CASE, THE <MESSAGE> WILL BE SENT TO THE SPECIFIED LINE.

IF THERE IS NO SUCH LINE ON THE SYSTEM,

LINE <LOGICAL LINE NUMBER> NOT AVAILABLE

WILL BE TYPED AT THE SPD AND IF THE LINE IS NOT DIALED-UP, THE MESSAGE TYPED IS:

LINE <LOGICAL LINE NUMBER> NOT DIALED-UP

IN THE SECOND CASE, THE <MESSAGE> WILL BE SENT TO ALL USERS WITH THE SPECIFIED <USER-CODE>. IF THERE ARE NONE, THE OUTPUT MESSAGE AT THE SPD WILL BE

<USER CODE> NOT DIALED-UP

IN THE THIRD CASE, THE <MESSAGE> WILL BE SENT TO ALL LOGGED-IN STATIONS IF CANDE IS RUNNING AND TO ALL DIALED-UP STATIONS OTHERWISE.

THE POSITION OF THE FIRST CHARACTER OF THE <MESSAGE> IS DETERMINED AS FOLLOWS. IF A <USER CODE> IS GIVEN, THE <MESSAGE> BEGINS WITH THE FIRST SPECIAL CHARACTER OR WITH THE EIGHTH CHARACTER. IF A <LOGICAL LINE NUMBER> IS USED, THE <MESSAGE> STARTS WITH THE FIRST NON-NUMERIC CHARACTER OR WITH THE THIRD CHARACTER, IF ALL IS USED, THE <MESSAGE>

BEGINS IMMEDIATELY AFTER THE SECOND L. IT IS SUGGESTED THAT THE MESSAGE ALWAYS BE STARTED WITH A BLANK, FOR EXAMPLE,

SS JONES NO SUCH TAPE

THE US MESSAGE

THE US MESSAGE "TURNS OFF" THE SPO OR AN ALTERNATE SPO. A TELETYPE IS TURNED OFF FOR BOTH INPUT AND OUTPUT. THE SPO CAN STILL BE USED FOR INPUT BUT WILL NO LONGER RECEIVE SPO OUTPUT. THE FORMATS ARE

US SPO

US <LOGICAL LINE NUMBER>

THE WU MESSAGE

THE WU MESSAGE PROVIDES A WAY TO DISCOVER WHICH USERS ARE DIALED IN, ARE ON GIVEN LINES, OR ARE ATTACHED TO GIVEN PROGRAMS. THE RESULTS OF THE WU MESSAGE DEPEND ON ITS FORMAT. IF THE FORMAT IS EITHER

WU

OR

WU<LOGICAL LINE NUMBER>

THE FOLLOWING WILL BE TYPED AT THE SPO FOR EACH USER ON THE SYSTEM OR FOR THE USER OF THE SPECIFIED LINE:

<USERCODE> IS ON LINE <LOGICAL LINE NUMBER>

IF THERE ARE NO USERS, THE MESSAGE IS

NULL WU.

IF THE FORMAT IS

<MIX> WU

THE USER CODE, LOGICAL LINE NUMBER (IF APPROPRIATE) AND JOB SPECIFIER
OF THE JOB WITH THE SPECIFIED MIX INDEX ARE OUTPUT AS

<USER CODE> ON LINE <LOGICAL LINE NUMBER> USING <JOB SPECIFIER>

THE OPTIONS

THE OPTIONS LISTED BELOW CAN BE INITIALLY SET IN THE COLD START DECK AS DESCRIBED FOR THE STANDARD MCP. THEREAFTER, THEY CAN BE SET, RESET AND TYPED BY THE SO, RO AND TO MESSAGES. OPTIONS MARKED BY ASTERISKS ARE ADDITIONS FOR THE TIME SHARING SYSTEM AND ARE DESCRIBED IN THE PARAGRAPHS FOLLOWING THE LIST.

47	USE DRA	34	TYPE CLOSE
46	USE DRB	33	TYPE ERRORMSG
45	TYPE BOJ	32	USE RET
44	TYPE EOJ	31	TYPE LIBMSG
43	TYPE OPN	30	TYPE SCHEDMSG
42	USE TERMNATE	29	TYPE SECMSG
41	TYPE DATE	28	USE DSKTOG
40	TYPE TIME	27	USE RELTOG
39 *	USE HALT	26	USE PBDREL
38	USE AUTOPRNT	25 *	TYPE CEMESS
37 *	USE REMOTE	24 *	TYPE DISKLOG
36 *	USE CHECK	23 *	TYPE DISKMSG
35	TYPE CMLPFILE	22 *	TYPE LIBERR

IF THE USE HALT OPTION IS SET WHEN A SLATE OVERFLOW OR AN INVALID ADDRESS IN THE MCP OCCURS, THE MCP WILL HANG AFTER TYPING EITHER

***SHOULD H/L: HALT CAUSED BY SLATE OVERFLOW

OR

***SHOULD H/L: HALT CAUSED BY INVALID ADDRESS

THE USE REMOTE OPTION MUST BE SET TO ALLOW DATA-COM I/O. IF THIS OPTION IS NOT SET, NO I/O TO OR FROM REMOTE UNITS WILL TAKE PLACE. THEREFORE, THIS OPTION SHOULD ALWAYS BE SET FOR NORMAL TIME SHARING OPERATION.

THE USE CHECK OPTION CAUSES A MEMORY LINK CHECK TO BE PERFORMED ON ENTRY TO AND EXIT FROM THE ROUTINE GETSPACE. IF AN INVALID LINK IS FOUND, THE SYSTEM HANGS. SINCE THIS CHECK IS RELATIVELY TIME CONSUMING, THIS OPTION IS USUALLY SET ONLY FOR DEBUGGING AND IS RESET DURING NORMAL SYSTEM OPERATION.

THE TYPE CEMESS OPTION IS USED IN CONJUNCTION WITH THE TYPE LIBMSG OPTION TO CONTROL THE PRINTING OF LIBRARY MAINTENANCE MESSAGES RESULTING FROM ACTIONS TAKEN BY CANDE. IF BOTH OPTIONS ARE SET, ALL SUCH MESSAGES WILL BE TYPED. IF TYPE CEMESS IS RESET, THE MESSAGES PERTAINING TO CANDE, AND THEREFORE TO REMOTE USERS, WILL NOT BE TYPED. SINCE CANDE PERFORMS AN IMMENSE AMOUNT OF LIBRARY MAINTENANCE, THIS OPTION IS NORMALLY RESET.

IF THE TYPE DISKLOG OPTION IS SET, THE LOG INFORMATION FOR DISK FILES WILL BE TYPED AT THE SPO. IF IT IS NOT SET, THE INFORMATION IS NOT TYPED.

WHEN THE TYPE DISKLOG OPTION IS SET, THE MCP WILL TYPE AN ERROR MESSAGE WHEN DISK READ OR WRITE ERRORS OCCUR.

THE TYPE LIBERR OPTION IS USED TO SUPPRESS THE LIBRARY MAINTENANCE IGNORED MESSAGES. WHEN THE OPTION IS SET THESE MESSAGES ARE NOT TYPED. WHEN IT IS RESET, THEY ARE.

PROGRAMMING LANGUAGES

THE LANGUAGES AVAILABLE WITH THE TIME SHARING SYSTEM ARE ALGOL, BASIC, COBOL, FORTRAN, TSPOL AND XALGOL (ALSO CALLED "SAFE" ALGOL). ALGOL AND TSPOL ARE NOT NORMALLY USED FROM A REMOTE TERMINAL. ALGOL IS BURROUGHS B5500 ALGOL AS IMPLEMENTED ON THE STANDARD SYSTEM AND TSPOL (TIME SHARING PROBLEM ORIENTED LANGUAGE) IS ALGOL WITH THE COMMUNICATE STATEMENT ADDED.

THIS LEAVES BASIC, COBOL, FORTRAN AND XALGOL AS THE LANGUAGES NORMALLY USED FROM A REMOTE UNIT. THE VERSIONS OF BASIC AND XALGOL PROVIDED ON THE TIME SHARING SYSTEM ARE DESCRIBED IN THE CORRESPONDING MANUALS. HOWEVER, THE TIME SHARING VERSIONS OF COBOL AND FORTRAN ARE SLIGHTLY MODIFIED VERSIONS OF THE LANGUAGES DESCRIBED IN THE MANUALS FOR THE STANDARD SYSTEM AND THEREFORE THE DIFFERENCES ARE SUMMARIZED BELOW.

ALL LANGUAGES IN THE TIME SHARING SYSTEM MAY BE USED EITHER FROM A REMOTE UNIT OR FROM THE CENTRAL SITE. HOWEVER, IF A PROGRAM RUNNING FROM THE CENTRAL SITE ATTEMPTS TO ACCESS A REMOTE UNIT FOR INPUT IT WILL BE DS-ED. OUTPUT INTENDED FOR A REMOTE UNIT WILL BE TYPED AT THE SPO.

WHEN USING A REMOTE TERMINAL, EACH LINE OF INPUT MUST BEGIN WITH A SEQUENCE NUMBER. THESE SEQUENCE NUMBERS ARE EQUIVALENT TO SEQUENCE NUMBERS PUNCHED IN COLUMNS 73 THROUGH 80 OF A DATA CARD. FOR REMOTE INPUT, THE SEQUENCE NUMBERS ARE CONSIDERED TO EXTEND TO THE FIRST NON NUMERIC CHARACTER, OR FOR A MAXIMUM OF EIGHT CHARACTERS. EVERYTHING

FOLLOWING THE SEQUENCE NUMBER IS TREATED AS NORMAL INPUT.

PROGRAMS INPUT FROM OR LISTED AT A REMOTE TERMINAL CANNOT USE THOSE SPECIAL CHARACTERS WHICH ARE ILLEGAL FOR REMOTE UNITS, (SEE THE TERMINAL USER-S GUIDE,) HOWEVER, SINCE THE SYNTAX OF ALL THE LANGUAGES INCLUDES SUBSTITUTES FOR THESE CHARACTERS, E.G. MNEMONICS FOR THE RELATIONAL OPERATORS, THE LANGUAGES THEMSELVES ARE NOT RESTRICTED.

FOR CARD READER INPUT A NEW \$ OPTION, SEQXEQ, HAS BEEN ADDED FOR XALGOL, COBOL, AND FORTRAN, AND IS DEFINED AS FOLLOWS:

SEQXEQ ::= EXECUTION TIME MESSAGES WILL BE IDENTIFIED BY SEQUENCE NUMBER INSTEAD OF RELATIVE ADDRESS.

THE SEQXEQ OPTION CANNOT BE RESET DURING THE COMPILATION. THAT IS, ANY ATTEMPT TO CHANGE IT AFTER PROGRAM CODE HAS BEEN ENCOUNTERED WILL BE IGNORED.

THIS OPTION IS AUTOMATICALLY SET FOR ALL REMOTE JOBS.

COBOL

DELETIONS

THE FOLLOWING CONSTRUCTS HAVE BEEN DELETED IN TIME SHARING COBOL.

1. "ASSIGN TO DATA."
2. "UNTIL"
3. "WAIT"
4. "PERFORM (,)")"
5. "PERFORM WITH "

REMOTE I/O

ACCEPT, DISPLAY, READ AND WRITE MAY ALL BE USED FOR INPUT FROM AND OUTPUT TO THE REMOTE UNIT. THE SYNTAX FOR USING ACCEPT AND DISPLAY FOR THE REMOTE UNIT IS:

"ACCEPT FROM REMOTE."

"ACCEPT ."

"DISPLAY UPON REMOTE."

"DISPLAY ."

THE SPD IS NOW ACCESSED WITH THE SYNTAX

"ACCEPT FROM KEYBOARD."

"DISPLAY UPON MESSAGE-PRINTER."

NOTE THAT DATA-NAME MUST BE AN 01 LEVEL NOT GREATER THAN 72 CHARACTERS WHEN USED WITH ACCEPT AND DISPLAY.

THE SYNTAX FOR USING READ AND WRITE IS:

```
SELECT          ASSIGN TO REMOTE.

READ            RECORD [INTO          ] [ ] AT END
                [ELSE                ]].

WRITE           [FROM                ] BEFORE ADVANCING
                LINES.
```

THE USE OF ACCEPT AND DISPLAY IS MUCH MORE EFFICIENT THAN THE USE OF READ AND WRITE SINCE ACCEPT AND DISPLAY DO NOT GO THROUGH THE INTRINSICS

INPUT FORMAT

IN ORDER TO MAKE IT EASIER TO USE THE REMOTE TERMINAL THE FOLLOWING CONVENTIONS HAVE BEEN ADOPTED IN PLACE OF THE STANDARD CONVENTIONS.

1. A CONTINUATION CARD IS SPECIFIED BY A HYPHEN IMMEDIATELY AFTER THE SEQUENCE NUMBER.
2. A DOLLAR SIGN IMMEDIATELY AFTER THE SEQUENCE NUMBER SPECIFIES A DOLLAR CARD.
3. ANY ALPHABETIC OR NUMERIC CHARACTER OCCURRING IMMEDIATELY AFTER THE SEQUENCE NUMBER IS ASSUMED TO BE A LABEL.
4. PROGRAM INSTRUCTIONS MUST BE SEPARATED FROM THE SEQUENCE

NUMBER BY A BLANK OR MAY FOLLOW A LABEL.

ALL OTHER CONVENTIONS ARE STANDARD. IT SHOULD ALSO BE NOTED THAT FOR PROGRAMS INPUT VIA THE CARD READER, ALL CONVENTIONS ARE STANDARD.

FORTRAN

REMOTE TERMINAL DELETIONS

THE FOLLOWING CONSTRUCTS ARE ILLEGAL WHEN USING A REMOTE TERMINAL AND WILL ALSO BE FLAGGED WHEN USING THE CARD READER AND THE TSSDIT OPTION. (SEE § OPTIONS.)

1. THE PAUSE STATEMENT
2. FORMAL PARAMETERS USED AS SUBPROGRAMS
3. HOLLERITH OR QUOTED STRINGS EXTENDED FROM ONE LINE TO ANOTHER
4. THE ZIP CONSTRUCT

NOTE THAT THE WORD "ZIP" IS NO LONGER RESERVED AND THEREFORE CAN BE USED LIKE ANY WORD. HOWEVER, SINCE THIS MAY BE CHANGED IN THE FUTURE, ITS USE IS DISCOURAGED.

REMOTE I/O

THE REMOTE UNIT DESIGNATOR HAS BEEN ADDED TO ALLOW INFORMATION TO BE READ FROM OR WRITTEN TO THE REMOTE UNIT. USE IN A SOURCE PROGRAM OF A FILE CARD CONTAINING "UNIT=REMOTE" WILL CAUSE RELEVANT I/O STATEMENTS TO REFERENCE THE REMOTE TERMINAL. NO BUFFER OR BLOCKING INFORMATION IS REQUIRED, AND IF PROVIDED IT WILL BE IGNORED.

INPUT FORMAT

IN ORDER TO ELIMINATE THE NECESSITY OF SPACING INPUT, TIME SHARING FORTRAN USES "REMOTE FREE FIELD FORMAT" INSTEAD OF THE ORDINARY CARD FORMAT. IN THE RULES GIVEN BELOW, COLUMN 1 REFERS TO THE FIRST COLUMN AFTER THE SEQUENCE NUMBER.

1. CONTINUATION CARDS CONTAIN A "-" IN COL. 1, AND THE CARD STARTS WITH THE FIRST NON-BLANK CHARACTER OR IN COL. 7, WHICHEVER COMES FIRST

COMMENT STARTING IN COL. 3.

3. FILE CARDS MUST START IN COL. 1, CONSEQUENTLY, ANY LINE STARTING WITH THE WORD FILE FOLLOWED BY TWO BLANKS MUST BE A FILE CARD.

4. LABELS MAY BE A MAXIMUM OF 5 COLUMNS LONG AND MAY CONTAIN EMBEDDED BLANKS. A NON-BLANK NON-NUMERIC CHARACTER, OR THE SIXTH COLUMN AFTER THE START OF THE LABEL, ENDS THE LABEL AND STARTS THE CARD TEXT. A LABEL MAY BE SEPARATED FROM THE SEQUENCE NUMBER BY ANY NUMBER OF BLANKS.

5. ONLY 66 COLUMNS OF CARD TEXT (SEE 1 AND 4 ABOVE) ARE ALLOWED ON ONE LINE. ADDITIONAL TEXT WILL ELICIT SYNTAX ERROR #149.

\$ OPTIONS

THE \$ OPTIONS AVAILABLE ON STANDARD FORTRAN MAY BE USED FROM THE REMOTE TERMINAL WITH THE FOLLOWING CHANGES AND ADDITIONS.

1. THE \$ TSSEDIT OPTION CAUSES THE COMPILER TO CONSIDER THE

FORMAT OF THE SOURCE FILE TO BE THE ORDINARY RESTRICTED FIELD FORMAT RATHER THAN THE REMOTE FREE FIELD FORMAT. THE OPTION IS TREATED AS ANY OTHER \$ OPTION AND, IF DESIRED, MUST BE RENEWED ON EACH \$ CONTROL CARD.

2. THE \$ ERRMES OPTION WILL CAUSE A DESCRIPTION OF THE ERROR TO BE TYPED NEXT TO THE ERRONEOUS CONSTRUCT IN THE ERROR MESSAGE.
3. THE \$ LIST OPTION IS INITIALLY SET "OFF" WHEN COMPILING FROM A REMOTE TERMINAL, USE OF THE OPTION WILL RESULT IN A LINE PRINTER LISTING IN THE ORDINARY FORMAT.
4. THE \$ ONSITE OPTION CAUSES PRINT STATEMENTS TO REFERENCE THE LINE PRINTER AND READ STATEMENTS WITHOUT FILE UNIT DIGITS TO REFERENCE THE CARD READER FILE "READER". IF THIS OPTION IS NOT USED, THESE STATEMENTS REFERENCE THE REMOTE TERMINAL BY DEFAULT.
5. THE \$ REMOTE OPTION (SEE 2 BELOW) SIMPLY RESETS THE \$ ONSITE OPTION
CARD READER.
1. IF THE \$ TSSEDIT OPTION IS USED, CONSTRUCTS WHICH WOULD BE ILLEGAL IF THE PROGRAM WERE BEING COMPILED FROM A REMOTE TERMINAL (SEE "DELETIONS") WILL BE FLAGGED WITH WARNING MESSAGES BUT WILL STILL BE COMPILED. IF THIS OPTION IS USED WITH THE \$ NEW TAPE OPTION, THE NEW SYMBOLIC FILE WILL BE IN REMOTE FREE FIELD FORMAT.
2. THE \$ REMOTE OPTION CAUSES PRINT STATEMENTS AND READ

STATEMENTS WITHOUT A FILE UNIT DIGIT TO REFERENCE A REMOTE
TERMINAL INSTEAD OF THE LINE PRINTER OR CARD READER. PROGRAMS
COMPILED UNDER THIS OPTION CAN BE RUN ONLY FROM A REMOTE TERMINAL.

3. THE \$ ONSITE OPTION (SEE 4 ABOVE) SIMPLY RESETS THE \$ REMOTE
OPTION

FIRST GROUP OF \$ CARDS, I.E., BEFORE ANY PROGRAM STATEMENTS. IF THEY
APPEAR AFTER PROGRAM STATEMENTS HAVE BEEN ENCOUNTERED THEY WILL BE
IGNORED.

TSPOL

THE DIFFERENCE BETWEEN TSPOL AND ALGOL IS THE COMMUNICATE STATEMENT,
THE SYNTAX FOR WHICH IS:

SYNTAX.

<COMMUNICATE STATEMENT> ::= COMMUNICATE (<COMMUNICATE NUMBER>)

<COMMUNICATE NUMBER> ::= <ARITHMETIC EXPRESSION>

SEMANTICS.

THIS STATEMENT CAUSES THE COMMUNICATE OPERATOR TO BE EXECUTED WITH
THE <COMMUNICATE NUMBER> AT THE TOP OF THE STACK. USE OF <COMMUNICATE
NUMBER>S FOR WHICH THERE IS NO MCP CODE RESULTS IN AN INVALID INDEX
IN THE MCP.

FILE SECURITY

THE TIME SHARING SYSTEM MAKES USE OF THE FILE SECURITY SYSTEM DEVELOPED FOR THE STANDARD MCP. THE USER CODE, PASSWORD AND LOCK FILE IN THE TIME SHARING SYSTEM ARE EQUIVALENT TO THE USER CODE, AUTHENTICATION CODE AND SECURITY FILE IN THE STANDARD SYSTEM. FILES CREATED FROM THE REMOTE UNIT BECOME SOLE USER FILES UNLESS SPECIFIED OTHERWISE VIA LOCK AND UNLOCK COMMANDS. LOCK (I.E. SECURITY) FILES CAN BE CREATED AND UPDATED AT THE CENTRAL SITE, OR FROM THE REMOTE TERMINAL USING THE GUARD COMMAND. IT SHOULD BE NOTED THAT SINCE ONE REMOTE USER IS NEVER ALLOWED TO MODIFY ANOTHERS FILES, SECONDARY AND TERTIARY USERS ARE EQUIVALENT FROM A REMOTE UNIT.

THE REMOTE/USERS FILE AND THE UPDATE/USERS PROGRAM HAVE BEEN REPLACED BY THE USERS/CANDE FILE AND THE USER/CANDE PROGRAM.

USER/CANDE

THE NAMES OF ALL AUTHORIZED USERS ARE KEPT IN THE FILE USERS/CANDE, WHICH IS CREATED AND UPDATED BY THE PROGRAM USER/CANDE. THE INPUT DECK FOR USER/CANDE CONSISTS OF \$ CARDS AND OPTION SPECIFIER CARDS.

THE \$ CARDS ARE

\$NEW SPECIFIES THAT A NEW FILE IS TO BE CREATED.

\$OPTIONS SPECIFIES THAT THE FOLLOWING SET OF OPTION SPECIFIER CARDS ARE TO BE USED TO FORM AN OPTION BLOCK.

\$USER "<USER CODE>" SPECIFIES THAT THE FOLLOWING SET OF OPTION SPECIFIER CARDS APPLY TO THE NAMED USER.

IF A \$NEW CARD IS USED, IT MUST BE THE FIRST CARD IN THE DECK. IF THERE IS NO \$NEW CARD, THE CURRENT USERS/CANDE WILL BE UPDATED.

THE OPTION BLOCK DEFINES THE DEFAULT OPTIONS. IT IS CREATED AND CHANGED BY USE OF THE \$OPTION CARD. THE OPTIONS SPECIFIED BETWEEN A \$OPTION CARD AND THE NEXT \$ CARD ARE STORED IN THE OPTION BLOCK.

THE \$USER CARD CAUSES THE NAMED USER TO BE ADDED TO THE FILE WITH THE SPECIFIED OPTIONS IF IT IS A NEW FILE OR IF HE WAS NOT IN IT. OTHERWISE THE \$USER CARD CAUSES THE OPTIONS FOR THE SPECIFIED USER TO BE CHANGED. IN EITHER CASE, ALL OPTIONS SPECIFIED BETWEEN THE \$USER CARD AND THE NEXT \$ CARD WILL BE STORED IN THAT USER-S BLOCK. OPTIONS NOT SPECIFIED ARE TAKEN FROM THE OPTION BLOCK.

THE OPTION SPECIFIER CARDS ARE:

PASSWORD "<1-7 CHARACTERS>"

SUPPLIES THE USERS PASSWORD

NAME "<ANY STRING>"

SUPPLIES THE USER NAME WHICH IS USED ONLY AT LOG-IN.

ONE OF THE FOLLOWING THREE CARDS SHOULD BE USED TO SPECIFY THE
OPTIONS FOR THE CHARGE CODE.

USE CHARGE "<1-7 CHARACTERS>"

THE CHARGE CODE WILL BE USED FOR ALL SESSIONS OF THIS USER
REQUEST CHARGE

THE USER WILL BE ASKED TO SUPPLY A CHARGE CODE DURING THE LOG-
IN PROCEDURE.

NO CHARGE

A CHARGE CODE IS NOT TO BE USED FOR THIS USER.

TIME "<A STRING OF UP TO 24 1-S AND 0-S>"

SPECIFIES THE HOURS OF THE DAY DURING WHICH THE USER WILL BE
ALLOWED TO RUN. A ONE IN THE STRING SPECIES HOURS WHICH MAY
BE USED, AND ZERO INDICATES HOURS WHICH MAY NOT BE USED. THE
FIRST DIGIT IS FOR THE HOUR FROM MIDNIGHT TO 1AM, THE SECOND

FOR 1AM TO 2AM, ETC. IF THE STRING HAS FEWER THAN 24 CHARACTERS
IT IS FILLED OUT WITH ZEROS ON THE RIGHT.

LANGUAGES [ALGOL][BASIC][COBOL][FORTRAN][TSPOL][XALGOL]

THE USER WILL BE ALLOWED TO USE ONLY THE SPECIFIED LANGUAGES,
WHICH MAY APPEAR IN ANY ORDER ON THE CARD.

TELEPHONE <TELEPHONE NUMBER>

THE USERS PHONE NUMBER MAY BE ENTERED. IT SHOULD BE ENTERED
AS EITHER 7 OR 10 DIGITS, WHICH MAY BE SEPARATED BY HYPHENS.
THE WORD PHONE MAY BE USED IN PLACE OF TELEPHONE.

RESTRICT <LIST OF VERBS>

THE USER WILL NOT BE ALLOWED TO USE THE VERBS IN THE LIST.

IT SHOULD BE NOTED THAT NO PROVISION HAS BEEN MADE FOR DELETING ENTRIES
HOWEVER, A USER CAN BE DENIED ACCESS TO THE SYSTEM BY SETTING HIS
PASSWORD TO A LEFT ARROW (GROUP MARK). TO REMOVE ALL TRACE OF A USER
FROM THE FILE, A NEW FILE MUST BE CREATED.

USERS/CANDE

THE FILE USERS/CANDE CONSISTS OF 465 SEGMENTS WHICH ARE UTILIZED AS FOLLOWS:

SEGMENT 0	POINTS TO THE FIRST AVAILABLE USER BLOCK.
SEGMENTS 1-29	CONTAIN THE USER CODE TABLES
SEGMENTS 30-464	CONTAIN THE USER BLOCKS.

THE USER CODE TABLES CONSIST OF TWO WORDS FOR EACH USER, CONTAINING THE USER CODE AND THE RELATIVE ADDRESS OF THE USER BLOCK FOR THIS USER. THUS, THERE ARE 15 USERS IN A SEGMENT AND 29 SEGMENTS IN THE TABLE ALLOWING A MAXIMUM OF 435 QUALIFIED USERS.

FORTUNATELY, THERE ARE ALSO EXACTLY 435 SEGMENTS FOR USER BLOCKS. THE WORDS IN A USER BLOCK CONTAIN:

WORD 0	MAIL
1	USER CODE
2	PASSWORD
3	CHARGE CODE OPTION
4	
5	USER NAME
6	TELEPHONE NUMBER
7	TIME WORD
8	COMPILER RESTRICTIONS
9	VERB RESTRICTIONS
10-29	RESERVED FOR EXPANSION

SINCE ENTRIES ARE NOT REMOVED FROM THE FILE, THE UNUSED USER BLOCKS ARE ALL AT THE END OF THE FILE, THE RELATIVE ADDRESS OF THE FIRST UNUSED BLOCK IS STORED IN WORD 1 OF SEGMENT 0. THE REST OF SEGMENT 0 IS, CURRENTLY UNUSED.

THE LOG

THE TSS LOG KEEPS AN ACCOUNT OF ALL TIME AND DISK SPACE USED BY THE REMOTE USERS, WITH SEPARATE ENTRIES FOR THE ELAPSED, THE I/O AND THE PROCESSOR TIMES. THE PROCESSOR TIME INCLUDES TIME USED FOR LIBRARY MAINTENANCE, FILE EDITING, AND COMMAND LANGUAGE PROCESSING, AS WELL AS THAT RELATED TO THE RUNNING OF PROGRAMS.

THE LOG IS KEPT IN THE FILE LOG/DISK, WHICH IS SIMILAR TO SYSTEM/LOG ON THE STANDARD MCP IN THAT LIBRARY MAINTENANCE CANNOT BE PERFORMED ON IT AND IT MUST BE PROVIDED BY THE INSTALLATION. LOG/DISK CONTAINS ALL THE SPO MESSAGES GENERATED BY OR INPUT TO THE SYSTEM. THE USE OF OPTIONS TO SUPPRESS SPO MESSAGES DOES NOT AFFECT THE CONTENTS OF THE LOG, WHICH ALWAYS CONTAINS ALL THE MESSAGES THAT WOULD APPEAR AT A SPO WITH ALL MESSAGE OPTIONS SET. IN ADDITION, THERE ARE TWO BINARY "MESSAGES" WHICH APPEAR ONLY IN THE LOG AND CONTAIN INFORMATION NOT INCLUDED IN THE SPO MESSAGES.

LOG/DISK CONSISTS OF 10 WORD LOGICAL RECORDS BLOCKED THREE TO A PHYSICAL RECORD. WORD 0 IS USED TO IDENTIFY THE MESSAGE WHICH APPEARS IN WORDS 1-9. THE MESSAGE IS STORED IN UNCOMPACTED FORM; THAT IS, IT HAS NOT BEEN SCANNED TO REMOVE EXTRANEIOUS BLANKS. THE FORMAT OF WORD 0 IS:

2:5 MIX INDEX OF THE JOB TO WHICH THIS ENTRY APPLIES. IF THERE IS NO SPECIFIC JOB TO WHICH IT APPLIES, THIS FIELD IS ZERO.

7:1 REMOTE-LOCAL BIT, IT IS:
1 IF THE MESSAGE APPLIES TO A REMOTE USER.
0 OTHERWISE.

8:8 LOGICAL LINE NUMBER OF THE REMOTE USER IF THE REMOTE-
LOCAL BIT IS EQUAL TO 1. OTHERWISE THIS FIELD IS ZERO.

16:7 TYPE OF MESSAGE,

24:1 SPO PRINTOUT BIT, IT IS:
1 IF THE MESSAGE WAS PRINTED AT THE SPO
0 IF IT WAS NOT

25:23 TIME OF DAY IN SIXTIETHS OF A SECOND.

FOR THE LAST RECORD IN THE FILE, BITS 1-47 OF WORD 0 ARE SET TO 1 AND
THE REST OF THE RECORD IS LEFT EMPTY.

THE OPERATOR IS KEPT INFORMED OF THE FULLNESS OF LOG/DISK BY THE
MESSAGE

LOG <PERCENTAGE> % FULL

WHICH IS PRINTED EVERY 5 PER CENT. THE OPERATOR CAN CREATE AN EMPTY
LOG/DISK WITH THE LN MESSAGE. WHEN LN IS TYPED, THE NAME OF THE CURRENT
LOG/DISK IS CHANGED TO <NUMBER>/LOG AND A NEW FILE NAMED LOG/DISK IS
CREATED. ALL ENSUING SPO MESSAGES WILL BE STORED IN THE NEW LOG/DISK,
LEAVING <NUMBER>/DISK TO BE PROCESSED LATER. THE NUMBER HAS 7 DIGITS
WHERE:

DIGITS 0-1 CONTAIN THE MONTH
DIGITS 2-3 CONTAIN THE DATE
DIGITS 4-6 ARE THE SERIAL NUMBER.

THE SERIAL NUMBERS ARE LIKE THOSE USED FOR PRINTER BACK-UP FILES ON DISK. ON A GIVEN DAY, THE FIRST FILE IS NUMBERED 001, THE SECOND 002, AND SO ON. IF LOG/DISK BECOMES 95 PER CENT FULL, AN LN IS AUTOMATICALLY INITIATED AND THE MESSAGE IS

LOG 95% FULL (AUTO LN)

BURROUGHS SUPPLIES A PROGRAM, LOGANN/DISK, TO PROCESS THE FILES <NUMBER>/DISK. IT SHOULD BE ADEQUATE FOR MOST INSTALLATIONS AND CAN BE USED AS A MODEL IF A DIFFERENT LOGGING PROGRAM IS DESIRED. LOGANN/DISK IS INITIATED IN THE USUAL WAY BY MEANS OF AN EXECUTE CARD. WHEN THE PROGRAM STARTS IT ASKS VIA THE SPO FOR THE MULTIPLE FILE IDENTIFICATION OF THE FILE TO BE ANALYZED. THE OPERATOR MUST THEN TYPE IN

<MIX> AX <MULTIPLE FILE IDENTIFICATION>

IF THE FILE IS NOT ON DISK THE PROGRAM WILL NOTIFY THE OPERATOR WITH THE FOLLOWING MESSAGE

NO FILE: <MULTIPLE FILE IDENTIFICATION>/LOG

AND WILL AGAIN ASK FOR THE MULTIPLE FILE IDENTIFICATION. AFTER PROCESSING THE FILE, THE PROGRAM TERMINATES AND MUST THEREFORE BE RERUN IN ORDER TO PROCESS ANOTHER FILE. OUTPUT FROM LOGANN/DISK IS ON FILE PRNTPS WHICH IS NORMALLY THE LINE PRINTER. HOWEVER, SINCE THE PROGRAM DOES NOT START PRINTING IMMEDIATELY, IT IS SUGGESTED THAT THE BACK UP OPTION BE USED.

MESSAGES APPEARING IN THE LOG

<u>TYPE</u> -----	<u>CONTENTS</u> -----
0	A MESSAGE NOT APPLICABLE TO LOGGING
1	A MESSAGE TYPED IN FROM THE SPO
2	BOJ MESSAGE
3	EOJ MESSAGE
4	PBEDJ MESSAGE (PRINTER BACK-UP END OF JOB)
5	FILE OPEN MESSAGE
6	FILE CLOSE MESSAGE
7	HALT/LOAD MESSAGE
8	EOJ STATISTICS, A BINARY RECORD WHICH ALWAYS OCCURS IMMEDIATELY IN FRONT OF A EOJ MESSAGE AND PROVIDES TIMING INFORMATION FOR THAT JOB IMMEDIATELY BEHIND A FILE CLOSE MESSAGE AND PROVIDES INFORMATION CONCERNING THAT FILE,
10	ON MESSAGE, INDICATES A SUCCESSFUL LOG-IN,
11	OFF MESSAGE,
12	CHARGE MESSAGE. (OUTPUT AFTER A CHARGE COMMAND.)
13	DISK CHARGES MESSAGE, (OUTPUT WHEN A FILE IS REMOVED.)
14	DATE MESSAGE
15	TIME MESSAGE

THE FORMATS OF THE VARIOUS MESSAGES ARE LISTED BELOW, THE DEFINITIONS
USED IN THE B5500 OPERATION MANUAL ARE USED HERE, WITH THE FOLLOWING
ADDITIONS:

<TIME OF DAY> TIME OF DAY IN HOURS AND MINUTES IN THE FORM HHMM.

<TIME> A TIME GIVEN AS HOURS, MINUTES AND SECONDS,
SEPARATED BY COLONS IN THE USUAL WAY.

THE BOJ MESSAGE HAS THREE FORMS:

<PROGRAM SPECIFIER>/<USER CODE>=<MIX INDEX>BOJ<TIME OF DAY>

<LOGICAL LINE NUMBER>:<PROGRAM SPECIFIER>=<MIX INDEX>

BOJ<TIME OF DAY>

<LOGICAL LINE NUMBER>:<PROGRAM SPECIFIER>/<USER CODE>=<MIX INDEX>

BOJ<TIME OF DAY>

THE FIRST FORM IS USED FOR LOCAL JOBS, THE SECOND FORM IS USED FOR
REMOTE JOBS WHERE THE <PROGRAM IDENTIFIER SUFFIX> IS EQUAL TO THE <
USER CODE>, I.E., USER PROGRAMS. THE THIRD FORM IS USED FOR OTHER
REMOTE JOBS SUCH AS CANDE PROGRAMS.

THE EOJ MESSAGE HAS TWO FORMS:

<PROGRAM SPECIFIER>=<MIX INDEX>,PST=<TIME>EOJ

<PROGRAM SPECIFIER>/<USER CODE>=<MIX INDEX>,PST=<TIME>EOJ

THE FIRST FORM IS USED FOR REMOTE JOBS WHERE THE <PROGRAM IDENTIFIER
SUFFIX> IS EQUAL TO THE <USER CODE>. THE SECOND FORM IS USED IN ALL
OTHER CASES. THE <TIME> IS PROCESSOR TIME. DS-ED, ES-ED, ETC. MAY
APPEAR INSTEAD OF EOJ.

PBEOJ MESSAGE

PRNPBT FOR <PROGRAM SPECIFIER>,PST=<TIME>,IOT=<TIME>JEOJ

WHERE PST IS THE PROCESSOR TIME AND IOT IS THE I/O TIME.

FILE OPEN MESSAGE

<UNIT MNEMONIC><IN-OUT><DATA FILE DESIGNATOR><RDC>:<JOB SPECIFIER>

WHERE <IN-OUT> IS IN OR OUT. THIS MESSAGE IS IDENTICAL TO THE MESSAGE ON THE STANDARD MCP.

FILE CLOSE MESSAGE

<UNIT MNEMONIC>REL<DATA FILE DESIGNATOR><RDC>:<JOB SPECIFIER>

IDENTICAL TO STANDARD MCP MESSAGE.

HALT MESSAGE

-H/L MARK TS=MCP ,<PATCH LEVEL>F=<FENCE LOCATION>

[MODS=<MEMORY MASK>]-

E0J STATISTICS:

WORD 1 CONTAINS THE PROCESSOR TIME IN SIXTIETHS OF A SECOND.

WORD 2 CONTAINS THE I/O TIME IN SIXTIETHS OF A SECOND.

WORD 3 CONTAINS THE NUMBER OF WORDS OF CORE USED.

FILE CLOSE STATISTICS

THIS MESSAGE TAKES 5 WORDS AND IS IDENTICAL TO A FILE-INFORMATION
RECORD ON SYSTEM/LOG FOR THE STANDARD MCP.

ON MESSAGE

<USER CODE>ON<LOGICAL LINE NUMBER>

<USER CODE>ON<LOGICAL LINE NUMBER> (<CHARGE CODE>)

THE FORM USED DEPENDS ON WHETHER OR NOT A CHARGE CODE IS BEING USED
FOR THIS USER.

OFF MESSAGE

<USER CODE>OFF<LOGICAL LINE NUMBER>

<USER CODE>OFF<LOGICAL LINE NUMBER> (<CHARGE CODE>)

CHARGE MESSAGE

FOR <USER CODE>ON<LOGICAL LINE NUMBER>CHARGE<CHARGE CODE>

DISK CHARGES MESSAGE

<FILE SPECIFIER>/<USER CODE>=<NUMBER OF SEGMENTS>SEGS--CREATED

<DATE>AT<TIME>

DATE MESSAGE

DATE IS <DAY OF WEEK>,<MONTH>/<DAY>/<YEAR>

TIME MESSAGE

TIME IS <TIME OF DAY>

PART II

PROGRAMMING INFORMATION

THE MCP

THE TIME SHARING MCP IS SIMPLY A MODIFIED VERSION OF THE STANDARD MARK VIII MCP. MUCH OF THE TSS-MCP IS THEREFORE ALMOST IDENTICAL TO THE CORRESPONDING PARTS OF THE STANDARD MCP. A GOOD EXAMPLE OF THIS IS THE WAY IN WHICH NON-DATA-COMMUNICATIONS I/O IS PROCESSED. OTHER AREAS, HOWEVER, HAVE BEEN EXTENSIVELY MODIFIED. FOR INSTANCE, EVERYTHING IN THE STANDARD MCP PERTAINING TO DATA COMMUNICATIONS HAS BEEN REMOVED ALTHOUGH PARTS OF IT HAVE BEEN REPLACED BY NEW ROUTINES AND PROCEDURES WHERE NECESSARY. THE DESCRIPTIONS CONTAINED HEREIN COVER ONLY THOSE PARTS OF THE TSS-MCP WHICH ARE NEW OR DIFFER SUBSTANTIALLY FROM THEIR COUNTERPARTS IN THE STANDARD MCP.

SEGMENTED SAVE AREAS

SEGMENTED SAVE AREAS ARE USED THROUGHOUT THE MCP TO PROVIDE A WAY IN WHICH PROCEDURES CAN GET 5, 10 OR 20 WORDS OF SPACE WITHOUT LOSING CONTROL. THEY ARE USED FOR SUCH THINGS AS THE EVENT QUEUE AND THE BUFFERS FOR REMOTE I/O. EACH AREA IS IDENTIFIED IN BITS 1-3 OF THE FIRST WORD OF THE AREA WHERE:

BIT 1 IS THE OCCUPIED BIT. IT IS 1 IF THIS AREA IS AVAILABLE,
0 IF IT IS IN USE.

BITS 2-3 ARE A CODE FOR THE SIZE OF THE AREA WHERE
0 MARKS A 5 WORD AREA
1 INDICATES A 10 WORD AREA
2 SIGNIFIES A 20 WORD AREA.

ALL THE AVAILABLE AREAS OF A GIVEN SIZE ARE KEPT IN A QUEUE, THE HEAD AND TAIL OF WHICH ARE KEPT IN SPACER[I], WHERE I IS THE SIZE CODE OF THE AREAS. THE QUEUE IS LINKED BY THE FIRST WORD IN EACH AREA, THE 18:15 FIELD POINTS TO THE PREVIOUS AREA IN THE QUEUE AND THE 33:15 FIELD POINTS TO THE NEXT AREA. THE QUEUE IS ORDERED ACCORDING TO THE CORE ADDRESSES OF THE AREAS, WITH AREAS IN LOWEST CORE AT THE HEAD OF THE QUEUE.

DURING INITIALIZATION, 160 WORDS ARE GOTTEN AND DIVIDED INTO 20 WORD AREAS. WHEN A PROCEDURE NEEDS AN AREA IT CALLS GETAREA AND PASSES IT THE SIZE CODE OF THE AREA. IF THE QUEUE FOR AREAS OF THAT SIZE IS NOT EMPTY, THE FIRST AREA IN THE QUEUE IS RETURNED TO THE CALLING PROCEDURE. IF THE QUEUE IS EMPTY, THE QUEUES OF THE LARGER AREAS ARE TESTED UNTIL

A NON-EMPTY QUEUE IS FOUND, THE FIRST AREA IN THAT QUEUE IS THEN SPLIT BY HALVING UNTIL AN AREA OF THE DESIRED SIZE IS OBTAINED. THE PIECES THUS CREATED ARE LINKED INTO THE APPROPRIATE QUEUES.

GETAREA ALSO CHECKS THE NUMBER OF 20 WORD AREAS IN THE QUEUE. IF IT IS LESS THAN FOUR, IT CALLS FOR MOREAREAS AS AN INDEPENDENT RUNNER. MOREAREAS OBTAINS AN ADDITIONAL 160 WORDS WHICH IT DIVIDES INTO 20 WORD AREAS AND ADDS TO THE QUEUE. RUNNING OUT OF AREAS IS A CAPITAL OFFENSE; THAT IS, IF GETAREA IS CALLED AND AN AREA OF THE PROPER SIZE CANNOT BE OBTAINED, THE SYSTEM IS HUNG.

WHEN A PROCEDURE IS THROUGH WITH AN AREA, IT RETURNS IT BY PASSING THE SIZE CODE AND ADDRESS OF THE AREA TO FORGETAREA, WHICH RELINKS THE AREA INTO THE PROPER QUEUE. HOWEVER, IF THE AREA IS LESS THAN 20 WORDS LONG, IT FIRST CHECKS TO SEE IF THE AREA WHICH FORMS THE OTHER HALF OF THE NEXT LARGER AREA IS ALSO AVAILABLE, IN WHICH CASE IT DELINKS THAT AREA, AND COMBINES THE TWO HALVES TO FORM A LARGER AREA. WHEN NO FURTHER RECOMBINING CAN BE DONE, THE AREA IS ADDED TO THE APPROPRIATE QUEUE.

MCP TABLES

MANY OF THE TABLES IN THE TIME SHARING SYSTEM ARE THE SAME AS THE CORRESPONDING TABLES IN THE STANDARD MCP AND THEREFORE ONLY THOSE TABLES WHICH HAVE BEEN ADDED OR CHANGED ARE DESCRIBED HERE. IN ADDITION, THE DEFINES USED TO ACCESS FIELDS WITHIN THE TABLES ARE ALSO LISTED. IN GENERAL, THESE DEFINES ARE OF TWO TYPES. FIRST, A DEFINE MAY BE USED FOR A PARTIAL WORD DESIGNATOR, FOR INSTANCE,

ADINFO,CANDETHRU

SECOND, PARAMETRIC DEFINES ARE USED TO ACCESS FIELDS AS IF THEY WERE THEMSELVES TABLES, FOR EXAMPLE, SC[I] IS DEFINED TO BE SQ[I],[36:6]. THE TYPE OF DEFINE BEING USED WITH A GIVEN TABLE IS EXPLAINED IN THE DESCRIPTION OF THAT TABLE.

IN ORDER TO SIMPLIFY THE TASK OF SWAPPING, MCP PARAMETERS PERTAINING TO A SPECIFIC JOB ARE KEPT IN THAT JOBS AREA, FOR INSTANCE, THE JAR ROW FOR EACH JOB IS KEPT IN ITS AREA, MOST SUCH PARAMETERS, HOWEVER, ARE DEFINED TO BE A PART OF THE UV TABLE,

UV ARRAY

THE UV ARRAY CONSISTS OF 15 ENTRIES FOR EACH JOB, ALL OF WHICH ARE NORMALLY ACCESSED BY PARAMETRIC DEFINES. FOR INSTANCE, THE TOP OF STACK VALUE FOR JOB WITH MIX INDEX I IS ACCESSED AS TOPSK[I]. IN SOME CASES THE UV ENTRY IS A DESCRIPTOR POINTING TO AN ARRAY ROW IN THE JOBS AREA. THESE ARRAYS CAN BE ACCESSED JUST AS THEY ARE ON THE STANDARD SYSTEM.

THE CONTENTS OF THE UV ARRAY ROW FOR A GIVEN JOB ARE:

<u>WORD</u> -----	<u>DEFINE</u> -----	<u>CONTENTS</u> -----
0	ELAPSEDLIMIT	MAXIMUM ELAPSED TIME ALLOWED FOR THIS JOB BEFORE IT IS SWAPPED OUT.
1	PROCLIMIT	MAXIMUM PROCESSOR TIME ALLOWED FOR THIS JOB BEFORE IT IS SWAPPED OUT.
2	IOCOUNT	NUMBER OF I/O OPERATIONS IN PROGRESS FOR THIS JOB.
3	TOPSK	ADDRESS OF TOP OF JOB-S STACK WHEN IT IS SWAPPED OUT.
4	USERCODE	USER-CODE FOR THIS JOB
5	PRYOR	PRIORITY OF THIS JOB
6	FS	DESCRIPTOR TO FS ROW FOR THIS JOB
7	FPBD	FILE BLOCK DATA DESCRIPTOR FOR THIS JOB
8	SEGD	SEGMENT DICTIONARY NAME DESCRIPTOR FOR THIS JOB.
9	SINFO	1:17 CLOCK AT BOJ

18:15 ESTIMATED CORE REQUIREMENTS

33:15 BOTTOM OF STACK

10	DALOC	DESCRIPTOR TO ROW OF DALOC CONTAINING ENTRIES PERTAINING TO THIS JOB.
11	IOTIME	I/O TIME USED FOR THIS TIME
12	INTABLE	DESCRIPTOR TO INTABLE ROW FOR THIS JOB
13	PROCTIME	PROCESSOR TIME USED FOR THIS JOB.
14	EVENT	HEAD OF EVENT QUEUE FOR THIS MIX INDEX (INCLUDED FOR USE WITH MULTIPLE C&E-S.)

TABLES USED IN SWAPPING

THERE ARE THREE ARRAYS USED IN SWAPPING, THROUGH THE USE OF PARAMETRIC DEFINES, FIELDS WITHIN THE WORDS IN THESE ARRAYS ARE THEMSELVES TREATED AS ARRAYS. THE DESCRIPTIONS SHOW FOR EACH ARRAY, THE FIELDS OF THE ARRAYS DEFINED TO BE IN THEM.

SQ[0:MIXMAX+1]

<u>FIELD</u> -----	<u>DEFINE</u> -----	<u>CONTENTS</u> -----
1:1		INTERLOCK FOR SWAPPINGIO
13:2	EXPAND[I]	FLAGS FOR EXPANDING A JOBS AREA.
15:3	SLN[I]	NUMBER OF SWAPS FOR JOB WITH MIX INDEX I.
18:6	STASUS[I]	STATUS OF JOB I.
24:6	COUNT[I]	(NUMBER OF CHUNKS POSSESSED BY JOB I)-1.
30:6	LC[I]	CHUNK NUMBER OF THE LAST CHUNK FOR JOB I.
36:6	SC[I]	CHUNK NUMBER OF THE FIRST CHUNK FOR JOB I.
42:6	LINK[I]	MIX INDEX OF JOB FOLLOWING JOB I IN THE SWAP OR READY QUEUE.

CT [0: NUMBER OF CHUNKS]

<u>FIELD</u> -----	<u>TABLE</u> -----	<u>CONTENTS</u> -----
30:6	POSSESS[J]	MIX INDEX OF JOB POSSESSING CHUNK J, OR ZERO IF CHUNK IS UNPOSSESSED.
36:6	ACTIVE[J]	NUMBER OF JOBS READY TO RUN USING CHUNK J

42:6 TOTAL[J] TOTAL NUMBER OF JOBS ASSIGNED TO CHUNK J.

DAT[0: MIXMAX+1]

FIELD -----	TABLE -----	CONTENTS -----
8:25	DISKSTORE[I]	DISK ADDRESS OF SWAP AREA FOR JOB I
33:15	ACTLEN[I]	ACTUAL LENGTH OF STUFF SWAPPED FOR JOB I.

TABLES USED BY LINE MAINTENANCE

ADINFO

ADINFO IS A ONE DIMENSIONAL TABLE LMAX+1 LONG. WORDS 1 THROUGH LMAX ARE USED TO STORE INFORMATION ABOUT THE CORRESPONDING LINE. THE DEFINES USED TO ACCESS SOME OF THE FIELDS IN ADINFO ARE ALSO LISTED. THOSE MARKED BY ASTERISKS ARE PARAMETRIC DEFINES AND ARE USED AS IF THEY WERE INDIVIDUAL TABLES. THE OTHERS ARE USED AS PARTIAL WORD DESIGNATORS IN CONJUNCTION WITH ADINFO,

FIELD -----	DEFINE -----	CONTENTS -----
1:1	DIALEDUP	DIAL-UP FLAG. 1 IF LINE IS CONNECTED, 0 IF NOT.
2:1	CANDEFLAG	C&E-TYPE-JOB-ATTACHED FLAG. IF THIS FIELD IS 1, INPUT GOES TO C&E.
3:5	MIXNR	MIX INDEX OF JOB TO WHICH THIS LINE IS ATTACHED. (INPUT GOES TO THIS JOB ONLY IF 2:1 IS OFF.)
8:1	PINGPING	PING-PING BUFFERS FLAG.
9:4		TERMINAL UNIT NUMBER
13:1		GROUP MARK FLAG. 1 IF GROUP MARKS SHOULD BE IGNORED. 0 IF GROUP MARKS TERMINATE I/O.
14:4		BUFFER ADDRESS
18:2	BUFSIZE	BUFFER-SIZE FLAG 0 FOR 28 CHARACTER BUFFERS

1 FOR 56 CHARACTER BUFFERS.

2 FOR 112 CHARACTER BUFFERS.

MULTIPLY THIS BY 2 FOR WRITES IF PING-PING
BUFFERS ARE USED.

20:3

ADAPTER TYPE

0 = TSS (MODE)

1 = 980 ADAPTER

2 = BIDS ADAPTER

23:3

TERMINAL TYPE

0 = MODEL 37 TELETYPE

1 = MODEL 33 TELETYPE

2 = MODEL 35 TELETYPE

3 = BIDS

26:1 INPUTANKING

DISK INPUT TANKING FLAG.

27:1

B249 TRANSLATION

1 IF THE BCL TO INTERNAL CODE TRANSLATION IN
THE DCTU SHOULD BE USED.

0 IF TRANSLATION IS BYPASSED

28:1 OUTPUTANKING

DISK OUTPUT TANKING FLAG. ON WHENEVER THERE
IS OUTPUT IN THE TANK FOR THIS LINE.

29:1 *PAPERTAPE

PAPER TAPE FLAG. WHEN ON, THE CARRIAGE
RETURN, LINE FEED RESPONSE IS SUPPRESSED
AND ALL OUTPUT GOING THROUGH DCWRITE IS
STOPPED (USED WHEN CONTINUOUS PAPER TAPE
IS BEING READ OR TO STOP OUTPUT WHILE
DISCARDING OUTPUT TANK AND QUEUE DURING

DISCONNECT, BREAK AND WRU PROCESSING.)

30:1 *IOINPROGRESS I/O IN PROGRESS FLAG. TURNED ON AT IIO TIME
AND OFF AT I/O FINISHED TIME.

31:1 CANDETHRU OFF IF C&E IS WORKING ON DISCONNECTING THIS
LINE.

32:1 DIRECTLINE DIRECT CONNECTION (NON DIAL-UP) FLAG.

33:15 ADDRESS OF LAST BUFFER SEGMENT (IE SEGMENTED
SAVE AREA) READ INTO BUT NOT YET LINKED
INTO THE WORKER QUEUE.

THE FORMAT OF ADINFO[0] IS:

14:2 WORD INDEX MINUS 1 OF TAIL OF ANSWERING QUEUE, I.E. IF
THIS FIELD IS A 1, WORD 2 OF THE AREA POINTED TO BY 18:
15 CONTAINS THE LAST ENTRY IN THE QUEUE

16:2 WORD INDEX MINUS 1 OF HEAD OF ANSWERING QUEUE.

18:15 ADDRESS OF THE AREA CONTAINING THE TAIL OF THE ANSWERING
QUEUE.

33:15 ADDRESS OF THE AREA CONTAINING THE HEAD OF THE ANSWERING
QUEUE.

LINETABLE

THE LINETABLE, WHICH IS LMAX/4 LONG, IS USED AFTER PASSIVE INTERROGATES
TO RELATE THE TERMINAL UNIT AND BUFFER NUMBER TO THE LOGICAL LINE
NUMBER. THE TERMINAL UNIT NUMBER AND THE FIRST TWO BITS OF THE BUFFER
NUMBER ARE USED AS AN INDEX INTO THE TABLE. FOR INSTANCE, LINETABLE[
0] CONTAINS LOGICAL LINE NUMBERS FOR TERMINAL UNIT 0, BUFFERS 0-3.

THE FINAL TWO BITS OF THE BUFFER ADDRESS ARE USED TO SPECIFY WHICH TWO CHARACTER FIELD CONTAINS THE LOGICAL LINE NUMBER FOR THIS BUFFER.

SEQARRAY

THE SEQARRAY, WHICH IS LMAX+1 LONG, IS USED WHEN DOING AUTOMATIC SEQUENCING OR WHEN READING PAPER TAPE. FOR SEQUENCING THE WORD FOR EACH LINE CONTAINS

1:1 =0, INDICATES WORD IS BEING USED FOR SEQUENCING.
2:19 INCREMENT
21:27 NEXT SEQUENCE NUMBER TO BE USED.

WHEN A PAPER TAPE IS BEING READ, THE SEQARRAY ENTRY FOR THAT LINE CONTAINS

<u>FIELD</u>	<u>DEFINE</u>	<u>CONTENTS</u>
1:1	PAPERTAPEFLAG	=1 TO INDICATE THAT THE WORD IS BEING USED FOR PAPER TAPE.
2:1		INDICATES WHICH OF THE 32 WORD BUFFERS IS BEING USED.
3:6	OFFSET	INDEX OF NEXT AVAILABLE WORD IN THE BUFFER.
9:1		ON WHEN DOING DISK I/O FOR PAPER TAPE. OFF OTHERWISE.
10:8	ROWNR	NUMBER OF ROWS IN DISK TANK FOR THIS LINE.
18:15		RELATIVE ADDRESS OF FIRST UNUSED SEGMENT IN THE CURRENT ROW.
33:15		ADDRESS OF BUFFERS.

INPUTANK

THE ARRAY INPUTANK IS MIXMAX+1 LONG AND CONTAINS INFORMATION ABOUT THE INPUT TANKS FOR EACH JOB IN THE MIX. THE DEFINES USED WITH IT ARE USED AS PARTIAL WORD DESIGNATORS,

FIELD	DEFINE	CONTENTS
-----	-----	-----
17:1	NDSABLE	ON IF JOB MAY NOT BE DS-ED, E.G. A CANDE JOB WHICH HAS TO FINISH ITS FILE MANIPULATION.
18:1		INTERLOCK BIT. IF THIS BIT IS OFF, THE TANK MAY NOT BE TOUCHED.
19:1	INPUTREADY	ON WHEN THE JOB IS SWAPPED OUT WAITING FOR INPUT.
20:1	SLOWDOWN	USER HAS BEEN TOLD TO STOP HIS INPUT SINCE THE TANK IS ALMOST FULL.
21:6	INPUTN	THE NUMBER OF MESSAGES IN THIS JOBS TANK.
27:6	INPUTL	ADDRESS WITHIN CLUMP OF NEXT MESSAGE TO BE PASSED TO JOB I.E. RELATIVE ADDRESS OF "OLDEST" INPUT.

TANKS

THE TANKS ARRAY, WHICH IS LMAX+1 LONG, IS USED TO STORE INFORMATION ABOUT THE OUTPUT TANKS FOR THE VARIOUS LINES, EXCEPT FOR DISCONNECTING, WHICH IS USED AS A PARTIAL WORD DESIGNATOR, THE DEFINES USED WITH TANKS ARE PARAMETRIC DEFINES.

<u>FIELD</u>	<u>DEFINE</u>	<u>CONTENTS</u>
1:1	DISCONNECTING	DISCONNECTING FLAG. ON IF NOW IN THE PROCESS OF THROWING AWAY OUTPUT FOR THIS LINE DUE TO DISCONNECT, BREAK OR WRU.
2:1		ACTIVITY BIT, SET AT EACH SWAP AND DC INTERRUPT.
3:3		POINTER TO WHICH 7 WORD (56 CHARACTER) PART OF THE SEGMENT BEING DETANKED IS NEXT.
6:1		CANDE SHUT-UP FLAG, TURNED ON WHEN A SHUT-UP EVENT IS ADDED TO THE QUEUE.
7:1		ON IF CANDE JOB HAS HAD BREAK WHILE RUNNING
10:8	TANKLINE	DETANKING QUEUE. THIS CONTAINS THE LINE NUMBER OF THE NEXT LINE AFTER THIS LINE FOR WHICH DETANKING IS TO BE DONE. (0 IF TAIL OF QUEUE OR NOT IN QUEUE.)
18:1		INTERLOCK BIT. OUTPUT TANKS MAY NOT BE TOUCHED IF THIS BIT IS 0.
19:1	TANKSWAP	ON WHEN THE OBJECT JOB IS SWAPPED OUT FOR ANY REASON.
20:1	TANKFUL	TANK FULL BIT. ON WHEN THE OBJECT JOB IS SWAPPED OUT TO WAIT FOR TANK TO EMPTY.
21:6	TANKN	NUMBER OF SEGMENTS USED IN THE TANK FOR THIS LINE.
27:6	TANKL	ADDRESS WITHIN TANK (I.E. WITHIN THIS USER-S GLOM) OF SEGMENT OUT OF WHICH NEXT DATA IS TO BE TAKEN, I.E. RELATIVE ADDRESS OF SEGMENT

BEING DETANKED.

33:15 TANKA

CORE ADDRESS OF TANK IN JOB-S AREA USED FOR
PACKING, 0 IF THERE IS NO SUCH AREA, I.E.
IF OUTPUT IS FROM A CANDE TYPE JOB.

THE FORK QUEUE: ITS STRUCTURE AND OPERATION

THE FORK QUEUE IS A QUEUE OF REQUESTS TO RUN INDEPENDENT PROCESSES. ENTRIES ARE PUT IN THE QUEUE BY FORK, AND ARE INITIATED BY THE CONTROL SECTION OF THE MCP NEAR THE LABELS NOTHINGTODD AND SLATESTARTER. THE ADDRESS OF THE HEAD AND TAIL OF THE QUEUE ARE KEPT IN FORKQUE, WHICH HAS THE FORMAT:

0:3 =5, MARKS THIS AS A PRESENT DATA DESCRIPTOR,
9:9 =777 (OCTAL)
18:15 CONTAINS THE ADDRESS OF THE LAST ENTRY IN THE QUEUE.
33:15 CONTAINS THE ADDRESS OF THE FIRST ENTRY IN THE QUEUE.

IF THE QUEUE IS EMPTY, THE TWO ADDRESS FIELDS POINT TO FORKQUE. NOTE THAT SINCE FORKQUE IS AN ARRAY DESCRIPTOR, THE CONTENTS OF THE FIRST ENTRY MAY BE ADDRESSED AS THE ARRAY FORKQUE[*].

ENTRIES TO THE FORK QUEUE ARE STORED IN 5 WORD SEGMENTED SAVE AREAS, WHERE:

WORD 0 9:9 CONTAINS THE PRIORITY+64,
 18:15 IS THE ADDRESS OF THE PRECEDING ENTRY OR OF
 FORKQUE IF THIS IS THE FIRST ENTRY,
 33:15 IS THE ADDRESS OF THE NEXT ENTRY, OR OF FORKQUE
 IF THIS IS THE LAST ENTRY.
WORD 1 18:15 CONTAINS LOGLINE,
 33:15 IS THE ADDRESS OF THE DESCRIPTOR FOR THE
 ROUTINE TO BE

WORD 3 ZERO IF STACK SPACE HAS BEEN OBTAINED, OTHERWISE,
 THE AMOUNT OF STACK SPACE NEEDED,
WORD 4 ADDRESS OF STACK SPACE IF ALREADY OBTAINED AND ZERO
 IF STACK SPACE IS NEEDED.

THE ENTRIES IN THE FORK QUEUE ARE ORDERED ACCORDING TO PRIORITY AND
ARE FIRST-IN, FIRST-OUT AMONG EQUAL PRIORITY REQUESTS.

WHEN CALLED, FORK IS PASSED THE ADDRESS OF THE PROCEDURE DESCRIPTOR,
A PARAMETER FOR THE PROCEDURE, THE PRIORITY, THE AMOUNT OF STACK SPACE
TO BE USED, AND A WORD CONTAINING EITHER THE ADDRESS OF STACK SPACE
ALREADY OBTAINED OR THE VALUE 1. AFTER GETTING AN AREA, LINKING IT
INTO THE QUEUE AND STORING THE INFORMATION, FORK RETURNS TO THE CALLING
PROCEDURE. HOWEVER, FOR PROCEDURES LACKING STACK SPACE, FORK FIRST
CHECKS THE STACK QUEUE, WHICH IS A QUEUE OF AREAS RESERVED BY INITIALIZE
FOR USE AS STACKS FOR INDEPENDENT RUNNERS. THERE ARE NUMSTACK STACKS
(CURRENTLY DEFINED IN INITIALIZE TO BE 2) AND EACH STACK IS STANDARDSTAC
LONG (CURRENTLY 90 WORDS). THE STACK AREAS WHICH ARE NOT BEING USED
ARE LINKED BY THE FIRST WORD OF THE AREA, AND THE ADDRESS OF THE FIRST
AREA IS KEPT IN STACKQ. IF THERE IS SPACE IN THE STACK QUEUE, FORK
DELINKS THE FIRST AREA, STORES ITS ADDRESS IN WORD 4 OF THE ENTRY AND
SETS WORD 3 TO ZERO. IF THE QUEUE IS EMPTY, WORD 3 IS SET TO STANDARDSTA K
AND WORD 4 IS SET TO ZERO.

IN ORDER TO START PROCEDURES IN THE FORK QUEUE, NOTHINGTODO CHECKS
THE QUEUE IMMEDIATELY AFTER INTERROGATING INTERRUPTS. IF THE QUEUE
IS NOT EMPTY, IT TAKES THE FIRST ENTRY, STORES THE INFORMATION CONTAINED
THEREIN, DELINKS IT AND FORGETS IT, IF THE PROCEDURE DOES NOT HAVE

STACK SPACE, IT CALLS GETSPACE BEFORE INITIATING THE PROCEDURE. IF SPACE CANNOT BE OBTAINED, IT CALLS FORK, WHICH LINKS THE ENTRY BACK INTO THE QUEUE BEHIND ANY OTHER ENTRIES WITH THE SAME PRIORITY, AND THEN RETURNS TO NOTHINGTODO. IF THE QUEUE IS EMPTY, NOTHINGTODO GIVES UP FORKING AND STARTS TO LOOK FOR SOMETHING IN THE BED.

WHEN AN INDEPENDENT RUNNER WHICH DOES NOT HAVE ITS OWN STACK IS FINISHED, IT CALLS KILL, PASSING IT THE ADDRESS OF THE STACK. IF THE STACK WAS OBTAINED BY GETSPACE IT IS RETURNED. IF THE STACK IS PART OF THE STACK QUEUE, IT IS LINKED INTO THE FRONT OF THE QUEUE. ITS MURDEROUS TASK COMPLETE, KILL GOES TO NOTHINGTODO.

THE BED: SLEEPING AND WAKING

THE BED IS A LINKED LIST OF CONTROL LINES, ENTRIES TO WHICH ARE MADE BY SLEEP. JOBS ARE PUT TO SLEEP TO WAIT FOR AN EXOGENOUS EVENT TO OCCUR - FOR EXAMPLE, AN I/O OPERATION TO FINISH, A TOGGLE TO BE RELEASED, OR A GIVEN LENGTH OF TIME TO PASS. CLEARING OF INTERRUPTS (SLEEP(1,1)) AND EUTHANASIA (SLEEP(0,0)) ARE ALSO POSSIBLE. THE TWO PARAMETERS PASSED TO SLEEP ARE A DESCRIPTOR, POINTING TO THE WORD TO BE TESTED AND A MASK TO TEST IT WITH. THE JOB IS AWAKENED IF ANY OF THE BITS WHICH ARE ON IN THE MASK ARE ALSO ON IN THE TEST WORD. IN ADDITION, THERE IS A VARIABLE, CLICK, IN WHICH A PROCEDURE CAN STORE A TIME OUT VALUE. IF THIS IS DONE, THE PROCEDURE WILL BE AWAKENED WHEN THE ACTUAL TIME EXCEEDS THE TIME OUT VALUE. IF IT IS NOT DONE, A VALUE OF 318145:43:32.25 IS USED. WHICH, IN EFFECT, ELIMINATES THE TEST. AS ANOTHER VARIATION, A PROCEDURE CAN PASS A BOOLEAN EXPRESSION TO COMPLEXSLEEP AND SLEEP UNTIL IT IS TRUE.

THE BED ENTRY FOR A PARTICULAR JOB IS KEPT IN THAT JOBS STACK. THE TWO PARAMETERS ARE SIMPLY LEFT IN THE STACK AND CONTAIN THE FOLLOWING INFORMATION:

PARAMETER 1 DESCRIPTOR POINTING TO THE WORD TO BE TESTED
 OR A 1 IF THIS ENTRY COMES FROM COMPLEX SLEEP.
PARAMETER 2 MASK OR ACCIDENTAL ENTRY PROGRAM DESCRIPTOR
 FOR COMPLEXSLEEP ENTRIES.

THEY ARE FOLLOWED BY THE RCW AND THEN SIX MORE WORDS ADDED BY SLEEP, WHICH CONTAIN:

WORD 0 2:1 1
 9:9 777 (OCTAL)
 18:15 PRIORITY + 64
 33:15 ADDRESS OF FIRST WORD ABOVE RCW IN THE NEXT
 ENTRY.
 WORD 2 MIX INDEX ASSOCIATED WITH THIS ENTRY.
 WORD 3 TIME OUT VALUE
 WORD 4 LOGLINE
 WORD 5 F REGISTER SETTING OF THE SLEEP ROUTINE WHEN IT
 MADE THE ENTRY (POINTS TO THE RCW).

THE HEAD AND TAIL OF THE QUEUE ARE KEPT IN BED AND BED1 WHICH HAVE
 THE FORMATS OF THE FORWARD AND BACK LINKS (WORDS 0 AND 1 ABOVE)
 RESPECTIVELY. THE PRIORITY FIELD OF BED IS 77777 (OCTAL). THE ENTRIES
 TO THE BED ARE ORDERED BY PRIORITY AND ARE FIRST-IN FIRST-OUT AMONG
 ENTRIES OF EQUAL PRIORITY. THE BACK LINK OF THE FIRST ENTRY AND THE
 FORWARD LINK OF THE LAST ENTRY BOTH POINT TO BED.

THE WORDS ABOVE THE RCW ARE NORMALLY REFERENCED AS AN ARRAY. THE BACK
 LINK OF AN ENTRY MAY BE USED TO ACCESS THE PRECEDING ENTRY AS A SINGLY
 DIMENSIONED ARRAY AND THE FOLLOWING ENTRY AS A DOUBLY DIMENSIONED
 ARRAY. THIS MAKES THE LINKING AND DELINKING OF AN ENTRY EXTREMELY
 SIMPLE.

JOBS ARE AWAKENED BY NOTHINGTODO, WHICH GOES TO THE BED AFTER PROCESSING
 THE FORK QUEUE. IT LINKS THROUGH THE BED, DOING AN INTERROGATE
 INTERRUPTS BEFORE PROCESSING ANY ENTRY, AND THEN SETTING THE S REGISTER

TO THE TOP OF THE ENTRY AND THE F REGISTER TO THE RCW. IF THE ENTRY
PASSES THE TESTS, IT IS DELINKED AND INITIATED VIA AN EXIT OPERATION.
IF IT DOES NOT PASS, THE NEXT ENTRY IS TESTED.

ASSIGNING JOBS TO CORE

THE AREA ABOVE THE FENCE IS DIVIDED INTO CHUNKS OF 1024 WORDS. WHEN A PROGRAM ENTERS THE MIX IT IS ASSIGNED A SET OF CONTIGUOUS CHUNKS IN WHICH TO RUN. THE NUMBER OF CHUNKS REQUIRED IS CALCULATED FROM THE ESTIMATE SUPPLIED BY THE COMPILER. ONCE A JOB HAS BEEN ASSIGNED TO AN AREA, IT IS ALWAYS RUN WITHIN THAT AREA.

THE AREA IN WHICH A JOB RUNS IS DIVIDED INTO MEMORY LINKS EXACTLY AS IT IT WERE ALL OF CORE ON THE STANDARD SYSTEM. WHEN GETSPACE AND FORGETSPACE ARE CALLED, THEY ARE GIVEN THE MIX INDEX OF THE JOB, WHICH THEY USE TO FIND THE AREA WITHIN WHICH SPACE NEEDS TO BE GOTTEN OR RETURNED. THUS, WHILE A JOB IS RUNNING, ITS STORAGE CAN BE OVERLAID OR MADE PRESENT IN THE STANDARD WAY. THE DIFFERENCE IS THAT IT IS CONFINED TO RUN IN A GIVEN AREA AND THAT NO OTHER PROGRAM CAN USE ANY OF THAT AREA.

SINCE A JOB ALWAYS OCCUPIES THE SAME AREA, NON-OVERLAYABLE STORAGE, SUCH AS THE PRT, CAN BE SWAPPED; THAT IS, IT CAN BE ROLLED OUT AND THEN ROLLED IN WITH THE REST OF THE PROGRAM. OF COURSE, SINCE IT IS NON-OVERLAYABLE, IT REMAINS IN CORE WHENEVER THE PROGRAM IS IN CORE.

THE ALGORITHM FOR ASSIGNING A PROGRAM TO AN AREA ATTEMPTS TO MINIMIZE THE CONFLICT FOR SPACE BETWEEN THAT PROGRAM AND THE OTHER PROGRAMS IN THE MIX. TO DO THIS, IT EXAMINES EACH POTENTIAL AREA, I.E. EACH SET OF THE NECESSARY NUMBER OF CONTIGUOUS CHUNKS, AND CALCULATES T , THE MAXIMUM NUMBER OF USERS OF ANY ONE CHUNK WITHIN THE AREA. THE JOB IS ASSIGNED TO THE AREA WITH THE MINIMUM T , AND INTO THE AREA LOWEST IN CORE AMONG AREAS WITH EQUAL T .

SWAPPING

ONE OF THE GOALS OF THE TIME SHARING SYSTEM IS TO MAINTAIN EFFICIENT USE OF CORE WHILE ENSURING A QUICK RESPONSE TO EVERY USER. THIS IS DONE BY PERIODICALLY ROLLING JOBS OUT OF CORE TO MAKE ROOM FOR OTHER JOBS. THERE ARE THREE REASONS FOR WHICH A JOB MAY BE ROLLED OUT WHILE IT IS RUNNING.

FIRST, EACH JOB IS ASSIGNED A TIME SLICE WHEN IT IS ROLLED IN. THIS TIME SLICE VARIES DEPENDING ON THE JOBS SIZE AND PRIORITY AND ON THE LENGTH OF TIME IT HAS BEEN RUNNING. WHEN A JOB HAS USED ITS ALLOTTED TIME SLICE, IT IS SWAPPED OUT.

SECOND, A JOB IS SWAPPED OUT IF THE OPERATOR STOPS IT WITH AN ST MESSAGE OR IF IT HAS TO WAIT FOR TELETYPE I/O, WHICH IS EXTREMELY SLOW RELATIVE TO THE COMPUTER. IT SHOULD BE NOTED THAT ON INPUT THE USER IS ALLOWED TO TYPE AHEAD OF HIS JOB, WHICH IS NOT SWAPPED OUT UNTIL IT HAS TO WAIT FOR DATA. ON OUTPUT, BEFORE IT IS ROLLED OUT, THE USERS PROGRAM IS ALLOWED TO FILL A TANK WHICH HOLDS APPROXIMATELY 10 MINUTES OF OUTPUT.

THIRD, WHEN A NEW JOB ENTERS THE SYSTEM, OR WHEN A JOB WHICH HAS BEEN WAITING FOR TELETYPE I/O RETURNS TO THE SYSTEM, AN ATTEMPT IS MADE TO GIVE IT AN IMMEDIATE TIME SLICE. THIS MEANS THAT JOBS ALREADY RUNNING MAY HAVE TO BE SWAPPED OUT. JOBS WHICH ARE SWAPPED OUT FOR THIS REASON ARE SWAPPED BACK IN AHEAD OF JOBS WHICH HAD USED THEIR FULL TIME SLICE.

NEW AND REENTERING JOBS ARE ALLOWED IMMEDIATE ACCESS TO CORE IN ORDER

TO GIVE THEM A CHANCE TO SEND A RESPONSE TO THE USER. SINCE THEIR FIRST TIME SLICE IS REALTIVELY SMALL, THEY ARE QUICKLY SWAPPED OUT, SO THAT OTHER USERS SUFFER ONLY A NEGLIGIBLE DELAY. THIS IMMEDIATE ACCESS IS ESPECIALLY VALUABLE TO PROGRAMS THAT GENERATE A GREAT DEAL OF OUTPUT IN A RELATIVELY SHORT TIME, SUCH AS THE LIST PROGRAM FOR CANDE.

THE PROCEDURES WHICH HANDLE THE ACTUAL SWAPPING OPERATE ON JOBS PLACED IN THE SWAP QUEUE BY THE OTHER PARTS OF THE MCP. IF A JOB IN THE SWAP QUEUE NEEDS TO BE SWAPPED OUT, IT IS DEALT WITH IMMEDIATELY, BUT IF IT IS READY TO BE SWAPPED IN, IT IS PLACED IN THE READY QUEUE, WHICH IS USED TO HOLD JOBS WAITING TO BE SWAPPED IN.

WHEN A JOB IS SWAPPED OUT, ITS AREA BECOMES AVAILABLE AND SO THE CHUNKS WITHIN IT ARE ASSIGNED TO THE JOBS HIGHEST IN THE READY QUEUE WHICH REQUIRE THEM. THE CHUNKS THUS ASSIGNED ARE SAID TO BE POSSESSED BY THE JOB TO WHICH THEY ARE ASSIGNED. WHEN A JOB POSSESSES A CHUNK, IT PREVENTS ALL OTHER JOBS FROM USING THAT CHUNK. AS A JOB SITS IN THE READY QUEUE IT ACCUMULATES CORE UNTIL IT POSSESSES THE AREA IT NEEDS. THEN IT IS SWAPPED IN AND ALLOWED TO RUN. WHEN IT IS SWAPPED OUT IT LOSES POSSESSION OF ITS CHUNKS, WHICH ARE REASSIGNED TO OTHER JOBS IN THE QUEUE.

THE EFFECT OF THIS IS THAT A JOB IN THE READY QUEUE CANNOT BE RUN IF ANY JOB ABOVE IT IN THE QUEUE REQUIRES ANY OF ITS CHUNKS. CONVERSELY, A JOB IN THE QUEUE IS RUN AHEAD OF ANY JOBS BELOW IT IN THE QUEUE WHICH REQUIRE A PART OF ITS AREA. THIS ENSURES EVERY JOB A CHANCE TO RUN NO MATTER HOW LARGE IT IS OR LOW ITS PRIORITY.

THERE ARE THREE ENTRY POINTS TO THE READY QUEUE, THEY ARE:

- 1) READYEND, INTO WHICH ARE PLACED NEW JOBS, JOBS WHICH HAVE WAITED FOR TELETYPE I/O, AND JOBS WHICH HAVE BEEN OK-ED AFTER BEING STOPPED.
- 2) FORCEND, INTO WHICH ARE PLACED JOBS WHICH ARE FORCED OUT BY JOBS AT READYEND.
- 3) RDRPTEND, INTO WHICH JOBS WHICH HAVE USED THEIR TIME SLICE ARE PLACED.

WHEN A JOB IS PLACED AT READYEND OR FORCEND, IT IS BEING PLACED IN FRONT OF OTHER JOBS IN THE QUEUE. IT IS THEREFORE ALLOWED TO TAKE CHUNKS FROM THE JOBS BELOW IT, THAT IS, IF A JOB BELOW THE NEW ENTRY TO THE QUEUE POSSESSES A CHUNK NEEDED BY THE NEW ENTRY, THAT CHUNK IS REASSIGNED TO THE NEW ENTRY. THUS, CHUNKS ARE ALWAYS POSSESSED BY THE HIGHEST JOB IN THE QUEUE WHICH CAN USE THEM.

FUTHERMORE, WHEN A JOB IS PLACED AT READYEND IT FORCES JOBS WHICH ARE USING CHUNKS IN ITS AREA OUT OF CORE, PROVIDED THAT THOSE JOBS DID NOT THEMSELVES ENTER THE QUEUE AT READYEND. WHEN THE FORCED OUT JOB IS ACTUALLY SWAPPED OUT, ITS CHUNKS ARE REASSIGNED, AND, SINCE READYEND IS THE HIGHEST ENTRY POINT INTO THE QUEUE, JOBS AT READYEND GET THE CHUNKS THEY NEED. IN THIS WAY, UNLESS PART OF THEIR AREA IS POSSESSED BY OTHER JOBS GETTING THEIR INITIAL TIME SLICE, JOBS WHICH ENTER AT READYEND GAIN POSSESSION OF THEIR AREA ALMOST IMMEDIATELY.

IT SHOULD BE NOTED THAT THE PRIORITY OF A JOB IS NOT A FACTOR IN THIS ALGORITHM. ALL JOBS, REGARDLESS OF THEIR PRIORITY, GET A TURN ON THE

SYSTEM. HOWEVER, THE PRIORITY OF A JOB DOES AFFECT THE LENGTH OF ITS TIME SLICE. THE LENGTH OF THE TIME SLICE IS ALSO AFFECTED BY THE NUMBER OF TIMES THE JOB HAS BEEN SWAPPED SINCE ITS LAST INTERACTION, I.E., SINCE IT WAS LAST PUT AT READYEND.

THE FORMULA FOR COMPUTING THE TIME SLICE IS

$$T = (1 + N + (9-P)/4 + C/4000) \times 64$$

WHERE

T IS THE PROCESSOR TIME LIMIT IN SIXTIETHS OF A SECOND.

N IS THE NUMBER OF TIMES THE JOB HAS BEEN SWAPPED SINCE ITS LAST INTERACTION,

P IS THE PRIORITY OF THE JOB.

C IS THE NUMBER OF WORDS OF CORE REQUIRED FOR THE JOB.

THE VARIABLE, N, IS NOT ALLOWED TO EXCEED SEVEN. THUS, ONCE A JOB HAS BEEN SWAPPED SEVEN TIMES, ITS TIME SLICE HAS REACHED THE MAXIMUM AND IS NO LONGER INCREASED BY FURTHER SWAPPING.

A JOB IS ALSO GIVEN AN ELAPSED TIME LIMIT OF ONE AND A HALF TIMES ITS PROCESSOR TIME LIMIT. IF IT EXCEEDS EITHER OF THESE LIMITS, IT IS SWAPPED OUT AND PLACED IN THE READY QUEUE AT RDYRPTEND.

IN SUMMARY, THE REASONS FOR WHICH A JOB CAN BE PLACED IN THE SWAP QUEUE AND THE ACTIONS TAKEN AS A RESULT ARE:

- 1) BEGINNING OF JOB. THE JOB IS PLACED IN THE READY QUEUE AT READYEND

- 2) END OF JOB, THE CHUNKS POSSESSED BY THE JOB ARE REASSIGNED TO JOBS IN THE READY QUEUE.
- 3) WAIT FOR EXTERNAL CONDITION. THE JOB IS SWAPPED OUT AND ITS CHUNKS ARE REASSIGNED.
- 4) EXTERNAL CONDITION SATISFIED, THE JOB IS ADDED TO THE READY QUEUE AT READYEND.
- 5) FORCE OUT, THE JOB IS ADDED TO THE READY QUEUE AT FORCEND AND THEN ITS CHUNKS ARE REASSIGNED.
- 6) END OF TIME SLICE, THE JOB IS ADDED TO THE READY QUEUE AT RDRYPTEND AND THEN ITS CHUNKS ARE REASSIGNED.

NOTE THAT JOBS WHICH ARE FORCED OUT OR WHICH HAVE USED THEIR TIME SLICE ARE ADDED TO THE QUEUE BEFORE THEIR CHUNKS ARE REASSIGNED. THIS MAKES IT POSSIBLE FOR THEM TO GET THEIR CHUNKS BACK IF NO OTHER JOBS CAN USE THEM.

SWAPPING STATUSES

FROM THE TIME IT ENTERS THE MIX UNTIL THE TIME IT LEAVES, EACH JOB HAS A STATUS ASSOCIATED WITH IT. THE STATUSES SPECIFY WHAT THE JOB IS DOING AT ANY GIVEN TIME. THE STATUSES ARE:

- 0 - TIMEND THE JOB IS WAITING TO BE SWAPPED OUT AFTER USING ITS TIME SLICE.
- 1 - WAITSWAP THE JOB IS WAITING TO BE SWAPPED OUT TO WAIT FOR AN EXTERNAL CONDITION.
- 2 - BOJSTATE NEW JOB WAITING TO BE ADDED TO READY QUEUE.
- 3 - SATISFY THE JOB IS WAITING TO BE ADDED TO THE READY QUEUE AFTER AN EXTERNAL CONDITION HAS BEEN SATISFIED.
- 4 - EOJSTATE THE JOB HAS REACHED COMPLETION AND IS WAITING TO HAVE ITS CHUNKS REASSIGNED.
- 5 - FORCESWAP THE JOB IS WAITING TO BE SWAPPED OUT DUE TO A FORCE OUT.
- 6 - TRANSIT THE JOB IS BEING SWAPPED IN OR OUT.
- 8 - WAITSTATE THE JOB HAS BEEN SWAPPED OUT ON DISK AND IS WAITING FOR AN EXTERNAL CONDITION.
- 9 - READystate JOB IS IN THE READY QUEUE AFTER WAITING FOR AN EXTERNAL CONDITION.
- 10 - RDYPPT JOB IS IN THE READY QUEUE AFTER BEING FORCED OUT OR AFTER USING ITS TIME SLICE.
- 11 - READYB NEW JOB IN THE READY QUEUE
- 16 - RUNNING JOB IS IN CORE.
- 32 - SELECTING NEW JOB BEING PROCESSED AFTER GAINING POSSESSION

OF ALL OF ITS CHUNKS.

56 - STABLE JOB HAS LOST ITS CHUNKS AFTER AN END OF JOB,
ALSO USED TO MASK FOR ALL STATUSES BELOW 7.

NOTE THAT STATUSES 0 - 5 APPLY TO JOBS IN THE SWAP QUEUE AND STATUSES
9 - 11 APPLY TO JOBS IN THE READY QUEUE.

THE SWAPPING PROCEDURES

THE PROCEDURES WHICH DEAL WITH SWAPPING ARE:

INITIALSWAP	ASSIGNS AN AREA TO A JOB, SETS UP MEMORY LINKS AND GETS A SWAP AREA ON DISK.
SWAP	ENTERS A JOB IN THE SWAP QUEUE AND GIVES IT THE APPROPRIATE STATUS.
BRINGBACK	CALLED TO ENTER A JOB IN THE SWAP QUEUE AFTER THE EXTERNAL CONDITION FOR WHICH IT HAS BEEN WAITING IS SATISFIED.
SWAPPER	HANDLES THE ACTUAL SWAPPING.
SWAPPINGIO	DOES I/O FOR SWAPPER.
UNHOOKANDWAIT	DISCONNECTS A JOB ABOUT TO BE SWAPPED OUT.
HOOKUPMCP	RECONNECTS A JOB THAT HAS BEEN SWAPPED IN.
REENTER	USED BY HOOKUPMCP.

WHEN IT FINDS A JOB IN THE SWAP QUEUE, SWAPPER USES ITS STATUS TO DETERMINE WHAT ACTION TO TAKE, AND THEN DEALS WITH IT AS DESCRIBED UNDER "SWAPPING." IF THE JOB REQUIRES SWAPPING OUT, SWAPPER CALLS UNHOOKANDWAIT WHICH STORES THE LOCATION OF THE TOP OF THE PROGRAMS STACK, DELINKS THE BED ENTRY FOR THE JOB, REMOVES THE JOB FROM THE MCP AND INTRINSIC TABLES AND THEN SLEEPS UNTIL ALL I/O FOR THE JOB IS COMPLETE. SWAPPER THEN CALLS SWAPPINGIO AND THE JOB IS WRITTEN INTO THE AREA RESERVED FOR IT ON DISK. IF APPROPRIATE, THE JOB IS LINKED INTO THE READY QUEUE AND THEN ITS CHUNKS ARE REASSIGNED ONE AT A TIME. AS SOON AS A JOB HAS GAINED POSSESSION OF ITS ENTIRE AREA, IT IS SWAPPED IN, AND THEN CHUNK REASSIGNMENT CONTINUES.

TO SWAP A JOB IN, SWAPPER FIRST CALLS SWAPPINGIO TO READ THE JOB BACK INTO CORE AND THEN HOOKUPMCP, WHICH IN TURN CALLS FOR REENTER TO RUN INDEPENDENTLY AND THEN SLEEPS UNTIL REENTER HAS RUN. REENTER SETS THE S AND F REGISTERS TO THE TOP OF THE JOBS STACK AND THEN TRANSFERS (I. E. NOT A CALL) TO SLEEP, WHICH WILL USE THE S AND F REGISTERS TO MAKE A BED ENTRY FOR THE JOB. SINCE THE BED ENTRY FOR HOOKUPMCP PRECEDES THE ONE FOR THE JOB, HOOKUPMCP IS AWAKENED IN TIME TO PUT THE JOB BACK INTO THE MCP TABLES. IT THEN RETURNS TO SWAPPER WHICH CONTINUES TO REASSIGN CHUNKS, AFTER PROCESSING ALL THE AVAILABLE CHUNKS, IT GOES TO THE NEXT JOB IN THE SWAP QUEUE, OR, IF THE SWAP QUEUE IS EMPTY, COMMITS SUICIDE BY CALLING KILL. EVENTUALLY, AS THE MCP PROCESSES THE BED, IT WILL FIND THE JOB AND RESTART IT AS IF IT HAD NEVER BEEN SWAPPED.

WHEN OTHER PROCEDURES IN THE MCP DISCOVER THAT A JOB NEEDS TO BE SWAPPED, THEY CALL BRINGBACK IF THE JOB IS SWAPPED OUT AND CAN NOW BE SWAPPED BACK IN AND SWAP FOR ALL OTHER CASES. THESE ROUTINES LINK THE JOB INTO THE SWAP QUEUE AND CHECK TO SEE IF SWAPPER IS DEAD OR ALIVE. IF IT IS DEAD, IT IS REINCARNATED AS AN INDEPENDENT RUNNER AND MARKED AS BEING ALIVE. THIS ALLOWS SWAPPER TO BE OVERLAID WHEN IT HAS NOTHING TO DO.

WHEN A JOB ENTERS THE MIX, THE SELECTRUN PROCEDURE, THE EQUIVALENT OF THE SELECTION PROCEDURE IN THE STANDARD MCP, IS CALLED AS DESCRIBED IN THE NARRATIVE DESCRIPTION OF THE DFMCP, BUT, INSTEAD OF CALLING GETSPACE, SELECTRUN CALLS INITIALSWAP, WHICH ASSIGNS THE JOB TO AN AREA, CALLS SWAP TO QUEUE IT INTO THE SWAP QUEUE, AND THEN SLEEPS

UNTIL THE JOB HAS GAINED POSSESSION OF ITS AREA. WHEN A NEW JOB ACCUMULATES ALL OF ITS CHUNKS, SWAPPER REMOVES IT FROM THE READY QUEUE AND GIVES IT THE STATUS SELECTING. THIS AWAKENS INITIALSWAP SO THAT IT CAN SET UP THE MEMORY LINKS AND GET A SWAP AREA ON DISK. IT THEN RETURNS TO SELECTRUN WHICH PROCEEDS TO INITIATE THE JOB IN THE USUAL WAY.

EXPANDING AN AREA

ALTHOUGH THE CORE AREA INITIALLY ASSIGNED TO A JOB WILL ORDINARILY BE SUFFICIENT, CASES CAN ARISE IN WHICH A JOB NEEDS MORE SPACE. THIS CONDITION SHOWS UP WHEN GETSPACE IS UNABLE TO OBTAIN ENOUGH SPACE TO SATISFY A REQUEST. IN THAT CASE, IF THE JOB IS NOT ALREADY ASSIGNED TO THE ENTIRE AREA ABOVE THE FENCE, GETSPACE WILL INITIATE AN EXPANSION OF ITS AREA BY CALLING SWAP TO FORCE THE JOB OUT (I.E., THE JOB WILL ENTER THE READY QUEUE AT FORCEND) AND THEN SLEEP UNTIL THE JOB IS AGAIN RUNNING, AT WHICH TIME IT WILL HAVE AN EXTRA CHUNK AND GETSPACE CAN TRY AGAIN. THIS IS REPEATED UNTIL SUFFICIENT SPACE IS OBTAINED OR UNTL THE JOB IS ASSIGNED THE MAXIMUM NUMBER OF CHUNKS. WHEN A JOB RUNS OUT OF SPACE AND NO CORE IS AVAILABLE FOR EXPANSION, GETSPACE SLEEPS AND THEN TRIES AGAIN. IF SPACE IS NOT AVAILABLE ON THE THIRD TRY, THE JOB IS TERMINATED.

DURING THE SWAPPING OF A JOB WHICH IS BEING EXPANDED, SWAPPINGIO CALLS EXPANDER. WHEN THE JOB IS BEING SWAPPED OUT, EXPANDER CHOOSES THE DIRECTION IN WHICH THE JOBS AREA SHOULD BE EXPANDED IN ORDER TO MINIMIZE CONFLICT WITH OTHER JOBS, CHANGES SC OR LC ACCORDINGLY AND EXITS. THE JOB THEN SITS IN THE READY QUEUE UNTIL IT POSSESSES ITS AREA, WHICH IS NOW ONE CHUNK LARGER THAN IT WAS. WHEN THE JOB IS SWAPPED IN, EXPANDER IS CALLED AGAIN. IT SETS UP THE MEMORY LINKS TO INCLUDE THE NEW AREA AND, IF THE JOB WAS ADDED IN FRONT, THE JAR AND UV TABLE ENTRIES FOR THE JOB ARE MOVED TO THE FRONT OF THE NEW AREA.

LINE MAINTENANCE

ONE OF THE PROBLEMS FACING ANY TIME SHARING SYSTEM IS THAT PEOPLE AND TELETYPES HAVE I/O REQUIREMENTS THAT ARE QUITE DIFFERENT FROM THOSE OF COMPUTERS. THE REMOTE USERS MUST BE ABLE TO TRANSMIT AND RECEIVE A RELATIVELY SLOW BUT STEADY STREAM OF INFORMATION, WHEREAS PROGRAMS REQUIRE SHORT BURSTS OF MUCH FASTER I/O. IN THE B5500 TIME SHARING SYSTEM THE NEEDS OF BOTH THE USER AND THE PROGRAM ARE MET BY PERFORMING THEIR I/O INDEPENDENTLY. IN ORDER TO DO THIS AS EFFICIENTLY AS POSSIBLE, BOTH INPUT AND OUTPUT MAKE USE OF QUEUEING STRUCTURES AND DISK TANKS.

AS DATA COMES IN FROM A TELETYPE, IT IS COLLECTED IN 28 CHARACTER BUFFERS IN THE DTTU. WHEN A BUFFER IS FULL IT IS READ INTO A SEGMENTED SAVE AREA AND LINKED TO ANY OTHER BUFFERS WHICH ARE PART OF THE SAME MESSAGE. THE COMPLETE MESSAGE, AS INDICATED BY A GROUP MARK, IS THEN LINKED INTO THE WORKER QUEUE, WHICH CONSISTS OF BUFFERS LINKED BY THE FIRST WORD, WHERE

- 2:2 SPECIFIES THE SIZE OF THE AREA
- 4:1 IS THE END OF MESSAGE FLAG, INDICATING THAT THIS IS THE LAST BUFFER OF A MESSAGE
- 7:1 IS USED TO IDENTIFY THE SPECIAL MESSAGE WHICH MARKS THE OCCURRENCE OF A DISCONNECT.
- 9:1 IS THE END OF QUEUE FLAG.
- 10:8 IS THE LOGICAL LINE NUMBER
- 18:15 CONTAINS THE ADDRESS OF THE PREVIOUS BUFFER IN THIS MESSAGE OR ZERO IF THIS IS THE FIRST BUFFER.

33:15 CONTAINS THE ADDRESS OF THE NEXT BUFFER IN THE QUEUE OR
ZERO IF THIS IS THE LAST BUFFER.

THE ADDRESSES OF THE HEAD AND TAIL OF THE QUEUE ARE KEPT IN WORKERINQ.
NOTE THAT BUFFERS ARE LINKED BACK ONLY TO A MESSAGE BOUNDARY, BUT ARE
LINKED FORWARD ACROSS MESSAGE BOUNDARIES.

THE BUFFERS IN THE WORKER QUEUE ARE CONSOLIDATED INTO MESSAGES AND
THEN STORED IN THE APPROPRIATE DISK TANK. INPUT FROM ALL USERS TO
CANDE IS STORED IN A TANK WHICH CONSISTS OF CHUNKS CONTAINING
TANKCHUNKSIZE (CURRENTLY 256) SEGMENTS OF USER DISK. THE MCP OBTAINS
ADDITIONAL CHUNKS AS THEY ARE NEEDED AND CANDE RETURNS THE CHUNKS
WHEN IT IS THROUGH WITH THEM. WHEN A SEGMENT IS WRITTEN ON DISK, IT
IS ALSO LEFT IN CORE UNTIL CANDE HAS DONE ITS INITIAL SCANNING. THE
FORMAT OF A SEGMENT ON DISK IS

WORD 0 3:15 RELATIVE DISK ADDRESS WITHIN THIS CHUNK
 18:15 CORE ADDRESS OF NEXT SEGMENT
 40:8 NUMBER OF WORDS OF DATA IN THIS SEGMENT
WORDS 1-29 MESSAGES

IN ADDITION, IN CORE TWO FURTHER WORDS ARE USED.

WORD -1 DECIMAL DISK ADDRESS OF THE SEGMENT.
WORD -2 I/O DESCRIPTOR USED TO WRITE IT ONTO DISK

EACH MESSAGE CONSISTS OF ONE INFORMATION WORD PLUS AS MANY WORDS AS
ARE REQUIRED TO HOLD IT. THE FORMAT OF THE INFORMATION WORD IS:

1:1 SPECIAL MESSAGE FLAG

10:8 LOGICAL LINE NUMBER

40:8 NUMBER OF CHARACTERS IN MESSAGE

THE CORE AND DISK ADDRESSES OF A SEGMENT ARE THEN PASSED TO CANDE IN A DATA EVENT.

WHEN THE USER ENTERS DATA FROM A PAPER TAPE, IT IS STORED IN A TANK FILE CREATED FOR THE LINE BY THE TAPE COMMAND. THIS FILE IS SET UP EXACTLY LIKE THE CANDE TANKS; THAT IS, IT CONTAINS ROWS OF TANKCHUNKSIZE SEGMENTS AND THE SEGMENTS, BOTH IN CORE AND ON DISK, HAVE THE FORMAT DESCRIBED ABOVE. HOWEVER, FOR PAPERTAPE THE SEGMENTS IN CORE ARE USED ONLY AS BUFFERS AND THEIR CONTENTS ARE NOT FURTHER SAVED.

IF THE USER IS CONNECTED TO A USER PROGRAM, HIS DATA IS STORED ACCORDING TO THE MIX INDEX OF THE PROGRAM. THE DISK AREA FOR THIS, WHICH IS GOTTEN DURING INITIALIZATION, CONTAINS CLUMPSIZE×MIXMAX SEGMENTS, THUS ALLOWING CLUMPSIZE (CURRENTLY 16) SEGMENTS TO EACH USER. THE ADDRESS OF THE BASE OF THIS AREA IS KEPT IN PROGTANK AND A USER USES THE AREA BEGINNING AT RELATIVE SEGMENT CLUMPSIZE×MIX. THE TANK FOR ANY GIVEN JOB IS USED CYCLICALLY. WHEN A PROGRAM ASKS FOR DATA IT IS READ OFF OF THE DISK AND INTO BUFFERS IN THE PROGRAMS AREA.

OUTPUT DATA FOR A GIVEN LINE IS NORMALLY STORED IN THE OUTPUT TANK FOR THAT LINE. THE OUTPUT TANKS ARE KEPT IN A DISK AREA OF LMAX×GLOMSIZE SEGMENTS WHICH IS OBTAINED DURING INITIALIZATION. (CURRENTLY, GLOMSIZE IS 32.) THE AREA FOR A SPECIFIC LINE BEGINS AT GLOMSIZE×LL, WHERE LL IS THE LINE NUMBER. MESSAGES FROM CANDE AND THE MCP ARE WRITTEN DIRECTLY INTO A SEGMENT WITHOUT BEING PACKED, BUT MESSAGES

FROM PROGRAMS RUNNING ABOVE THE FENCE ARE PACKED INTO A 30 WORD ARRAY WITHIN THAT PROGRAMS AREA, AND THEN WRITTEN ONTO DISK AS A FULL SEGMENT. THE FIRST WORD OF ANY SEGMENT CONTAINS THE NUMBER OF CHARACTERS IN THE SEGMENT AND THE REMAINING 29 WORDS CONTAIN OUTPUT WITH A SINGLE GROUP MARK AT THE END. WHEN A JOB IS SWAPPED OUT OR WHEN AN ARRAY GETS FULL (I.E. THE NEXT MESSAGE REQUIRES MORE SPACE THAN IS LEFT IN THE ARRAY) IT IS WRITTEN ONTO THE DISK.

THE PROCEDURE WHICH PERFORMS THE ACTUAL OUTPUT OF INFORMATION TAKES DATA FOR A GIVEN LINE FROM THAT LINE-S LINE QUEUE. WHEN THE LINE QUEUE IS EMPTY, DATA IS TAKEN FROM THE TANK 56 CHARACTERS AT A TIME AND ADDED TO THE LINE QUEUE. OUTPUT DATA IS ADDED DIRECTLY TO THE LINE QUEUE ONLY IF BOTH THE TANK AND THE LINE QUEUE ARE EMPTY.

THE LINE QUEUE FOR A GIVEN LINE CONSISTS OF BUFFERS WHICH ARE THE CORRECT SIZE FOR THE ADAPTER ON THAT LINE. THEY ARE STORED IN SEGMENTED SAVE AREAS AND ARE LINKED BY AN INFORMATION WORD WHICH HAS THE FORMAT

4:1 END OF MESSAGE FLAG
10:8 LOGICAL LINE NUMBER
18:15 ADDRESS OF PREVIOUS ENTRY IN THE LINE QUEUE OR ZERO IF
 THIS IS THE FIRST ENTRY.
33:15 ADDRESS OF THE NEXT ENTRY IN THE LINE QUEUE OR ZERO IF
 THIS IS THE LAST ENTRY.

THE ADDRESS OF THE HEAD AND TAIL OF THE LINE QUEUE FOR LINE I IS KEPT IN DCREQUEST[I]. DCREQUEST IS ALSO USED TO KEEP THE READY QUEUE, A LINKED LIST OF LINES WHICH HAVE I/O-S (DATA TO BE WRITTEN, BLAST READS

OR ACTIVE INTERROGATES) READY FOR OUTPUT IN THEIR LINE QUEUES. THE
FORMAT OF AN ENTRY IN DCREQUEST IS

1:9 STATUS OF THE LINE.

10:8 LINE NUMBER OF THE NEXT LINE IN THE READY QUEUE OR ZERO
IF THIS LINE IS NOT IN THE READY QUEUE OR IS THE LAST
LINE IN THE READY QUEUE.

18:15 ADDRESS OF THE TAIL OF THE LINE QUEUE OR ZERO IF THE LINE
QUEUE IS EMPTY.

33:15 ADDRESS OF THE HEAD OF THE LINE QUEUE OR ZERO IF THE LINE
QUEUE IS EMPTY.

THE LINE NUMBERS OF THE HEAD AND TAIL OF THE READY QUEUE ARE KEPT IN
DCREQUEST[0].

LINE MAINTENANCE PROCEDURES

THE MCP PROCEDURES THAT PERFORM LINE MAINTENANCE CAN BE ROUGHLY DIVIDED INTO THREE GROUPS. THE PROCEDURES IN THE FIRST GROUP SERVICE THE TERMINAL UNIT BUFFERS AND OPERATE PRIMARILY AS A PART OF IOFINISH. THE PROCEDURES IN THE THIRD GROUP INTERFACE DIRECTLY WITH THE NORMAL STATE PROGRAMS AND ARE MOSTLY CALLED VIA COMMUNICATES. THE SECOND GROUP OF PROCEDURES OPERATES BETWEEN THE OTHER TWO AND IS THEREFORE RESPONSIBLE FOR MOST OF THE TANKING.

THE PROCEDURES WHICH FALL INTO THE FIRST GROUP ARE:

ENTERLINEQ	ENTERS DATA INTO THE LINE QUEUE.
ENTEREADYQ	ENTERS A LINE INTO THE READY QUEUE IF THE LINE IS NOT ALREADY IN THE READY QUEUE.
NEXTDCIO	PERFORMS WRITES AND SOME INTERROGATES.
INTERROGATOR	HANDLES INQUIRY REQUEST INTERRUPTS.
DCIOFINISH980	HANDLES ALL I/O FINISHED INTERRUPTS FOR 980 ADAPTERS.
HELLO	HANDLES DIAL-UPS.
WRURRESPONSE	RESPONDS TO WRU AND IS USED AT DIAL-UP TIME.
QUITTER	USED DURING THE PROCESSING OF DISCONNECTS, WRUS AND BREAKS.
GIVEAWAY	LINKS MESSAGES INTO THE WORKER QUEUE.

THERE ARE TWO WAYS IN WHICH THIS SET OF ROUTINES CAN GET CONTROL. FIRST, DATA MAY BE PASSED TO ENTERLINEQ FOR OUTPUT. THIS CAUSES ENTEREADYQ TO BE CALLED, AND POSSIBLY NEXTDCIO. SECOND, THEY GET

CONTROL AS A RESULT OF INTERRUPTS GENERATED BY THE TERMINAL UNITS.

THERE ARE TWO KINDS OF INTERRUPTS GENERATED BY THE TERMINAL UNITS. THE FIRST IS AN INQUIRY REQUEST INTERRUPT, WHICH INFORMS THE MCP THAT THE STATUS OF A BUFFER HAS CHANGED. THE MCP MUST THEN INITIATE A PASSIVE INTERROGATE TO FIND THE ADDRESS OF THE BUFFER IN QUESTION AND ITS NEW STATUS. THE SECOND IS AN I/O FINISHED INTERRUPT, WHICH INFORMS THE MCP THAT A READ, WRITE OR INTERROGATE IS COMPLETE AND PASSES BACK INFORMATION ABOUT THE COMPLETED OPERATION IN THE I/O RESULT DESCRIPTOR. THE I/O FINISHED INTERRUPT FOR A READ OR A WRITE OPERATION SIGNALS ONLY THAT THE DATA HAS BEEN TRANSFERRED BETWEEN CORE AND THE TERMINAL UNIT BUFFERS. THE TRANSMISSION OF DATA TO AND FROM THE REMOTE UNIT IS HANDLED INDEPENDENTLY BY THE ADAPTERS, AND THE MCP IS INFORMED OF ITS COMPLETION VIA AN INQUIRY REQUEST INTERRUPT AND A PASSIVE INTERROGAT RETURNING AN IDLE INTERRUPT.

THIS SYSTEM OF INTERRUPTS AND INTERROGATES ALLOWS THE MCP TO KEEP MANY BUFFERS BUSY SIMULTANEOUSLY. FOR INSTANCE, AFTER IT RECEIVES THE I/O FINISHED INTERRUPT FOR A WRITE, IT CAN DO I/O-S ON OTHER LINES WHILE THE DATA IS BEING TRANSMITTED TO THE USER. THEN, WHEN IT FINDS OUT THAT THE TRANSMISSION IS COMPLETE, IT CAN INITIATE FURTHER I/O ON THAT LINE.

THE PROCEDURE WHICH HANDLES THE INQUIRY REQUEST IS INTERRAGATOR. IF THERE IS NO I/O CHANNEL ASSIGNED TO DATA COMMUNICATIONS, IT ENTERS AN INTERROGATE IN THE I/O QUEUE BY CALLING IDREQUEST. OTHERWISE, IT INFORMS NEXTDCIO THAT AN INTERROGATE NEEDS TO BE DONE.

WHEN NEXTDCIO GETS CONTROL, IF A CHANNEL IS AVAILABLE AND AN INTERROGATE NEEDS TO BE DONE, THE INTERROGATE IS INITIATED. OTHERWISE, THE FIRST BUFFER LOAD IN THE LINE QUEUE OF THE FIRST LINE IN THE READY QUEUE IS WRITTEN, USING THE AVAILABLE CHANNEL IF THERE IS ONE OR, IF NO CHANNEL IS AVAILABLE, BY CALLING IOREQUEST.

NEXTDCIO GETS CONTROL FROM TWO SOURCES. ONE IS ENTERREADYQ, WHICH CALLS NEXTDCIO IF NO DATA COMMUNICATIONS I/O IS IN PROGRESS. IN THIS CASE, AN I/O CHANNEL MAY OR MAY NOT BE AVAILABLE. THE OTHER SOURCE FROM WHICH IT GETS CONTROL IS DCIOFINISH980, IN WHICH CASE AN I/O CHANNEL IS AVAILABLE SINCE DCIOFINISH980 GETS CONTROL BEFORE ANYTHING FURTHER IS INITIATED ON THE CHANNEL THAT GENERATED THE I/O FINISHED INTERRUPT. THIS ABILITY OF LINE MAINTENANCE TO KEEP AN I/O CHANNEL ALLOWS IT TO INITIATE ITS OWN I/O UNTIL IT HAS NO MORE TO DO WITHOUT WAITING FOR THE I/O QUEUEING STRUCTURE OR SHARING THE CHANNEL WITH THE REST OF THE SYSTEM. WHEN DCIOFINISH980 GETS CONTROL, IT STARTS AN I/O AS SOON AS POSSIBLE, EITHER BY INITIATING IT ITSELF OR BY CALLING NEXTDCIO.

SINCE IT PROCESSES ALL OF THE DATA COMMUNICATIONS I/O FINISHES, DCIOFINISH980 IS THE PRIMARY CONTROLLING ROUTINE FOR THE FIRST GROUP. FOR SOME OF THE SPECIAL CASES REQUIRING RELATIVELY EXTENSIVE PROCESSING IT CALLS SUCH ROUTINES AS WRURESPONSE AND THE INDEPENDENT RUNNERS, HELLO AND QUITTER. MOST CASES, HOWEVER, CAN BE HANDLED ALMOST ENTIRELY WITHIN DCIOFINISH980.

WHEN A PASSIVE INTERROGATE REPORTS A READ READY BUFFER, DCIOFINISH980 IMMEDIATELY INITIATES A READ. WHEN IT GETS THE I/O FINISHED INTERRUPT FROM THAT, IT CHECKS FOR A GROUP MARK. IF THERE IS ONE, IT GENERATES

THE CARRIAGE RETURN, THE LINE FEED, AND, IF THE SEQ COMMAND IS BEING USED, THE SEQUENCE NUMBER, LINKS THEM INTO THE FRONT OF THE LINE QUEUE AND CALLS ENTEREADYQ. IT THEN LINKS THE BUFFER TO THE REST OF THE MESSAGE AND CALLS GIVEAWAY AND NEXTDCIO. IF THERE IS NO GROUP MARK, IT SIMPLY CALLS NEXTDCIO AND LINKS THE BUFFER TO THE MESSAGE.

WHEN DCIOFINISH980 RECIEVES THE I/O FINISHED INTERRUPT AFTER A WRITE, IT FIRST CHECKS TO SEE IF THE BUFFER WAS ALREADY BUSY, THIS CAN OCCUR IF THE USER HAS STARTED TYPING BUT HAS NOT YET FILLED THE BUFFER OR TYPED A GROUP MARK OR IF THE TRANSMISSION OF AN EARLIER WRITE TO THE USER IS NOT YET COMPLETE. IF THE BUFFER WAS BUSY, THE WRITE COULD NOT BE DONE AND SO IT IS RELINKED INTO THE FRONT OF THE LINE QUEUE. IF THE BUFFER WAS NOT BUSY, THE CORE BUFFER IS FORGOTTEN AND THE LINE QUEUE IS CHECKED. IF THERE IS MORE DATA IN THE LINE QUEUE, ENTEREADYQ IS CALLED. IF THERE IS NOTHING IN THE LINE QUEUE, BUT THERE IS DATA IN THE TANK, THIS LINE IS ADDED TO THE DETANKING QUEUE. LATER, WHEN THE TRANSMISSION OF DATA TO THE USER IS COMPLETE, DCIOFINISH980 LINKS THE LINE INTO THE READY QUEUE IF THERE IS DATA IN THE LINE QUEUE AND THEN CALLS NEXTDCIO

OLDWEIRDHAROLD TANKS INPUT AND DETANKS OUTPUT
NOTIFYCANDE QUEUES UP DATA EVENTS.
DCWRITE OUTPUTS MESSAGES.
DCDISKIO WRITES DATA IN CANDE-S TANKS AND GETS NEW CHUNKS
 WHEN NEEDED.
PAPERTAPEDISKIO WRITES DATA IN PAPER TAPE TANKS AND GETS NEW ROWS
 WHEN NEEDED.

CALLITOFF HANDLES THE END OF PAPER TAPE INPUT.

THE MOST IMPORTANT OF THESE PROCEDURES IS OLDWEIRDHAROLD.

OLDWEIRDHAROLD, ALIAS THE WORKER, DOES MANY STRANGE AND WONDROUS THINGS SOME OF WHICH ARE USEFUL. IF HE HAS NOT BEEN FORKED ALREADY, HE IS FORKED WHENEVER SOMETHING IS ADDED TO ONE OF HIS QUEUES. WHEN GIVEN CONTROL HE FIRST PROCESSES THE WORKER QUEUE, ONE MESSAGE AT A TIME, AND THEN, WHEN IT IS EMPTY, DETANKS DATA FOR THE FIRST LINE IN THE DETANKING QUEUE. AFTER HANDLING A LINE IN THE DETANKING QUEUE, HE PROCESSES ANYTHING WHICH WAS ADDED TO THE WORKER QUEUE WHILE HE WAS DETANKING. HE CONTINUES TO PROCESS THE WORKER QUEUE AND THEN DETANK DATA FOR A LINE UNTIL HE HAS DETANKED DATA FOR ALL LINES IN THE QUEUE, AFTER WHICH HE GOES TO NOTHINGTODO.

TO PROCESS THE WORKER QUEUE, OLDWEIRDHAROLD SCANS DOWN THE BUFFERS LOOKING FOR A LEFT ARROW, PROCESSING BACKSPACES AND CONSOLIDATING AND FORGETTING BUFFERS AS HE GOES. WHEN HE HAS A COMPLETE MESSAGE, HE CHECKS TO SEE IF THE LINE IS CONNECTED TO CANDE, IS BEING USED FOR PAPER TAPE INPUT OR IS CONNECTED TO A USER PROGRAM.

SINCE CANDE EXPECTS TO FIND INPUT MESSAGES BOTH IN CORE AND ON DISK, THEY ARE WRITTEN INTO 30 WORD ARRAYS, THE NEXT ARRAY TO BE PASSED TO CANDE IS FIRSTSEG AND THE ARRAY INTO WHICH DATA IS CURRENTLY BEING PLACED IS LASTSEG. THE FIRST WORD OF THE DATA TO BE PASSED NEXT IS POINTED TO BY FIRSTOFFSET AND THE NEXT AVAILABLE WORD IN LASTSEG IS POINTED TO BY LASTOFFSET. IN A DATA EVENT, CANDE IS PASSED THE ADDRESS OF FIRSTSEG, FIRSTOFFSET, AND THE NUMBER OF WORDS OF DATA. THUS, IF

FIRSTSEG IS NOT EQUAL TO LASTSEG, CANDE IS PASSED THE ENTIRE SEGMENT, AND FIRSTSEG AND FIRSTOFFSET ARE ADJUSTED TO POINT TO THE NEXT SEGMENT. IF FIRSTSEG EQUALS LASTSEG, CANDE IS PASSED ALL AVAILABLE INPUT (IE EVERYTHING BETWEEN FIRSTOFFSET AND LASTOFFSET) AND FIRSTOFFSET IS UPDATED. IF THERE IS NO DATA AVAILABLE, THE FLAG CANDEINPUTREADY IS SET.

WHEN HARRY HAS A MESSAGE FOR CANDE, HE CHECKS TO SEE IF IT WILL FIT INTO THE SPACE REMAINING IN LASTSEG. IF IT WONT, HE SETS UP A NEW ARRAY BEFORE CONTINUING. HE THEN WRITES THE MESSAGE INTO LASTSEG AND, IF CANDEINPUTREADY IS ON, CALLS NOTIFYCANDE.

NOTIFYCANDE USES DCDISKID TO WRITE FIRSTSEG ONTO DISK (POSSIBLY WRITING OVER A PREVIOUS VERSION OF THIS SEGMENT WHICH WAS NOT AS FULL) AND QUEUES UP A DATA EVENT. IN THIS WAY, CANDE WILL FIND THE DATA BOTH IN CORE AND ON DISK. IT SHOULD BE NOTED THAT SINCE A DISK I/O MUST BE DONE EVERYTIME CANDE REQUESTS DATA, FEWER I/OS ARE PERFORMED WHEN CANDE IS AT LEAST A FULL SEGMENT BEHIND LINE MAINTENANCE. HOWEVER, AS CANDE FALLS FURTHER BEHIND, MORE CORE IS USED TO HOLD CANDE-S TANKS. WIN A FEW, LOSE A FEW.

MESSAGES FROM PAPER TAPE ARE STORED IN ONE OF TWO 30 WORD AREAS WHICH ARE PART OF A 65 WORD AREA GOTTEN FOR THE LINE WHEN THE TAPE COMMAND IS RECEIVED. THIS AREA IS SET UP AS FOLLOWS:

WORD -1	1:20	ADDRESS OF HEADER IN DIRECTORY
	21:27	BASE DISK ADDRESS OF CURRENT ROW
WORDS 0-31		FIRST 30 WORD BUFFER ARRAY, INCLUDING THE TWO

ADDITIONAL WORDS DESCRIBED ABOVE.

WORDS 32-63 SECOND 30 WORD BUFFER ARRAY

WORD 64 I/O FINISH MASK

THE TWO BUFFER AREAS ARE USED ALTERNATELY IN MUCH THE SAME MANNER AS PING-PONG BUFFERS. WHEN, HARRY HAS A MESSAGE FROM A PAPER TAPE, HE FIRST CHECKS TO SEE IF IT WILL FIT INTO THE CURRENT SEGMENT. IF NOT, HE CALLS PAPERTAPEDISKIO WHICH GETS A NEW ROW IF NECESSARY, WRITES THE CURRENT SEGMENT ONTO DISK, SWITCHES THE AREAS AND THEN SLEEPS UNTIL ANY PREVIOUS I/O ON THE NEW AREA IS FINISHED. HARRY THEN CHANGES LASTSEG TO POINT TO THE CURRENT AREA AND LASTOFFSET TO THE OFFSET IN SEQARRY. WHEN HE IS THROUGH, HE RESETS LASTSEG AND LASTOFFSET AND UPDATES OFFSET.

WHEN HE RECEIVES THE SIGNAL INDICATING THE END OF PAPER TAPE INPUT, HE CALLS CALLITOFF, WHICH CALLS PAPERTAPEDISKIO TO WRITE THE REMAINING DATA ONTO DISK AND RESETS THE ADINFO AND SEQARRY ENTRIES FOR THE LINE. THEN, SINCE THE FLAGS ARE RESET, HARRY SENDS THE END OF PAPER TAPE MESSAGE ON TO CANDE, WHICH FURTHER PROCESSES THE DISK FILE BEFORE SENDING A NUMBER SIGN TO THE USER.

IF THE JOB IS CONNECTED TO A USER, HARRY WRITES THE MESSAGE INTO THE USERS TANK, AND CALLS BRINGBACK IF THE JOB WAS SWAPPED OUT TO WAIT FOR IMPUT, IF, ON THE OTHER HAND, THERE IS ROOM FOR ONLY 3 MORE MESSAGES IN THE TANK, THE USER IS ASKED TO "PLEASE WAIT".

WHEN DETANKING OUTPUT FOR A GIVEN LINE, HARRY FIRST CHECKS TO SEE IF THERE IS ANYTHING IN THE DISK TANKS, IF NOT, HE TAKES EVERYTHING IN

THE JOBS CORE TANKS. (NOTE THAT WHEN A JOB IS SWAPPED OUT ITS CORE TANK IS WRITTEN INTO ITS DISK TANK.) IF THERE IS DATA ON DISK, HE TAKES THE NEXT 56 CHARACTERS. FOR THIS PURPOSE, A POINTER IS KEPT WHICH POINTS TO THE NEXT 56 CHARACTER HUNK TO BE OUTPUT. HE THEN CALLS DCWRITE, WHICH BREAKS THE MESSAGE INTO BUFFERS OF THE PROPER SIZE FOR THE LINE, STORES THEM IN SEGMENTED SAVE AREAS AND CALLS ENTERLINEQ.

THE PROCEDURES IN THE THIRD GROUP ARE:

COMMS	HANDLES CANDES INPUT REQUEST COMMUNICATE.
COMM13	HANDLES INPUT REQUESTS FROM USER PROGRAMS.
TWXOUT	HANDLES EVERYBODYS OUTPUT.
OUTRAN980	FORMATS OUTPUT FOR 980 ADAPTERS.
CLEARTANK	CALLED DURING A SWAP OUT TO WRITE THE USERS OUTPUT ARRAY INTO HIS DISK TANK.

COMMS SIMPLY FORGETS OLD DATA ARRAYS WHEN CANDE IS THROUGH WITH THEM AND THEN CALLS NOTIFYCANDE. INTERLOCKS BETWEEN COMMS AND OLDWEIRDHAROLD ARE AVOIDED BY JUDICIOUSLY CHOOSING WHERE EACH PROCEDURE LOSES CONTROL SO THAT ONE MAY RUN EVEN WHILE THE OTHER SLEEPS. COMM13 CHECKS TO SEE IF THERE IS ANYTHING IN THE USERS INPUT TANKS. IF NOT, IT CALLS SWAP TO SWAP THE JOB OUT. OTHERWISE IT GETS THE NEXT MESSAGE FROM DISK, AND, IF THE USER HAD BEEN TOLD TO WAIT AND HIS TANK IS NOW HALF EMPTY, IT CALLS TWXOUT TO PRINT THE MESSAGE "CONTINUE TYPING".

TWXOUT FIRST CHECKS TANKS [7:1] WHICH IS SET WHEN BREAK IS TYPED. IF IT IS ON, TWXOUT EXITS WHICH IN EFFECT THROWS AWAY THE OUTPUT. THEN IT CHECKS THE OUTPUT TANKS. IF ONLY NINE SEGMENTS ARE LEFT AND THE

LINE IS CONNECTED TO A JOB ABOVE THE FENCE, SWAP IS CALLED. IF ONLY SIX SEGMENTS ARE LEFT, CANDE IS PASSED A SHUT-UP EVENT. IF THE TANK IS FULL, TWXOUT SLEEPS UNTIL THERE IS ROOM FOR THE MESSAGE

ANY CHARACTERS NOT IN THE 980 CHARACTER SET BY QUESTION MARKS, ADDS A CARRIAGE RETURN IF REQUESTED, FOLLOWED BY AS MANY LINE FEEDS AS REQUESTED, AND THEN FOLDS THE LINE IF NECESSARY. IF THERE IS NOTHING IN THE LINE QUEUE OR THE TANK FOR THIS LINE, DCWRITE IS CALLED, OTHERWISE, THE OUTPUT IS TANKED. IF THE LINE IS CONNECTED TO A CANDE TYPE JOB RUNNING ABOVE THE FENCE, THERE IS NO CORE TANK FOR PACKING AND SO THE MESSAGE IS WRITTEN INTO THE NEXT DISK SEGMENT. IF THE LINE IS CONNECTED TO A JOB ABOVE THE FENCE, THE MESSAGE IS PACKED INTO THE CORE TANK UNLESS IT WILL NOT FIT, IN WHICH CASE THE TANK IS WRITTEN ONTO THE NEXT DISK SEGMENT AND THE MESSAGE IS WRITTEN INTO THE FRONT OF THE CORE TANK.

DELETE

QUESTION MARKS, EXCLAMATION POINTS AND EOTS ARE ALL SIGNALLED BY THE ABNORMAL FLAG IN THE I/O RESULT DESCRIPTOR FOR A READ. DCIOFINISH980 MUST, THEREFORE, SCAN THE BUFFER. IF THERE ARE NO EXCLAMATION POINTS (BCL \geq), THE BUFFER SIMPLY CONTAINS QUESTION MARKS, AND IS HANDLED IN THE USUAL WAY. IF THERE ARE EXCLAMATION POINTS, AN ACTIVE INTERROGATE MUST BE DONE TO FIND OUT IF IT WAS A DELETE OR AN EOT. IF THE BUFFER IS "NOT READY" IT IS AN EOT AND IS HANDLED LIKE A DISCONNECT AS DESCRIBED BELOW. OTHERWISE, IT IS A DELETE AND SO THE BUFFER OR BUFFERS IN THIS MESSAGE ARE FORGOTTEN AND THE "DEL" MESSAGE IS LINKED INTO

THE FRONT OF THE LINE QUEUE.

DIAL UP

ONE OF THE BITS IN THE ADINFO TABLE SPECIFIES WHETHER OR NOT A LINE IS DIALED-UP. THIS APPLIES TO LINES WHICH ARE DIRECTLY CONNECTED AS WELL AS TO THOSE WHICH ARE NOT. IF INPUT IS RECEIVED FROM A DIRECT LINE WHICH IS NOT MARKED AS DIALED-UP IT IS TREATED AS A NEW LINE. FOR THIS PURPOSE, A CHECK IS MADE DURING THE PROCESSING OF A READ READY INTERRUPT TO SEE IF THE LINE IS DIALED-UP. SIMILARLY, A WRITE READY ABNORMAL INTERRUPT, WHICH CAN BE EITHER A DIAL-IN OF A WRU, IS ASSUMED TO BE A DIAL-IN IF THE LINE IS NOT MARKED AS BEING DIALED-UP. IT IS THEREFORE GOOD PRACTICE TO INITIATE A DIRECT LINE WITH A WRU SO THAT IT WILL BE TREATED EXACTLY LIKE A DIAL-IN.

WHEN IT RECEIVES A DIAL-IN, DCIOFINISH980 ADDS A NEW LINE TO THE ANSWERING QUEUE, AND, IF IT HAS NOT BEEN STARTED ALREADY, HELLO IS FORKED. AN ENTRY IN THE ANSWERING QUEUE CONSIST OF ONE WORD PER LINE WHERE:

10:8 CONTAINS THE LOGICAL LINE NUMBER

18:30 CONTAINS THE TIME OF DIAL-UP (PLUS FOUR SECONDS FOR
 LINES NOT DIRECTLY CONNECTED)

THE QUEUE IS KEPT IN 5 WORD SEGMENTED SAVE AREAS, THE FIRST WORD OF AN AREA CONTAINS THE ADDRESS OF THE NEXT AREA, IF THERE IS ONE, IN THE 33:15 FIELD. THE REMAINING FOUR WORDS ARE ENTRIES TO THE QUEUE. AFTER THE FOURTH ENTRY IN AN AREA IS PROCESSED, THE AREA IS FORGOTTEN.

NEW AREAS ARE OBTAINED WHEN A LINE DIALS-UP AND EITHER THERE ARE NO OTHER ENTRIES IN THE QUEUE OR THE FOURTH ENTRY OF THE PREVIOUS AREA HAS BEEN USED.

HELLO TAKES THE FIRST ENTRY IN THE QUEUE AND CHECKS TO SEE IF THE CURRENT TIME IS GREATER THAN THAT STORED IN THE ENTRY. IF THE LINE IS DIRECT, THIS IS ALWAYS THE CASE, BUT IF THE LINE REALLY DID DIAL-IN, IT MEANS THAT FOUR SECONDS HAVE ELAPSED WHICH PACIFIES ONE OF THE GLITCHES IN THE ADAPTER. IF THE CURRENT TIME IS NOT GREATER, HELLO SLEEPS UNTIL IT IS. IT THEN CALLS WRURESPONSE WHICH SENDS OUT THE SHARINGLINE AS DIALED-UP IN ADINFO AND IN THE FILE SYSTEM/DISK. AFTER ALL ENTRIES IN THE QUEUE ARE PROCESSED, IT CALLS KILL, LEAVING HARD-HEARTED CANDE TO HANDLE THE LOG-IN THROUGH THE USUAL I/O CHANNELS.

BREAK, WRU, EOT AND DISCONNECT

THE OCCURRENCE OF A BREAK, WRU, EOT OR DISCONNECT IS SIGNALLED BY AN INTERRUPT AND THEREFORE IS RECOGNIZED BY DCIOFINISH980, WHICH QUEUES THE APPROPRIATE EVENT FOR CANDE, BLOCKS FURTHER OUTPUT TO THE LINE BY SETTING THE PAPERTAPE FLAG, FORKS THE PROCEDURE QUITTER, THROWS AWAY ANY INPUT NOT YET ADDED TO THE WORKER QUEUE, DELINKS THE LINE FROM THE READY QUEUE AND THROWS AWAY THE LINE QUEUE, AND FINALLY CALLS NEXTDCIO. IN ADDITION, IF A WRU IS BEING PROCESSED, WRURESPONSE IS CALLED. IF A DISCONNECT, I.E. DISCONNECT OR EOT, IS BEING PROCESSED, THE DIALEDUP AND CANDETHRU FLAGS ARE TURNED OFF, THE DISCONNECT BIT IN TANKS IS TURNED ON, AND THE MESSAGE "BYE++++" IS ADDED TO THE WORKER QUEUE TO MARK THE SPOT AT WHICH THE DISCONNECT OCCURRED.

WHEN QUITTER GETS CONTROL AFTER A WRU OR A DISCONNECT, IT TERMINATES THE JOB TO WHICH THE LINE IS ATTACHED IF IT CAN BE DS-ED. AFTER A BREAK OR IF THE JOB IS NOT DS-ABLE, QUITTER SETS THE 7:1 BIT IN TANKS TO STOP OUTPUT, CLEARS THE INPUT TANK BY RESETTING INPUTANK AND, IF THE JOB WAS SWAPPED OUT TO WAIT FOR INPUT OR FOR THE OUTPUT TANKS TO CLEAR, IT IS BROUGHT BACK AND ALLOWED TO RUN. THEN, IN ALL CASES, IF OUTPUT IS BEING TANKED, THE LINE IS DELINKED FROM THE DETANKING QUEUE, THE TANKS ARE FORGOTTEN BY RESETTING TANKS, AND, IF IT IS A BREAK OR WRU AND CANDE HAD BEEN PASSED A SHUT-UP EVENT, IT IS NOW TOLD TO GO AHEAD. (ON DISCONNECT, QUITTER SLEEPS UNTIL CANDE SETS THE CANDETHRU FLAG BEFORE PROCESSING THE OUTPUT TANKS.) NEXT QUITTER RESETS PAPERTAPE AND, FOR A BREAK OR WRU SENDS CANDE A LINE CLEAR EVENT WHEREAS ON A DISCONNECT IT UPDATES SYSTEM/DISK. QUITTER THEN QUILTS VIA KILL.

AFTER RECEIVING A BREAK OR WRU EVENT, CANDE DISCARDS ANY FURTHER EVENTS FOR THE LINE UNTIL IT HAS RECEIVED THE LINE CLEAR EVENT AND, IF A JOB WAS RUNNING ABOVE THE FENCE, THE EOJ MESSAGE EVENT. IT THEN SENDS THE POUND SIGN RESPONSE TO INDICATE THAT THE USER CAN CONTINUE. AFTER A DISCONNECT, CANDE THROWS AWAY EVENTS UNTIL IT HAS THE "BYE<<<<" MESSAGE AND, IF APPROPRIATE, THE EOJ MESSAGE. IT THEN MARKS THE LINE DISCONNECTED IN ITS OWN ARRAYS AND SETS THE CANDETHRU FLAG FOR THE MCP VIA A -16 COMMUNICATE.

SYSTEM/DISK, AND HALT/LOAD RECOVERY

THE FILE SYSTEM/DISK CURRENTLY CONTAINS THE FOLLOWING INFORMATION FOR EACH LINE ON THE SYSTEM:

WORD 0 ADINFO TABLE ENTRY FOR THE LINE

WORD 1 USER-CODE IF LOGGED-ON, 0 IF NOT

WORD 2 TIME OF LAST ACTIVITY

WORD 3 LENGTH OF TIME BEFORE AUTOMATIC DISCONNECT.

THE FILE IS INITIALIZED BY THE PROGRAM SYSDISK/MAKER, WHICH READS IN THE INFORMATION FOR THE ADINFO ENTRIES AND ESTABLISHES THE BLOCKING FOR THE FILE. I/O TO AND FROM SYSTEM/DISK IS HANDLED BY THE ROUTINE SYSDISKIO, WHICH ENSURES THAT THE FILE IS ACCESSED BY ONLY ONE PROCEDURE AT A TIME AND ALSO, IF POSSIBLE, USES THE BLOCKING TO MINIMIZE THE NUMBER OF I/O-S PERFORMED.

WORDS 2 AND 3 ARE USED DURING NORMAL SYSTEM OPERATION TO CAUSE AN AUTOMATIC DISCONNECT ON LINES WHICH HAVE BEEN IDLE AN EXCESSIVE LENGTH OF TIME, CURRENTLY 1 MINUTE DURING LOG-ON AND AFTER A LOG-OFF AND 10 MINUTES OTHERWISE. THIS TIME IS SET BY CANDE AND STORED IN WORD 3. THERE IS ALSO A BIT IN THE MCP WHICH IS SET WHENEVER AN ACTIVITY OCCURS FOR THE LINE, FOR INSTANCE, AN I/O OR A SWAP. WHEN A TIMER INTERRUPT OCCURS, IF THE BIT IS ON, THE TIME IS STORED IN WORD 2. IF THE BIT IS OFF, AND THE TIME SINCE THE LAST ACTIVITY IS GREATER THAN THAT IN WORD 3, AN AUTOMATIC DISCONNECT OF THE USER IS INITIATED.

AFTER A HALT/LOAD, INITIALIZE CALLS SPREADTHEWORD WHICH USES SYSTEM/

DISK TO UPDATE ADINFO AND TO SEND THE "PLOP" MESSAGE TO ALL LINES DIALED-UP. THIS IS ALL THAT CAN BE DONE WITHOUT CANDE. THE REST OF HALT/LOAD RECOVERY IS CONCERNED WITH RECOVERING FROM THE LOSS OF CANDE AND THEREFORE APPLIES WHENEVER THE CE MESSAGE IS TYPED.

AFTER A CE MESSAGE, THE PROCEDURE STARTCANDY IS CALLED. IF THERE ARE NO TANKS STARTCANDY GETS DISK SPACE FOR A CHUNK, AND, IN ALL CASES QUEUES A TANK CHUNK EVENT. IF THE TANKS EXISTED PRIOR TO THE HALT/LOAD, CANDE CAN FIND OUT WHERE THE CHUNKS WERE FROM THIS EVENT. THEN, FOR EACH LINE WHICH WAS LOGGED-ON, STARTCANDE QUEUES UP A RESTART EVENT.

A RESTART EVENT IS PROCESSED LIKE A STANDARD DIAL-IN. IF A RESTART COMMAND IS ENTERED CANDE ASKS FOR THE OLD LINE NUMBER WHICH IS USED TO FIND THE FILES WHICH MAKE UP THE WORK FILE. IF THE OLD LINE NUMBER AND NEW LINE NUMBER ARE DIFFERENT, THE NAMES OF THE FILES ARE CHANGED TO REFLECT THE NEW LINE NUMBER. THE FILE 1P <LOGICAL LINE NUMBER> /< USER CODE> (SEE "THE WORK FILE") IS SEARCHED TO FIND THE END OF THE POINTERS, WHICH IS MARKED BY AN ENTRY OF 10E8. FOLLOWING THIS ARE AN ADDITIONAL 30 WORDS OF INFORMATION CONCERNING THE USER. CANDE USES THESE TO RESET ITS INTERNAL PARAMETERS AND TO TELL THE USER THE SEQUENCE NUMBER OF THE LAST LINE WHICH WAS SAVED. SINCE THE POINTERS AND THE OTHER INFORMATION ARE WRITTEN INTO THE FILE EVERY 10 LINES, AT MOST 10 LINES OF INPUT CAN BE LOST ON A HALT/LOAD.

MOVING THE FENCE

WHEN AN MF MESSAGE IS TYPED, THE PROCEDURE FENCEMOVER IS CALLED. IF NECESSARY IT ROUNDS THE VALUE ENTERED FOR THE FENCE UP SO THAT THERE WILL BE AN INTEGRAL NUMBER OF CHUNKS ABOVE THE FENCE. THEN, IF THE RESULTANT VALUE IS IN THE RANGE FROM 8184 TO 28644, THE VALUE IS WRITTEN ONTO DISK AT DIRECTORYTOP+19. IF THE VALUE IS OUT OF RANGE, AN INV KBD MESSAGE IS OUTPUT.

AFTER A HALT/LOAD, THE PROCEDURE INITIALIZE COMPARES DIRECTORYTOP+19 TO THE PREVIOUS VALUE OF THE FENCE. IF THEY ARE DIFFERENT AND DIRECTORYTOP+19 IS NOT EQUAL TO ZERO, THE VALUE OF FENCE IS CHANGED, BOTH IN CORE AND ON MCP DISK. INITIALIZE IS THEN RESTARTED WITH THE NEW FENCE VALUE.

OPERATION OF CANDE

THE COMMAND AND EDIT LANGUAGE PROCESSOR, CANDE/TSHARER, IS AN INDEPENDENT NORMAL STATE PROGRAM WHICH RUNS BELOW THE FENCE. IT IS STARTED DURING INITIALIZATION AND CONTINUES TO RUN INDEFINITELY THEREAFTER.

SINCE CANDE IS A NORMAL STATE PROGRAM, IT MUST GET INFORMATION ABOUT THE STATUS OF JOBS AND LINES FROM THE MCP. THIS IS DONE IN THE FORM OF EVENTS, WHICH ARE QUEUED BY THE MCP IN THE ORDER OF THEIR OCCURRENCE AND ARE PASSED TO CANDE WHEN IT REQUESTS ONE VIA A COMMUNICATE. AN EVENT IS A FIVE WORD ARRAY. THE FIRST WORD CONTAINS THE LOGICAL LINE NUMBER TO WHICH THE EVENT BELONGS AND THE TYPE OF EVENT. THE REMAINING WORDS ARE USED TO PASS ADDITIONAL INFORMATION AS REQUIRED BY THE DIFFERENT EVENT TYPES. THE POSSIBLE EVENTS ARE LISTED IN TABLE 1.

IN ADDITION, THE MCP COLLECTS INPUT FROM THE USERS IN TANKS AND THEN ADDS A DATA EVENT TO THE QUEUE WHEN CANDE ASKS FOR DATA. THUS, CANDE CAN CONTROL BOTH THE RATE AT WHICH IT RECEIVES INFORMATION AND THE RATE AT WHICH IT RECEIVES NEW DATA. THIS ALLOWS IT TO PROCESS EACH EVENT TO COMPLETION AND ALSO TO PROCESS ONE LINE OF INPUT AS FAR AS POSSIBLE BEFORE STARTING ON ANOTHER.

WHEN CANDE RECEIVES A DATA EVENT, WHICH MAY CONTAIN MANY MESSAGES, IT STORES IT AND CONTINUES TO PROCESS THE EVENT QUEUE. WHEN THE EVENT QUEUE IS EXHAUSTED, CANDE PROCESSES THE FIRST MESSAGE IN THE DATA EVENT BY GENERATING A PSEUDO DATA EVENT CONTAINING ONLY THAT MESSAGE. WHEN THE QUEUE IS AGAIN EMPTY, IT PROCESSES THE NEXT MESSAGE, CONTINUING

IN THIS FASHION UNTIL IT HAS PROCESSED ALL THE MESSAGES IN THE DATA EVENT, AT WHICH TIME IT ASKS THAT A NEW DATA EVENT BE ADDED TO THE QUEUE AND THEN GETS THE NEXT EVENT FROM THE QUEUE, THUS, SINCE CANDE CONTROLS THE ADDITION OF DATA EVENTS TO THE EVENT QUEUE AND CAN THEREFORE POSTPONE THE PROCESSING OF NEW DATA UNTIL THE EVENT QUEUE IS EMPTY, EVENTS RESULTING FROM COMMANDS IN PROCESS ARE, IN EFFECT, GIVEN PRIORITY OVER NEW COMMANDS. THIS ENSURES A SMOOTH FLOW OF INFORMATION THROUGH THE SYSTEM.

THE MAIN BODY OF CANDE CONSISTS OF A NON-TERMINATING LOOP WHICH REQUESTS DATA WHEN IT IS NEEDED, GETS AN EVENT AND THEN CALLS HANDLETHISLINE. HANDLETHISLINE TRANSFERS CONTROL TO THE APPROPRIATE ROUTINES THROUGH A CASE STATEMENT IN WHICH EACH STATEMENT IS A PROCEDURE STATEMENT. CALLS TO PROCEDURES WHICH HANDLE EVENTS ARE POSITIONED SO THAT THE NUMBER OF A PROCEDURE STATEMENT WITHIN THE CASE STATEMENT IS EQUAL TO THE NUMBER OF THE EVENT WHICH THE PROCEDURE HANDLES. PROCEDURES WHICH ARE NOT DIRECTLY ASSOCIATED WITH AN EVENT, SUCH AS THOSE TO HANDLE SPECIFIC COMMANDS, ARE PLACED AFTER THE PROCEDURES TO HANDLE EVENTS. NEARLY ALL TRANSFERS OF CONTROL FROM ONE PROCEDURE TO ANOTHER ARE DONE USING THIS CASE STATEMENT.

THE PROCESSING OF ONE EVENT OFTEN INVOLVES SEVERAL STEPS AND MAY LEAD TO FURTHER EVENTS, FOR INSTANCE, A COMMAND IS PASSED TO CANDE AS A DATA EVENT AND MUST BE IDENTIFIED BEFORE IT CAN BE PROCESSED. PROCESSING MAY REQUIRE SEVERAL PROCEDURES AND IT MAY INVOLVE WAITING WHILE SOMETHING, SUCH AS A COMPILE, IS DONE ELSEWHERE. FOR INSTANCE, THE PROCEDURE TO HANDLE A RUN COMMAND HAS TO WAIT WHILE THE PROGRAM IS

EXECUTED AND THEN MUST REGAIN CONTROL WHEN AN END-OF-JOB EVENT OCCURS. HANDLETHISLINE MUST THEREFORE BE ABLE TO CALL THE APPROPRIATE PROCEDURES AND, WHEN THEY OCCUR, TO RECOGNIZE EVENTS FOR WHICH SOMEONE IS WAITING. TO DO THIS, FOR EACH LINE CONNECTED TO THE SYSTEM THERE IS AN ARRAY WHICH IS USED AS A STACK FOR THAT LINE. THE VARIOUS PROCEDURES USE IT FOR SCRATCH AND TO PASS PARAMETERS. IT IS ALSO USED TO STORE CONTROL WORDS INFORMATION NEEDED TO START OR RESTART THAT PROCEDURE. (SEE TABLE 2). IN ADDITION, WHEN A PROCEDURE GIVES UP CONTROL, IT USES THREE VARIABLES TO INDICATE WHAT FURTHER PROCESSING, IF ANY, IS REQUIRED.

IAM INDICATES WHETHER OR NOT THE PROCEDURE WISHES TO REGAIN CONTROL.

WANT SPECIFIES WHICH OTHER PROCEDURE, IF ANY, SHOULD GET CONTROL NEXT

GIVEN TO THE PROCEDURE SPECIFIED BY WANT OR RETURNED TO THE PROCEDURE SPECIFIED IN IAM.

WHEN WAITFOR IS NON-ZERO, IT IS STORED AND HANDLETHIS LINE IS EXITED SO THAT ANOTHER EVENT CAN BE PROCESSED. THEN, WHEN THE EVENT SPECIFIED BY WAITFOR OCCURS, CONTROL IS TRANSFERRED TO THE MOST RECENT RCW. THE POSSIBLE COMBINATIONS OF IAM, WANT AND WAITFOR ARE LISTED IN TABLE 3.

WHEN AN EVENT WHICH IS NOT THE RESULT OF SOME PREVIOUS EVENT OCCURS, SUCH AS A DIAL-UP OR A DATA EVENT, AN RCW FOR THE FINISHED PROCEDURE IS STORED IN THE STACK ARRAY AT ITS HIGHEST LOCATION. THEREAFTER, ANY FURTHER RCW-S ARE STORED BELOW THE PREVIOUS RCW. THE VARIABLE PREVRCW

ALWAYS POINTS TO THE RCW MOST RECENTLY ADDED TO THE STACK.

THE LOWER PART OF THE STACK IS USED FOR SCRATCH AND FOR PARAMETERS. WHEN A NEW PROCEDURE IS GIVEN CONTROL BY HANDLETHISLINE, THE VARIABLE, BASE, IS SET TO POINT TO THE FIRST AVAILABLE SCRATCH LOCATION IN THE STACK, THE PROCEDURE THEN USES THE VARIABLE SREG TO POINT TO THE LOCATION IT IS CURRENTLY USING, BASE IS CHANGED ONLY WHEN CONTROL IS TRANSFERRED FROM ONE PROCEDURE TO ANOTHER. THE SPACE IMMEDIATELY ABOVE THE BASE IS USED FOR TEMPORARY STORAGE BY THE PROCEDURE AND THE NUMBER OF WORDS USED IS STORED IN TEMP. ABOVE ITS TEMPORARY STORAGE, THE PROCEDURE STORES ANY PARAMETERS THAT NEED TO BE PASSED TO THE NEXT PROCEDURE AND PLACES THE NUMBER OF PARAMETERS IN PARAM. BEFORE ANOTHER PROCEDURE IS GIVEN CONTROL, PARAM IS ALSO STORED AT THE NEW BASE.

TABLE 1. EVENTS

FOR ALL EVENTS, THE RELEVANT PARTS OF WORD 0 ARE:

18:7 EVENT NUMBER
 25:8 LOGICAL LINE NUMBER

THE CONTENTS OF THE REMAINING WORDS IN THE VARIOUS EVENTS ARE:

<u>EVENT</u> <u>-----</u>	<u>WORD</u>	<u>CONTENTS</u> <u>-----</u>
1 DIAL-UP	1-4	NOT USED
2 BREAK	1-4	NOT USED
3 WHO ARE YOU	1-4	NOT USED
4 LINE CLEAR AFTER BREAK OR WRU	1-4	NOT USED
5 DISK CHUNK	1	DISK ADDRESS OF CHUNK
	2	DISK ADDRESS OF MESSAGE/CANDE
	3	ADDRESS OF HEADER OF TANK FILE
	4	DISK ADDRESS OF USERS/CANDE
6 DISCONNECT	1-4	NOT USED
7 DATA EVENT	1	FIRSTOFFSET, I.E. RELATIVE ADDRESS IN SEGMENT OF FIRST WORD OF DATA.
	2	NUMBER OF WORDS OF DATA
	3	RELATIVE DISK ADDRESS OF SEGMENT
	4	NOT USED
8 NOT USED		

9 HALT/LOAD RECOVERY	1	USER CODE
	2-4	NOT USED
10 SHUT-UP	1	NUMBER OF EVENT TO WAIT FOR BEFORE CONTINUING (USUALLY ALSO 10 IN WHICH CASE THIS WORD IS 0 FOR THE CONTINUE EVENT.)
	2-4	NOT USED
11		NOT USED
12		NOT USED
13		NOT USED
14 MCP MESSAGE	1	MESSAGE TYPE (SEE BELOW)
	2-4	NOT USED
15 PSUEDO DATA EVENT FOR ? MESSAGE	1	=0 IF THIS MESSAGE CONSISTS SOLELY OF A QUESTION MARK. #0 IF THIS IS A MESSAGE DIRECTED TO CANDE (E.G. STATUS), IN WHICH CASE WORDS 1-4 ARE THE SAME AS FOR A DATA EVENT
	2-4	SEE THE DESCRIPTION OF WORD 1
16 PSEUDO DATA EVENT	1-4	SAME AS FOR A DATA EVENT
* DISK I/O COMPLETE	1-4	NOT USED

* THIS EVENT MAY HAVE ANY EVENT NUMBER BUT USUALLY HAS THE NUMBER OF THE PROCEDURE WHICH INITIATED THE I/O IF IT WAS A READ OR A 13, WHICH IS DISCARDED, IF IT WAS A WRITE.

THE MCP MESSAGE TYPES ARE:

0	EOJ - SYNTAX ERROR	6	FILE REMOVED
1	EOJ - OK	7	FILE-NAME CHANGED
2	EOJ - DS-ED	8	SECURED FILE
3	DS MESSAGE	9	ZIP ERROR
4	BOJ	10	FILE NOT ON DISK
5	LIB. MAINT. IGNORED	11	NOT AN OBJECT CODE FILE

TABLE 2 FORMAT OF AN RCW

BIT 0 OFF MARKS THIS WORD AS AN OPERAND

BITS 1-2 CONTAIN A 2 TO MARK THE WORD AS AN RCW

BITS 3-10 CONTAIN THE VALUE OF SREG

BITS 11-17 CONTAIN THE VALUE OF PARAM

BITS 18-24 CONTAIN THE VALUE OF TEMP

BITS 25-32 CONTAIN THE VALUE OF BASE

BITS 33-39 CONTAIN THE PROCEDURE-S CASE NUMBER, USED TO TELL IT
 WHERE TO RESTART.

BITS 40-47 SPECIFY THE PROCEDURE

NOTE: IF THE PROCEDURE HAS NEVER HAD CONTROL, SREG, PARAM, AND TEMP
 ARE EQUAL TO ZERO.

TABLE 3

<u>IAM</u>	<u>WANT</u>	<u>WAITFOR</u>	<u>ACTION</u>
=0	=0	=0	GIVE CONTROL TO THE PROCEDURE SPECIFIED IN THE FIRST RCW.
=0	≠0	=0	CALL THE PROCEDURE SPECIFIED BY WANT
=0	≠0	≠0	STORE AN RCW FOR THE PROCEDURE SPECIFIED BY WANT, SAVE WAITFOR AND EXIT
≠0	≠0	=0	STORE AN RCW FOR IAM, THEN TRANSFER TO WANT
≠0	=0	≠0	STORE AN RCW FOR IAM, SAVE WAITFOR AND EXIT
≠0	≠0	≠0	STORE AN RCW FOR IAM, STORE AN RCW FOR WANT, SAVE WAITFOR AND EXIT

AN EXAMPLE OF THE COMPLEXITY INVOLVED IN PROCESSING A RELATIVELY SIMPLE COMMAND IS THE LIST COMMAND WHEN USED TO LIST THE WORK FILE. (SEE FIG. 1.) A LIST COMMAND IS RECEIVED AS A DATA EVENT, WHICH IS STORED UNTIL ALL OTHER EVENTS HAVE BEEN PROCESSED, THEN, AN RCW IS STORED FOR THE FINISHED ROUTINE. THIS WILL ENSURE THAT THINGS ARE PROPERLY WRAPPED UP AFTER ALL PROCESSING IS FINISHED. CONTROL IS THEN GIVEN TO INPUT WHICH IDENTIFIES THE COMMAND, STORES THE COMMAND AND ITS PARAMETERS IN THE STACK, AND THEN CALLS FOR PROCEDURE LISTIT BY SETTING WANT=33 AND IAM AND WAITFOR EQUAL TO ZERO.

LISTIT ANALYZES THE PARAMETERS AND THEN ARRANGES FOR CLOSEWORKTABLE TO BE CALLED, AND AFTER THAT, FOR DISPATCH TO BE CALLED. IT DOES THIS BY SETTING IAM=17, FOR DISPATCH, AND WANT=25, FOR CLOSEWORKTABLE. AN RCW IS STORED FOR DISPATCH AND CONTROL GOES TO CLOSEWORKTABLE, WHICH CLOSES THE WORK TABLE AS DESCRIBED LATER UNDER THE HEADING "WORK FILE." TO DO THIS, IT MAY HAVE TO READ DATA FROM DISK, WHICH IT DOES BY CALLING DISKREAD WITH IAM=25 SO THAT IT GETS CONTROL BACK.

DISKREAD USES 31 WORDS OF TEMPORARY STORAGE INTO WHICH IT READS A SEGMENT. IT THEN STORES IT IN ANOTHER ARRAY AND EXITS WITH IAM, WANT AND WAITFOR EQUAL TO ZERO. CONTROL IS THEREFORE GIVEN TO THE PROCEDURE SPECIFIED IN THE MOST RECENT RCW, WHICH IN THIS CASE IS CLOSEWORKTABLE. THE BASE IS RESET TO THE VALUE IN THE RCW, SO THAT THE STACK ALWAYS LOOKS THE SAME BEFORE AND AFTER A PROCEDURE TEMPORARILY LOSES CONTROL. CLOSEWORKTABLE FINISHES UP AND EXITS WITH IAM, WANT AND WAITFOR EQUAL TO ZERO. CONTROL IS GIVEN TO DISPATCH DUE TO THE RCW STORED WHEN LISTIT WAS EXITED, DISPATCH CAUSES THE LIST PROGRAM TO BE STARTED

ABOVE THE FENCE AND THEN EXITS WITH IAM EQUAL TO 0 AND WANT AND WAITFOR EQUAL TO 14, WHICH IS THE NUMBER OF AN MCP MESSAGE EVENT.

SINCE IT IS NECESSARY TO WAIT, AN RCW IS STORED FOR MCPMSG AND CANDE GOES ON TO PROCESS OTHER EVENTS. WHEN IT RECEIVES AN MCP-MESSAGE EVENT FOR THIS LINE, HANDLETHISLINE WILL TRANSFER TO MCPMSG ON THE BASIS OF THE RCW. THE FIRST MESSAGE WILL BE A BOJ, SO MCPMSG CONTINUES TO WAIT BY EXITING WITH IAM AND WAITFOR EQUAL TO 14. AFTER THE EOJ MESSAGE, IT EXITS WITH ALL THREE VARIABLES EQUAL TO 0. THIS WILL NOW CAUSE FINISHED TO BE CALLED WHICH WILL CAUSE AN EXIT.

THE WORK FILE

IF, WHEN PROCESSING A DATA EVENT, CANDE FINDS THAT THE MESSAGE BEGINS WITH A SEQUENCE NUMBER OR IS A FIX COMMAND, IT BUILDS A POINTER TO THE DISK TANK ADDRESS OF THE MESSAGE. (THE ADDRESS IS RELATIVE TO THE BASE ADDRESS OF A CHUNK, WHICH IS PASSED TO CANDE IN A NEW-DISK-CHUNK EVENT.) THE FORMAT OF THE POINTER IS:

1:2 FLAG FOR ENTRY TYPE. THIS FIELD
=1 IF THE MESSAGE IS OF THE FORM <SEQUENCE-NUMBER>+,
=2 IF IT IS A NORMAL ENTRY TO THE WORK FILE,
=3 IF IT IS A FIX COMMAND,
4:12 DISKLOC, IN WHICH
4:4 IS THE CHUNK NUMBER
8:8 IS THE RELATIVE SEGMENT
16:5 DKOFFSET, SPECIFIES THE WORD IN THE SEGMENT AT WHICH THE
MESSAGE STARTS,
21:17 SEQUENCE NUMBER IN BINARY.

THE POINTERS ARE ACCUMULATED IN CORE UNTIL THERE ARE 10 OF THEM, AT WHICH TIME THEY ARE WRITTEN INTO THE FILE,

1P<LOGICAL-LINE-NUMBER>/<USER-CODE>.

THIS FILE, WHICH IS BLOCKED (10,30), IS OPENED FOR EACH USER WHEN HE LOGS IN. NOTE THAT SINCE NO MORE THAN 10 POINTERS ARE IN CORE AT ONE TIME, A USER CAN LOSE AT MOST 10 RECORDS DURING A HALT/LOAD.

WHEN A COMMAND AFFECTING THE WORK FILE IS ISSUED, THE WORK FILE IS

SORTED AND UPDATED. FIX COMMANDS ARE PROCESSED AND, IN THE CASE OF RECORDS WITH DUPLICATED SEQUENCE NUMBERS, ONLY THE MOST RECENTLY ENTERED IS RETAINED. SEQUENCE NUMBERS AND LEADING BLANKS ARE STRIPPED FROM THE FRONT OF THE RECORD. IF NECESSARY, PRECEDING ZEROES ARE ADDED TO SEQUENCE NUMBERS TO MAKE THEM EIGHT DIGITS LONG, AND THEN THEY ARE STORED IN CHARACTER POSITIONS 73-80. ENOUGH BLANKS ARE ADDED TO THE END OF THE MESSAGE TO MAKE IT 72 CHARACTERS LONG. THE RESULTING RECORD CONSISTS OF 10 WORDS WHERE

WORDS 0-8 CONTAIN THE ALPHANUMERIC MESSAGE

WORD 9 CONTAINS THE ALPHANUMERIC SEQUENCE NUMBER

THE SORTED, UPDATED RECORDS ARE STORED IN FILE

1S<LOGICAL-LINE-NUMBER>/<USER-CODE>

POINTERS TO THE NEW SOURCE FILE ARE STORED IN THE FILE

1T<LOGICAL-LINE-NUMBER>/<USER-CODE>

IF MORE DATA IS ENTERED AFTER THE WORK FILE HAS BEEN SORTED AND UPDATED, POINTERS CAN AGAIN BE STORED IN 1P<LOGICAL-LINE-NUMBER>/<USER-CODE>. WHEN THE WORK FILE IS AGAIN UPDATED, THESE POINTERS CAN BE SORTED AND THEN MERGED WITH THOSE IN 1T<LOGICAL-LINE-NUMBER>/<USER-CODE>. FROM THE RESULT, A NEW SOURCE FILE AND A NEW FILE OF POINTERS CAN BE CONSTRUCTED IN 1S<LOGICAL-LINE-NUMBER>/<USER-CODE> AND 1T<LOGICAL-LINE-NUMBER>/<USER-CODE> RESPECTIVELY.

THUS, THE WORK FILE CONSISTS OF FOUR PARTS:

- 1) MESSAGES IN THE DISK INPUT TANKS.
- 1) THE FILE 1P<LOGICAL-LINE-NUMBER>/<USER-CODE> WHICH CONTAINS POINTERS TO THE MESSAGES IN THE TANKS.
- 3) THE SORTED, UPDATED SOURCE FILE.
- 4) THE FILE 1T<LOGICAL-LINE-NUMBER>/<USER-CODE> WHICH CONTAINS POINTERS TO THE SOURCE FILE.

ORDINARILY, THE SOURCE FILE IS 1S<LOGICAL-LINE-NUMBER>/<USER-CODE> BUT AFTER A LOAD COMMAND THE FILE SPECIFIED IN THE COMMAND IS USED AS THE SOURCE AND THE POINTERS POINT TO IT.

THE WORK FILE IS SORTED AND UPDATED BY PROGRAMS WHICH RUN ABOVE THE FENCE, THIS FREES CANDE TO SERVICE OTHER USERS AND ALSO ENSURES THAT ANYTIME A USER REQUIRES A SIGNIFICANT AMOUNT OF PROCESSING, HE SHARES TIME AND CORE WITH OTHER USERS THROUGH THE SWAPPING MECHANISM, THE VERBS WHICH CAUSE THESE PROGRAMS TO BE RUN ABOVE THE FENCE ARE LOAD, LIST, COPY, RESEQ, APPEND, MERGE, DELETE AND, IF THE WORK FILE IS NOT SORTED WHEN THE COMMAND IS ISSUED, SAVE.

TO FREE CANDE EVEN FURTHER FROM THE MENIAL CHORES OF FILE HANDLING, ALL OF ITS FILE OPERATIONS ARE DONE BY THE MCP. USING COMMUNICATES, CANDE TELLS THE MCP TO OPEN AND CLOSE FILES, TO PERFORM READS AND WRITES AND TO CHECK THE EXISTENCE AND SECURITY STATUS OF FILES. IT PASSES SUCH NECESSARY INFORMATION AS FILE NAMES, DISK AND CORE ADDRESSES AND THE NUMBER OF THE PROCEDURE REQUIRING THE FILE OPERATION. WHEN THE OPERATION IS FINISHED, THE MCP QUEUES AN EVENT WHICH IT IDENTIFIES BY THE PROCEDURE NUMBER WHICH WAS PASSED TO IT IN THE COMMUNICATE. THIS EVENT TELLS CANDE THAT THAT PROCEDURE CAN CONTINUE ITS PROCESSING. IN THIS WAY, CANDE CAN PROCESS OTHER EVENTS WHILE I/O OPERATIONS ARE BEING PERFORMED FOR IT.

THE NEGATIVE COMMUNICATES

THE FOLLOWING IS A LIST OF THE NEGATIVE COMMUNICATES, WHICH HAVE BEEN ADDED TO THE MCP FOR TIME SHARING. OF THESE, ONLY -11 AND -13 ARE NOT USED BY CANDE. FOR THOSE COMMUNICATES WHICH SIMPLY PROVIDE ACCESS TO MCP PROCEDURES, THE PROCEDURE IDENTIFIER IS LISTED.

<u>COMMUNICATE</u> -----	<u>USE</u> ---
-1	CANDE-S DISK I/O
-2	REQUEST FOR NEXT EVENT
-3	GETESPDISK
-4	FORGETESPDISK
-5	DATA EVENT REQUEST
-6	GETUSERDISK
-7	FORGETUSERDISK
-8	DISKWAIT
-9	CONTROL CARDS
-10	USER CODES
-11	TWXOUT FOR USER PROGRAMS
-12	TWXOUT FOR CANDE
-13	USER PROGRAMS INPUT REQUEST
-14	AUTOMATIC SEQUENCING
-15	FILE CREATION, SECURITY MAINTENANCE, LOG-ONS, LOG-OFFS, LIBRARY MAINTENANCE AND PAPER TAPE
-16	DISCONNECTS
-17	CHARGE CODES

WHEN A PROCEDURE IN CANDE NEEDS TO PERFORM A COMMUNICATE, IT CALLS ANOTHER PROCEDURE CONSISTING ONLY OF THE COMMUNICATE STATEMENT AND PASSES IT A LIST OF PARAMETERS. THUS, WHEN SHORTCOMMUNICATES GETS CONTROL TO PROCESS THE COMMUNICATE, THE FOLLOWING ITEMS ARE IN THE STACK:

MARK STACK CONTROL WORD	FROM THE CALL ON THE COMMUNICATE
PARAMETERS	PROCEDURE
RETURN CONTROL WORD	
INTERRUPT RETURN CONTROL WORD	FROM THE INTERRUPT GENERATED BY
INTERRUPT CONTROL WORD	THE COM OPERATOR
MARK STACK CONTROL WORD	FROM THE CALL ON SHORTCOMMUNICATES
RETURN CONTROL WORD	

SHORTCOMMUNICATES USES F-RELATIVE ADDRESSING TO ACCESS THE PARAMETERS. IT PROCESSES THE -10, -14, AND -16 COMMUNICATES ITSELF AND CALLS OTHER PROCEDURES FOR THE OTHER COMMUNICATES.

THE ACTIONS PERFORMED BY THE -10, -14, -16 AND -17 COMMUNICATES ARE RELATIVELY SIMPLE. A -10 COMMUNICATE JUST STORES THE USER CODE IN USERCODE. A -14 IS USED TO PASS THE INFORMATION FOR AUTOMATIC SEQUENCING TO SEQARRAY. A -16 RESULTS IN CANDETHRU BEING SET TO ONE, THE VALUE OF WHICH IS EXPLAINED IN THE DESCRIPTION OF DISCONNECTS. A -17 SIMPLY SPOUTS A CHANGE OF CHARGE CODE MESSAGE, WHICH IS ENTERED IN THE LOG. THE -5 AND -13 COMMUNICATES ARE HANDLED BY COMM5 AND COMM13 RESPECTIVELY WHICH ARE EXPLAINED IN THE DISCUSSION OF LINE MAINTENANCE.

A -1 COMMUNICATE IS HANDLED BY COMM1, WHICH DOES DISK READS AND WRITES FOR CANDE. IN ADDITION TO THE CORE AND DISK ADDRESSES, THE SIZE AND

A READ/WRITE FLAG, CANDE ALSO PASSES COMM1 THE LOGICAL LINE NUMBER AND A "REASON", I.E. THE NUMBER OF THE CANDE PROCEDURE DOING THE I/O. THIS ALLOWS THE MCP TO PASS CANDE AN I/O COMPLETE EVENT WITH THAT NUMBER SO THAT THE PROCEDURE MAY PROCEED.

COMM1 GETS AN AREA FOR THE EVENT, PUTS THE REASON AND LINE NUMBER IN WORD 0, ARRANGES TO HAVE THE I/O RESULT DESCRIPTOR IN WORD 1, CALLS DISKIO, SNEAKILY FLAGGING IT AS A CANDE I/O, AND RETURNS. WHEN THE I/O IS COMPLETE, THE EQUALLY DEVIUS IOFINISH PROCEDURE RECOGNIZES THIS AS A CANDE I/O AND QUEUES UP THE EVENT.

A -2 COMMUNICATE, WHICH IS CANDES WAY OF ASKING FOR ANOTHER EVENT, IS HANDLED BY COMM2. EVENTS ARE KEPT IN SEGMENTED SAVE AREAS AND ARE LINKED BY THE FIRST WORD OF THE AREA, WHICH HAS THE FORMAT.

1:1 OCCUPIED BIT
2:2 SIZE CODE
18:7 EVENT NUMBER, I.E. THE REASON
25:8 LOGICAL LINE NUMBER
33:15 ADDRESS OF NEXT EVENT IN THE QUEUE

THE HEAD AND TAIL OF THE EVENT QUEUE ARE KEPT IN EVENT.

COMM2 FORGETS THE AREA FOR THE PREVIOUS EVENT AND THEN CHECKS THE EVENT QUEUE. IF THERE ARE NO EVENTS, IT EITHER RETURNS OR SLEEPS, DEPENDING ON WHETHER OR NOT CANDE INDICATED THAT IT NEEDED TO WAIT FOR AN EVENT. IF THERE IS AN EVENT, OR AFTER WAITING, THE NEXT EVENT AND THE CURRENT TIME ARE GIVEN TO CANDE.

CANDE USES A -9 COMMUNICATE TO PASS THE MCP CONTROL CARDS, WHICH IT USES FOR SUCH THINGS AS INITIATING JOBS ABOVE THE FENCE AND LIBRARY MAINTENANCE. COMM9 SIMPLY FORKS CONTROLCARD AND GIVES IT THE INFORMATION CANDE ALSO PASSES THE REASON WHICH IT WANTS USED ON THE MCP MESSAGE EVENTS WHICH RESULT FROM THE PROCESSING OF THE CONTROL CARD. SINCE THE REASON IS ALWAYS THE SAME, THERE IS NO NEED TO GO INTO THE DETAILS OF HOW THIS IS DONE, BESIDES, IT WOULD SPOIL THE READERS FUN.

A -15 COMMUNICATE IS USED BY CANDE FOR NINE DIFFERENT THINGS:

- 0) TO PERFORM A DIRECTORY SEARCH FOR A FILE AND A SECURITY CHECK ON THE USER WHO WISHES TO ACCESS IT.
- 1) TO CREATE A FILE.
- 2) TO RECORD LOG-ONS AND LOG-OFFS.
- 3) TO REMOVE A FILE.
- 4) TO REPLACE A CURRENT FILE BY A NEW FILE.
- 5) TO CHANGE THE NAME OF A FILE.
- 6) TO START A JOB.
- 7) TO CHANGE THE SECURITY STATUS OF A JOB.
- 8) TO START A PAPER TAPE.

THE INFORMATION PASSED TO COMM15 INCLUDES A FLAG SPECIFYING WHICH TYPE OF COMMUNICATE THIS IS, THE USER CODE, FILE-NAMES, THE LOGICAL LINE NUMBER, AND A REASON TO BE USED IN THE RETURN EVENT. THIS INFORMATION IS STORED IN A SEGMENTED SAVE AREA WHICH IS ADDED TO THE INDIAN QUEUE. THE FIRST WORD OF THIS AREA CONTAINS A LINK TO THE NEXT ENTRY IN THE QUEUE IN 33:15, AND THE REASON AND LINE NUMBER IN 18:15. THUS, THE SAME AREA CAN BE USED FOR THE ANSWERING EVENT. THE HEAD AND

TAIL OF THE INDIAN QUEUE ARE KEPT IN INDIAN.

THE QUEUE IS PROCESSED BY INDIANBOY, WHICH, IF IT IS NOT ALREADY RUNNING OR IN THE FORK QUEUE, IS ADDED TO THE FORK QUEUE WHENEVER SOMETHING IS ADDED TO THE INDIAN QUEUE. FOR AN ENTRY OF THE FIRST TYPE, IE TYPE 0, HE JUST CALLS DIRECTORYSEARCH AND, IF THE FILE IS THERE, SECURITYCHECK. THE RETURN EVENT HAS THE FOLLOWING INFORMATION:

WORD 1	PROTECT CODE, SAME AS ALGOL SEARCH STATEMENT.
WORD 2	1:1 INTERLOCK
	18:15 HEADER ADDRESS
	36:6 FILE TYPE
	42:6 OPEN COUNT.
WORD 3	EOF COUNT.
WORD 4	DISK ADDRESS OF FIRST RECORD.

IF THE SPECIFIED FILE DOES NOT EXIST, WORD 1 CONTAINS A -1 AND THE REMAINING WORDS ARE UNDEFINED.

FOR THE SECOND TYPE, INDIANBOY SETS UP THE FILE HEADER AND DOES A DIRECTORYSEARCH TO SEE IF THERE IS ALREADY A FILE BY THAT NAME, IN WHICH CASE HE REMOVES IT. HE THEN CALLS ENTERUSERFILE TO ENTER THE FILE IN THE DIRECTORY AND PASSES BACK TO CANDE AN EVENT CONTAINING THE DISK ADDRESS OF THE FILE IN WORD 1 AND THE ADDRESS OF THE HEADER IN WORD 2.

FOR THE THIRD TYPE, INDIANBUY SIMPLY MAKES THE NECESSARY CHANGES IN SYSTEM/DISK AND SPOUTS THE APPROPRIATE MESSAGE.

CANDE USES TYPES 3, 4 AND 5 FOR LIBRARY MAINTENANCE. TO REMOVE A FILE (TYPE 3) INDIANBOY CALLS DIRECTORYSEARCH. TO REPLACE A FILE (TYPE 4) HE USES DIRECTORYSEARCH TO REMOVE THE OLD FILE (IF PRESENT) AND THEN TO CHANGE THE NAME OF THE NEW FILE TO THAT OF THE OLD FILE. THIS IS USED FOR SUCH THINGS AS SAVE COMMANDS, WHICH REQUIRE THAT THE CURRENT DISK FILE BE REPLACED BY THE CURRENT WORK FILE. A CHANGE OF NAME (TYPE 5) IS AGAIN DONE USING DIRECTORYSEARCH. AFTER COMPLETING ONE OF THESE OPERATIONS, INDIANBOY QUEUES AN EVENT USING THE MCP MESSAGE CODES 5, 6 AND 7 (SEE TABLE 1) TO INDICATE THE RESULTS.

TO START A JOB FOR CANDE (TYPE 6), INDIANBOY CHECKS TO SEE IF THE FILE IS THERE, IF IT IS OBJECT CODE, AND IF THE USER CAN ACCESS IT UNDER THE FILE SECURITY SYSTEM. IF THE JOB FAILS ANY OF THESE TESTS, AN APPROPRIATE MCP MESSAGE EVENT IS QUEUED, OTHERWISE THE JOB IS ADDED TO THE SCHEDULE AND SELECTRUN IS FORKED.

FOR TYPE 7, INDIANBOY IS PASSED THE WORDS TO BE USED IN WORDS 2, 5 AND 6 OF THE FILE HEADER. IF THE FILE IS PRESENT, HE SIMPLY MAKES THE INDICATED CHANGES IN THE HEADER. IF THE REQUESTOR IS CANDE HE QUEUES UP EITHER AN IGNORED EVENT (MCP MESSAGE #5) OR A CHANGED EVENT (#6) WHEN HE IS FINISHED.

FOR TYPE 8, INDIANBOY INITIALIZES SEQARRAY AND THEN CREATES THE FILE FOR TANKING AS HE WOULD FOR A TYPE 1 REQUEST. HE ALSO GETS 65 WORDS FOR THE CORE TANKS, INITIALIZES THEM AND SENDS THE OK MESSAGE AND AN X-ON.

INDIANBOY CONTINUES TO PROCESS THE INDIAN QUEUE UNTIL IT IS EXHAUSTED, AND THEN CALLS KILL.

APPENDIX A

DESCREPANCIES BETWEEN THE INITIAL RELEASE AND THIS MANUAL

THE FOLLOWING ITEMS WHICH ARE DESCRIBED IN THIS MANUAL ARE EITHER NOT IN THE INITIAL RELEASE OR ARE IMPLEMENTED DIFFERENTLY THAN DESCRIBED.

1. ALTERNATE SPO. IN THE INITIAL SYSTEM, THE ALTERNATE SPO CAN BE USED ONLY FOR OUTPUT. THE REAL SPO MUST THEREFORE BE USED FOR ALL KEYBOARD INPUT MESSAGES, EVEN IF IT HAS BEEN USED. AN ATTEMPT TO TYPE IN ANYTHING FROM AN ALTERNATE SPO WILL CAUSE, AT BEST, CONFUSION AND, AT WORST, A HUNG SYSTEM.
2. USER/CANDE. INITIALLY ONLY THE PASSWORD AND TIME OPTION CARDS WILL BE IMPLEMENTED. THE OTHER OPTION CARDS WILL BE IGNORED IF INPUT.
3. FORTRAN. LABELS FOR REMOTE FREE FIELD FORMAT MUST BE IN COLUMNS 1 THROUGH 5 FOLLOWING THE SEQUENCE NUMBER. SINCE THE LABEL MUST BE SEPARATED FROM A SEQUENCE NUMBER LESS THAN 8 CHARACTERS LONG, LABELS CAN BE AT MOST 4 CHARACTERS LONG WHEN SEQUENCE NUMBERS SHORTER THAN 8 CHARACTERS ARE USED.
4. MESSAGES. THE IN AND OT MESSAGES CANNOT BE USED AT ALL.