Burroughs

B 5700

TIME SHARING SYSTEM

REFERENCE MANUAL

PREFACE

THIS DOCUMENT DESCRIBES THE DIFFERENCES BETWEEN THE TIME SHARING
SYSTEM (TSSMCP) AND THE STANDARD SYSTEM (DCMCP). THE READER IS
ASSUMED TO BE FAMILIAR WITH OPERATION OF THE B 5500 DCMCP AS
DESCRIBED IN THE OPERATIONS MANUAL. PART II, PROGRAMMING
INFORMATION, CONTAINS DETAILED DESCRIPTIONS OF THE PROCEDURES AND
PROGRAMS IN THE TSSMCP WHICH ARE NOT A PART OF THE DCMCP. SINCE IT
IS INTENDED PRIMARILY FOR THE SYSTEMS PROGRAMMER, THE READER IS
EXPECTED TO BE FAMILIAR WITH THE CODING OF THE DCMCP.

TABLE OF CONTENTS

PART I

OPERATION OF THE TIME SHARING SYSTEM

INTRODUCTION

THE PRIMARY OBJECTIVE OF THE STANDARD DATACOM MCP IS TO MAXIMIZE THE THROUGHPUT OF THE COMPUTER. THIS IS ACCOMPLISHED BY DISTRIBUTING RESOURCES AMONG THE JOBS RUNNING, AND BY RUNNING ADDITIONAL JOBS ONLY WHEN THERE ARE SUFFICIENT AVAILABLE RESOURCES. IN THIS WAY, THE COMPUTER CAN ACCOMPLISH THE MOST IN THE LEAST AMOUNT OF TIME.

THE TIME SHARING MCP, HOWEVER, WAS DESIGNED TO MAXIMIZE THE THROUGHPUT OF A LARGE NUMBER OF INTERACTING, REMOTELY LOCATED PEOPLE. IT DOES THIS BY ALLOWING EACH TO ENTER DATA AND COMPILE AND EXECUTE JOBS AT THE SAME TIME. SINCE THE B5700 IS NOT LARGE ENOUGH TO HANDLE THEM SIMULTANEOUSLY (E.G. TO HAVE 25 JOBS IN CORE AT ONCE), IT SHARES ITS TIME AND RESOURCES AMONG THE USERS. THE USERS ARE VIRTUALLY UNAWARE OF SUCH A DISTRIBUTION BECAUSE THE COMPUTER OPERATES AT A MUCH FASTER RATE THAN THEY. IT IS ABLE TO DO A LITTLE BIT FOR EACH USER IN TURN AND STILL GET BACK TO THE FIRST USER BEFORE HE CAN TELL THAT IT LEFT HIM. THUS, BY SACRIFICING SOME OF ITS OWN THROUGHPUT TO THE OVERHEAD OF SWITCHING ITSELF BETWEEN MANY USERS, THE COMPUTER MAKES IT POSSIBLE FOR THOSE USERS TO OPERATE MORE EFFICIENTLY THEMSELVES.

IN ORDER TO SERVICE THE USERS, THERE ARE TWO MAIN DIFFERENCES BETWEEN THE TIME SHARING MCP AND THE BATCH MCP, AND MOST OF THE VARIANCES OF OPERATION ARISE FROM THESE TWO MAJOR DIFFERENCES. THE FIRST OF THESE IS THE OPERATION OF DATACOM. THE TSSMCP AUTOMATICALLY TAKES CARE OF ALL I/O-S TO DATACOM, JUST AS IT DOES FOR THE OTHER PERIPHERAL UNITS. IN ADDITION, THERE IS A PROGRAM, CANDE/TSHARER, WHICH SERVES AS THE HANDLER FOR THE REMOTE USERS. IT TAKES CARE OF LOGGING THEM ON AND OFF, BUILDING THEIR FILES, AND RUNNING THEIR JOBS, AS DESCRIBED IN THE TERMINAL USERS GUIDE. CANDE IS INSEPARABLY INTERWOVEN INTO THE TSSMCP AND MUST BE RUNNING TO ALLOW REMOTE OPERATION.

THE SECOND MAJOR DIFFERENCE IS THE MANNER IN WHICH CORE IS HANDLED. FOR TIME SHARING, CORE IS DIVIDED INTO TWO PARTS. THE LOWER PART IS RESERVED FOR SYSTEM FUNCTIONS. THE MCP AND CANDE RESIDE THERE, AND PRINTER BACK UP AND LOAD CONTROL RUN THERE. THE UPPER PART IS RESERVED FOR OBJECT JOBS. ALL JOBS USE THIS UPPER PART EXCLUSIVELY AND ARE NEVER ALLOWED TO USE SPACE IN THE LOWER PART. THE DIVISION BETWEEN THE PARTS OF CORE IS CALLED THE FENCE. IT CAN BE SET IN THE COLD AND COOL START DECKS AND MAY BE CHANGED BY THE "MF" MESSAGE.

THE REASON FOR THE FENCE IS THAT JOBS ABOVE THE FENCE ARE SWAPPED. THAT IS, EACH JOB IS ASSIGNED A SUB-AREA IN WHICH IT RUNS EXCLUSIVELY. AFTER RUNNING A FEW SECONDS (OR WHEN IT STOPS TO WAIT FOR A FILE OR INPUT OR SOME SUCH), ITS CORE IS WRITTEN TO DISK, AND ANOTHER JOB (OR JOBS) IS BROUGHT IN TO RUN. IN TURN, THAT JOB WILL BE SWAPPED OUT AND FURTHER JOBS BROUGHT IN. EVENTUALLY, AFTER EACH JOB THAT NEEDS A TURN HAS HAD ONE, THE FIRST JOB IS BROUGHT IN AGAIN FOR A SECOND TURN. IN THIS WAY, THE JOBS SHARE THE AVAILABLE TIME

AND CORE SPACE, ALLOWING MANY JOBS TO BE IN THE MIX AT ONE TIME, AND
MANY USERS TO BE SERVICED SIMULTANEOUSLY. IN MANY WAYS, SWAPPING IS
LIKE THE PAGING USED BY COMPUTERS WHICH HAVE NOT DISCOVERED THE
BENEFITS OF OVERLAY, EXCEPT THAT OVERLAY STILL OCCURS WITHIN EACH
JOBS AREA, ALLOWING IT TO MAKE MUCH MORE EFFICIENT USE OF THAT AREA.

MUCH OF THE DESCRIPTION THAT FOLLOWS IN THIS PART OF THE MANUAL
DETAILS THE CHANGES IN OPERATION NECESSITATED BY SWAPPING AND BY THE
PRESENCE OF CANDE.

OPERATING THE TSSMCP

THE OPERATION OF THE TSSMCP IS QUITE SIMILAR TO THAT OF THE DCMCP. FOR INSTANCE, MOST OF THE KEYBOARD INPUT MESSAGES ARE THE SAME WITH SEVERAL ADDITIONAL ONES . THUS, ANYONE FAMILIAR WITH THE OPERATION OF THE DCMCP SHOULD HAVE NO TROUBLE OPERATING THE TSSMCP.

IN ORDER TO USE REMOTE TERMINALS, THE DTCU MUST BE IN REMOTE AND THE FILE SYSTEM/DISK, CONTAINING A DESCRIPTION OF THE DATA COMMUNICATIONS HARDWARE CONFIGURATION OF THE SYSTEM, 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 ON THE SPO. THE FILE SYSTEM/DISK CAN BE CREATED BY RUNNING THE PROGRAM SYSDISK/MAKER.

AS ON THE DCMCP, THE INTRINSICS FILE SHOULD ALSO BE ON DISK WHEN A HALT/LOAD IS DONE. TO COMPILE THE INTRINSICS FOR TIME SHARING, THE FOLLOWING $ CARD SHOULD BE USED:

    $ SET TIMESHARING = TRUE

IF THIS OPTION IS SET FALSE, THE RESULTING INTRINSICS WILL BE THOSE FOR THE DCMCP.

AFTER THE HALT/LOAD HAS BEEN PERFORMED, IN ORDER TO DO ANY REMOTE 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. IT IS RECOMMENDED THAT CANDE BE COMPILED WITH A CORE ESTIMATE OF 4000 AND THAT CORE AND STACK CARDS BE USED WITH SOME OF THE CANDE PROGRAMS. THE CANDE PROGRAMS AND RECOMMENDED CORE AND STACK VALUES ARE:

SOURCE	OBJECT	CORE	STACK
SYMBOL/APPEND	APPEND/CANDE	4000	256
SYMBOL/COPY	COPY/CANDE	4000	256
SYMBOL/DELETE	DELETE/CANDE	2000	256
SYMBOL/FIND	FIND/DISK	3000	256
SYMBOL/GUARD	GUARD/DISK	3000	256
SYMBOL/HARD	HARD/CANDE		256
SYMBOL/LFILES	LFILES/CANDE	3000	256
SYMBOL/LIST	LIST/CANDE	4000	256
SYMBOL/LOAD	LOAD/CANDE	3000	256
SYMBOL/MERGE	MERGE/CANDE	4000	256

SYMBOL/PAPER	PAPER/CANDE	4000	256
SYMBOL/PUNCH	PUNCH/CANDE	2000	256
SYMBOL/QUIKLST	QUIKLST/CANDE	2000	256
SYMBOL/REPLACE	REPLACE/CANDE	3000	256
SYMBOL/RESEQ	RESEQ/CANDE	3000	256
SYMBOL/RESEQB	RESEQB/CANDE	4000	256
SYMBOL/SCHEDUL	SCHEDUL/CANDE	2000	256
SYMBOL/CANDE	CANDE/TSHARER	4000	

IN ADDITION, THE FILES

MESAGE/CANDE, CONTAINING THE ERROR MESSAGES USED BY CANDE,

AND

USERS/CANDE, CONTAINING INFORMATION IDENTIFYING AUTHORIZED USERS
OF THE SYSTEM,

MUST ALSO BE PRESENT ON DISK. ALL CANDE PROGRAMS ARE COMPILED IN
TSPOL. ALL CANDE OBJECT PROGRAMS MUST HAVE THE NAMES GIVEN ABOVE
SINCE THE SYSTEM REFERENCES THE PROGRAMS BY THESE NAMES.

IF THE FILE

NEWS/CANDE

IS PRESENT ON DISK, THEN CANDE WILL USE THE FIRST 72 CHARACTERS OF
THE FIRST RECORD OF THIS FILE AS A MESSAGE OF THE DAY, PRINTING IT
ON THE TERMINAL OF ANY USER WHO HAS JUST LOGGED-IN. IF THIS FILE IS
NOT ON DISK, THEN NO MESSAGE OF THE DAY IS PRINTED AT LOG-IN TIME.

WHEN A "CE" IS KEYED-IN TO START CANDE/TSHARER AND EITHER OF THE
FILES MESAGE/CANDE OR USERS/CANDE IS NOT ON DISK, THEN ONE OF THE
FOLLOWING MESSAGES IS DISPLAYED:

#MESAGE FILE NOT ON DISK
#USERS FILE NOT ON DISK

AND CANDE/TSHARER IS ES-ED.

ALSO, IF A "CE" IS KEYED-IN AND THERE IS NOT SUFFICIENT DISK SPACE
FOR TANKING TERMINAL INPUT/OUTPUT, THE FOLLOWING MESSAGE IS
DISPLAYED:

#NO USER DISK FOR DATACOM TANKS

AND CANDE IS ES-ED.

IF, AFTER A HALT/LOAD, THE FILE TANK/DISK HAS BEEN REMOVED FOR ANY
REASON , AND A "CE" IS THEN KEYED-IN, THE FOLLOWING MESSAGE IS
DISPLAYED:

#TANK FILE NOT ON DISK

AND CANDE IS ES-ED. THE SYSTEM MUST THEN BE HALT/LOADED.

SYMBOL/USER, SYMBOL/HARD, AND SYMBOL/SYSDSK COMPILE INTO USER/CANDE, HARD/CANDE, AND SYSDISK/MAKER, RESPECTIVELY, AND ARE DESCRIBED ELSEWHERE.

THE CORE FACTOR AFFECTS THE USE OF CORE BELOW THE FENCE THE SAME AS ON THE BATCH MCP. ABOVE THE FENCE, IT AFFECTS THE AMOUNT OF OVERLAY ALLOWED BEFORE THE AREA ASSIGNED TO A JOB IS EXPANDED.

RUNNING BATCH JOBS

JOBS MAY BE ENTERED AND RUN VIA THE CARD READER IN THE USUAL WAY. WHEN AN EXECUTE CARD OR A COMPILE CARD IS USED, JOBS ARE RUN ABOVE THE FENCE AND ARE THEREFORE SUBJECT TO SWAPPING WITH OTHER JOBS. JOBS CAN BE RUN BELOW THE FENCE BY USING A RUN CARD. HOWEVER, RUNNING JOBS BELOW THE FENCE CAUSES A PLETHORA OF NO MEM-S AND SERIOUSLY HAMPERS THE TSSMCP AND CANDE.

POSITIONING THE FENCE

THE MCP WILL AUTOMATICALLY SET THE FENCE AT 16000 UNLESS INSTRUCTED TO DO OTHERWISE BY A "MF" MESSAGE OR BY A FENCE CARD IN THE COLD OR COOL START DECK. THIS VALUE HAS BEEN FOUND TO BE BEST FOR MOST INSTALLATIONS, ALTHOUGH, DEPENDING ON THE NUMBER OF USERS AND THE TYPE OF WORK THEY DO, IT MAY BE ADVISABLE FOR SOME INSTALLATIONS TO CHANGE THIS VALUE. IN GENERAL, IT IS BEST TO PLACE THE FENCE AS LOW AS POSSIBLE WITHOUT CAUSING NO-MEMS. THIS ALLOWS MAXIMUM ROOM FOR JOBS ABOVE THE FENCE.

NOTE: PLACING THE FENCE BELOW 10000 MAY CAUSE NO-MEMS DURING HALT/ LOAD, NECESSITATING A COOL START TO MOVE THE FENCE BACK UP TO A BETTER LOCATION.

TSSMCP MODULARITY

AS IN THE DCMCP, MODULARITY IN THE TSSMCP IS ACHIEVED BY SETTING
CERTAIN OPTIONS WHEN THE TSSMCP IS COMPILED. THE OPTIONS AVAILABLE
IN THE TSSMCP ARE:

    CHECKLINK
    B6500LOAD
    DUMP
    DEBUGGING
    DFX
    SAVERESULTS
    SHAREDISK
    STATISTICS
    TWXONLY
    DCP

B6500LOAD, CHECKLINK, DEBUGGING, DUMP, DFX, SHAREDISK, AND
STATISTICS ARE ANALOGOUS TO THEIR COUNTERPARTS IN THE DCMCP.

IF SAVERESULTS IS SET TRUE, THE RESULTHOLDER ARRAY OF THE TSSMCP
WILL BE INCLUDED. THIS ARRAY STORES IMPORTANT DATACOM INFORMATION
CYCLICALLY FOR HARDWARE AND/OR SOFTWARE DEBUGGING PURPOSES ONLY.
WHEN SET FALSE, THIS ARRAY IS OMITTED. SEE APPENDIX A FOR
INFORMATION ON THE CONTENTS OF THE RESULTHOLDER ARRAY.

IF TWXONLY IS SET TRUE, THE ONLY TERMINALS HANDLED BY THE TSSMCP ARE
MODEL 33 AND MODEL 35 TELETYPES. WHEN SET FALSE, THE SYSTEM WILL
HANDLE B9352, B9353, AND TC500 TERMINALS AS WELL AS TELETYPES.

IF DCP IS SET TRUE, THE RESULTING TSSMCP WILL INTERFACE WITH THE
DATA COMMUNICATIONS PROCESSOR (DCP). IF DCP IS SET FALSE, THE
TSSMCP WILL USE THE B487 DATACOMM SUBSYSTEM.

SYSDISK/MAKER

THE PROGRAM SYSDISK/MAKER IS USED TO CREATE THE FILE SYSTEM/DISK. THE SYSTEM/DISK FILE DESCRIBES THE DATACOMM HARDWARE CONFIGURATION OF A SYSTEM AND MUST BE CREATED BY THE INSTALLATION.

THE INPUT FOR SYSDISK/MAKER CONSISTS OF TWO KINDS OF CARDS: "STATION" CARDS AND "LINE" CARDS. LINE CARDS DESCRIBE ADAPTERS, THE LINE DISCIPLINE, ETC., AND STATION CARDS DESCRIBE THE REMOTE TERMINALS ATTACHED TO THE LINE.

THE FORMAT OF A LINE CARD IS:

LINE,<TERMINAL UNIT NO>,<BUFFER NUMBER>,<BUFFER SIZE>, <PING-PING FLAG>,<ADAPTER TYPE>,<LINE DISCIPLINE>, <DIRECT CONNECT FLAG>,

NOTE THAT EACH ITEM IS FOLLOWED BY A COMMA, INCLUDING THE LAST. THE WORD "LINE" IDENTIFIES THIS AS A LINE CARD.

<TERMINAL UNIT> AND <BUFFER NUMBER>	SELF EXPLANATORY.
<BUFFER SIZE>	28,56, OR 112.
<PING-PING FLAG>	1 IF IT IS A PING-PING BUFFER, 0 OTHERWISE.
<ADAPTER TYPE>	0 FOR A 980 1 FOR A 992
<LINE DISCIPLINE>	0 FOR TELETYPE 1 FOR CONTENTION 2 FOR MULTIPOINT 7 FOR SCHEDULE
<DIRECT CONNECT FLAG>	1 FOR DIRECT CONNECT LINES 0 FOR DIAL-UP.

THE FORMAT OF A STATION CARD IS:

STA,<TYPE>,<LINE LENGTH>,<PAGE SIZE>,<NAK MAX>,"<ADDRESS 1>", "<ADDRESS 2>",<NOT EQUAL FLAG 1>,<NOT EQUAL FLAG 2>,

NOTE AGAIN THAT EVERY ITEM IS FOLLOWED BY A COMMA. THE WORD "STA" IDENTIFIES THIS AS A STATION CARD.

<TYPE>	0 FOR TELETYPE 1 FOR B9352 2 FOR TC500 3 FOR B9353

<LINE LENGTH>   NUMBER OF CHARACTERS IN A LINE.

<PAGE SIZE>   NUMBER OF LINES ON A PAGE.

<NAK MAX>   MAXIMUM NUMBER OF "NAK"S ALLOWED BEFORE
                                AN ERROR IS CONSIDERED IRRECOVERABLE.

<ADDRESS 1> AND <ADDRESS 2>   THE TWO CHARACTER ADDRESSES OF
                                THE STATION. EACH CHARACTER MUST BE
                                ENCLOSED IN QUOTES.

<NOT EQUAL FLAG 1> AND <NOT EQUAL FLAG 2>   IF ON, INDICATE
                                THAT A NOT-EQUAL SIGN (CHANGE-MODE
                                CHARACTER) SHOULD BE INSERTED BEFORE
                                THE CORRESPONDING ADDRESS CHARACTER.

FIELDS THAT ARE NOT APPLICABLE FOR A GIVEN STATION SHOULD BE SET TO
0. LINE LENGTH AND PAGE SIZE APPLY ONLY TO SCREEN DEVICES, AND
ADDRESS CHARACTERS AND FLAGS APPLY ONLY TO MULTIPOINT STATIONS.

THE ADDRESSING IS ALWAYS ASSUMED TO START IN CONTROL MODE. THUS, IF
THE LAST FOUR ENTRIES ON A STATION CARD WERE:

    "1", "2", 1, 0,   THE ADDRESS WOULD BE TEXT 12, I.E., 12
    "1", "2", 0, 0,   WOULD BE CONTROL 1, CONTROL 2
    "1", "2", 0, 1,   WOULD BE CONTROL 1, TEXT 2
    "1", "2", 1, 1,   WOULD BE TEXT 1, CONTROL 2

THE INPUT TEXT CONSISTS OF A LINE CARD FOR EACH LINE ON THE SYSTEM,
FOLLOWED BY STATION CARDS AS NECESSARY. ALL SCHEDULE LINES MUST BE
DECLARED FIRST. STATION CARDS ARE NOT REQUIRED WITH SCHEDULE LINES,
AND WILL BE IGNORED IF USED.

THE LINES ARE DECLARED IN INCREASING ORDER OF TERMINAL UNIT AND
BUFFER NUMBER, WITH BUFFER NUMBER THE FASTER MOVING. EACH LINE CARD
MAY HAVE ONE OR MORE STATION CARDS AFTER IT, DESCRIBING THE DEVICES
ON THE LINE. THE FOLLOWING RULES APPLY:

    1) MORE THAN ONE STATION CARD IS NOT ALLOWED UNLESS THE LINE
        DISCIPLINE IS MULTIPOINT. A MAXIMUM OF 3 STATIONS ARE
        ALLOWED ON LINES WITH 28 CHARACTER BUFFERS AND 7 ON OTHER
        LINES.
    2) THE STATION CARD MAY BE OMITTED IF THE LINE DISCIPLINE IS
        TELETYPE. A CARD CONSISTING OF:

        STA, 0, 0, 0, 0, "0", "0", 0, 0,

        WILL BE ASSUMED.
    3) IF THE STATION CARD IS OMITTED FOR NON-TELETYPE LINES A
        WARNING WILL BE GIVEN, BUT THE PROGRAM WILL USE THE LAST
        STATION CARD ENTERED.

ANY TEXT CHARACTER EXCEPT <NOT-EQUAL> AND <QUOTE> MAY BE USED AS AN

ADDRESS CHARACTER ON A MULTIPOINT LINE. THE CONTROL CHARACTERS WHICH MAY NOT BE USED ARE:

CHARACTER	MEANING
<NOT-EQUAL SIGN>	CHANGE MODE
≠	ETX
%	ENQ
<GREATER THAN OR EQUAL SIGN>	SOH
&	ACK
5	NAK
P	POL
Q	SEL
T	BSL

FOR POINT-TO-POINT LINES, AN ADDRESS OF <NOT-EQUAL><NOT-EQUAL> SHOULD BE USED.

A SAMPLE DECK FOR:

1. ONE SCHEDULE LINE.
2. ONE DIAL-UP MODEL 35 TELETYPE ON 1/0 WITH 28-CHARACTER PING-PING BUFFERS.
3. ONE DIRECTLY-CONNECTED MODEL 33 TELETYPE ON 1/2 WITH A SINGLE 56-CHARACTER STRAIGHT BUFFER.
4. ONE DIAL-UP TC500 ON 1/4 OPERATING IN MULTIPOINT MODE WITH ADDRESS "11" WITH 56-CHARACTER PING-PING BUFFERS.
5. ONE DIRECTLY-CONNECTED B9352 ON 1/8 OPERATING IN POINT-TO-POINT WITH 56-CHARACTER PING-PING BUFFERS.
6. ONE DIAL-UP MODEL 35 TELETYPE ON 2/0 WITH 28-CHARACTER PING-PING BUFFERS.

WOULD BE SET UP AS FOLLOWS:

```
? EXECUTE SYSDISK/MAKER
?DATA CARD
LINE,0,0,112,0,0,7,0,
LINE,1,0,28,1,0,0,0,
STA,0,0,0,0,"0","0",0,0,
LINE,1,2,56,0,0,0,1,
STA,0,0,0,0,"0","0",0,0,
LINE,1,4,56,1,1,2,0,
STA,2,0,0,7,"1","1",1,0,
LINE,1,8,56,1,1,1,1,
STA,1,80,12,7,"≠","≠",0,0,
LINE,2,0,28,1,0,0,0,
STA,0,0,0,0,"0","0",0,0,
?END
```

KEYBOARD INPUT MESSAGES

THE KEYBOARD INPUT MESSAGES WHICH ARE THE SAME IN BOTH THE DCMCP AND THE TSSMCP ARE:

AX	BK	CC	CD	CI	CM
CT	DD	DT	DP	ED	EI
EX	FM	FR	IL	IN	LD
LF	LR	LS	MR	MX	OF
OK	OL	OU	PD	PG	PI
PR	QT	RD	RM	RO	RW
SF	SI	SO	ST	SY	TI
TF	TL	TO	UL	WD	WI
WM	WY	XD	XT		

THE MESSAGES IN THE DCMCP WHICH ARE NOT INCLUDED IN THE TSSMCP ARE:

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

THE DS MESSAGE IS INCLUDED WITH THE RESTRICTION THAT IT CANNOT BE USED IN THE FORM:

    DS<PROGRAM SPECIFIER>

THE ONLY FORM ALLOWED IS:

    <MIX INDEX>DS

THE MESSAGES WHICH HAVE BEEN MODIFIED FOR TIME SHARING ARE:

BS	CL	ES	LN	RS	RY
SM	SS	SV	TS	US	WU
XS					

THE MESSAGES WHICH HAVE BEEN ADDED ARE:

CE	CX	LNDK	MF

THESE MESSAGES ARE DESCRIBED BELOW.

THE BS MESSAGE

THE BS MESSAGE IS USED TO DESIGNATE A TERMINAL AS AN ALTERNATE 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 <STATION NUMBER>

ONLY ONE TERMINAL AT A TIME CAN BE USED AS AN ALTERNATE SPO. IT MUST BE DIALED-UP BUT NOT LOGGED IN AT THE TIME OF THE BS. 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 STATION OR TO CLEAR A PERIPHERAL UNIT. THE FORMATS ARE:

    CL <UNIT MNEMONIC>
    CL <STATION NUMBER>
    CL <STATION NUMBER>$

WHEN A UNIT IS CLEARED WITH THE FIRST FORMAT, THE JOB WHICH IS USING THAT UNIT, IF ANY, IS DS-ED.

WHEN A STATION IS CLEARED WITH THE SECOND FORMAT, A "BLAST READ" IS PERFORMED ON THAT LINE NO MATTER WHAT ITS CURRENT STATUS IS, CLEARING THE LINE AND MAKING IT IDLE.

WHEN A STATION IS CLEARED WITH THE THIRD FORMAT, THE STATION IS DISCONNECTED, THE USER IS LOGGED-OUT, AND IF THE USER WAS RUNNING A JOB, THAT JOB IS DS-ED.

WHEN USING THE SECOND FORMAT, MORE THAN ONE STATION NUMBER CAN BE INCLUDED IN A SINGLE CL MESSAGE. THE NUMBERS MAY BE SEPARATED BY ANY STRING OF NON-NUMERIC CHARACTERS OR THEY MAY BE RUN TOGETHER, IN WHICH CASE THEY WILL BE ASSUMED TO BE TWO DIGITS LONG. HOWEVER, STATION NUMBERS CANNOT BE MIXED WITH UNIT MNEMONICS.

IF THE "STATION NUMBER" REFERS TO A SCHEDULE LINE, PROCESSING OF THAT LINE WILL CEASE. IN ADDITION, IF THE SCHEDULE LINE NUMBER IS IMMEDIATELY FOLLOWED BY A $, THE LINE WILL BE SAVED. THEREFORE, CL3 $ IS EQUIVALENT TO:

    CL3

    SV3

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>

    E.G. CX MAKE TEST COBOL

THE ES MESSAGE

THE ES MESSAGE IS IDENTICAL TO THAT OF THE DCMCP EXCEPT THAT THE FORMAT:

    ES <SCHEDULE TASK NUMBER>

IS ALSO ALLOWED. THIS MESSAGE WILL ELIMINATE THE DESIGNATED TASK FROM THE SCHEDULED TASK QUEUE.

THE LN MESSAGE

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

    LN

REFER TO THE SECTION ON THE LOG.

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 A DECIMAL 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 OR AS NEAR TO THAT LOCATION AS POSSIBLE, PROVIDED THERE IS AN INTEGRAL NUMBER OF CHUNKS ABOVE THE FENCE.

THE MR MESSAGE

THE MR MESSAGE, AS IN THE DCMCP, CAUSES THE FILE RESERVE/DISK TO BE CREATED, PROVIDED IT DOES NOT ALREADY EXIST. THE FILE IS USED TO ATTEMPT RECOVERY FROM NO USER DISK SITUATIONS. THE MR MESSAGE FORMAT IS:

MR

IF THE FILE RESERVE/DISK IS CREATED, THE SYSTEM WILL RESPOND WITH:

RESERVE/DISK CREATED

IF THE FILE RESERVE/DISK WAS PREVIOUSLY CREATED, THE SYSTEM WILL RESPOND WITH:

RESERVE/DISK ALREADY PRESENT

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 OR WHEN CANDE/TSHARER IS NOT RUNNING. ITS FORMATS ARE:

    RS <STATION NUMBER> <MESSAGE>
    RS <USERCODE> <MESSAGE>
    RS ALL <MESSAGE>

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

THE RY MESSAGE

THE RY MESSAGE IS IDENTICAL TO THAT OF THE DCMCP EXCEPT THAT IT MAY BE USED TO READY SCHEDULE LINES AS WELL AS PERIPHERAL UNITS. SCHEDULE LINES MUST BE RY-ED AFTER EVERY HALT/LOAD OR SCHEDULED TASKS WILL NOT BE RUN. THE FORMAT IS:

    RY <STATION NUMBER>

SEVERAL STATION NUMBERS MAY APPEAR IN ONE MESSAGE. FOR EXAMPLE:

    RY 1
    RY 1,2,3,4
    RY 1 2 3 4

THE RS MESSAGE

DEPENDING ON ITS FORMAT, THE RS MESSAGE IS USED TO OBTAIN A SUMMARY OF THE ACTIVITIES ON THE SYSTEM, ON A GIVEN STATION, OR FOR A GIVEN

JOB. IF THE FORMAT IS:

  SM A6

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:

       <USERCODE> ON <STATION NUMBER>
       <USERCODE> ON <STATION 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:

       <USERCODE> 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<STATION NUMBER>

A SUMMARY OF THE ACTIVITIES FOR THAT STATION 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
STATION. IF CANDE IS NOT RUNNING, THE MESSAGE IS OUTPUT IMMEDIATELY.

THE SS MESSAGE HAS THREE FORMATS.

   SS <STATION NUMBER><MESSAGE>
   SS <USERCODE><MESSAGE>
   SS ALL <MESSAGE>

THE <MESSAGE> WILL BE SENT TO THE SPECIFIED STATION, OR TO ALL USERS

WITH THE SPECIFIED <USERCODE>.

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 <USERCODE> IS GIVEN, THE <MESSAGE> BEGINS WITH THE FIRST SPECIAL CHARACTER OR WITH THE EIGHTH CHARACTER. IF A <STATION 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 SV MESSAGE

THE SV MESSAGE IS IDENTICAL TO THAT OF THE DCMCP EXCEPT THAT IF AN SV <STATION NUMBER> IS GIVEN AND THE STATION IS A SCHEDULE LINE, THAT LINE WILL BE MARKED NOT READY. THIS WILL NOT AFFECT A TASK CURRENTLY USING THAT LINE, BUT IT WILL PREVENT TASKS FROM BEING INITIATED ON THAT LINE. E.G.

    SV 1
    SV 1,2,3,4
    SV 1 2 3 4

THE TS MESSAGE

THE TS MESSAGE WILL RESPOND WITH A LIST OF SCHEDULED JOBS, IF ANY, FOLLOWED BY A LIST OF SCHEDULED TASKS, IF ANY, AND THE SCHEDULE LINE ASSOCIATED WITH EACH. IF THERE ARE NO SCHEDULED JOBS AND NO SCHEDULED TASKS, THE RESPONSE WILL BE:

    NULL SCHEDULE
    NULL TASK SCHEDULE

OTHERWISE, SCHEDULED JOBS WILL BE LISTED AS WITH THE DCMCP, AND SCHEDULED TASKS WILL BE LISTED IN THE ORDER THEY WILL BE PROCESSED, WITH THE TASK NUMBER, USERCODE, AND THE EARLIEST THE JOB CAN BE RUN (IF SO SPECIFIED BY A USER AT A REMOTE TERMINAL) APPEARING FOR EACH TASK.

THE US MESSAGE

THE US MESSAGE "TURNS OFF" THE SPO OR AN ALTERNATE SPO AND IS THE CONVERSE OF THE BS MESSAGE. IF THE SPO IS US-ED, IT CAN STILL BE USED FOR INPUT BUT WILL NO LONGER DISPLAY OUTPUT IF AN ALTERNATE SPO HAS BEEN DESIGNATED. THE FORMATS ARE:

```
US SPO
US <STATION NUMBER>
```

THE WU MESSAGE

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

```
WU
```

OR

```
WU<STATION NUMBER>
```

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

```
<USERCODE> ON <STATION NUMBER>
```

IF THERE ARE NO USERS, THE MESSAGE IS

```
NULL WU
```

IF THE FORMAT IS:

```
<MIX> WU
```

THE USERCODE, STATION NUMBER (IF APPROPRIATE) AND JOB SPECIFIER OF
THE JOB WITH THE SPECIFIED MIX INDEX ARE OUTPUT AS

```
<USERCODE> ON <STATION NUMBER> USING <JOB SPECIFIER>
```

THE XS MESSAGE

THE XS<SCHEDULE TASK NUMBER> MESSAGE WILL PUT THE DESIGNATED TASK AT
THE FRONT OF THE SCHEDULE QUEUE. IF A "TIME AFTER" WAS SPECIFIED,
IT WILL BE CHANGED TO ZERO.

THE <SCHEDULE INDEX>XS FORM OF THE MESSAGE CAN BE USED TO START
EXECUTION OF A BACKGROUND "COMPILE" OR "EXECUTE" JOB (OR LIBRARY
MAINTENANCE) WHICH WAS PLACED IN THE SCHEDULE BECAUSE ITS CORE
REQUIREMENTS OVERLAPPED THOSE OF BACKGROUND JOBS ALREADY RUNNING.
THE CHUNKS USED BY AN XS-ED JOB WILL COUNT AS FOREGROUND CHUNKS AND
NOT AS CHUNKS USED BY A BACKGROUND JOB. THUS, THE RESTRICTION THAT
A NEWLY ENTERING BACKGROUND JOB CAN ONLY ENTER IF THERE ARE ENOUGH
UNUSED CHUNKS IS CIRCUMVENTED.

NOTE: THIS RESTRICTION CAN BE REMOVED ALTOGETHER BY SETTING THE
OPTION "NOBATCH". SEE "OPTIONS".

THE XS MESSAGE CAN ALSO BE USED TO START EXECUTION OF A JOB THAT WAS
INITIATED WITH A "RUN" CARD, AND WAS SCHEDULED BECAUSE THERE WASN'T
SUFFICIENT ROOM FOR IT BELOW THE FENCE. HOWEVER, THE JOB WILL RUN
BELOW THE FENCE, AND HAS A HIGH PROBABILITY OF CAUSING A SYSTEM HANG
DUE TO RUNNING OUT OF MEMORY. THE USE OF THE XS MESSAGE FOR THIS
PURPOSE IS THEREFORE NOT RECOMMENDED.

THE OPTIONS

THE OPTIONS LISTED BELOW CAN BE INITIALLY SET IN THE COLD OR COOL START DECK AS DESCRIBED FOR THE DCMCP. THEREAFTER, THEY CAN BE SET, RESET AND TYPED BY THE SO, RO AND TO MESSAGES. OPTIONS MARKED BY THE LETTER "B" ARE USED ONLY BY THE BATCH MCP. OPTIONS MARKED BY THE LETTER "T" ARE EXCLUSIVE TO THE TSSMCP AND ARE DESCRIBED IN THE PARAGRAPHS FOLLOWING THE LIST.

47		USE DRA	34		TYPE CLOSE	21		USE PBDONLY
46		USE DRB	33		TYPE ERRORMSG	20		USE SAVEPBT
45		TYPE BOJ	32		USE RET	19		TYPE RSMSG
44		TYPE EOJ	31		TYPE LIBMSG	18		USE AUTOUNLD
43		TYPE OPEN	30		TYPE SCHEDMSG	17		USE RNALL
42		USE TERMNATE	29		TYPE SECMSG	16	B	USE CODEOLAY
41		TYPE DATE	28		USE DSKTOG	15	B	USE COREST
40		TYPE TIME	27		USE RELTOG	14	B	USE DATAOLAY
39	B	USE ONEBREAK	26		USE PBDREL	13	T	USE HALT
38		USE AUTOPRNT	25		USE CHECK	12	T	USE REMOTE
37	B	USE CLEARWRS	24	B	USE DISKMSG	11	T	TYPE CEMESS
36	B	USE DISCONDC	23	T	TYPE DISKLOG	10	T	USE BATCHZIP
35		TYPE CMPLFILE	22	T	TYPE LIBERR	9	T	USE NOBATCH

THE "USE HALT" OPTION CAN BE USED TO FORCE A SYSTEM HALT IF CERTAIN ERROR CONDITIONS OCCUR. IF THE "USE HALT" OPTION IS RESET THE TSSMCP WILL TYPE:

> ***SHOULD H/L: HALT CAUSED BY <REASON>

BUT THE SYSTEM WILL CONTINUE RUNNING. IF HALT IS SET, THE MESSAGE WILL NOT PRINT OUT, AND THE ERROR CONDITION WILL CAUSE A SYSTEM HALT.

THE "USE REMOTE" OPTION MUST BE SET TO ALLOW DATACOM 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, AND IS AUTOMATICALLY SET WHEN A HALT/LOAD IS DONE IF ALL IS IN ORDER AT THAT TIME.

THE "USE CHECK" OPTION CAUSES A MEMORY LINK CHECK OF ALL LINKS IN MEMORY EVERY TIME THE ROUTINES GETSPACE AND FORGETSPACE ARE CALLED. IF AN INVALID LINK IS FOUND, THE SYSTEM HANGS. A SIMILAR CHECK IS MADE OF THE QUEUES OF SEGMENTED SAVE AREAS EVERY TIME A FORGETAREA IS DONE. SINCE THESE CHECKS ARE CONSUMING, THIS OPTION IS USUALLY SET ONLY FOR DEBUGGING AND IS RESET DURING NORMAL SYSTEM OPERATION. IN ORDER TO USE THIS OPTION THE TSSMCP MUST BE COMPILED WITH CHECKLINK SET TRUE. IF NOT, THE SETTING OF THE CHECK OPTION IS IRRELEVANT.

THE "TYPE CEMESS" OPTION IS USED TO OVERRIDE THE SETTING OF THE LIBMSG, BOJ, AND EOJ OPTIONS FOR CANDE-RELATED MESSAGES. THAT IS, IF CEMESS IS RESET, BOJ, EOJ, AND LIBRARY MAINTENANCE MESSAGES

ARISING FROM CANDE-S ACTIVITIES WILL BE SUPRESSED REGARDLESS OF THE
SETTING OF THE OTHER OPTIONS. IF CEMESS IS SET, THESE MESSAGES WILL
BE PRINTED SUBJECT TO THE CONTROL OF THE OTHER OPTIONS, AS IS THE
CASE FOR NON-CANDE MESSAGES. SINCE NORMAL CANDE OPERATION GENERATES
AN ENORMOUS NUMBER OF THESE MESSAGES, THIS OPTION IS USUALLY 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.

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 TYPED.

THE "BATCHZIP" OPTION IS USED WITH THE SHARED DISK VERSION OF THE
TSSMCP. IF IT IS SET, BACKGROUND JOBS WILL BE RUN ON THE OTHER
SYSTEM.

THE "NOBATCH" OPTION CONTROLS THE SCHEDULING OF "COMPILE" AND
"EXECUTE" BACKGROUND JOBS. IF "NOBATCH" IS RESET, NEWLY ENTERING
BACKGROUND JOBS WILL BE STARTED ONLY IF THEIR CORE REQUIREMENTS DO
NOT OVERLAP THOSE OF BACKGROUND JOBS ALREADY RUNNING. THIS PREVENTS
BACKGROUND JOBS FROM BEING SWAPPED WITH ONE ANOTHER. IF "NOBATCH"
IS SET, THIS RESTRICTION IS REMOVED AND NEWLY ENTERING BACKGROUND
JOBS WILL BE STARTED IMMEDIATELY THE SAME AS REMOTE JOBS.

THE ALTERNATE SPO

THE ALTERNATE SPO CAN BE USED WITH TIME SHARING TO ENTER INPUT SPO MESSAGES, AS WELL AS TO PRINT THEM OUT. THE BS <STATION NUMBER> COMMAND IS USED TO DESIGNATE A STATION AS THE BACKUP SPO, WHILE US <STATION NUMBER> IS USED TO REMOVE THE LINE AS A SPO. (NOTE: TO BS OR US THE REAL SPO USE STATION NUMBER 0 OR THE WORD "SPO" AS IN BSO, USO, BS SPO, OR US SPO.)

THERE CAN ONLY BE ONE BACK UP SPO WITH TIME SHARING. IF A BS IS DONE TO ANOTHER STATION (OTHER THAN 0) AND THERE ALREADY IS A BACK UP SPO, THE NEW STATION WILL REPLACE THE OLD ONE AS THE BACK UP SPO. IF A US TO A STATION OTHER THAN 0 OR SPO IS DONE, THE REAL SPO BECOMES THE ONLY SPO, EVEN IF IT HAS BEEN US"ED IN THE PAST. IF THE BACK UP SPO DISCONNECTS, THE ACTION TAKEN IS THE SAME AS IF A US HAD BEEN DONE FOR THAT STATION.

IN ORDER TO DO A BS TO A STATION, IT MUST BE DIALED-UP, BUT NOT LOGGED ON TO CANDE. IF EITHER OF THESE TWO REQUIREMENTS IS NOT MET, AN "INV KBD" WILL BE GIVEN. ANY OF THE DEVICES IN THE CURRENT MCP MAY BE USED AS A BACK UP SPO (I.E., TELETYPE, B9352, TC500).

IT IS EXPECTED THAT THE USE OF A BACK UP SPO WOULD BE TO ATTEMPT TO SAVE I/O CHANNELS FROM BEING TIED UP BY THE SPO. TO USE THIS, THE OPERATOR SHOULD BS A STATION AND THEN USO OR USSPO. THIS SHOULD ELIMINATE MOST OF THE SPO I/O. SOME OF THE DISCUSSION TO FOLLOW IS CONCERNED WITH THIS OPERATION CONFIGURATION.

A PROBLEM CAN ARISE WITH THE USE OF A BACK UP SPO BY FLOODING THE SYSTEM WITH KEY IN MESSAGES BEFORE IT CAN FINISH PROCESSING THEM. THIS COULD ESPECIALLY HAPPEN WITH BUFFERED DEVICES LIKE THE TC500 OR B9352. IF THE MCP FINDS MORE THAN THREE MESSAGES FROM THE BACK UP SPO WAITING TO BE PROCESSED, IT WILL THROW AWAY ANY FURTHER MESSAGES, UNTIL THE NUMBER OF WAITING MESSAGES DECREASES. THIS IS DONE IN AN ATTEMPT TO PREVENT NO MEMS BEING CAUSED BY THE BACK UP SPO.

IF A CONTROL CARD IS ENTERED FROM A BACK UP SPO AND AN ERROR IS MADE, THE MCP WILL PRINT OUT THE ERROR AND THEN ASSUME A "CC END." CARD FOLLOWS. THIS WILL ALSO BE DONE IF AN INCOMPLETE CARD IS ENTERED, ALTHOUGH "CONTROL CARD ERROR" MAY NOT BE PRINTED. THUS, THE OPERATOR CAN TAKE ADVANTAGE OF THIS AT A BACK UP SPO AND LEAVE OFF THE "; END." AS IN "CC LOAD FROM SYSTEM MCP/DISK.".

IF THE OPTION "TERMNATE" IS NOT SET OR CANDE GETS DS-ED, THE MCP WILL WAIT FOR AN INVALID SPO MESSAGE TO TERMINATE THE JOB. IF THE SPO HAS NOT BEEN US-ED THE INPUT READY LIGHT WILL COME ON AND THE MCP WILL WAIT FOR AN INVALID KEY IN. IF THE SPO HAS BEEN US-ED, THE MCP WILL SEND A "?" TO THE BACK UP SPO AND WAIT FOR AN INVALID RESPONSE FROM THE BACK UP SPO.

IT MAY HAPPEN THAT WHILE THE BACK UP SPO IS IN THE PROCESS OF
WRITING SEVERAL MESSAGES, THE OPERATOR WISHES TO INTERVENE IN ORDER
TO ENTER A KEY IN FROM THE BACK UP SPO. THE BREAK KEY CAN BE USED
IN THIS CASE AS THE INPUT REQUEST KEY. WHEN IT IS USED, THE MCP
WILL STOP SENDING MESSAGES TO THE BACK UP SPO UNTIL A KEY IN IS MADE.

BK OR <MIX> BK MESSAGES MAY STILL BE USED FROM THE BACK UP SPO TO
STOP OUTPUT TO THE SPO OR BACK UP SPO. HOWEVER, IN THE CASE WHERE
OUTPUT IS GOING ONLY TU THE BACK UP SPO, THE BK OR <MIX> BK WILL NOT
BE ABLE TO DISTINGUISH THE <MIX> (OR LACK OF) AND WILL THROW ALL OF
THE OUTPUT AWAY.

THE SPO WILL ALWAYS BE ABLE TO ENTER KEY INS IF THE INPUT REQUEST
BUTTON IS PRESSED, EVEN IF IT HAS BEEN US-ED.

PROGRAMMING LANGUAGES

THE LANGUAGES AVAILABLE WITH THE TIME SHARING SYSTEM ARE ALGOL, BASIC, COBOL, COBOL68, FORTRAN, TSPOL, AND XALGOL (COMPATIBLE ALGOL). ALGOL AND TSPOL SHOULD NOT NORMALLY BE USED FROM A REMOTE TERMINAL. ALGOL IS BURROUGHS B 5500 EXTENDED ALGOL AND TSPOL (TIME SHARING PROBLEM ORIENTED LANGUAGE) IS ALGOL WITH THE COMMUNICATE STATEMENT ADDED.

THIS LEAVES BASIC, COBOL, COBOL68, 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 DCMCP 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 OPENS A REMOTE FILE, IT WILL BE DIVERTED TO THE SPO.

WHEN ENTERING SOURCE CODE FROM A REMOTE TERMINAL, EACH LINE OF INPUT MUST BEGIN WITH A SEQUENCE NUMBER. THE SEQUENCE NUMBERS ARE EQUIVALENT TO THOSE APPEARING IN COLUMNS 73 TO 80 OF AN ALGOL OR FORTRAN PROGRAM OR TO THOSE IN COLUMNS 1 TO 6 IN A STANDARD COBOL PROGRAM. THE SYSTEM WILL AUTOMATICALLY PLACE THE SEQUENCE NUMBER RIGHT JUSTIFIED IN COLUMNS 73 TO 80 OF THE CARD IMAGE ON DISK. 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, COBOL68, 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 DATA-NAME TO DATA.
    2. ASSIGN DATA-NAME TO MEMORY.
    3. UNTIL (DATA-NAME, MASK)
    4. WHEN DATA-NAME
    5. PERFORM("PROGRAM-PREFIX", "PROGRAM-SUFFIX")
    6. PERFORM WITH DATA-NAME.

COBOL/COBOL68 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 DATA-NAME FROM REMOTE.

    ACCEPT DATA-NAME.

    DISPLAY DATA-NAME UPON REMOTE.

    DISPLAY DATA-NAME.

THE SPO IS NOW ACCESSED WITH THE SYNTAX

    ACCEPT DATA-NAME FROM KEYBOARD.

    DISPLAY DATA-NAME UPON KEYBOARD.

NOTE THAT DATA-NAME MUST BE AN 01 ITEM IN WORKING-STORAGE AND MAY NOT BE GREATER THAN 72 CHARACTERS LONG WHEN USED WITH ACCEPT AND DISPLAY.

THE SYNTAX FOR USING READ AND WRITE IS:

    SELECT FILE-NAME ASSIGN TO REMOTE.

    READ FILE-NAME RECORD[INTO ...][BEFORE ADVANCING DATA-NAME
    LINES] [ON ERROR ANY STATEMENT[ELSE ANY STATEMENT]]

    WRITE RECORD-NAME[FROM DATA-NAME][BEFORE ADVANCING DATA-NAME
    LINES] [ON ERROR ANY STATEMENT[ELSE ANY STATEMENT]].

THE "ON ERROR" BRANCH IS TAKEN ON THE NEXT WRITE AFTER BREAK IS PUSHED.

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.

COBOL/COBOL68 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. NO SPACES MAY SEPARATE A SECTION NAME, DIVISION NAME, OR PARAGRAPH NAME FROM THE SEQUENCE NUMBER.

4. PROGRAM INSTRUCTIONS MUST BE SEPARATED FROM THE SEQUENCE NUMBER BY A BLANK OR MAY FOLLOW A LABEL.

FOR PROGRAMS INPUT THROUGH THE CARD READER, ALL COBOL CONSTRUCTS ARE STANDARD.

SINCE CANDE SEQUENCES IN COLUMNS 73 TO 80 AND COBOL SEQUENCES IN COLUMN 1 TO 6 , SOLT DISK FILES CREATED ON THE DCMCP THROUGH USE OF FILE EQUATION OR SOLT TAPE FILES FROM STANDARD SYSTEM MUST BE REGENERATED. THERE ARE THREE $ OPTIONS WHICH PERTAIN TO THESE DIFFERENCES:

```
$FREE ::= THE PRIMARY SOURCE IS IN CANDE FORMAT.
$TSSEDIT ::= THE OUTPUT (NEWTAPE) IS TO BE IN CANDE FORMAT.
$TSSTAPE (COBOL) OR $TSSMERGE (COBOL68) ::= THE SECONDARY
                                SOURCE IS IN CANDE FORMAT.
```

FORTRAN

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 TSSEDIT 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 WHEN USED FROM REMOTES AND THEREFORE CAN BE USED LIKE ANY WORD.

FORTRAN 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.

FORTRAN 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.

2. COMMENT CARDS CONTAIN A "C" IN COL. 1, A "-" IN COL. 2, AND THE 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.

FORTRAN $ 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.

THE FOLLOWING ADDITIONAL OPTIONS APPLY TO PROGRAMS COMPILED THROUGH
THE 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.

IF THE $ ONSITE OR $ REMOTE OPTIONS ARE USED, THEY MUST APPEAR IN
THE 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 PRIMARY 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 TSSMCP CODE RESULTS IN AN INVALID INDEX IN THE TSSMCP.

FILE SECURITY

THE B-5700 TSSMCP USES THE FILE SECURITY SYSTEM DEVELOPED FOR THE DCMCP. THE USERCODE, PASSWORD AND GUARD FILE IN THE TSSMCP ARE EQUIVALENT TO THE USERCODE, AUTHENTICATION CODE AND SECURITY FILE IN THE DCMCP. THIS SYSTEM RECOGNIZES ONE PRIVILEGED USER WHO IS ALLOWED ACCESS TO ALL FILES ON THE SYSTEM. THIS USERCODE IS "SITE". THE SPO IS AUTOMATICALLY ASSIGNED THE USERCODE "SITE".

ALL REMOTE USERS ARE SUBJECT TO THE CONSTRAINTS OF THE FILE SECURITY SYSTEM. FOR THEM, THERE ARE FOUR LEVELS OF FILE SECURITY:

1. A LOCKED FILE MAY BE ACCESSED ONLY BY THE PERSON WHO CREATED IT. THIS TYPE OF SECURITY IS CREATED BY THE USE OF THE "SAVE" COMMAND. ALL FILES START AS LOCKED FILES.

2. A GUARDED FILE MAY BE ACCESSED BY THE USERS AND/OR PROGRAMS LISTED IN THE GUARD FILE ASSOCIATED WITH THE FILE. A GUARDED FILE CAN BE CREATED BY THE USE OF THE "GUARD" AND THE "LOCK" COMMANDS.

3. AN UNLOCKED FILE CAN BE READ (OR EXECUTED, IF THE FILE HAS AN OBJECT VERSION) BY ANYONE BUT IT CAN ONLY BE CHANGED BY THE PERSON WHO CREATED IT. UNLOCKED FILES ARE CREATED BY THE USE OF THE "UNLOCK" COMMAND.

4. A PUBLIC FILE, CREATED BY THE USE OF THE "PUBLIC" COMMAND, MAY BE READ OR WRITTEN BY ANYONE.

THE SECURITY STATUS OF A FILE IS TREATED IN MUCH THE SAME WAY AS THE FILE NAME OR THE FILE TYPE. IN EFFECT, THE SECURITY STATUS IS LOADED WITH THE FILE AND, IF NOT CHANGED BY THE USER, REMAINS WITH IT WHEN THE FILE IS SAVED, EVEN IF THE FILE NAME HAS BEEN CHANGED.

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 $ CARD OPTIONS AND USER OPTIONS.

1) $ CARD OPTIONS.

A "$ OPTION" CARD CONTAINS THE CHARACTER "$" IN COLUMN 1, FOLLOWED BY AN OPTION TYPE. IN SOME INSTANCES, THE OPTION TYPE MUST BE FOLLOW BY A DESCRIPTION, DELIMITED WITH QUOTE (") MARKS.

THE USER/CANDE PROGRAM CURRENTLY RECOGNIZES THE FOLLOWING $ CARD OPTIONS:

```
$ NEW
$ OPTIONS
$ USER "USRCODE"
$ REMOVE "USRCODE"
$ PRINT
$ LINK
```

A) $ NEW

THE $ NEW OPTION CAUSES THE USER/CANDE PROGRAM TO CREATE A NEW USERS/CANDE FILE. IF A USERS/CANDE FILE ALREADY RESIDES ON THE DISK, IT IS REMOVED AND REPLACED WITH A SKELETON FILE, CONTAINING NO USERS.

THE $ NEW CARD MUST BE THE FIRST CARD READ BY THE USER/CANDE PROGRAM. IF THIS CARD APPEARS AFTER THE FIRST CARD HAS BEEN READ, AN ERROR MESSAGE WILL BE PRINTED ON THE LINE PRINTER, AND THE INSTRUCTION IGNORED.

B) $ OPTIONS

THE USER/CANDE PROGRAM WILL PLACE ALL USER OPTIONS APPEARING BETWEEN THE $ OPTIONS CARD AND THE NEXT $ CARD IN THE "OPTIONS" BLOCK (THESE ARE THE DEFAULT OPTIONS FOR USERS TO BE ADDED TO THE FILE, OR CORRECTED WITHIN THE FILE). ONCE THESE OPTIONS HAVE BEEN SET, ALL USERS REFERENCED BY A $ USER CARD, WHETHER NEW USERS OR USERS ALREADY PRESENT IN THE USERS/CANDE FILE, ARE ASSIGNED THOSE OPTIONS SPECIFIED IN THE OPTIONS BLOCK.

ANY USER OPTIONS ASSIGNED FOLLOWING THE $ USER CARD WILL TAKE PRECEDENCE OVER THOSE IN THE OPTIONS BLOCK.

EXAMPLE:

$ OPTIONS

```
REQUEST CHARGE
PASSWORD "B5700"
$ USER "USRCODE"
NO CHARGE
```

THE CARDS
```
    REQUEST CHARGE
AND
    PASSWORD "B5700"
```
APPEARING AFTER THE $ OPTIONS CARD, WILL RESULT IN A CHARGE
CODE OF ZERO AND A PASSWORD "B5700" BEING PLACED IN THE
OPTIONS BLOCK.

IMMEDIATELY AFTER THE $ USER "USRCODE" CARD IS READ, THESE
OPTIONS ARE ASSIGNED TO THE USER "USRCODE". THE "NO CHARGE"
CARD WILL THEN BE READ AND A "-1   " PLACED IN THE CHARGE
CODE FIELD FOR THIS USER.

THE NET RESULT WILL BE THAT THIS USER WILL BE ASSIGNED A
PASSWORD OF "B5700" AND A CHARGE CODE OF "-1   ".

NOTE:
IT IS SOMETIMES DESIRABLE TO CHANGE ONLY ONE FIELD IN A USERS
RECORD, SUCH AS A CHANGE IN THE PHONE WORD ONLY. THIS CAN BE
ACCOMPLISHED BY ENTERING:

```
$ USER "USRCODE"
PHONE "PHONEWD"
```

PROVIDED THAT THE OPTIONS BLOCK HAS NOT BEEN SET. IF THE
DEFAULT OPTIONS HAVE ALREADY BEEN SET, THESE OPTIONS WILL
AUTOMATICALLY BE ASSIGNED TO THE USER AT THE TIME THE $ USER
CARD IS REFERENCED.

C) $ USER "USRCODE"

THIS CARD CAUSES THE RECORD BELONGING TO THE PREVIOUS USER (IF
ANY) REFERENCED IN THE USER/CANDE PROGRAM, TO BE WRITTEN ON
DISK. A SEARCH OF THE USERS/CANDE FILE IS THEN MADE TO
DETERMINE WHETHER THE NEW USER HAS ALREADY BEEN ENTERED IN THE
FILE. IF THE SPECIFIED USER IS ALREADY IN THE FILE, HIS
EXISTING RECORD IS COPIED INTO CORE. IF THE USER IS NOT IN
THE FILE, SPACE IS ALLOCATED FOR THIS USER IN THE USERS/CANDE
TABLE. ALL USER OPTIONS APPEARING AFTER THE $ USER CARD AND
BEFORE THE NEXT $ CARD WILL BE ASSIGNED TO THIS USER.

D) $ REMOVE "USRCODE"

TO FACILITATE THE REMOVAL OF RECORDS FROM THE USERS/CANDE
FILE, A "LINK" MUST BE ESTABLISHED BETWEEN A USERS SEGMENT
(CONTAINING THE USERS PASSWORD, CHARGE CODE, NAME, ETC.) AND
THE "SCRAMBLE TABLE".

THE USER/CANDE PROGRAM WILL ESTABLISH THIS LINK WHENEVER A NEW USER IS ADDED TO THE FILE. HOWEVER, OLDER USERS/CANDE FILES (CREATED PRIOR TO THE ADDITION OF THE LINK OPTION) WILL NOT BE PROPERLY LINKED, AND THE $ REMOVE CARD WILL RESULT IN AN ERROR MESSAGE.

ANOTHER OPTION, REQUIRING A CARD OF THE FORM
    $ LINK
HAS THEREFORE BEEN PROVIDED TO ESTABLISH THE "LINKS" IN EXISTING USERS/CANDE FILES.

NOTE:
THE $ LINK OPTION NEED ONLY BE USED ONCE. AFTER THE "LINKS" HAVE BEEN ESTABLISHED, THE USER/CANDE PROGRAM WILL PRESERVE THEM AND/OR ADD THEM, AS REQUIRED.

E) $ PRINT

THIS OPTION CAUSES THE PRINTING OF THE CONTENTS OF THE USERS/ CANDE FILE ON THE LINE PRINTER. THE "SCRAMBLE TABLE" IS PRINTED FIRST, FOLLOWED BY THE USERS OPTION SEGMENTS, IN ALPHABETICAL ORDER BY USER CODES.

THE "$PRINT" OPTION PRINTS THE USERS/CANDE FILE AS IT EXISTS AT THE TIME THE "$PRINT" IS READ. THEREFORE, THIS OPTION SHOULD NORMALLY BE THE LAST OPTION IN THE DECK TO OBTAIN A TRUE COPY OF THE UPDATED USERS/CANDE FILE.

2) USER OPTIONS

THE FOLLOWING OPTIONS MAY BE ASSIGNED TO A USER FOLLOWING A $ USER "USRCODE" CARD.

PASSWORD "PASSWRD"

THIS CONSTRUCT INSERTS THE CHARACTER STRING CONTAINED BETWEEN QUOTES INTO THE USERS PASSWORD FIELD. THE PASSWORD MAY BE ANY STRING FROM 1 TO 7 CHARACTERS IN LENGTH. IF FEWER THAN 7 CHARACTERS ARE USED, THEY ARE PLACED IN THE FIELD LEFT JUSTIFIED WITH TRAILING BLANKS.

NAME "USERNAME"

THIS CONSTRUCT INSERTS THE CHARACTER STRING CONTAINED BETWEEN QUOTES INTO THE USERS NAME FIELD. THE NAME MAY BE ANY CHARACTER STRING FROM 1 TO 15 CHARACTERS IN LENGTH. IF FEWER THAN 15 CHARACTERS ARE USED, THEY ARE PLACED IN THE NAME FIELD LEFT JUSTIFIED WITH TRAILING BLANKS.

TIME "A STRING OF 24 1-S OR 0-S"

EACH OF THE 24 DIGITS BETWEEN QUOTES REPRESENTS THE HOURS OF THE DAY, THE FIRST DIGIT BEING THE HOUR FROM MIDNIGHT TO 1 A.M.

A "1" INDICATES THE HOUR IN WHICH THE SYSTEM MAY BE USED, A "0" THE HOUR IN WHICH THE SYSTEM MAY NOT BE USED.

IF FEWER THAN 24 DIGITS APPEAR IN THE STRING, THE STRING IS FILLED OUT TO 24 DIGITS WITH ZEROS.

IF ALL 24 DIGITS ARE ZEROS, **NO** TIME RESTRICTION IS PLACED ON THE USER.

CHARGE "CHRGWRD"

THIS CONSTRUCT PLACES THE CHARACTER STRING CONTAINED WITHIN THE QUOTES IN THE USERS CHARGE FIELD. THE CHARGE CODE MAY BE ANY STRING FROM 1 TO 7 CHARACTERS IN LENGTH. IF FEWER THAN 7 CHARACTERS ARE USED, THEY ARE PLACED IN THE CHARGE FIELD LEFT JUSTIFIED WITH TRAILING BLANKS.

REQUEST CHARGE

THIS CONSTRUCT PLACES A STRING OF ZEROS IN THE USERS CHARGE FIELD. THE ZEROS ARE A FLAG TO CANDE TO REQUEST A CHARGE FROM THE USER AT LOG-IN TIME.

NO CHARGE

THIS CONSTRUCT PLACES THE 7 CHARACTER STRING "-1   " IN THE USERS CHARGE FIELD.

LANGUAGES [LANGUAGE LIST]

THIS CONSTRUCT PLACES IN THE USERS LANGUAGE FIELD THE LIST OF COMPILERS WHICH MAY **NOT** BE USED.

EXAMPLE: LANGUAGES COBOL, ALGOL

THE CONSTRUCT:
    LANGUAGES NONE
MAY BE USED TO REMOVE ALL LANGUAGE RESTRICTIONS FOR A USER.

VERBS [VERB LIST]

THIS CONSTRUCT PLACES IN THE USERS VERB FIELD THE LIST OF CANDE VERBS WHICH MAY **NOT** BE USED.

EXAMPLE: VERBS EXECUTE,COMPILE,RUN,DO

THE CONSTRUCT
    VERBS NONE
CAN BE USED TO REMOVE ALL VERB RESTRICTIONS FOR A USER.

PHONE "PHONEWRD"

THIS CONSTRUCT PLACES THE CHARACTER STRING CONTAINED BETWEEN

QUOTES IN THE USERS PHONE WORD FIELD. THE PHONEWORD MAY BE ANY STRING FROM 1 TO 7 CHARACTERS IN LENGTH. IF FEWER THAN 7 CHARACTERS ARE USED, THE STRING IS PLACED IN THE PHONE WORD LEFT JUSTIFIED WITH TRAILING BLANKS.

USERS/CANDE

THE FILE "USERS/CANDE" IS USED BY CANDE/TSHARER TO OBTAIN THE SECURITY AND ACCOUNTING INFORMATION ASSOCIATED WITH EACH USER. THE FILE IS COMPRISED OF THREE SECTIONS:

A) SEGMENT ZERO, WHICH CONTAINS THE RECORD COUNT AND DATE OF LAST ALTERATION OF THE FILE. THE NEXT AVAILABLE SEGMENT POSITION IS KEPT IN WORD 1 OF SEGMENT ZERO, AND THE DATE OF LAST ALTERATION IS KEPT IN WORD 2 OF SEGMENT ZERO.

B) THE INITIAL SCRAMBLE TABLE SECTION, LOCATED IN SEGMENTS 1 THROUGH 29 OF THE FILE. EACH SCRAMBLE SEGMENT CONSISTS OF 15 TWO WORD ENTRIES, THE FIRST WORD CONTAINING THE USERCODE AND THE [33:15] FIELD OF THE SECOND WORD CONTAINING THE RELATIVE DISK ADDRESS OF THE SECURITY SEGMENT ASSOCIATED WITH THAT USERCODE. THE [18:15] FIELD OF WORD 1 OF EACH SCRAMBLE SEGMENT IS A LINK TO THE NEXT SCRAMBLE SEGMENT.

THE ENTRY FOR A GIVEN USER IS PLACED IN THE SEGMENT FOUND BY ARITHMETICALLY ADDING THE FIRST THREE AND A HALF CHARACTERS OF THE USERCODE TO THE LAST THREE AND A HALF CHARACTERS, TAKING THE RESULT MODULO 29 AND ADDING 1.

WHEN AN ATTEMPT IS MADE TO ADD A NEW USERCODE TO THE SCRAMBLE BLOCK SECTION, AND THAT BLOCK IS ALREADY FULL, THE USER CANDE PROGRAM WILL CREATE A NEW SCRAMBLE BLOCK IN THE USERS/CANDE FILE, AND WILL PLACE THE LOCATION OF THE LINKED BLOCK IN THE [18:15] FIELD OF WORD 1 OF THE CURRENTLY ACCESSED BLOCK. THE [18:15] FIELD OF WORD 29 OF EACH LINKED SCRAMBLE BLOCK IS USED AS A BACK LINK TO THE PREVIOUS SCRAMBLE BLOCK FOR PURPOSES OF FILE MAINTENANCE.

C) THE BODY OF THE USERS/CANDE FILE, LOCATED IN SEGMENTS 30 AND BEYOND. THIS SECTION OF THE FILE IS COMPRISED OF BOTH LINKED SCRAMBLE BLOCKS AND INDIVIDUAL USERS SECURITY RECORDS. WORD 29 OF EACH SECURITY RECORD CONTAINS A BACK LINK TO THE SCRAMBLE BLOCK WHICH POINTS TO THAT RECORD. THE [33:15] FIELD OF WORD 29 POINTS TO THE RELATIVE DISK ADDRESS OF THE SCRAMBLE BLOCK, AND THE [18:15] FIELD POINTS TO THE WORD LOCATION OF THE USERCODE WITHIN THE SCRAMBLE BLOCK.

THE SIGN BIT OF WORD 29 OF EACH OF THESE SEGMENTS IS USED TO INDICATE WHETHER THE SEGMENT IS A LINKED SCRAMBLE BLOCK, OR A USERS SECURITY RECORD. THE SIGN BIT IS ON WHEN THE SEGMENT IS A SCRAMBLE BLOCK, AND IT IS OFF WHEN THE SEGMENT IS A USERS SECURITY RECORD.

HARD/CANDE

WHEN A COPY TO <HARDWARE> COMMAND IS ENTERED BY A USER, AN ADDITIONAL COPY OF THE FILE IS PLACED ON DISK AND IDENTIFIED BY PLACING EITHER A 2, 3 OR 4 (FOR FILES TO BE PRINTED, PUNCHED OR PLACED ON TAPE) IN FRONT OF THE FILE NAME. THESE FILES CAN BE PROCESSED BY THE SYSTEM OPERATOR USING THE PROGRAM HARD/CANDE.

> NOTE: THE COPY/CANDE PROGRAM CONTAINS A VARIABLE CALLED "PRINTSET" WHICH CONTROLS THE ACTION TAKEN ON A USERS "COPY TO PRINTER" COMMAND. IF PRINTSET IS DEFINED TO BE TRUE, THE FILE WILL BE PRINTED DIRECTLY. IF PRINTSET IS DEFINED TO BE FALSE, THE FILE WILL BE COPIED ON DISK TO AWAIT PROCESSING BY HARD/CANDE.

THE PROGRAM HARD/CANDE IS INITIATED BY AN EXECUTE CARD. IF THERE ARE NO FILES TO COPY, THE PROGRAM NOTIFIES THE OPERATOR AND TERMINATES. IF THERE ARE FILES, THE PROGRAM ASKS:

    FILE NAMES?

AND THE OPERATOR ENTERS EITHER YES OR NO VIA AN AX MESSAGE, DEPENDING UPON WHETHER OR NOT HE WISHES A LIST OF THE NAMES OF ALL FILES WAITING TO BE COPIED.

THE PROGRAM WILL THEN ASK:

    WHATS NEXT?

THE ACCEPTABLE RESPONSES ARE:

    1)   ALL
    2)   STOP
    3)   <HARDWARE TYPE>
    4)   <USERCODE>
    5)   <FILE NAME>/<USERCODE>
    6)   <HARDWARE TYPE><USERCODE>
    7)   <HARDWARE TYPE><FILE NAME>/<USERCODE>

THE ACTIONS THEN TAKEN BY THE PROGRAM ARE:

    1)   COPY ALL FILES
    2)   TERMINATE
    3)   COPY ALL FILES DESIGNATED FOR THE SPECIFIED UNIT
    4)   COPY ALL FILES FOR THIS USER
    5)   COPY THE DESIGNATED FILE (MAY BE TO MORE THAN ONE UNIT)
    6)   COPY ALL FILES BELONGING TO THE USER WHICH ARE DESIGNATED
          FOR THE SPECIFIED UNIT.
    7)   COPY THE FILE

THE ACCEPTABLE ENTRIES FOR <HARDWARE TYPE> ARE PRINTER, PUNCH OR TAPE. WHEN FILES ARE COPIED TO TAPE, THEY ARE BLOCKED (56,10). ALL

THE FILES FOR A GIVEN USER ARE PLACED ON ONE TAPE USING THE USERCODE AS THE MULTI-FILE IDENTIFICATION AND THE FILE NAME AS THE FILE IDENTIFICATION. THUS, A FILE IDENTIFIED ON DISK AS <FILE-NAME>/<USERCODE> WILL APPEAR ON TAPE AS <USERCODE>/<FILE-NAME>. WHEN A FILE IS PUNCHED, IT IS PRECEDED BY A BLANK CARD, A CARD IDENTIFYING THE FILE AND ANOTHER BLANK CARD.

AFTER A FILE IS COPIED, THE ADDITIONAL COPY ON DISK IS REMOVED. THE PROGRAM WILL CONTINUE TO ASK WHATS NEXT UNTIL TOLD TO STOP OR UNTIL ALL FILES ARE COPIED.

NOTE: THE OPTION "DSKTOG", OPTION 28, MUST BE RESET DURING EXECUTION OF HARD/CANDE, SINCE THE PROGRAM ACCESSES THE DISK DIRECTORY.

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, IO, AND 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 DCMCP IN THAT LIBRARY MAINTENANCE CANNOT BE PERFORMED ON IT. IF LOG/DISK IS NOT DECLARED IN THE COLD START DECK, THE TSSMCP WILL CREATE ONE WITH ONE ROW OF 300 SEGMENTS. 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 EXTRANEOUS 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	STATION NUMBER OF THE REMOTE USER IF THE REMOTE-LOCAL BIT IS EQUAL TO 1. OTHERWISE THIS FIELD IS IGNORED.
16:7	TYPE OF MESSAGE.
23:1	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 A NEW 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 TYPED IS:

LOG 95% FULL (AUTO LN)

MESSAGES APPEARING IN THE LOG

TYPE   CONTENTS
----   --------
0   A MESSAGE NOT APPLICABLE TO LOGGING
1   A MESSAGE TYPED IN FROM THE SPO
2   BOJ MESSAGE
3   EOJ MESSAGE
4   PBEOJ 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 IN
        FRONT OF A EOJ MESSAGE AND PROVIDES TIMING INFORMATION FOR
        THAT JOB.
9   FILE CLOSE STATISTICS, A BINARY RECORD WHICH ALWAYS
        APPEARS BEHIND A FILE CLOSE MESSAGE AND PROVIDES
        INFORMATION CONCERNING THAT FILE.
10   ON MESSAGE, INDICATES A SUCCESSFUL LOG-IN.
11   OFF MESSAGE.
12   CARRIAGE SIZE MESSAGE. (OUTPUT AFTER A CC COMMAND.)
13   DISK CHARGES MESSAGE. (OUTPUT WHEN A FILE IS REMOVED.)
14   DATE MESSAGE
15   TIME MESSAGE
16   CONTROL CARD FOR REMOTE OR LOCAL JOB.
17   HARDWARE MESSAGE.
19   STATISTICS

THE FORMATS OF THE VARIOUS MESSAGES ARE LISTED BELOW. THE
DEFINITIONS USED IN THE B 5500 OPERATION MANUAL ARE USED HERE, WITH
THE FOLLOWING CHANGES AND 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.
<PROGRAM SPECIFIER>   FOR CANDE JOBS IS GIVEN AS
                        <PROGRAM IDENTIFIER>/<FILE NAME> WHERE
                        <FILE NAME> IS THE FILE BEING MANIPULATED BY
                        THE CANDE PROGRAM.

THE BOJ MESSAGE

THE BOJ MESSAGE HAS THREE FORMS:

<PROGRAM SPECIFIER>/<USERCODE>=<MIX INDEX>BOJ<TIME OF DAY>
<STATION NUMBER>:<PROGRAM SPECIFIER>=<MIX INDEX>
     BOJ<TIME OF DAY>
<STATION NUMBER>:<PROGRAM SPECIFIER>/<USERCODE>=<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 <USERCODE>, I.E., USER PROGRAMS. THE THIRD FORM IS USED FOR OTHER REMOTE JOBS SUCH AS CANDE PROGRAMS.

THE EOJ MESSAGE

THE EOJ MESSAGE HAS TWO FORMS:

```
<PROGRAM SPECIFIER>=<MIX INDEX>,PST=<TIME>EOJ
<PROGRAM SPECIFIER>/<USERCODE>=<MIX INDEX>,PST=<TIME>EOJ
```

THE FIRST FORM IS USED FOR REMOTE JOBS WHERE THE <PROGRAM IDENTIFIER SUFFIX> IS EQUAL TO THE <USERCODE>. 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> EOJ

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 DCMCP.

FILE CLOSE MESSAGE

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

IDENTICAL TO DCMCP MESSAGE.

HALT/LOAD MESSAGE

-H/L WITH <TSSMCP NAME> MARK <PATCH LEVEL>, F=<FENCE LOCATION>
    [MODS=<MEMORY MASK>]-

EOJ 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 DCMCP.

ON MESSAGE

<USERCODE>ON<STATION NUMBER>
<USERCODE>ON<STATION NUMBER> (<CHARGE CODE>)

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

OFF MESSAGE

<USERCODE>OFF<STATION NUMBER> (<C&E TIME IN 10THS OF A SECOND>)

CARRIAGE SIZE MESSAGE

SHORT CARRIAGE FOR <USERCODE>ON<STATION NUMBER>
LONG CARRIAGE FOR <USERCODE>ON<STATION NUMBER>

DISK CHARGES MESSAGE

<FILE SPECIFIER>/<USERCODE>=<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>

CONTROL CARDS

FOR A JOB INITIATED FROM THE SITE, THIS IS THE EXECUTE OR COMPILE CARD.

FOR REMOTE COMPILES, THE MCP BUILDS A CONTROL CARD OF THE FORM:

    ?COMPILE <MULTI-FILE ID>/<USERCODE> <COMPILER NAME> LIBRARY

FOR REMOTE EXECUTES AND FOR THE "GO" PART OF ALL COMPILE-AND-GO

JOBS, THE MCP BUILDS A CONTROL CARD OF THE FORM:

    ?USER=<USERCODE>; EXECUTE <JOB SPECIFIER>

HARDWARE MESSAGES

ALL SPO OUTPUT MESSAGES CONCERNING HARDWARE FAILURES ARE ENTERED AS
TYPE 17 TO MAKE IT EASY TO DISTINGUISH THEM FROM TYPE 0 MESSAGES.

HANDLING OF INPUT FROM REMOTE TERMINALS

INPUT TO THE SYSTEM IS HANDLED IN RECORDS. IN INPUT FROM A TELETYPE, THE GROUP MARK SERVES AS THE END OF RECORD CHARACTER. IN INPUT FROM A B9352, A TC500, OR A B9353 UNIT THE END OF RECORD CHARACTERS ARE ETX AND THE ASCII CARRIAGE RETURN. THIS MAKES POSSIBLE THE TRANSMISSON OF MORE THAN ONE RECORD IN A BLOCK.

RECORDS CAN BE ANY LENGTH WITHOUT AFFECTING THE SYSTEM. HOWEVER, IF THE INPUT IS GOING TO CANDE, ALL CHARACTERS AFTER THE 224TH CHARACTER ARE IGNORED. THAT IS, IF 224 CHARACTERS ARE RECEIVED WITHOUT A RECORD SEPARATOR, THEY ARE SENT TO CANDE AND THE REMAINING CHARACTERS ARE SIMPLY DISCARDED. ON THE OTHER HAND, INPUT TO AN OBJECT JOB IS ACCUMULATED UNTIL THE SPACE IN THE TANK IS EXHAUSTED. RECORDS MAY THEREFORE BE SEVERAL THOUSAND CHARACTERS LONG.

THE REASON FOR THE DIFFERENCE BETWEEN CANDE AND OBJECT JOB INPUT IS TWO-FOLD. FIRST, CANDE CANNOT HANDLE RECORDS OVER 224 CHARACTERS LONG. SECOND, THE TANKING IS DIFFERENT. WHEREAS EACH OBJECT JOB HAS ITS OWN TANK INTO WHICH DATA CAN BE PLACED AS IT COMES IN, THERE IS ONLY ONE TANK FOR CANDE AND THE INFORMATION IN IT MUST BE ALREADY BROKEN INTO RECORDS. THE MCP THEREFORE ACCUMULATES AN ENTIRE TRANSMISSION IN CORE, THEN SCANS THROUGH IT TO BREAK IT INTO RECORDS FOR CANDE. THE EXTRA CODE REQUIRED TO LOOK FOR SIGNIFICANT DATA BEYOND THE 224TH CHARACTER (E.G., 120 BACK SPACES) USES MORE SPACE THAN IT IS WORTH.

SINCE THE MCP ACCUMULATES AN ENTIRE TRANSMISSION FOR CANDE BEFORE PROCESSING IT, THE RECORDS IN THAT TRANSMISSION ARE GIVEN TO CANDE AT THE SAME TIME, QUEUED ONE BEHIND THE OTHER. THIS MEANS THAT GREAT CARE MUST BE TAKEN WITH THOSE COMMANDS THAT CAUSE A JOB TO BE RUN. CANDE WILL SAVE ONLY THREE RECORDS BEYOND THE ONE IT IS CURRENTLY WORKING ON.

CONSIDER THE FOLLOWING EXAMPLE (<CR> IS A CARRIAGE RETURN AND <ETX> IS AN ETX):

```
LOAD X <CR>
MERGE Y <CR>
MERGE Z <CR>
APPEND Q <CR>
LIST <CR>
SAVE <ETX>
```

WOULD CAUSE CANDE TO RESPOND WITH TWO "NOT DONE" MESSAGES, ONE FOR THE "LIST" AND ONE FOR THE "SAVE". HOWEVER, IF THIS WERE ENTERED AS:

```
LOAD X; MERGE Y; MERGE Z <CR>
APPEND Q <CR>
LIST <CR>
```

```
     SAVE <ETX>
```

EVERYTHING WOULD GET DONE SINCE CANDE WOULD HAVE TO STORE ONLY THREE
RECORDS WHILE IT WAS PROCESSING THE "LOAD".

THE THINGS WHICH CAN BE PROCESSED FAST ENOUGH TO AVOID "NOT DONE"
PROBLEMS ARE DATA AND THE "WHATS", "SS", "TO", "TYPE", AND "SEQ"
COMMANDS. THUS, THE FOLLOWING WOULD NOT CAUSE ANY PROBLEMS.

```
     MAKE X <CR>
     SEQ <CR>
     A <CR>
     B <CR>
     C <CR>
     D <CR>
     E <CR>
     <CR>
     SAVE <ETX>
```

NOTE THAT SINCE CANDE CAN PROCESS THE "SEQ" COMMAND IMMEDIATELY IT
IS NOT NECESSARY TO WAIT FOR THE SEQUENCE NUMBERS TO BE OUTPUT
BEFORE ENTERING DATA. OF COURSE, THIS THROWS THE DATA OUT OF SYNC
WITH THE NUMBERS OUTPUT AFTER THE END OF THE TRANSMISSION. IN THIS
CASE THAT DOES NOT MATTER BECAUSE AUTOMATIC SEQUENCING WAS
TERMINATED BY ENTERING A NULL RECORD, I.E., ONLY A CARRIAGE RETURN.

THE VERBS WHICH HAVE BEEN MODIFIED FOR NON-TELETYPE DEVICES ARE:

1. TAPE   - ILLEGAL ON A B9352.
2. CC   - ILLEGAL ON ANY DEVICE OTHER THAN A TELETYPE.
3. SEQ   - ILLEGAL ON THE TC500 SINCE THE TC500 HANDLES IT
           INTERNALLY.
           - MODIFIED ON THE B9352 SO THAT TEN SEQUENCE NUMBERS
           ARE OUTPUT AT A TIME, SURROUNDED BY "SI" AND "SO"
           CHARACTERS. THIS CAUSES THE CURSOR TO AUTOMATICALLY
           POSITION ITSELF BEHIND EACH SEQUENCE NUMBER. WHEN
           CANDE RECEIVES 10 SEQUENCE NUMBERS, IT IMMEDIATELY
           WRITES THE NEXT 10 SEQUENCE NUMBERS. AUTOMATIC
           SEQUENCING IS ENDED BY A NULL RECORD, I.E., A
           CARRIAGE RETURN OR ETX IN THE FIRST POSITION AFTER A
           SEQUENCE NUMBER.

WHEN THE INPUT IS TO A USER PROGRAM, EACH READ ACCESSES A SEPARATE
RECORD. FOR INSTANCE, IF THE BUFFER FOR A PROGRAM WAS 10 WORDS
LONG, BUT THE RECORD WAS 100 CHARACTERS LONG, THE FIRST 80
CHARACTERS WOULD BE GIVEN TO THE PROGRAM AND THE REMAINING 20
DISCARDED. HOWEVER, IF THE BUFFER WAS 120 CHARACTERS LONG, ALL 100
CHARACTERS WOULD BE PASSED TO THE PROGRAM FOLLOWED BY 20 BLANKS TO
FILL OUT THE BUFFER. THE ONLY RESTRICTION ON THE LENGTH OF A RECORD
IS THE SIZE OF THE INPUT TANK.

PAPER TAPE INPUT

PAPER TAPE IS READ DIRECTLY BY OBJECT PROGRAMS, USING THE SAME
"READ" STATEMENTS USED FOR ORDINARY REMOTE I/O. ORDINARY INPUT CAN
BE ENTERED BEFORE AND/OR AFTER THE TAPE AND MORE THAN ONE TAPE CAN
BE READ DURING AN EXECUTION. A PROGRAM IS SIMPLY PASSED RECORDS IN
THE ORDER IN WHICH THEY ARE RECEIVED, WHETHER THEY ARE FROM PAPER
TAPE OR NOT. NO SPECIAL HANDLING OF ANY KIND IS REQUIRED OF THE
OBJECT JOB.

THE THINGS REQUIRED OF THE REMOTE USER ARE ALMOST AS SIMPLE. IN
ORDER TO INITIATE A PAPER TAPE WHILE A PROGRAM IS RUNNING HE SIMPLY
TYPES "?TAPE" AND THE SYSTEM RESPONDS WITH "OK" AND SENDS AN "X-ON"
CHARACTER. WHEN THE TAPE IS FINISHED, THE USER MUST ENTER A RECORD
WHICH STARTS WITH A QUESTION MARK. THIS RECORD IS TANKED AND PASSED
TO THE PROGRAM, MARKING THE END OF THE TAPE. IT ALSO TAKES THE MCP
OUT OF TAPE MODE.

WHEN THE MCP IS IN TAPE MODE FOR A DATACOMM LINE, ALL OUTPUT IS
DISCARDED. FURTHERMORE, INPUT IS PACKED IN THE TANK INSTEAD OF
WRITTEN ONE MESSAGE PER SEGMENT. ALSO, SINCE THE PROGRAM CANNOT
RESPOND INTERACTIVELY TO THE USER, IT IS NOT SWAPPED IN UNTIL THE
TANK IS ONE-HALF FULL OR THE TAPE ENDS. THIS ALLOWS IT TO READ MORE
THAN ONE RECORD DURING ITS TIME SLICE AND THEREBY RUN MORE
EFFICIENTLY.

PAPER TAPE IS READ INTO THE WORKFILE USING THE "TAPE" COMMAND (NO
QUESTION MARK). CANDE THEN RUNS THE "PAPER/CANDE" PROGRAM. WHICH
READS RECORDS FROM TAPE AND MERGES THEM INTO THE WORK FILE.

USE OF TELETYPES WITH TIME SHARING

THE USE OF TELETYPES AND TELETYPE-COMPATIBLE DEVICES IS DESCRIBED IN
THE "B5700 TIME SHARING TERMINAL USERS GUIDE", FORM NO. 1055944.

USE OF THE B9352 WITH TIME SHARING

TRANSMISSION OF DATA FROM THE B9352

THE MANNER IN WHICH DATA IS TRANSMITTED FROM A B9352 OR TC500 IS DIFFERENT FROM THAT OF A TELETYPE. THE LATTER SENDS EACH CHARACTER DOWN THE LINE AS IT IS TYPED. THE B9352, HOWEVER, STORES CHARACTERS AS THEY ARE TYPED AND DISPLAYS THEM ON THE SCREEN. ONLY WHEN THE USER TRANSMITS THIS DATA ARE CHARACTERS SENT OUT. THIS ALLOWS THE USER TO TOTALLY COMPOSE HIS OUTPUT AND LOOK AT IT BEFORE SENDING IT TO THE SYSTEM.

DATA IS ENTERED BY PUTTING THE DEVICE IN LOCAL (USING THE "LOC" KEY) AND POSITIONING THE CURSOR AT THE DESIRED STARTING PLACE USING THE VARIOUS CURSOR CONTROL KEYS (I.E., HOME, CLEAR, TAB, OR ONE OF THE FOUR ARROWS). CHARACTERS ARE THEN TYPED IN AS ON A TELETYPE. SINCE THE B9352 IS STORING ALL CHARACTERS TYPED, IT IS POSSIBLE AND HIGHLY USEFUL TO ENTER MORE THAN ONE LINE ON THE SCREEN. THE END OF A LINE IS SIGNIFIED BY THE CARRIAGE RETURN KEY WHICH DOES A LINE FEED AS WELL AS A CARRIAGE RETURN ON THE SCREEN. THIS CHARACTER REPLACES THE LEFT ARROW (GROUP MARK) KEY USED ON TELETYPES AS THE END OF LINE DELIMITER. IF ONLY ONE LINE IS DESIRED OR YOU HAVE TYPED THE LAST LINE YOU WISH TO TRANSMIT, USE THE END OF TEXT (ETX) CHARACTER TO TERMINATE THIS LINE. THIS CHARACTER SIGNIFIES THE END OF THIS TRANSMISSION. IT DOES AN AUTOMATIC HOME TO THE TOP OF THE SCREEN.

AFTER THE USER HAS CHECKED HIS DATA AND DECIDES HE WISHES TO TRANSMIT IT, HE POSITIONS THE CURSOR AT THE FIRST CHARACTER HE TYPED AND THEN PRESSES THE TRANSMIT (XMIT) KEY. THIS WILL SEND OUT ALL THE DATA TO THE SYSTEM AND THEN RETURN TO THE RECEIVE STATE -- READY TO ACCEPT DATA FROM THE SYSTEM. THE CURSOR WILL BE POINTING TO THE FIRST CHARACTER POSITION AFTER THE ETX. CAUTION: IF THE CURSOR IS NOT POSITIONED CORRECTLY BEFORE PRESSING TRANSMIT, INCORRECT DATA WILL UNDOUBTEDLY BE SENT. ALL OF THE CHARACTERS IN THE BCL SET MAY BE USED WITH THE B9352 EXCEPT THE LEFT ARROW, WHICH IS DISCARDED BY THE ADAPTERS.

RECEIVING OF DATA BY THE B9352

THE B9352 CAN RECEIVE CHARACTERS ONLY WHEN IT IS IN RECEIVE STATE. THIS STATE IS ENTERED BY USING THE RECEIVE KEY (REC) OR AUTOMATICALLY AT THE COMPLETION OF THE TRANSMISSION PROCESS. CHARACTERS ARE ENTERED STARTING AT WHERE THE CURSOR IS PRESENTLY POINTING AND CONTINUES FROM THERE. AS ON OUTPUT THE CARRIAGE RETURN IS USED TO SEPARATE LINES. THE ETX SIGNIFIES THE END OF TRANSMISSON AND THE CURSOR WILL POINT TO THE CHARACTER POSITION AFTER THE ETX. IF THE S202 TOGGLE SWITCH ON THE BACK OF THE B9352 IS PLACED IN THE "DOWN" STATE (THE SUGGESTED MODE OF OPERATION) THE B9352 WILL REMAIN IN THE RECEIVE STATE; OTHERWISE IT WILL RETURN TO THE LOCAL STATE.

SINCE THE CHARACTER WRITE SPEED IS EXTREMELY FAST AND THERE ARE A
LIMITED NUMBER OF LINES AND CHARACTER POSITIONS ON THE SCREEN, IF
THE TIME SHARING MCP WERE TO INDISCRIMINATELY SEND OUT DATA AS FAST
AS IT COULD, LINES WOULD BE STEPPED ON BY THE WRAP AROUND OF LATER
INPUT BEFORE THE USER HAD A CHANCE TO READ THEM ALL. THEREFORE,
TIME SHARING USES A PAGING SCHEME TO COUNT LINES AND STOP SENDING
DATA WHEN THE SCREEN IS FULL. THE USER CAN THEN LEISURELY READ THE
OUTPUT. HE SIGNIFIES THAT HE IS READY TO RECEIVE MORE DATA BY DOING
ANY TRANSMISSION OF DATA. THE MCP WILL THEN SEND WHATEVER ELSE IT
HAS (UP TO ANOTHER FULL PAGE) AND CONTINUE.

THE MCP ASSUMES THAT THE DEVICE IS AT THE TOP OF THE PAGE(I.E., THE
SCREEN IS EFFECTIVELY EMPTY) AT HALT/LOAD TIME, AFTER TRANSMISSION
OF DATA, OR WHEN IT DISCOVERS THAT THE DEVICE IS IN LOCAL WHEN IT IS
ATTEMPTING TO WRITE. THE INITIAL TRANSMISSION WILL START WITH A
CARRIAGE RETURN AND A LINE FEED AND THEN CONTINUE ON. IF THERE IS
ANOTHER TRANSMISSION AND THE DEVICE IS NOT AT THE TOP OF THE PAGE A
BACKSPACE WILL BE DONE TO ERASE THE <ETX> CHARACTER FROM THE
PREVIOUS DATA. WHEN THE COUNT OF OUTPUT LINES BECOMES GREATER THAN
THE TOTAL LINES (SPECIFIED FOR THE SCREEN IN SYSTEM/DISK) LESS 2,
THE PAGE IS CONSIDERED FULL. A HOME TO THE TOP OF THE SCREEN IS
DONE, THE <ETX> IS SENT, THE CURSOR POINTS TO THE SECOND CHARACTER
POSITION AND NO MORE OUTPUT WILL BE TRANSMITTED UNTIL THE USER
TRANSMITS DATA TO SIGNIFY HE IS READY FOR MORE OUTPUT.

THE BREAK OF OUTPUT THAT IS DONE ON A TELETYPE WITH THE BREAK KEY IS
ACCOMPLISHED BY TRANSMITTING "?BREAK" TO THE SYSTEM. SIMILARLY A
JOB CAN BE DS-ED (AS ON A TELETYPE WHERE CONTROL-WRU KEY IS TYPED)
BY TRANSMITTING "?WRU". A BREAK CAN ALSO BE DONE BY PUTTING THE
DEVICE IN LOCAL WHILE IT IS RECEIVING DATA.

DIFFERENCES IN THE OPERATION OF CANDE WITH THE B9352

   A. SEQUENCING - SEQUENCING IS HANDLED DIFFERENTLY FOR THE B9352 IN
       AN ATTEMPT TO USE THE ABILITY TO TRANSMIT MORE THAN ONE LINE AT
       ONCE. WHEN SEQUENCE MODE IS ENTERED CANDE WILL TRANSMIT A
       SCREEN CLEAR, A HOME, AND THE NEXT 10 SEQUENCE NUMBERS, ONE TO
       A LINE. EACH SEQUENCE NUMBER IS WITHIN SHIFT OUT AND SHIFT IN
       CHARACTERS. THIS TAKES ADVANTAGE OF THE FORMS MODE CAPABILITY.
       UNLESS THE DEVICE IS PUT INTO FORMS MODE (SHIFT FORM) THESE
       CHARACTERS WILL BE PROTECTED. THEY WILL NOT BE ERASED OR
       WRITTEN OVER AND WILL NOT BE TRANSMITTED WHEN THAT PROCESS IS
       DONE. THUS, ONE HAS A RELATIVELY PERMANENT INDICATION ON THE
       SCREEN OF THE PRESENT SEQUENCE NUMBER BEING TYPED. CANDE
       COUNTS THE NUMBER OF LINES SENT WITH EACH TRANSMISSION AND WHEN
       THE TOTAL PASSES 10 THE NEXT BLOCK OF 10 SEQUENCE NUMBERS IS
       WRITTEN OUT. THIS CONTINUES UNTIL A LINE WITH NO RECORDS
       (SINGLE CR OR ETX) IS ENTERED, AND THEN SEQUENCING IS EXITED.

   B. ?CONTINUE - SINCE TRANSMISSION OF DATA IS NECESSARY TO GET
       ANOTHER PAGE OF OUTPUT WHEN THE SCREEN IS FULL, AN INPUT WAS

CREATED THAT WOULD NOT CAUSE ANY ACTION TO EITHER AN OBJECT JOB OR CANDE. "?CONTINUE" IS THAT INPUT. IT WILL BE EFFECTIVELY DISCARDED IF TRANSMITTED.

THE NORMAL MODE OF OPERATION ONE WOULD USE IF USING THE DEVICE WHEN LARGE AMOUNTS OF OUTPUT IS EXPECTED IS TO PUT ?CONTINUE <ETX> ON THE SCREEN, STARTING IN THE SECOND CHARACTER POSITION IN THE TOP LINE. WITH THE HOME DONE BY THE MCP WHEN A PAGE IS FULL, THE USER NEED ONLY TO PRESS XMIT TO RECEIVE MORE OUTPUT AND NOT HAVE TO POSITION THE CURSOR.

C. DISPLAY - SINCE THE B9352 STORES THE CHARACTERS ON THE SCREEN WHETHER THEY ARE TYPED OR TRANSMITTED BY THE MCP, A USEFUL AND UNIQUE METHOD OF FILE MAINTENANCE CAN BE USED. THIS INVOLVES PRINTING OUT OF INFORMATION ON THE SCREEN, PARTIALLY MODIFYING IT, AND RETRANSMITTING IT BACK AS INPUT. IN ORDER TO GET THE MOST USEFUL CARD IMAGE THE "DISPLAY" COMMAND IS USED INSTEAD OF PRINT OF LIST. THIS ELIMINATES THE BLANK AFTER THE SEQUENCE NUMBER (WHICH WOULD BE AN EXTRA CHARACTER) AND DISPLAYS THE SEQUENCE NUMBER WITH 8 DIGITS (INCLUDING LEADING ZEROS) IN ORDER TO AVOID THE CONFLICT OF THE FIRST CHARACTER IN THE LINE BEING A DIGIT. NOTE: SINCE THE LEFT ARROW IS TOTALLY DISCARDED BY THE ADAPTER IT CAN BE USED IN DISPLAY SITUATIONS AND OTHER PLACES TO DELETE A CHARACTER FROM A LINE BY TYPING IT OVER THE BAD CHARACTER.

SPECIAL FEATURES OF THE B9352

A. TABS - A TAB CAN BE SET IN ANY OF THE CHARACTER POSITIONS ACROSS THE SCREEN (AS ON A TYPEWRITER). THE USE OF THESE CAN PROVE HIGHLY DESIRABLE IN SUCH ACTIVITIES AS LINING UP CARD IMAGES FOR CANDE FILES OR FOR INPUT ON HIGHLY INTERACTIVE PROGRAMS.

TABS ARE SET BY PLACING THE CURSOR IN THE DESIRED POSITION AND SIMULTANEOUSLY PRESSING THE SHIFT KEY AND THE TAB KEY. TO SPACE OVER TO THE TAB, MERELY PRESS THE TAB KEY - IT WILL SKIP TO THE NEXT SET TAB. MORE THAN ONE TAB MAY BE USED. IF THERE ARE NO MORE TABS SET ON THE LINE AND TAB IS PRESSED, THE CURSOR WILL DROP TO THE NEXT LINE, FIRST POSITION.

TO CLEAR A TAB, PLACE THE CURSOR AT THAT POSITION AND DEPRESS TAB CLEAR. PRESSING SHIFT AND TAB CLEAR TOGETHER WILL REMOVE ALL OF THE TABS.

B. FORMS - THESE HAVE BEEN PARTIALLY DISCUSSED UNDER SEQUENCING, AS THIS OPTION IS USED BY THE SEQUENCING OPERATION. THE USE OF THIS OPTION CREATES AREAS ON THE SCREEN THAT CANNOT BE WRITTEN OVER, ERASED, OR TRANSMITTED IN NORMAL B9352 OPERATION. HENCE, THEY CAN BE USED TO PUT A FIXED FORMAT AND DESCRIPTION ON THE SCREEN TO ALLOW THE UNTRAINED TO SIMPLY ENTER DATA.

TO USE THIS OPTION, THE DEVICE IS PUT IN FORMS MODE BY PRESSING
SHIFT AND FORM KEYS. THE LOCAL LIGHT SHOULD BE BLINKING IF IT
IS IN FORMS MODE. TO CREATE A PROTECTED AREA, THE CURSOR IS
PLACED IN THE DESIRED PLACE AND THE "RS" KEY IS PRESSED. THE
DESIRED DATA IS TYPED IN AND THEN THE "US" KEY SHOULD BE
DEPRESSED. (THESE TWO KEYS WILL ENTER THEIR CHARACTERS ONLY
WHILE IN FORMS MODE.) AS MANY PROTECTED AREAS AS ARE DESIRED
MAY BE PLACED ON THE SCREEN. TO LEAVE THE FORMS MODE, THE
"FORM" KEY IS PRESSED. THE "LOCAL" LIGHT SHOULD STOP BLINKING.
IN NORMAL MODE, IF THE CURSOR IS POSITIONED TO ENTER ANY AREA
BETWEEN (AND INCLUDING) THE "RS" AND THE "US", THE CURSOR WILL
SKIP AROUND THE ENTIRE AREA. HENCE, IT CANNOT BE ENTERED,
ERASED, OR WRITTEN OVER, AND IS THUS PROTECTED.

NOTE: IT IS POSSIBLE, WHILE IN FORMS MODE, TO SET UP RS"S AND
US"S SUCH THAT THE WHOLE SCREEN IS PROTECTED AND OFF LIMITS TO
THE CURSOR. WHEN THE B9352 IS RETURNED TO NORMAL MODE, THE
RATHER HUMOROUS SITUATION ARISES WHERE THE CURSOR IS
CONTINUALLY MOVING IN AN ATTEMPT TO FIND A PLACE TO STAY. IF
THIS HAPPENS, PLACE THE DEVICE BACK IN FORMS MODE AND RECOMPOSE
THE SCREEN SO THAT THIS DOESNT HAPPEN.

FORMS CAN BE WRITTEN OVER AND ERASED ONLY WHEN THE DEVICE IS IN
FORMS MODE.

IF THE DEVICE ATTEMPTS TO TRANSMIT DATA WHILE IN NORMAL MODE,
IT WILL NOT TRANSMIT ANY OF THE PROTECTED AREAS. THESE CAN BE
TRANSMITTED BY PLACING THE DEVICE IN FORMS MODE AND THEN
PRESSING XMIT. THIS IS, IN GENERAL, NOT RECOMMENDED.

WHEN THE DEVICE IS IN RECEIVE STATE, IT IS IN FORMS MODE (IN
ORDER TO ALLOW THE WRITING OF FORMS DATA TO IT). HENCE, IT IS
POSSIBLE TO HAVE PROTECTED AREAS WRITTEN OR ERASED WHEN IN
RECEIVE STATE.

C. MISCELLANEOUS

1.	CLEAR	THE SCREEN IS CLEARED AND THE CURSOR HOMED IF CLEAR AND ERASE LOCK ARE SIMULTANEOUSLY DEPRESSED. IF IN NORMAL MODE, FORMS WILL NOT BE ERASED. THEY WILL BE ERASED IF IN FORMS MODE. THIS FEATURE WILL WORK IN RECEIVE STATE, AND WILL ERASE FORMS IF IN RECEIVE STATE.
2.	LINE ERASE	PRESSING LINE ERASE AND ERASE LOCK WILL ERASE ALL CHARACTERS FROM THE CURSOR TO THE END OF THE LINE ON THAT LINE ONLY.
3.	THE FOUR ARROWS	THESE CHARACTERS WILL MOVE THE CURSOR IN THE DIRECTION POINTED WITHOUT CHANGING ANY OF THE CHARACTERS ON THE SCREEN.
4.	HOME	PLACES THE CURSOR AT THE TOP LEFT CORNER OF THE SCREEN.
5.	REPT	DEPRESSING THIS KEY AND HOLDING DOWN A

CHARACTER (OR CURSOR CONTROL) WILL REPEAT
THE CHARACTER UNTIL EITHER KEY IS RELEASED.
6. ERROR RESET   IF THE RED ERROR LIGHT COMES ON, THIS KEY
SHOULD BE PRESSED TO TURN IT OFF.

D. PAPER TAPE - THERE IS NO PAPER TAPE CAPABILITY WITH THE B9352.
USE OF TAPE COMMANDS WILL RESULT IN AN ERROR.

DIAL UP PROCEDURE

THE OPTIONS IN THE B9352 AND 992 ADAPTER DETERMINE HOW THE DIAL UP
PROCEDURE IS DONE. IF THE B9352 HAS THE POINT TO POINT CARD, IS
RUNNING THROUGH A DATA SET, AND THE ADAPTER IS PLUGGED FOR POINT TO
POINT AND CALL UP AND DISCONNECT, THEN ONE NEED ONLY DIAL UP THE
APPROPRIATE NUMBER FOR THE ADAPTER. THE SYSTEM RECEIVES A "DIAL-UP"
INTERRUPT, AND THE LOGGING ON PROCESS IS INITIATED.

ANY OTHER CONFIGURATIONS WILL NOT CAUSE A "DIALED-UP" INTERRUPT AND
HENCE, ARE CONSIDERED BY THE MCP TO BE DIRECT LINES. THESE ALL
REQUIRE THE TRANSMISSION OF DATA TO "DIAL IN" TO THE SYSTEM.

WITH THE POINT TO POINT OPTION IN THE B9352, PRESSING THE XMIT KEY
SHOULD CAUSE IMMEDIATE TRANSMISSION OF THE APPROPRIATE DATA AND
ACCOMPLISH THE "DIAL-IN". HOWEVER, SOMETIMES THE B9352 WILL BE TIED
UP AFTER JUST TURNING ON ITS POWER AND THE TRANSMIT WON"T WORK. THE
THE OPERATOR CAN THEN BE ASKED TO CLEAR (CL) YOUR LINE AND YOU WILL
THEN BE ABLE TO "DIAL IN" BY PRESSING XMIT AGAIN.

WITH THE MULTI-POINT (POLL AND SELECT) OPTION IN THE B9352 ,
OPERATIONS ARE DIFFERENT. AFTER XMIT IS PRESSED, A DELAY OF UP TO A
MAXIMUM OF A PREDETERMINED TIME SET IN THE MCP (CURRENTLY 30
SECONDS) WILL BE NOTICED. THIS WAIT IS THE TIME BETWEEN POLLS TO
YOUR PREVIOUSLY INACTIVE STATION.

IT IS SUGGESTED THAT THE INITIAL TRANSMISSION USED TO EFFECT THE
DIAL-UP BE THE CHARACTERS "?WRU". THIS WILL AVOID CONFUSION AS TO
WHICH WORD IS TO BE INTERPRETED AS THE USER CODE IN SUBSEQUENT
TRANSMISSIONS, AND WILL ALLOW THE LOG-ON PROCESS TO PROCEED SMOOTHLY.

IF THERE IS ANY TROUBLE IN THE DIAL-IN PROCESS THE OPERATOR CAN BE
REQUESTED TO CLEAR YOUR LINE, AND BY TRANSMITTING DATA THE DIAL-IN
SHOULD BE IMMEDIATELY ACCOMPLISHED.

FURTHER INFORMATION

THE B9352 INPUT AND DISPLAY TERMINAL REFERENCE INFORMATION MANUAL
CAN BE REFERRED TO FOR MORE EXPLANATION OF THE B9352 DISPLAY.

USE OF THE TC500 WITH TIME SHARING

A TC500 PROGRAM WHICH WILL ALLOW THE TC500 TO ACT AS A TIME SHARING TERMINAL IS AVAILABLE THROUGH NORMAL SYSTEM SUPPORT CHANNELS. THE FOLLOWING INFORMATION MAY BE HELPFUL TO THOSE USERS WHO WRITE THEIR OWN TC500 PROGRAMS.

A. WHEN OPERATING IN A MULTIPOINT (POLLING-SELECT) ENVIRONMENT, THE TC500 SEND ADDRESS AND RECEIVE ADDRESS MUST BE SET TO THE SAME CHARACTERS. THE TSSMCP WILL NOT ALLOW THE SEND ADDRESS TO BE DIFFERENT FROM THE RECEIVE ADDRESS.

B. THE TSSMCP ACCEPTS EITHER THE USASCII CARRIAGE RETURN CHARACTER (INTERNAL CODE "OD"), OR THE ETX CHARACTER (INSERTED BY THE FIRMWARE) AS DENOTING THE END OF A RECORD. INPUT RECORDS MAY BE OF ANY LENGTH (SUBJECT TO THE LIMITATION OF THE TC500 TRANSMIT BUFFER SIZE). HOWEVER, RECORDS INPUT TO CANDE WILL BE TRUNCATED TO 224 CHARACTERS, AND RECORDS INPUT TO A WORK FILE WILL BE TRUNCATED TO 72 CHARACTERS PLUS THE SEQUENCE NUMBER (80 CHARACTERS FOR TYPE DATA FILES).

C. THE TSSMCP DOES NOT RESPOND TO A TRANSMISSION FROM THE TC500 EXCEPT TO MAKE THE APPROPRIATE LINE DISCIPLINE RESPONSES. THE TC500 PROGRAM MUST PROVIDE THE NECESSARY CARRIAGE RETURN AND LINE ADVANCE INSTRUCTIONS TO POSITION THE PRINT BALL FOR THE NEXT OPERATION.

D. THE TSSMCP TERMINATES EACH OUTPUT RECORD WITH A CARRIAGE RETURN CHARACTER AND A LINE FEED CHARACTER, AND TERMINATES EACH TRANSMISSION WITH AN ETX CHARACTER. THE TC500 PROGRAM MUST TAKE THE APPROPRIATE ACTION UPON ENCOUNTERING THESE CHARACTERS IN THE RECEIVE BUFFER.

E. RECORDS OUTPUT TO THE TC500 BY CANDE OR THE CANDE PROGRAMS WILL BE A MAXIMUM OF 80 CHARACTERS LONG. RECORDS OUTPUT BY AN OBJECT PROGRAM MAY BE OF ANY LENGTH UP TO 200 CHARACTERS. IT IS THE RESPONSIBILITY OF THE TC500 PROGRAM TO HANDLE ANY RECORDS WHICH MAY EXCEED A SINGLE LINE OF PRINT.

HARDWARE CONFIGURATION FOR EXTENDED DATACOMM

I. 992 - ADAPTER OPTIONS

A. MULTI POINT

1. LIMITED DIRECT CONNECT - SEE TIN 4124 (8/14/69).
2. 202C OR 202D - 2 WIRE DATA SETS - SEE TIN 4124 (8/14/69).
3. 202D - 4 WIRE DATA SETS - SEE TIN 4124 (8/14/69). NOTE: CONX/ SHOULD NOT BE CUT TO ALLOW FASTER TURN-AROUND TIME.
4. DIRECT CONNECT - (TWISTED PAIR)
   A) SET - MPML,MACL,202L,2WRX,CAXX,IRLX BY STRL,CONX/, FLOATING SB.
   B) DONT SET - CDIL/,MACL/,AMPL,IDOL,BPOL,CADL,D12L,COAX.
   C) IN THE TC500 PROGRAM, OPERATION SHOULD BE SPECIFIED AS 4-WIRE.

B. POINT TO POINT

1. SET - CADL,CDIL/,202L,2WRX,CONX/,FLOATING SB,IRLX BY STRL.
2. DONT SET - MPML,MACL, MACL/,AMPL,BPOL,D12L,CAXX,COAX,IDOL.
3. TIMEOUT MULTI SHOULD BE SET TO MAXIMUM DELAY.

C. BUFFER SIZE

RECOMMENDED BUFFER SIZE IS 56 X 56 PING PING. 28 X 28 PING PING COULD ALSO BE USED.

II. B9352 PHASE II DISPLAY OPTIONS

A. SCREEN SIZE

1. 60 HZ 80 X 12 (B9352-2 (1706 8982)) IS RECOMMENDED AS MOST USEABLE.
2. 60 HZ 40 X 24 (B9352-1 (1706 8974)) CAN ALSO BE USED.

B. STATION DISCIPLINE

1. BSL POLLING AND SELECT COMM. (B9352-2 (1706 9139)) CAN BE USED WITH THE MULTIPOINT LINE DISCIPLINE. THIS OPTION IS REQUIRED IF MORE THAN ONE STATION IS ONE LINE.
2. POINT TO POINT COMM. (1706 9105) CAN BE USED WITH THE POINT TO POINT LINE DISCIPLINE. THIS OPTION IS REQUIRED IF THE ADAPTER IS SET UP FOR POINT TO POINT (I.E.,CADL,ETC. -SEE I-B) AND MAY BE USED WITH THE ADAPTER IN MULTI POINT.

C. LINE SPEED CARDS

1. 1200 BPS 60 HZ (202C/D) ASYNCH (B9352-7 (1706 9139))

SHOULD BE USED WHEN OPERATING THROUGH A DATA SET (202C OR 202D).

2. 1200 BPS 60 HZ (TDI-DIR.) ASYNCH (B9352-4 (1729 3515)) SHOULD BE USED WITH THE 2 WIRE TWISTED PAIR HOOKUP OR THE LIMITED DIRECT CONNECT (DSI CABLE WITH 8" COUPLER).

D. OTHER OPTIONS

1. PRINTER ADAPTER (B9352-1 (1706 9071)) CAN BE USED TO SAVE HARD COPY THE USER MIGHT WANT. IT IS NOT AN INTEGRAL PART OF THE SYSTEM, AND IT IS TO THE USERS DISCRETION WHETHER HE NEEDS IT OR NOT.
2. CONTROLLED FORMAT (B9352-3 (1722 4551)) IS REQUIRED. IT IS USED BY CANDE FOR AUTOMATIC SEQUENCING. IT IS EXPECTED THAT THE USER WOULD FIND IT USEFUL.

III. TC-500

A. FIRMWARE

1. ANY MAIN MEMORY DATACOMM FIRMWARE MAY BE USED TO RUN WITH THE TIME SHARING SYSTEM, DEPENDING ON THE CAPABILITIES DESIRED IN ADDITION TO DATACOMM.
2. TO RUN THE TC500 IN A MULTI POINT LINE DISCIPLINE USE E-MEMORY FIRMWARE 2-1044-006-00.
3. TO RUN THE TC500 IN A POINT TO POINT LINE DISCIPLINE USE E-MEMORY FIRMWARE 2-1044-004-00.

B. STATION DISCIPLINES

1. POINT TO POINT FIRMWARE (SEE III-A3) IS REQUIRED IF THE TC500 IS TO BE RUN THROUGH A DATA SET INTO A POINT TO POINT LINE DISCIPLINE ADAPTER. IT MAY ALSO BE RUN WITH MULTI POINT LINE DISCIPLINE.
2. MULTI POINT FIRMWARE CAN BE USED WITH A MULTI POINT LINE DISCIPLINE ADAPTER. IT MUST BE USED TO PUT MORE THAN ONE STATION ON A LINE.

C. CONNECTING LINES

1. TWISTED PAIR
   A) MULTI POINT REQUIRES TDI INTERFACE CARD.
   B) POINT TO POINT FIRMWARE WILL NOT RUN WITH TWISTED PAIR AND TIMESHARING.
   C) ONLY ONE STATION MAY BE ON A TWISTED PAIR MULTI POINT LINE.
   D) THE TC500 SHOULD SPECIFY 4 WIRE OPERATION.

2. LIMITED DIRECT CONNECT - 8" INTERFACE CABLE
   A) POINT TO POINT REQUIRES DSI INTERFACE CARD.
   B) MULTI POINT
      1) CAN BE ONLY ONE STATION WHICH HAS DSI CARD.

2) CAN HAVE MORE THAN ONE STATION CONCATENATED WITH CLC CARDS, THE LAST STATION HAVING A DSI CARD. FOR A TC500 TO OPERATE, EVERY PREVIOUS ONE ON THE LINE MUST BE TURNED ON.

3. DATA SET (202C OR 202D) THIS HOOKUP CAN BE PUT IN ANY OF THE METHODS OF THE LIMITED DIRECT CONNECT (SEE III-C2 ABOVE.)

PART II
---- --

PROGRAMMING INFORMATION
------------ -----------

THE TSSMCP

THE TIME SHARING MCP (TSSMCP) IS SIMPLY A MODIFIED VERSION OF THE
STANDARD MCP (DCMCP). MUCH OF THE TSSMCP IS THEREFORE ALMOST
IDENTICAL TO THE CORRESPONDING PARTS OF THE DCMCP. 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 DCMCP 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 TSSMCP WHICH ARE NEW OR DIFFER SUBSTANTIALLY
FROM THEIR COUNTERPARTS IN THE DCMCP.

SPACER AND SEGMENTED SAVE AREAS

SEGMENTED SAVE AREAS ARE USED THROUGHOUT THE TSSMCP 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. EXCEPT FOR THE [1:3] FIELD, THE
FIRST WORD OF AN AREA IS ZEROED BEFORE ITS ADDRESS IS PASSED TO A
PROCEDURE.

DURING INITIALIZATION, SPACE IS GOTTEN AND DIVIDED INTO 20-WORD
AREAS. THE AMOUNT OF SPACE GOTTEN IS DEPENDENT ON THE DEFINE
"MAXLMAX" (AT SEQ 00004500), WHICH IS THE MAXIMUM ALLOWABLE NUMBER
OF STATIONS. SINCE THE MCP WILL ADJUST THE SIZE OF THE DATACOM
TABLES ACCORDING TO INFORMATION SUPPLIED IN SYSTEM/DISK, MAXLMAX CAN
BE LARGER THAN THE ACTUAL NUMBER OF STATIONS WITHOUT AFFECTING THE
SIZE OF THE TABLES. IT CAN, THEREFORE, BE USED TO AFFECT THE
ALGORITHM FOR INITIALLY GETTING SPACE FOR SEGMENTED SAVE AREAS. THE
NUMBER OF WORDS OBTAINED AT HALT/LOAD TIME IS:

((MAXLMAX OR 1) + 1) X 10

EXPERIENCE INDICATES THAT ABOUT 80 EXTRA WORDS (8 LINES WORTH) ARE
REQUIRED ALTHOUGH THIS CAN AND SHOULD BE ADJUSTED BY EACH SITE.

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.

TSSMCP TABLES

MANY OF THE TABLES IN THE TIME SHARING SYSTEM ARE THE SAME AS THE
CORRESPONDING TABLES IN THE DCMCP 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, STATABLE[I].
CANDEFLAG. 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].

IN ORDER TO SIMPLIFY THE TASK OF SWAPPING, TSSMCP 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 19 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:

WORD	DEFINE	CONTENTS
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 CANDE-S.)
15		NOT USED
16	NEXT1	MULTI-FILE ID OF CHAINED JOB (IF ANY)
17	NEXT2	FILE ID OF CHAINED JOB
18	OLAYCTR	TIME SPENT OVERLAYING DURING CURRENT SWAP (USED FOR DYNAMIC EXPANSION).
19-32		INCLUDED AND USED ONLY IF THE TSSMCP WAS COMPILED WITH THE STATISTICS MODULE SET TRUE.

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]

FIELD	DEFINE	CONTENTS
1:1		INTERLOCK FOR SWAPPINGIO
7:1	CANTEXPAND[I]	IF ON, JOBS CORE AREA CANNOT BE EXPANDED.
8:4	NLS[I]	NUMBER OF TIMES JOB HAS BEEN PASSED OVER FOR SWAP-IN.
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]

FIELD	TABLE	CONTENTS
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
1:1		INTERLOCK BETWEEN SWAPPING AND INTRINSICS OVERLAY.
7:1	NSDABLE	IF ON, JOB MAY NOT BE DS-ED.
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

LINETABLE

LINETABLE IS A ONE DIMENSIONAL TABLE LMAX LONG. EACH WORD IN LINETABLE CONTAINS INFORMATION ABOUT THE CORRESPONDING LINE. THE DEFINES USED TO ACCESS SOME OF THE FIELDS ARE LISTED BELOW. THOSE MARKED BY ASTERISKS ARE PARAMETRIC DEFINES AND ARE USED AS IF THEY WERE INDIVIDUAL TABLES. THE OTHERS ARE USED AS PARTIAL WORD DESIGNATORS.

FIELD	DEFINE	CONTENTS
1:1	*LOCKED	LOCK BIT, USED BY DCWAIT.
2:1	DIRECTLINE	DIRECT CONNECT FLAG - ON IF DIRECT CONNECT.
3:3	*LINEDISC	
	LINEDIS	LINE DISCIPLINE:
		0 = TELETYPE
		1 = CONTENTION
		2 = MULTIPOINT
		7 = SCHEDULE
6:2	BUFSIZE	BUFFER SIZE:
		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.
9:4		TERMINAL UNIT.
13:1		GROUPMARK FLAG - 1 IF GROUP MARKS SHOULD BE IGNORED, 0 IF GROUP MARK TERMINATES I/O.
14:4		BUFFER NUMBER.
18:2		ADAPTER TYPE:
		0 = 980
		1 = 992
20:1	*THROWAWAY	
21:5	*LSTATUS	LINE STATUS (SEE BELOW).
26:1	*LONGCARRIAGE	IF 1, SUPPRESSES LINE FOLDING FOR TELETYPES.
33:15		ADDRESS OF HEAD OF INPUT QUEUE.

LINE STATUSES

VALUE	DEFINE	MEANING
WRITING:		
0	WRB	USED FOR ALL MESSAGE WRITES.
1	POLLING	USED WHEN A POLL IS BEING WRITTEN.
2	SELECT	USED WHEN A SELECT OR ENQ MESSAGE IS WRITTEN
3	ACKING	USED TO POSITIVELY ACKNOWLEDGE A MESSAGE.
4	NAKING	USED TO NEGATIVELY ACKNOWLEDGE A MESSAGE.
5	ACKINGENQ	POSITIVE REPLY TO AN ENQ (POINT TO POINT ONLY).
6	NAKINGENQ	NEGATIVE REPLY TO AN ENQ (POINT TO POINT ONLY). THIS MESSAGE IS SENT ONLY IF THE INPUT TANK IS FULL.
7	WRBUSY	TEMPORARY STATUS USED TO PREVENT OUTPUT FROM OCCURRING.
IDLE:		
8	IDL	NORMAL IDLE STATUS - LINE IS AVAILABLE FOR OUTPUT.
9	IDLPOLLING	A RECYCLING POLL IS IN THE ADAPTER, BUT THE LINE IS AVAILABLE FOR OUTPUT (AFTER AN ACTIVE INTERROGATE).
10	WAITING	ADAPTER WENT WRITE READY BECAUSE IT RECEIVED A FULL BUFFER WITHOUT A GROUP MARK. THE NEXT BUFFER LOAD WILL BE SENT NEXT.
11	WAITINGENQ	POINT TO POINT ONLY. A LINE IS IDLE WITH EITHER SIDE ABLE TO START IT UP WITH AN ENQ MESSAGE.
·READ READY:		
16	NORMAL	READ OF A MESSAGE OTHER THAN THE FIRST BUFFER.
17	FIRSTIME	READ OF THE FIRST BUFFER OF A MESSAGE.
18	SELECTANS	READ OF THE RESPONSE TO A SELECT OR ENQ MESSAGE.
19	ENQREAD	EXPECTED MESSAGE IS ENQ (POINT TO POINT ONLY).
20	BROKEN	A BREAK OF A MESSAGE HAS OCCURRED.
21	POLLTIMEOUT	TIMEOUT FROM A POLL. EXPECT TO CALL DCTIMEOUT.
22	TIMEOUT	TIMEOUT FROM A SELECT.
23	EOTREAD	EXPECT TO READ AN EOT.
24	RRA	READ READY ABNORMAL RECEIVED FOR TELETYPE LINE. EXPECT A "?", DELETE, OR DISCONNECT.
25	MSGANSWER	RESPONSE TO A MESSAGE OR SELECT (SHOULD BE AN ACK OR NAK).

STATABLE

STATABLE IS A ONE DIMENSIONAL TABLE STATIONMAX LONG. EACH WORD IN
STATABLE CONTAINS INFORMATION ABOUT THE CORRESPONDING STATION.

FIELD	DEFINE	CONTENTS
1:1	*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).
2:1	ACTIVITY	SET BY SWAP OR DCIO, RESET BY NSECOND.
3:1	SWAPPED	ON WHENEVER JOB IS SWAPPED OUT.
4:1	CANDEFLAG	IF ON, INPUT GOES TO CANDE.
4:6	MIXFLAG	
5:5	MIXNR	MIX NUMBER OF JOB TO WHICH STATION IS ATTACHED.
10:3	STATIONTYPE	STATION TYPE: 0 = TELETYPE 1 = B9352 2 = TC500 3 = B9353
13:1	OWHTHROWOUT	HARRY IS DISCARDING INPUT FROM THIS STATION.
14:1	*BREAK	ON IF BREAK HAS OCCURRED.
15:1	DIALEDUP	ON IF THIS STATION IS ALIVE.
16:1	*DISCONNECTING	OFF IF QUITTER OR JOB IS STILL RUNNING.
17:1	*QUITN	INTERLOCK FOR QUITTER.
22:1		TEXT MODE BIT FOR PREVIOUS OUTPUT BUFFER.
23:1		TEXT MODE BIT FOR CURRENT OUTPUT BUFFER.
24:1	OUTPUTANKING	ON WHENEVER THERE IS OUTPUT IN THE TANK.
25:8	LEENKER	READY QUEUE LINK IF STATION LEQ LMAX, LINE NUMBER IF STATION GTR LMAX.
33:15		ADDRESS OF HEAD OF OUTPUT QUEUE.

SEQARRAY

SEQARRAY IS A ONE DIMENSIONAL ARRAY STATIONMAX LONG. ITS CONTENTS
FOR EACH STATION DEPEND ON THE LINE DISCIPLINE.

FIELD	DEFINE	CONTENTS

TELETYPES:
1:1		ON IF IN SEQUENCING MODE.
2:19		INCREMENT.
21:27		CURRENT SEQENCE NUMBER.

MULTIPOINT OR CONTENTION:
1:1	SELECTED	SELECTED OR ENQ-ED.
2:1	*TANKOK	ON IF FULL INPUTANK.
3:1		STATION IS NOT IN POLLING LIST.
4:2		TIME OUT COUNT.
6:3	NAKMAX	MAXIMUM NUMBER OF NAK-S BEFORE ABORT.
9:3	NAKCNT	NUMBER OF NAK-S ON CURRENT IO.
12:12		ADDRESS CHARACTERS.
24:1		CHANGE-MODE BEFORE FIRST ADDRESS CHARACTER.
25:1		CHANGE-MODE BEFORE SECOND ADDRESS CHARACTER.
26:6		INDEX TO TNAOG

MULTIPOINT ONLY:
32:8	LINELINK	CIRCULAR LINK TO ALL STATIONS ON LINE.
40:8		STATION IN READY QUEUE (MASTER ONLY).

SCHEDULE LINES:
1:1		ON IF CANDE IS READY FOR INPUT FROM LINE.
2:1	*SCHEND	SCHEDULE TERMINATING.
33:15		ADDRESS OF 80 WORD BUFFER ARRAY.

INPUTANK

INPUTANK IS A ONE DIMENSIONAL ARRAY STATIONMAX LONG. IT KEEPS TRACK
OF INPUT TANKS FOR THE STATIONS.

FIELD	DEFINE	CONTENTS
1:1		LOCK BIT.
2:8		NEXT CHARACTER TO BE DETANKED.
10:8	INPUTL	OLDEST SEGMENT IN TANK.
18:15		HEAD OF QUEUE OF 30 WORD AREAS.
33:1	INPUTREADY	ON IF THE JOB IS SWAPPED OUT TO AWAIT INPUT.
34:1	SLOWDOWN	USER HAS BEEN TOLD TO STOP HIS INPUT SINCE THE TANK IS ALMOST FULL.
35:1	FIRSTBUF	1 IF PREVIOUS INPUT BUFFER CONTAINED AN ETX.
38:2		MODE BITS FOR INPUT SCANNING IN COMM13.
40:8	INPUTN	NUMBER OF SEGMENTS IN TANK.

TANKS

TANKS IS A ONE DIMENSIONAL ARRAY STATIONMAX LONG, USED TO HANDLE THE
OUTPUT TANKS FOR THE STATIONS.

FIELD	DEFINE	CONTENTS
1:1		TANK INTERLOCK BIT.
2:8	*TANKLINE	DETANKING QUEUE. THIS CONTAINS THE NUMBER OF THE NEXT STATION WHICH IS TO BE DETANKED (O IF TAIL OF QUEUE OR NOT IN QUEUE).
10:1		CANDE SHUT-UP FLAG. TURNED ON WHEN A SHUT-UP EVENT IS ADDED TO THE QUEUE.
11:1	*TANKFUL	TANK FULL BIT. ON WHEN THE OBJECT JOB IS SWAPPED OUT TO WAIT FOR THE TANK TO EMPTY.
12:1		SOH BIT. ON=SOH, OFF=ETX.
13:1		IF ON, LAST MESSAGE WAS NAK-ED.
14:5	TANKL	NEXT SEGMENT TO DETANK.
19:5		NEXT WORD OF SEGMENT.
24:3		NEXT CHARACTER OF WORD.
27:6	SOUSE	
27:6	TANKN	NO. OF SEGMENTS IN USE.
33:15	*TANKA	CORE ADDRESS OF TANK.

TNAOG

TNAOG (TRANSMISSION NUMBERS AND OTHER GOODIES) IS A ONE DIMENSIONAL
ARRAY WHICH CONTAINS A WORD FOR EACH MULTIPOINT OR CONTENTION
STATION. IT IS INDEXED FROM A FIELD IN SEQARRAY. ITS USE DEPENDS
ON DEVICE TYPE.

```
    FIELD   DEFINE        CONTENTS
    -----   ------        --------
    1:5                   LAST SEGMENT DETANKED.
    6:5                   LAST WORD OF SEGMENT.
    11:3                  LAST CHARACTER OF WORD.

TC500:
    28:10                 INPUT TRANSMISSION NUMBER.
    38:10                 OUTPUT TRANSMISSION NUMBER.

PAGED SCREEN DEVICES:
    14:6                  NUMBER OF LINES PER PAGE.
    20:8                  NUMBER OF CHARACTERS PER LINE.
    28:8                  NUMBER OF CHARACTERS ON CURRENT LINE.
    36:6                  NUMBER OF LINES ON PREVIOUS PAGE.
    42:6                  NUMBER OF LINES ON CURRENT PAGE.
```

THE FORMATS DESCRIBED ABOVE APPLY TO ENTRIES 1 AND ABOVE. IN
ADDITION, ENTRY 0 OF THE FOLLOWING TABLES IS ALSO USED.

LINETABLE[0]

      LINETIMER   USED BY NSECOND TO TIME THE INTERVAL
                          BETWEEN POLLS TO INACTIVE STATIONS.

STATABLE[0]

  18:15   TAIL OF READY QUEUE.
  33:15   HEAD OF READY QUEUE.

TANKS[0]

   2:8   HEADOUT   HEAD OF OUTPUT DETANKING QUEUE.
  33:15   TAILOUT   TAIL OF OUTPUT DETANKING QUEUE.

SEQARRAY[0]

   3:15   TAIL OF QUEUE OF NEWLY NAK-ING STATIONS.
  18:15   HEAD OF QUEUE OF NEWLY NAK-ING STATIONS.
  33:15   HEAD OF NAK QUEUE.

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 TSSMCP NEAR THE LABELS NOTHINGTODO 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 RUN
    WORD 2   CONTAINS A PARAMETER TO THE ROUTINE.
    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 THE STACK SPACE ALREADY OBTAINED OR THE VALUE 1, OR 0. AFTER GETTING AN AREA, LINKING IT INTO THE QUEUE AND STORING THE INFORMATION, FORK RETURNS TO THE CALLING PROCEDURE. HOWEVER, IF FORK WAS PASSED A 1 FOR A PROCEDURE LACKING STACK SPACE, IT 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 STANDARDSTACK 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 STANDARDSTACK (THE AMOUNT OF SPACE NEEDED) 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.

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 1           LINK TO PREVIOUS 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 ARRAY

EXTREMELY SIMPLE.

JOBS ARE AWAKENED BY NOTHINGTODO, WHICH PROCESSES 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 IF 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. FOR EACH AREA IT CALCULATES A NUMBER T, WHICH IS THE MAXIMUM OVER ALL CHUNKS IN THE AREA OF THE TOTAL NUMBER OF JOBS USING THE AREA PLUS 64 TIMES THE NUMBER OF CURRENTLY ACTIVE JOBS USING THE AREA. THE JOB IS ASSIGNED TO THE AREA WITH THE MINIMUM T. THUS, CONFLICT IS MINIMIZED AMONG ACTIVE JOBS FIRST AND TOTAL JOBS SECOND. IF SEVERAL AREAS HAVE THE SAME T, THE ONE HIGHEST IN CORE IS USED.

IN ADDITION, JOBS INITIATED IN ANY WAY OTHER THAN BY A REMOTE USER ARE NOT ALLOWED TO CONFLICT WITH EACH OTHER FOR CORE. THUS, IF THE NOBATCH OPTION IS RESET, BACKGROUND JOBS WILL BE SCHEDULED IF IT IS NOT POSSIBLE TO RUN THEM WITHOUT THEIR AREA OVERLAPPING THE AREA OF OTHER BACKGROUND JOBS. SINCE IT IS ASSUMED THAT BACKGROUND JOBS DO NOT INTERACT AND HENCE CAN BE RUN ANY TIME, THIS RESTRICTION ELIMINATES NEEDLESS SWAPPING.

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 IT MUST WAIT FOR HUMAN
INTERACTION, FOR INSTANCE, TO WAIT FOR A TAPE TO BE MOUNTED,
AFTER AN ST MESSAGE, OR FOR TELETYPE I/O. 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 RELATIVELY 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 TSSMCP. 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) RDYRPTEND, 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.

THE FIELD NLS IS USED TO COUNT THE NUMBER OF JOBS WHICH ARE PLACED
IN FRONT OF A JOB WHICH IS OTHERWISE AT THE FRONT OF THE QUEUE. IF
THIS EXCEEDS THE SLICE NUMBER, SLN, PLUS 2, READYEND IS MOVED BEHIND
THIS JOB SO THAT FURTHER READYEND JOBS WILL COME IN BEHIND IT AND IT
WILL BE ALLOWED TO RUN. THIS ASSURES ALL JOBS OF THEIR TIME IN CORE.

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

Formula is in HOOKUPMCP at 2111 1540

$$T = (N \times 4 + C - P) \times 8 + 208$$
$$E = 2 \times T$$

WHERE

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

N IS THE NUMBER OF TIMES THE JOB HAS BEEN SWAPPED FOR USING ITS
TIME SLICE SINCE ITS LAST INTERACTION.
P IS THE PRIORITY OF THE JOB.
C IS THE NUMBER OF CHUNKS 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.

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
   RDYRPTEND 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 QUE 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 WAITNG FOR AN EXTERNAL CONDITION.
9	- READYSTATE	JOB IS IN THE READY QUEUE AFTER WAITING FOR AN EXTERNAL CONDITION.
10	- RDYRPT	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
TSSMCP 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. THEN ITS CHUNKS ARE REASSIGNED
ONE AT A TIME. IF 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 TSSMCP 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 TSSMCP PROCESSES THE BED, IT WILL FIND THE JOB
AND RESTART IT AS IF IT HAD NEVER BEEN SWAPPED.

WHEN OTHER PROCEDURES IN THE TSSMCP 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 DCMCP, 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.

IN ADDITION, THE MCP USES OLAYCTR TO KEEP TRACK OF THE APPROXIMATE
NUMBER OF MILLISECONDS SPENT DOING OVERLAYS DURING THE JOBS TIME
SLICE, ASSUMING A 20 MILLISECOND LATENCY FOR EACH OVERLAY. IF, WHEN
A JOB IS BEING SWAPPED OUT, OLAYCTR EXCEEDS THE LENGTH OF THE TIME
SLICE TIMES THE MULTIPROCESSING FACTOR (TO GIVE THE OPERATOR SOME
CONTROL OVER THIS DYNAMIC EXPANSION) THE JOBS AREA IS EXPANDED
DURING THE SWAP.

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 CHUNK WAS ADDED IN FRONT, THE
JAR AND UV TABLE ENTRIES FOR THE JOB ARE MOVED TO THE FRONT OF THE
NEW AREA.

LINE MAINTENANCE TANKING AND QUEUEING STRUCTURES

ONE OF THE PROBLEMS FACING ANY TIME SHARING SYSTEM IS THAT PEOPLE 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 B5700 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.

INPUT

AS DATA COMES IN FROM A REMOTE DEVICE, IT IS COLLECTED IN BUFFERS IN THE B487. WHEN A BUFFER IS FULL, THE MCP READS IT INTO A SEGMENTED SAVE AREA OF THE APPROPRIATE SIZE AND LINKS IT TO ANY OTHER BUFFERS WHICH ARE PART OF THE SAME MESSAGE. WHEN THE END OF THE MESSAGE IS REACHED, AS INDICATED BY A GROUP MARK OR ETX, OR WHEN 224 CHARACTERS HAVE BEEN QUEUED, THE BUFFERS ARE LINKED INTO THE WORKER QUEUE, THE HEAD AND TAIL OF WHICH ARE KEPT IN THE MCP VARIABLE WORKERINQ. THE FORMAT OF A QUEUED INPUT BUFFER IS:

WORD		CONTENTS
0	1:1	0
	2:2	SIZE OF AREA.
	4:1	DELETE FLAG, IF ON, DELETE CHARACTER TRANSMISSION ERROR HAS OCCURRED.
	5:1	END OF MESSAGE FLAG, INDICATING THAT A GROUP MARK OR ETX IS IN THIS BUFFER.
	6:1	SPECIAL RECORD FLAG FOR CANDE.
	7:3	0
	10:8	STATION NUMBER.
	18:15	ADDRESS OF PREVIOUS BUFFER IN THIS MESSAGE OR ZERO IS THIS IS THE FIRST BUFFER.
	33:15	ADDRESS OF THE NEXT BUFFER IN THE QUEUE OR ZERO IF THIS IS THE LAST BUFFER.
1-END		DATA.

NOTE THAT BUFFERS IN THE WORKER QUEUE ARE LINKED BACK ONLY TO A MESSAGE BOUNDARY, BUT ARE LINKED FORWARD ACROSS MESSAGE BOUNDARIES.

THE BUFFERS IN THE WORKER QUEUE ARE CONSOLIDATED INTO COMPLETE MESSAGES (WHICH MAY CONTAIN MORE THAN ONE RECORD) AND THEN STORED IN THE APPROPRIATE DISK TANK. INPUT FROM ALL USERS TO CANDE IS STORED IN THE FILE TANK/DISK, WHICH HAS ROWS OF TANKCHUNKSIZE SEGMENTS. TANKCHUNKSIZE IS DEFINED AT 00324700 IN THE MCP AND IS NOMINALLY SET TO 256. THE MCP OBTAINS ADDITIONAL ROWS AS THEY ARE NEEDED AND

CANDE RETURNS THEM WHEN SHE IS THROUGH WITH THEM. WHEN A SEGMENT IS
WRITTEN ON DISK, A COPY IS LEFT IN CORE UNTIL CANDE HAS COMPLETED
HER INITIAL SCANNING. THE FORMAT OF A SEGMENT IN THIS CANDE INPUT
QUEUE IS:

WORD		CONTENTS
0	3:15	RELATIVE DISK ADDRESS OF SEGMENT WITHIN THE ROW.
	18:15	CORE ADDRESS OF NEXT SEGMENT IN THE QUEUE.
	40:8	NUMBER OF WORDS OF DATA IN THIS SEGMENT.
1-29		RECORDS

THE DATA STORED FOR EACH RECORD CONSISTS OF ONE INFORMATION WORD
FOLLOWED BY THE RECORD IN AS MANY WORDS AS ARE REQUIRED TO HOLD IT.
THE FORMAT OF THE INFORMATION WORD IS:

1:1	SPECIAL RECORD FLAG (QUESTION MARK OR BIT 6:1 OF BUFFER).
10:8	STATION NUMBER.
40:8	NUMBER OF CHARACTERS IN THE RECORD.

EACH RECORD ENDS IN A GROUP MARK AND IS ENTIRELY CONTAINED WITHIN A
SINGLE SEGMENT. RECORDS CONTAINING MORE THAN 224 CHARACTERS ARE
TRUNCATED.

THE CORE AND DISK ADDRESSES OF THE FIRST SEGMENT IN THE QUEUE ARE
PASSED TO CANDE IN A DATA EVENT WHENEVER SHE REQUESTS MORE INPUT
FROM THE MCP. THE MECHANISM FOR THIS IS DESCRIBED LATER.

IF THE USER IS CONNECTED TO A USER PROGRAM, AS DISTINCT FROM ONE OF
THE CANDE PROGRAMS, HIS DATA IS STORED IN THE INPUT TANK FOR HIS
STATION. THE DISK SPACE FOR THIS, WHICH IS GOTTEN DURING DATACOM
INITIALIZATION, CONTAINS (CLUMPSIZE X STAMAX) SEGMENTS, THUS
ALLOWING CLUMPSIZE SEGMENTS TO EACH USER. CLUMPSIZE IS DEFINED AT
00327300 AND HAS A NOMINAL VALUE OF 32. THE ADDRESS OF THIS AREA IS
KEPT IN THE MCP VARIABLE PROGTANK, AND A USER USES THE AREA
BEGINNING AT RELATIVE SEGMENT (CLUMPSIZE X STATION NUMBER). THE
TANK FOR A GIVEN USER IS USED CYCLICALLY. WHEN A PROGRAM ASKS FOR
DATA IT IS READ FROM DISK AND MOVED INTO THE PROGRAM-S BUFFERS.

THE FIRST WORD OF EACH SEGMENT OF INPUT CONTAINS INFORMATION ABOUT
THE DATA STORED IN THE REMAINING 29 WORDS OF THE SEGMENT. ONLY THE
FOLLOWING FIELDS OF THE FIRST WORD ARE SIGNIFICANT:

2:1	PACKING FLAG.
5:1	ON IF AND ONLY IF AN END OF MESSAGE OCCURS IN THE SEGMENT.
18:15	NUMBER OF CHARACTERS STORED IN THE REMAINDER OF THE SEGMENT.

DATA FOR OBJECT JOB INPUT IS STORED AS UNSCANNED MESSAGES IN ORDER

TO ALLOW THE PROGRAM TO SPECIFY WHAT SCANNING SHOULD BE DONE. A MESSAGE MAY EXTEND OVER AS MANY SEGMENTS AS REQUIRED. FOR NON-PAPERTAPE INPUT, EACH MESSAGE BEGINS AT THE START OF A SEGMENT AND THE PACKING FLAG IS RESET FOR ALL SEGMENTS. FOR INPUT RECEIVED IN PAPERTAPE MODE, THE PACKING FLAG IS SET AND A SEGMENT MAY CONTAIN MORE THAN ONE MESSAGE, IN WHICH CASE, THE MESSAGES ARE SEPARATED BY GROUP MARKS. HOWEVER, IF A MESSAGE ENDS IN A SEGMENT, ADDITIONAL MESSAGES IN THAT SEGMENT ARE NOT ALLOWED TO CONTINUE TO ADDITIONAL SEGMENTS.

OUTPUT

DATACOM OUTPUT IS ACCOMPLISHED BY SCANNING, TANKING AND DETANKING DATA PASSED TO THE MCP. TANKING AND DETANKING ARE DESIGNED TO ACHIEVE AN ORDERLY FLOW OF OUTPUT TO A MAXIMUM NUMBER OF TERMINAL DEVICES. SCANNING, WHICH WILL BE DESCRIBED LATER, IS DESIGNED TO RESOLVE THE UNIQUE REQUIREMENTS OF VARIOUS TERMINAL DEVICES (E.G., CONTROL CHARACTERS). AFTER THE DATA IS SCANNED, IT IS TANKED IF THERE IS OTHER OUTPUT PENDING FOR THE STATION, OTHERWISE IT IS IMMEDIATELY QUEUED FOR OUTPUT. FOR DEVICES OTHER THAN TELETYPES, THE DATA IS ALWAYS TANKED (EVEN IF IT IS ALSO QUEUED) IN ORDER TO PROVIDE PAGING AND RETRANSMISSION CAPABILITIES.

TWO LEVELS OF OUTPUT TANKING ARE MAINTAINED BY THE SYSTEM. THE FIRST LEVEL IS A 30 WORD CORE TANK PROVIDED WHEN THE STATION IS ATTACHED TO A JOB RUNNING ABOVE THE FENCE. THIS CORE TANK, WHICH IS IN THE JOB-S AREA AND HENCE SUBJECT TO SWAPPING IS USED TO PACK MESSAGES INTO A CONTINUOUS STREAM OF OUTPUT. WHEN A MESSAGE WILL NOT FIT INTO THE TANK, THE TANK IS WRITTEN INTO THE NEXT AVAILABLE SEGMENT IN THE DISK TANK FOR THE STATION, AND THE MESSAGE IS THEN PLACED INTO THE FRONT OF THE NOW EMPTY CORE TANK. BEFORE A JOB IS SWAPPED, ITS CORE TANK IS WRITTEN INTO THE DISK TANK. WHEN THE CORE TANK IS NOT PRESENT, EITHER BECAUSE THE STATION IS ATTACHED TO CANDE OR ITS JOB IS SWAPPED OUT, OUTPUT IS TANKED ON DISK, ONE MESSAGE PER SEGMENT.

THE OUTPUT TANKS ARE KEPT IN A DISK AREA OF (STAMAX X GLOMSIZE) SEGMENTS WHICH IS OBTAINED DURING INITIALIZATION. GLOMSIZE, WHICH IS NOMINALLY SET AT 32, IS DEFINED AT 00327800. THE AREA FOR A SPECIFIC STATION BEGINS AT (GLOMSIZE X LL), WHERE LL IS THE STATION NUMBER. THE FIRST WORD OF EACH DISK SEGMENT (AND OF THE CORE TANK) CONTAINS THE NUMBER OF CHARACTERS IN THE SEGMENT AND THE REMAINING 29 WORDS CONTAIN THE OUTPUT WITH A SINGLE GROUP MARK AT THE END.

WHEN A STATION IS DETANKED, THE OLDEST SEGMENT OF THE STATION-S DISK TANK IS BROUGHT INTO CORE AND QUEUED FOR OUTPUT. IF THE DISK TANK IS EMPTY, THE STATION-S CORE TANK IS USED, AND, IF THE STATION IS NOT A TELETYPE, THE CORE TANK IS WRITTEN TO DISK.

DATA IS QUEUED FOR OUTPUT BY MOVING IT INTO SEGMENTED SAVE AREAS WHICH ARE LINKED TOGETHER TO CONTAIN THE MESSAGE. THE B487 BUFFER SIZE FOR THE STATION DETERMINES THE SIZE OF THE AREAS USED. THE

BUFFERS ARE THEN LINKED INTO THE STATION-S OUTPUT QUEUE FOR TRANSMISSION. THE FIRST WORD OF EACH BUFFER CONTAINS:

1:1	0 INDICATES AREA IN USE.
2:2	AREA SIZE.
5:1	IF 1, AN ACTIVE INTERROGATE IS TO BE DONE.
6:1	IF 1, A BLAST READ IS TO BE DONE.
7:3	IF [5:2] = 0 THIS IS THE TYPE OF WRITE THAT IS BEING DONE - I.E., IT WILL BE DIRECTLY DIALED INTO LSTATUS.

0 - NORMAL WRITE
1 - POLL
2 - SELECT OR ENQ
3 - ACKING MESSAGE
4 - NAKING MESSAGE
5 - ACKING ENQ
6 - NAKING ENQ

10:8	STATION NUMBER
33:15	NEXT AREA IN OUTPUT QUEUE.

THE HEAD OF A STATION-S OUTPUT QUEUE IS KEPT IN STATABLE.

LINES WHICH HAVE STATIONS WITH OUTPUT ARE LINKED THROUGH THE READY QUEUE, WHICH IS KEPT IN THE 25:8 FIELD OF THE STATABLE ENTRY FOR THAT LINE NUMBER. IN THE CASE OF MULTIPOINT STATIONS, THE NUMBER OF THE STATION FOR WHICH OUTPUT IS TO BE DONE IS KEPT IN THE LINE-S SEQARRAY ENTRY. WHEN THAT STATION-S OUTPUT IS COMPLETE, ALL STATIONS ON THE LINE ARE CHECKED TO SEE IF THERE IS ONE WITH OUTPUT. IF SO, THE LINE IS RE-LINKED INTO THE READY QUEUE FOR THAT STATION. THE HEAD AND TAIL OF THE READY QUEUE ARE KEPT IN STATABLE[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.

GROUP I - DATACOM HARDWARE INTERFACE PROCEDURES

THE PROCEDURES WHICH FALL INTO THE FIRST GROUP ARE:

INITIATEDCIO	- INITIATES I/O-S FOR LINE MAINTENANCE PROCEDURES.
INTERROGATOR	- HANDLES INQUIRY REQUEST INTERRUPTS.
DCIOFINISH	- HANDLES I/O COMPLETE INTERRUPTS FOR DATACOM.
GIVEAWAY	- LINKS INPUT INTO THE WORKER QUEUE.
ENTERLINEQ	- LINKS BUFFERS INTO THE OUTPUT QUEUE FOR A STATION.
ENTEREADYQ	- ENTERS A LINE INTO THE READY QUEUE AND ALSO HANDLES SWITCHING BETWEEN STATIONS AND GENERATION OF POLLS FOR MULTIPOINT LINES.
NEXTDCIO	- PERFORMS I/O-S FROM THE READY QUEUE AND SOME INTERROGATES.
HELLO	- HANDLES NEW USERS.
QUITTER	- HANDLES DISCONNECTS, BREAKS, AND WRU.
DCTIMEOUT	- HANDLES TIMEOUT ERRORS AND RECOVERY FOR NON-TWX DEVICES.
NAKQUE	- HANDLES MESSAGE RETRY FOR NOT READY SPECIAL DEVICES.
DCERR	- SPOUTS MESSAGE TO SPO AFTER ERRORS.
DCBUSY	- SENDS MESSAGE TO USER AFTER BUFFER OVERFLOW.

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 THEN 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 IOFINISH 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 THE 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 INTERROGATE.

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 THAT HANDLES THE INQUIRY REQUEST IS INTERROGATOR. IF THERE IS NO I/O CHANNEL ASSIGNED TO DATA COMMUNICATIONS, IT ENTERS AN INTERROGATE IN THE I/O QUEUE BY CALLING IOREQUEST, 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 OUTPUT 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 ENTEREADYQ, IN WHICH CASE, AN I/O CHANNEL MAY OR MAY NOT BE AVAILABLE. THE OTHER IS DCIOFINISH, IN WHICH CASE AN I/O CHANNEL IS AVAILABLE SINCE DCIOFINISH 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 DCIOFINISH GETS CONTROL, IT STARTS AN I/O AS SOON AS POSSIBLE, EITHER BY INITIATING IT ITSELF OR BY CALLING NEXTDCIO.

THESE PROCEDURES, ENTEREADYQ, NEXTDCIO, AND DCIOFINISH, FORM THE HEART OF DATACOM I/O HANDLING. BETWEEN THEM, THEY HANDLE ALMOST ALL THE PROBLEMS OF THE VARIOUS LINE DISCIPLINES, E.G., POLLS, ACK-S, NAK-S, ETC. THE LINE STATUS (KEPT IN LINETABLE[I].[21:5]) IS USED TO KEEP TRACK OF WHICH OF THESE FUNCTIONS ARE BEING DONE. THIS IS VERY IMPORTANT SINCE THE SAME INTERRUPTS (E.G., READ READY ABNORMAL) HAVE DIFFERENT MEANINGS DEPENDING ON THE CONTEXT IN WHICH THEY OCCUR. THE STATUSES ARE DIVIDED INTO THREE GROUPS: READING, I. E., DATA IS BEING RECEIVED; WRITING, DATA IS BEING TRANSMITTED; AND IDLE. THE STATUSES AND THEIR USES ARE LISTED UNDER "TABLES USED BY LINE MAINTENANCE".

ENTEREADYQ IS CALLED TO START QUEUED I/O-S FOR A LINE. FOR INSTANCE, IT IS CALLED FROM ENTERLINEQ AFTER NEW DATA HAS BEEN ADDED TO A STATION-S OUTPUT QUEUE AND FROM DCIOFINISH AFTER THE B487 HAS FINISHED TRANSMITTING A BUFFER, FREEING IT TO ACCEPT THE NEXT BUFFER LOAD OF OUTPUT. ENTEREADYQ FIRST CHECKS TO SEE IF THE LINE IS IN AN IDLE STATUS. IF IT IS, AND THERE IS SOMETHING IN THE STATION OUTPUT QUEUE, THE LINE IS ADDED TO THE READY QUEUE. FOR MULTIPOINT LINES THE STATION BEING READIED IS DETERMINED BY THE 25:8 FIELD OF SEQARRAY. IF IT IS ZERO WHEN ENTEREADYQ IS CALLED, ENTEREADYQ LOOKS

FOR A STATION WITH OUTPUT TO MAKE THAT ONE THE ACTIVE STATION. IF THERE ARE NONE, IT GENERATES A POLL AND LINKS IT INTO THE MASTER STATION-S OUTPUT QUEUE. IN ALL CASES, ENTEREADYQ CALLS NEXTDCIO BEFORE EXITING.

IF THERE IS ALREADY A DATACOM I/O GOING ON, NEXTDCIO EXITS. IF THERE IS NOT, BUT ANY INTERROGATE NEEDS TO BE DONE, NEXTDCIO DOES THAT, OTHERWISE IT TAKES THE FIRST I/O IN THE OUTPUT QUEUE OF THE FIRST LINE IN THE READY QUEUE. HOWEVER, IF THAT LINE IS POLLING, IT DOES AN ACTIVE INTERROGATE INSTEAD. THIS WILL STOP THE POLL, AT WHICH TIME THE LINE WILL AGAIN BE READIED AND THE I/O CAN BE DONE.

DCIOFINISH HANDLES ALL DATACOM I/O COMPLETE INTERRUPTS AND THEREFORE ACTS AS THE PRIMARY DRIVER FOR THE REST OF THE LINE MAINTENANCE CODE. USING THE LINE STATUS AND THE INFORMATION IN THE RESULT DESCRIPTOR, IT DETERMINES WHAT HAS HAPPENED AND WHAT NEEDS TO BE DONE NEXT. IN MOST CASES, IT JUST HAS TO UPDATE THE STATUS AND CALL EITHER ENTEREADYQ OR NEXTDCIO. FOR MORE COMPLICATED CASES, IT FORKS VARIOUS INDEPENDENT RUNNERS. IN ALL CASES, IT EITHER STARTS AN I/O ITSELF OR CALLS NEXTDCIO. IF AN I/O DOES NOT GET INITIATED, DCIOFINISH RETURNS THE IOQUE SPACE ASSIGNED TO DATACOM BEFORE IT EXITS.

BASICALLY, THERE ARE TWO PARTS TO DCIOFINISH. THE FIRST PART HANDLES INTERROGATE RESULTS. THIS INCLUDES MOST OF THE ERROR AND TIME-OUT HANDLING. ALSO, IF A READ READY INTERRUPT OCCURS DCIOFINISH IMMEDIATELY INITIATES A READ. THE SECOND PART HANDLES READ AND WRITE COMPLETES. THIS INCLUDES THE SCANNING AND GENERATION OF THE APPROPRIATE LINE CONTROL CHARACTERS, I.E., ACK-S, NAK-S, AND EOT-S. FOR INSTANCE, AFTER TRANSMISSION OF A MESSAGE, THE TERMINAL RESPONDS WITH AN ACK OR NAK. WHEN THIS IS READ, DCIOFINISH FINDS OUT WHICH IT IS, AND EITHER RETURNS THE LINE TO A IDLE STATUS OR CAUSES THE MESSAGE TO BE RETRANSMITTED. FOR TELETYPES, THIS CODE GENERATES THE CARRIAGE RETURN, LINE FEED AND, IF NECESSARY, SEQUENCE NUMBER AFTER A GROUP MARK. IN ADDITION, DCIOFINISH QUEUES THE BUFFER LOADS OF INPUT, CALLING GIVEAWAY TO PUT THEM INTO THE WORKER QUEUE AFTER SUCCESSFUL COMPLETION OF THE MESSAGE OR WHEN IT HAS ACCUMULATED 224 CHARACTERS. IN ALL CASES, THE LINE STATUS IS UPDATED BEFORE DCIOFINISH EXITS.

THE PROCEDURES FORKED BY DCIOFINSH ARE HELLO, QUITTER, NAKQUE, AND DCTIMEOUT. HELLO IS FORKED WHEN A STATION BECOMES ACTIVE AS SIGNALED BY EITHER A DIAL-IN OR ACTIVITY ON A PREVIOUSLY INACTIVE DIRECT CONNECT STATION. HELLO MERELY SENDS THE GREETING MESSAGE, UPDATES THE STATIONS SYSTEM/DISK ENTRY AND QUEUES AN EVENT TO CANDE.

QUITTER IS USED FOR BREAKS, WRU-S, AND DISCONNECTS. WHEN ONE OF THOSE HAPPENS, DCIOFINISH, IN ADDITION TO FORKING QUITTER, QUEUES AN APPROPRIATE EVENT TO CANDE, UPDATES THE STATUS AND, IF THE LINE IS IN THE READY QUEUE, DELINKS IT. THIS BLOCKS FURTHER OUTPUT. WHEN IT RUNS, QUITTER DISCARDS THE INPUT AND OUTPUT QUEUES AND TANKS AND RESETS THE PAGING COUNTERS. FOR A DISCONNECT, A "BYE" WITH THE SPECIAL MESSAGE FLAG ON IS QUEUED TO CANDE, THUS MARKING THE SPOT IN

CANDE-S INPUT AT WHICH THE DISCONNECT OCCURRED, AND THE STATION IS MARKED DISCONNECTED IN SYSTEM/DISK. QUITTER THEN GIVES THE LINE AN IDLE STATUS AND DOES AN ENTEREADYQ SO THAT MULTIPOINT LINES CAN BE INITIATED FOR A NEW STATION OR COMMENCE POLLING.

TIMEOUT

DCTIMEOUT IS FORKED WHEN A STATION DOES NOT RESPOND TO A POLL OR SELECT. THE PROCEDURE CALCULATES WHICH STATION TIMED OUT (IN CASE IT WAS ON A MULTI-POINT LINE) AND ATTEMPTS TO RECOVER FROM THE ERROR BY DETERMINING IF THE LINE IS PERMANENTLY DEAD OR WAS JUST TEMPORARILY DEAD. IT RETRIES 4 TIMES AND IF THE STATION IS STILL TIMING OUT, DCTIMEOUT MARKS THE STATION AS DISCONNECTED, FORKS QUITTER AND REMOVES THE STATION FROM THE POLL LIST BY SETTING BIT 3: 1 OF SEQARRAY.

STATIONS WHICH HAVE BEEN REMOVED FROM THE POLL LIST ARE POLLED EVERY 30 SECONDS TO FIND OUT IF THE HAVE AGAIN BECOME ACTIVE. FOR THIS PURPOSE, A TIMER IS KEPT IN LINETABLE[0]. IF, WHEN NSECOND RUNS, THE TIME IS GREATER THAN THE VALUE OF LINETABLE[0], NSECOND ADDS 30 SECONDS TO LINETABLE[0] AND THEN RESETS THE 3:1 BIT OF SEQARRAY FOR ALL MULTIPOINT STATIONS WHICH HAVE IT ON. THUS, ALL STATIONS WILL BE POLLED ROUGHLY EVERY 30 SECONDS, DEPENDING ON WHEN NSECOND RUNS.

NAK HANDLING AND THE NAK QUEUE

THERE ARE TWO KINDS OF NAK-S THAT CAN BE RECEIVED FROM THE NON-TELETYPE DEVICES HANDLED BY THE TSS MCP. THEY ARE:

1) NAK TO A MESSAGE,
2) NAK TO A SELECT.

A NAK TO A MESSAGE INDICATES THAT THE DEVICE SENSED PARITY ON THE MESSAGE SENT, AND IS REQUESTING THAT THE MESSAGE BE RETRANSMITTED. THE MCP THEREFORE SETS THE PRESENT TANK POINTERS TO POINT TO THE LAST MESSAGE AND FIRES UP THE DETANKING PROCEDURE. THIS HAS THE EFFECT OF SENDING THE PREVIOUS MESSAGE AGAIN. THERE IS A SPECIFIED MAXIMUM NUMBER OF RETRIES ALLOWED, NAKMAX, WHICH IS ESTABLISHED IN THE SYSTEM/DISK SPECIFICATION FOR EACH LINE. IF THERE ARE NAKMAX CONSECUTIVE NAK-S RECEIVED FROM THE STATION, THE LINE WILL BE DISCONNNECTED.

THE PURPOSE OF THE SELECT MESSAGE IS TO ASK THE DEVICE IF IT IS READY TO RECEIVE INFORMATION. THE POSITIVE REPLY IS AN ACK, INDICATING THAT THE DEVICE IS READY TO RECEIVE. THE NEGATIVE RESPONSE IS A NAK, INDICATING THAT THE DEVICE IS NOT READY AND THEREFORE THE SELECT NEEDS TO BE RETRIED LATER. WHEN THE MCP RECEIVES A NAK, IT IDLES THE LINE AND ENTERS THE STATION INTO A QUEUE OF 5 WORD SEGMENTED SAVE AREAS, CONTAINING THE FOLLOWING:

```
WORD                        CONTENTS
----                        --------
0       33:15               ADDRESS OF THE NEXT ENTRY IN THE QUEUE OR
                            0 IF THIS IS THE LAST ENTRY.
1                           STATION NUMBER.
2                           TIME AT WHICH NAK WAS RECEIVED.
```

THE ADDRESS OF THE FIRST ENTRY IN THE QUEUE IS KEPT IN SEQARRAY[0].
[3:15] AND THE ADDRESS OF THE LAST ENTRY IS IN SEQARRAY[0].[18:15].

AFTER QUEUEING THE STATION, THE MCP FORKS NAKQUE IT IF IS NOT
ALREADY RUNNING. NAKQUE PICKS UP ENTRIES FROM THE QUEUE OF NAKKED
STATIONS, STORES THE TIME THE STATION SHOULD BE RETRIED IN WORD 2
AND THEN REQUEUES THEM, ORDERED BY WORD 2, THE RETRY TIME. THE HEAD
OF THE ORDERED QUEUE IS KEPT IN SEQARRAY[0].[33:15]. WHEN EACH OF
THE TIMES EXPIRE, THE LINE IS AWAKENED BY REQUESTING DETANKING. THE
LINE WILL BE AWAKENED EARLIER IF DATA IS RECEIVED FROM THE DEVICE.

THE TIME BETWEEN SUCCESSIVE TRANSMISSIONS OF THE SELECT MESSAGE IS
CALCULATED AS FOLLOWS:

    1) FOR A TC500 THE TIME INTERVAL IS 1/2 SECOND.

    2) FOR A B9352 OR B9353 THE TIME IS INCREASING INTERVALS OF 3
       SECONDS (I.E., 3 SECONDS, THEN 6 SECONDS, AND SO ON) UP TO A
       MAXIMUM OF 24 SECONDS.

BLASTREAD

BLASTREAD IS A PROCEDURE THAT ATTEMPTS TO CLEAR OR IDLE A LINE WHEN
IT IS IN AN UNKNOWN STATE. IT IS CALLED BY SPREADTHEWORD TO CLEAR
EACH LINE AFTER A HALT/LOAD, BY NSECOND IN PREPARATION FOR THE
DISCONNECT OF AN INACTIVE LINE, AND BY LINECLEAR AFTER A CL. THE
ACTION TAKEN IS DEPENDENT UPON THE LINE DISCIPLINE OF THE LINE THAT
IS TO BE CLEARED:

    1. TELETYPE
        A. WRITE A CARRIAGE RETURN, LINE FEED, GROUP MARK. THIS
            IDLES THE ADAPTER IF IT WERE IN A WRITE READY STATE.
        B. READ THE BUFFER - THIS WILL CLEAR THE ADAPTER IF IT IS IN
            A READY READY STATE.
    2. POINT TO POINT
        A. READ THE BUFFER - CLEAR UP THE READ READY STATE.
        B. WRITE AN EOT - CLEAR UP THE STATE OF THE DEVICE. THIS IS
            ESPECIALLY NEEDED FOR A B9352 AFTER IT HAS JUST BEEN
            POWERED ON.
    3. MULTI POINT
        A. ACTIVE INTERROGATE - THIS SHUTS DOWN ANY RECYCLING POLLS
            THAT MAY BE IN THE ADAPTER.
        B. READ THE BUFFER - CLEAR UP THE READ READY STATE.

THESE ACTIONS WILL, IN MOST CASES, REDUCE THE LINE TO AN IDLE STATE,
ENABLING FURTHER ACTION TO BE TAKEN.

BLASTREAD CAN BE CALLED WITH THE REQUEST TO DISCONNECT THE LINE
AFTER IT IS IDLED. IN THIS CASE, AFTER CLEARING THE LINE AS
DESCRIBED ABOVE, BLASTREAD SENDS THE MESSAGE "PLEASE CALL BACK
LATER" AND CAUSES THE LINE TO BE DISCONNECTED.

GROUP II - TANKING PROCEDURES

THE PROCEDURES IN THE SECOND GROUP ARE:

    OLDWEIRDHAROLD   - TANKS INPUT AND DETANKS OUTPUT.
    PAPERTAPEIO   - PERFORMS SPECIAL HANDLING FOR INPUT FROM PAPER
                        TAPE.
    INPUTSCAN   - SCANS INPUT TO BREAK IT INTO RECORDS AND TO
                        ELIMINATE CONTROL CHARACTERS.
    DCWRITE   - PREPARES OUTPUT FOR TRANSMISSION.

THE MOST IMPORTANT OF THESE PROCEDURES IS OLDWEIRDHAROLD.

OLDWEIRDHAROLD, ALIAS THE WORKER, DOES MANY STRANGE AND WONDROUS
THINGS, SOME OF WHICH ARE USEFUL. HE OPERATES ON TWO QUEUES, THE
WORKER QUEUE IN WHICH BUFFERS OF INPUT ARE PLACED, AND THE DETANKING
QUEUE, A QUEUE OF STATIONS FOR WHICH OUTPUT MUST BE TAKEN FROM THE
TANKS. IF HE HAS NOT BEEN FORKED ALREADY, HE IS FORKED WHENEVER
SOMETHING IS ADDED TO ONE OF THE QUEUES. WHEN GIVEN CONTROL, HE
PROCESSES THE WORKER QUEUE UNTIL IT IS EMPTY AND THEN DETANKS DATA
FOR THE FIRST STATION IN THE DETANKING QUEUE. AFTER DETANKING A
STATION, HE AGAIN CHECKS THE WORKER QUEUE FOR INPUT. HE CONTINUES
IN THIS FASHION UNTIL BOTH QUEUES ARE EMPTY, AFTER WHICH HE GOES TO
NOTHINGTODO.

THE WAY IN WHICH INPUT FROM THE WORKER QUEUE IS PROCESSED DEPENDS ON
THE TYPE OF STATION FROM WHICH IT CAME AND WHETHER IT IS INTENDED
FOR CANDE OR AN OBJECT JOB. INPUT FROM A BACK-UP SPO IS TREATED
LIKE INPUT TO CANDE, EXCEPT THAT IT IS PASSED OFF TO KEYIN INSTEAD
OF BEING QUEUED IN CANDE-S INPUT TANK.

INPUT FOR AN OBJECT JOB IS MOVED UNSCANNED INTO 30 WORD ARRAYS.
THIS CONSOLIDATES THE DATA IN THE SEGMENTED SAVE AREAS IN THE WORKER
QUEUE INTO A SMALLER NUMBER OF LARGER AREAS IN ORDER TO FREE THE
SEGMENTED SAVE AREAS FOR FURTHER USE. THIS IS VERY IMPORTANT SINCE
A B9352 CAN TRANSMIT UP TO 960 CHARACTERS (120 WORDS) AT ONE TIME.
WHEN THERE IS NOT ENOUGH ROOM IN THE ARRAY TO HOLD THE NEXT BUFFER,
AN I/O IS INITIATED TO WRITE THE ARRAY INTO THE STATION-S INPUT TANK
AND A NEW ARRAY IS GOTTEN. WHEN THIS ARRAY IS FILLED, HARRY
INITIATES A DISK WRITE ON IT, SLEEPS UNTIL THE I/O ON THE PREVIOUS
ARRAY IS COMPLETE AND THEN REUSES THE OLD ARRAY. THE NUMBER OF
SEGMENTS (ARRAYS) WRITTEN TO DISK IS KEPT IN THE FIRST WORD OF THE
ARRAY SO THAT THE INPUTANK ENTRY FOR THE STATION CAN BE UPDATED
APPROPRIATELY WHEN THE ENTIRE MESSAGE IS RECEIVED. AT THAT TIME, IF
THE JOB IS WAITING FOR INPUT, IT IS SWAPPED IN. IF THE MESSAGE MUST
BE DELETED, DUE TO RECEIPT OF A DELETE CHARACTER OR DUE TO A
TRANSMISSION ERROR, THE ARRAYS ARE SIMPLY FORGOTTEN, AND, SINCE
INPUTANK WAS NOT UPDATED, ANY INFORMATION ALREADY WRITTEN TO DISK IS
LOST.

PAPERTAPE INPUT IS HANDLED IN THE SAME WAY EXCEPT WHEN A DELETE OR
END-OF-MESSAGE OCCURS. IN THESE CASES OLDWEIRDHAROLD CALLS
PAPERTAPEIO. FOR AN END-OF-MESSAGE, PAPERTAPEIO SCANS FOR THE GROUP

2-45

MARK AND ADJUSTS THE CHARACTER COUNT IN WORD O OF THE ARRAY TO POINT
TO IT. IT ALSO STORES THE CHARACTER COUNT SO THAT IT CAN RESTORE IT
IF A DELETE SUBSEQUENTLY OCCURS. IF THE TANK IS LESS THAN HALF
FULL, IT THEN RETURNS TO OLDWEIRDHAROLD, WHO PROCEEDS AS IF END-OF-
MESSAGE HAD NOT OCCURRED. THUS, TO HARRY, PAPERTAPE INPUT LOOKS
LIKE ONE LONG MESSAGE. HOWEVER, IF THE TANK IS OVER HALF FULL AND
THE JOB IS WAITING FOR INPUT, PAPERTAPEIO UPDATES INPUTANK AND
CAUSES THE JOB TO BE SWAPPED IN. WAITING UNTIL THE TANK IS HALF
FULL REDUCES SWAPPING BY ALLOWING THE PROGRAM TO PROCESS MANY
RECORDS EACH TIME IT IS BROUGHT IN. THE USER DOES NOT NOTICE SINCE
PAPERTAPE IS A NON-INTERACTIVE MODE OF INPUT.

THE ADDRESS OF THE ARRAY CURRENTLY BEING USED IS KEPT IN THE
INPUTANK ENTRY FOR THE STATION. RELATIVE TO THAT ADDRESS, THE ARRAY
IS FORMATTED AS FOLLOWS:

WORD		CONTENTS
-3		COPY OF WORD O FOR PAPERTAPE DELETE.
-2		I/O COMPLETE WORD.
-1		RESERVED FOR ADDRESS FOR DISK I/O.
0	1:1	FLAG TO INDICATE THAT INPUTANK SHOULD BE UPDATED WHEN THE ARRAY IS WRITTEN TO DISK.
	2:1	PACKING (PAPERTAPE) FLAG.
	3:2	NOT USED.
	5:1	FLAG TO INDICATE THAT END OF MESSAGE OCCURS IN THIS BUFFER.
	6:4	NOT USED.
	10:8	NUMBER OF THIS BUFFER IN CURRENT MESSAGE (STARTING FROM ZERO).
	18:15	NUMBER OF CHARACTERS IN THIS BUFFER.
	33:15	ADDRESS OF PREVIOUS BUFFER.
1-29		DATA

NOTE THAT THE FORMAT OF WORDS 0-29 MATCH THE FORMAT OF A SEGMENT IN
THE INPUT TANK SO THAT NO REFORMATTING IS REQUIRED.

INPUT TO CANDE FROM NON-TWX STATIONS IS ALSO ACCUMULATED IN THESE 30
WORD ARRAYS. HOWEVER, INSTEAD OF BEING WRITTEN ONTO DISK, THEY ARE
QUEUED IN CORE IN REVERSE TIME ORDER THROUGH THE 33:15 FIELD OF WORD
0, WITH A ZERO ENTRY TO MARK THE END OF THE QUEUE. WHEN THE MESSAGE
IS COMPLETED, THE LINKING IS REVERSED AND THE BUFFERS ARE TREATED
LIKE INPUT FROM TWX STATIONS, WHICH IS PROCESSED DIRECTLY FROM THE
WORKER QUEUE. THE REASON NON-TWX DATA MUST BE ACCUMULATED IS THAT,
IF A ERROR OCCURS, IT WILL BE RETRANSMITTED, THEREFORE, IT CANNOT BE
PROCESSED UNTIL THE END OF THE MESSAGE IS SUCCESSFULLY REACHED.
PROCEDURE INPUTSCAN IS USED TO MOVE AND SCAN DATA FROM THE BUFFERS,
WHICH MAY BE EITHER ARRAYS OR SEGMENTED SAVE AREAS, INTO ANOTHER
ARRAY. SCANNING STOPS AT THE END OF A RECORD OR WHEN 224 CHARACTERS
(THE MOST THAT CAN BE PASSED TO CANDE) ARE PROCESSED. IF IT IS FROM
A BACK-UP SPO, THE RECORD IS PASSED TO KEYIN, OTHERWISE IT IS QUEUED
FOR CANDE. OLDWEIRDHAROLD THEN PROCEEDS TO THE NEXT RECORD OR THE

NEXT MESSAGE IN THE WORKER QUEUE.

WHEN HE HAS A RECORD FOR CANDE, HARRY CHECKS TO SEE IF IT WILL FIT
INTO THE LAST ARRAY IN CANDE-S INPUT QUEUE, WHICH IS POINTED TO BY
LASTSEG. IF IT WON-T, HE GETS A NEW AREA, LINKS IT INTO THE QUEUE
VIA THE 33:15 FIELD OF WORD 0 OF LASTSEG AND THEN CHANGES LASTSEG TO
POINT TO THE NEW AREA. HE THEN PUTS THE MESSAGE INTO THE ARRAY
POINTED TO BY LASTSEG AND, IF CANDEINPUTREADY IS ON, FORKS
NOTIFYCANDE, A PROCEDURE IN THE THIRD GROUP OF LINE MAINTENANCE
PROCEDURES.

WHEN DETANKING OUTPUT FOR A STATION, HARRY CHECKS TO SEE IF THERE IS
ANYTHING IN THE DISK TANKS. IF THERE IS NOT, THE ENTIRE CORE TANK
IS USED AND ITS CHARACTER COUNT IS SET TO ZERO. IF THERE IS DATA ON
DISK, IT IS DETANKED 56 CHARACTERS AT A TIME FOR TWX STATIONS AND
ONE SEGMENT AT A TIME FOR OTHER STATIONS. THIS IS DONE SINCE
TELETYPES ARE SO MUCH SLOWER THAN THE OTHER DEVICES. THE DATA IS
THEN PASSED TO DCWRITE. IF THE STATION IS NOT CONNECTED, OR IS IN
PAPERTAPE MODE, DCWRITE EXITS, CAUSING THE DATA TO BE LOST. TO
PREPARE THE DATA FOR TRANSMISSION, IT MOVES IT INTO SEGMENTED SAVE
AREAS, FORMATTED FOR INCLUSION IN THE STATION-S OUTPUT QUEUE. IF
REQUIRED BY THE LINE DISCIPLINE, IT PUTS A TERMINAL SELECT MESSAGE
IN A SEPARATE BUFFER AT THE START AND PUTS A MESSAGE HEADING AT THE
FRONT OF THE MESSAGE. FOR SCREEN DEVICES, CONTROL CHARACTERS TO
PERFORM PAGING ARE INSERTED. THE AREAS ARE THEN PASSED TO
ENTERLINEQ.

GROUP III - NORMAL STATE PROGRAM INTERFACE PROCEDURES

THE PROCEDURES IN THE THIRD GROUP ARE:

TWXOUT   - HANDLES ALL OUTPUT.
OUTRAN980   - FORMATS OUTPUT FOR 980 (TELETYPE) ADAPTERS.
OUTRANTC   - FORMATS OUTPUT FOR TC500-S.
OUTRANBIDS   - FORMATS OUTPUT FOR B9352 AND B9353 DISPLAYS.
CLEARTANK   - WRITES THE CORE TANK IN THE DISK OUTPUT TANK
                    WHEN A JOB IS SWAPPED.
COMM13   - HANDLES THE -13 COMMUNICATE, THE OBJECT
                    JOB INPUT REQUESTS.
COMM5   - HANDLES THE -5 COMMUNICATE, THE CANDE INPUT
                    REQUEST.
NOTIFCANDE   - PASSES INPUT TO CANDE.

TWXOUT IS THE PROCEDURE THROUGH WHICH ALL OUTPUT NOT GENERATED BY
LINE MAINTENANCE PROCEDURES IS DONE. IF THE STATION IS BROKEN OR
DISCONNECTED WHEN IT IS CALLED, TWXOUT EXITS, EFFECTIVELY THROWING
AWAY THE OUTPUT. NEXT IT CHECKS THE OUTPUT TANK FOR THE STATION.
IF THERE ARE ONLY NINE SEGMENTS LEFT, AND THE STATION IS ATTACHED TO
A JOB RUNNING ABOVE THE FENCE, THE JOB IS SWAPPED OUT.
(OLDWEIRDHAROLD SWAPS IT BACK IN AFTER ENOUGH IS DETANKED SO THAT
THE TANK IS LESS THAN HALF FULL). IF THERE ARE FEWER THAN SIX
SEGMENTS LEFT, CANDE IS PASSED A SHUT-UP EVENT. IF THE TANK IS
FULL, TWXOUT SLEEPS UNTIL THERE IS A FREE SEGMENT. AFTER THESE
TESTS, TWXOUT CALLS THE APPROPRIATE OUTPUT SCANNING PROCEDURE.
THEN, IF THERE IS NOTHING IN THE OUTPUT QUEUE OR TANKS FOR THIS
STATION, THE MESSAGE IS PASSED TO DCWRITE FOR TRANSMISSION. IF
THERE IS OUTPUT PENDING, OR IF THE STATION IS NOT A TWX, THE MESSAGE
IS TANKED AND TANKS IS UPDATED. IF THERE IS NO CORE TANK, THIS IS
DONE BY WRITING THE MESSAGE INTO THE NEXT SEGMENT ON DISK. IF THERE
IS A CORE TANK, TWXOUT CHECKS TO SEE IF THERE IS ROOM FOR THE
MESSAGE. IF SO IT MOVES IT INTO THE CORE TANK, OTHERWISE IT WRITES
THE CORE TANK TO DISK BEFORE MOVING THE MESSAGE INTO THE NEWLY
EMPTIED TANK.

OUTPUT SCANNING AND FORMATTING IS PERFORMED BY THE THREE DEVICE
ORIENTED PROCEDURES OUTRAN980, OUTRANBIDS, OUTRANTC. OUTRAN980
REPLACES CHARACTERS NOT IN THE 980 CHARACTER SET BY QUESTION MARKS,
INSERTS CARRIAGE RETURN AND LINE FEED CHARACTERS AS REQUESTED AND
THEN FOLDS THE LINE IF NECESSARY. OUTRANBIDS AND OUTRANTC INSET THE
NECESSARY UPPER AND LOWER CASE SHIFT CODES AND ALSO ADD CARRIAGE
RETURNS AND LINE FEEDS. IN ADDITION, OUTRANBIDS ADDS NULL
CHARACTERS FOR TIMING AND LINE ERASE CHARACTERS. ALL OF THESE
PROCEDURES RETURN TO TWXOUT THE REFORMATTED MESSAGE AND THE NUMBER
OF CHARACTERS IT CONTAINS.

COMM13 IS USED BY OBJECT JOBS TO REQUEST THE NEXT RECORD OF INPUT.
IF THE INPUT TANK FOR THE STATION IS EMPTY WHEN THE COMMUNICATE IS
DONE, COMM13 SENDS OUT A QUESTION MARK, CAUSES THE JOB TO BE SWAPPED
OUT AND SETS INPUTREADY SO THAT OLDWEIRDHAROLD WILL SWAP IT BACK IN
AS SOON AS HE PUTS SOME INPUT INTO THE TANKS. THIS CAUSES COMM13 TO

RUN AGAIN.

TO DETANK DATA, COMM13 FIRST READS IN A SEGMENT FROM DISK AND USES
INPUTSCAN TO MOVE THE NEXT RECORD, WITH TRANSLATION IF NECESSARY,
INTO AN ARRAY, FROM WHICH IT IS MOVED INTO THE PROGRAM-S BUFFER,
WITH ANY EXCESS TRUNCATED. IF THERE IS ROOM FOR MORE DATA IN THE
BUFFER, AND AN END OF RECORD HAS NOT BEEN REACHED, THE NEXT SEGMENT
IS READ AND PROCESSED. IF, ON THE OTHER HAND, THE BUFFER IS FILLED
BUT THE END OF RECORD DOES NOT OCCUR IN THE CURRENT SEGMENT,
ADDITIONAL SEGMENTS ARE SCANNED UNTIL THE END OF THE RECORD IS FOUND.
AFTER EACH SEGMENT IS READ AND PROCESSED, INPUTANKS IS UPDATED.

THE PROCEDURES COMM5 AND NOTIFYCANDE PROCESS THE CANDE INPUT QUEUE.
THE MCP VARIABLES FIRSTSEG AND FIRSTOFFSET ARE THE ARRAY AND INDEX
INTO IT OF THE HEAD OF THE QUEUE, AND LASTSEG AND LASTOFFSET ARE THE
TAIL OF THE QUEUE. DATA IS PASSED TO CANDE FROM FIRSTSEG BEGINNING
AT FIRSTOFFSET, AND OLDWEIRDHAROLD ADDS NEW INPUT AT LASTOFFSET OF
LASTSEG. IT IS POSSIBLE FOR FIRSTSEG TO EQUAL LASTSEG AND, IF THE
QUEUE IS EMPTY, FOR FIRSTOFFSET ALSO TO EQUAL LASTOFFSET.

NOTIFYCANDE, WHICH IS FORKED ONLY IF THE QUEUE IS NOT EMPTY, IS USED
TO PASS DATA TO CANDE. FIRST, IT WRITES FIRSTSEG INTO CANDE-S DISK
TANK, ASSURING THAT CANDE CAN ALSO FIND THE DATA ON DISK. IF
NECESSARY IT GETS A NEW ROW OF DISK AND SO INFORMS CANDE. NEXT, IT
QUEUES A DATA EVENT TO CANDE. THIS CONTAINS FIRSTOFFSET AND THE
NUMBER OF WORDS OF DATA WHICH BEGIN AT THAT INDEX. IF LASTSEG IS
NOT EQUAL TO FIRSTSEG, THIS IS ALL OF THE DATA LEFT IN FIRSTSEG, SO
NOTIFYCANDE CHANGES FIRSTSEG TO POINT TO THE NEXT ARRAY IN THE
QUEUE, INCREASES THE DISK ADDRESS FOR THE CANDE TANK BY ONE, AND
SETS FIRSTOFFSET TO 1. ON THE OTHER HAND, IF LASTSEG EQUALS
FIRSTSEG, THE NUMBER OF WORDS IS JUST THE DIFFERENCE BETWEEN
LASTOFFSET AND FIRSTOFFSET, SO, AFTER QUEUEING THE EVENT,
NOTIFYCANDE SETS FIRSTOFFSET EQUAL TO LASTOFFSET, THUS MAKING THE
QUEUE EMPTY. IF MORE INPUT IS RECEIVED, IT CAN BE PUT INTO THE SAME
ARRAY. NOTIFYCANDE WILL THEN WRITE THE NEW VERSION OVER THE OLD AND
PASS CANDE A DATA EVENT BEGINNING AT THE NEW FIRSTOFFSET. NOTE THAT
AS CANDE FALLS BEHIND FEWER DISK I/O-S ARE NEEDED, SINCE MORE DATA
IS PASSED WITH EACH EVENT. HOWEVER, MORE CORE IS USED TO HOLD THE
ARRAYS. WIN A FEW, LOSE A FEW.

WHEN CANDE HAS PROCESSED ALL THE DATA PASSED TO HER IN AN EVENT, SHE
DOES A -5 COMMUNICATE. IF THE QUEUE IS EMPTY, COMM5 SETS
CANDEINPUTREADY SO THAT OLDWEIRDHAROLD WILL FORK NOTIFYCANDE WHEN HE
GETS INPUT. IF THE QUEUE IS NOT EMPTY, COMM5 CHECKS TO SEE IF
CANDE-S TTYINPUT ARRAY DESCRIPTOR, WHICH IS PASSED BY THE
COMMUNICATE, POINTS TO FIRSTSEG. IF IT DOES NOT, THE AREA IT POINTS
TO, THE PREVIOUS ARRAY IN THE QUEUE, IS FORGOTTEN AND THE DESCRIPTOR
IS CHANGED TO POINT TO FIRSTSEG. IN THIS WAY THE ARRAYS ARE
FOGOTTEN AS CANDE FINISHES WITH THEM. NOTIFYCANDE IS THEN FORKED TO
PASS MORE DATA TO CANDE. NOTE THAT INTERLOCKS BETWEEN COMM5,
NOTIFYCANDE, AND OLDWEIRDHAROLD ARE AVOIDED BY JUDICIOUSLY CHOOSING
WHERE EACH PROCEDURE LOSES CONTROL.

SYSTEM/DISK, HALT/LOAD RECOVERY

EACH LINE DECLARED IN THE FILE SYSTEM/DISK IS ASSIGNED A LOGICAL
LINE NUMBER, IN ASCENDING ORDER OF TERMINAL UNIT - BUFFER NUMBER
STARTING WITH 1. IN ADDITION, EACH STATION DECLARED IN SYSTEM/DISK
IS ASSIGNED A UNIQUE STATION NUMBER, WHICH IS NOT NECESSARILY THE
SAME AS ITS LINE NUMBER. STATION NUMBERS ARE USED TO IDENTIFY
REMOTE USERS, FOR EXAMPLE IN THE "SS" AND "WU" MESSAGES AND IN THE
MCP VARIABLE"LOGLINE". LINE NUMBERS ARE USED ONLY BY THE DATACOM
HANDLERS IN THE MCP AND BY THE "CL" MESSAGE.

THE FIRST STATION DECLARED FOR A LINE HAS A STATION NUMBER EQUAL TO
ITS LINE NUMBER. ANY OTHER STATIONS ON THE LINE HAVE STATION
NUMBERS GREATER THAN THE HIGHEST LINE NUMBER. SUCH STATIONS ARE
ASSIGNED NUMBERS IN THE ORDER IN WHICH THEY APPEAR IN SYSTEM/DISK.
FOR INSTANCE, IF A SYSTEM HAD ONE STATION ON 1/0 AND 1/2, TWO
STATIONS ON 1/8 AND THREE ON 1/4, STATION NUMERS WOULD BE ASSIGNED
AS FOLLOWS:

ADAPTER	LINE NUMBER	STATION NUMBERS
1/0	1	1
1/2	2	2
1/4	3	3,5,6
1/8	4	4,7

NOT TR...
A-...
SEE INSIDE
FRONT COVE...
OF LISTING
BINDER FOR
MCP PATCH...
& MAINTENAN...

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

STATION INFORMATION

WORD 0	2:8	LINE NUMBER (IF DIFFERENT FROM STATION NUM.)
	10:3	STATION TYPE
	15:1	DIALEDUP
	17:3	NAKMAX
	20:6	(NUMBER OF LINES PER PAGE) - 2
	26:8	LINE LENGTH
	34:1	CHANGE-MODE FLAG FOR FIRST ADDRESS CHARACTER
	35:1	CHANGE-MODE FLAG FOR SECOND ADDRESS CHARACTER
	36:6	FIRST ADDRESS CHARACTER
	42:6	SECOND ADDRESS CHARACTER
WORD 1		USER CODE IF LOGGED ON,OTHERWISE 0
WORD 2		TIME OF LAST ACTIVITY
WORD 3		LENGTH TIME ALLOWED BEFORE DISCONNECT

LINE INFORMATION

WORD 4	2:1	DIRECT CONNECT FLAG
	3:3	LINE DISCIPLINE
	6:2	BUFFER SIZE (0=28,1=56,2=112)
	8:1	PING-PING FLAG
	9:4	TERMINAL UNIT NUMBER

```
14:4        BUFFER NUMBER
18:2        ADAPTER TYPE
```

THE FIELDS IN WORD 0 ARE IGNORED IF THEY ARE NOT APPLICABLE TO THE STATION TYPE. FOR EXAMPLE, ONLY THE STATION TYPE AND DIALED-UP FLAGS ARE SIGNIFICANT FOR TELETYPES.

WORD 4 IS ZERO FOR STATIONS WHOSE NUMBER IS GREATER THAT THE NUMBER OF LINES.

IN ADDITION, THE FOLLOWING INFORMATION IS KEPT IN RECORD 0:

```
WORDS 0-2            NOT USED
WORD 3              SPOWORD
WORD 4      24:8    NUMBER OF NON-TWX LINES (FOR TNAOG).
            32:8    NUMBER OF STATIONS
            40:8    NUMBER OF LINES
```

THE FILE IS INITIALIZED BY THE PROGRAM SYSDISK/MAKER, WHICH READS IN THE INFORMATION FOR THE 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 TSSMCP 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 ITS TABLES 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, STARTCANDY QUEUES UP A RESTART EVENT, (EVENT NUMBER 9).

INITIALLY, A RESTART EVENT IS TREATED LIKE AN ORDINARY DIAL IN EXCEPT THAT THE USER IS GIVEN MORE TIME IN WHICH TO LOG-ON SUCCESSFULLY. AFTER THE LOG-ON IS COMPLETED, CANDE COMPARES THE NEW USER CODE TO THE ONE IN THE RESTART EVENT. IF THEY MATCH, CANDE

CHECKS FOR THE FILE 1P <STATION NUMBER>/<USER CODE> AND , IF IT IS THERE, SEARCHES IT 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 DISK. INITIALIZE IS THEN RESTARTED WITH THE NEW FENCE VALUE.

IF THE FENCE IS IN A MISSING MEMORY MOD, THE MCP AUTOMATICALLY MOVES IT UP TO THE BOTTOM OF THE NEXT HIGHER AVAILABLE MOD.

OPERATION OF CANDE

THE COMMAND AND EDIT LANGUAGE PROCESSOR, CANDE/TSHARER, IS AN INDEPENDENT NORMAL STATE PROGRAM WHICH RUNS BELOW THE FENCE. IT IS DESIGNED TO ACCEPT, TO ANALYZE AND TO RESPOND TO A USERS INPUT IN A MANNER WHICH MINIMIZES THE "IDLE" TIME FOR THE OVERALL TIME SHARING SYSTEM. IT IS STARTED BY THE "CE" MESSAGE AND CONTINUES TO RUN INDEFINITELY THEREAFTER.

THE CANDE PROGRAM IS CODED IN TSPOL, AND IS COMPRISED OF A SERIES OF PROCEDURES. MOST OF THESE PROCEDURES ARE DIRECTLY RELATED TO A CANDE "VERB". FOR EXAMPLE, THE CANDE PROCEDURE "LISTIT" IS CALLED IN RESPONSE TO A USERS "LIST" OR "PRINT" REQUEST.

THESE "VERB RELATED" PROCEDURES ARE ONLY CALLED IN RESPONSE TO A USERS REQUEST. THERE ARE, HOWEVER, OTHER MORE GENERAL PROCEDURES, SUCH AS THE CONTROL ROUTINES, WHICH ARE CALLED FROM VARIOUS OTHER POINTS IN THE PROGRAM. CANDE DEPENDS ENTIRELY UPON THE TIME SHARING SYSTEM MCP FOR ALL I/O INCLUDING THAT FOR REMOTE TERMINALS AS WELL AS FOR THE PERIPHERALS AT THE COMPUTER SITE. INPUT FROM REMOTE USERS IS REQUESTED FROM THE MCP BY CANDE PERIODICALLY VIA A NEGATIVE COMMUNICATE, AND OUTPUT DIRECTED TO A REMOTE TERMINAL IS QUEUED UP BY MEANS OF ANOTHER NEGATIVE COMMUNICATE.

EACH USER IS IDENTIFIED BY HIS "LOGICAL LINE NUMBER", THAT IS, THE STATION NUMBER. THE TSS MCP PREFIXES EACH USERS INPUT WITH THE NUMBER OF THE LINE FROM WHICH THE INPUT CHARACTER STRING ORIGINATED. IN ADDITION, ALL TRANSACTIONS BETWEEN CANDE AND THE MCP ARE "TAGGED" WITH THIS LOGICAL LINE NUMBER IN ORDER TO IDENTIFY THE OWNER OF THE REQUEST.

CANDE MAINTAINS A "STACK" FOR EACH LOGICAL LINE IN THE SYSTEM. A USERS "STACK" IS ACTUALLY A ROW IN THE TWO DIMENSIONAL ARRAY "LINE" [0:MAXLINES, 0:LLWORDS]. "MAXLINES" AND "LLWORDS ARE NOMINALLY DEFINED AS 48 AND 100 RESPECTIVELY. HOWEVER, ONCE THE ROW HAS BEEN IDENTIFIED BY CANDE, IT IS PASSED (AS A FORMAL PARAMETER TO A PROCEDURE) AS THOUGH IT WERE A ONE-DIMENSIONAL ARRAY FOR EASE OF REFERENCE. THE "STACK" IS GENERALLY REFERRED TO WITHIN CANDE AS THE ARRAY "A".

A USERS "STACK" IS USED TO STORE INFORMATION PERTINENT TO THAT PARTICULAR USERS OPERATION. THE FIRST ELEVEN WORDS OF THE STACK ARE USED TO STORE "POINTERS" TO THE WORKFILE. THESE "POINTERS" CONTAIN INFORMATION REGARDING THE LOCATION AND NATURE OF THE MOST RECENTLY ENTERED WORKFILE RECORDS.

THE DEFINED QUANTITY "SBASE" IS USED TO MARK THE LOCATION OF THE BEGINNING OF THE "SCRATCH PAD" AREA OF THE STACK. THIS AREA IS USED FOR TEMPORARY STORAGE OF USERS REQUESTS AND FOR STORAGE OF INTERMEDIATE RESULTS.

THE TOP PORTION OF THE STACK IS USED TO STORE "CONTROL" INFORMATION
FOR THE USER. THE CANDE PROGRAM IS DESIGNED SO THAT IT IS NOT
NECESSARY TO SUSPEND PROCESSING WHILE WAITING FOR I/O OR FOR A JOB
TO FINISH RUNNING. WHENEVER IT IS NECESSARY TO WAIT FOR A RESULT
BEFORE CONTINUING ON WITH A USERS REQUEST, CANDE STORES ALL THE
INFORMATION NECESSARY TO RE-START THE USERS JOB IN THE TOP OF THE
STACK, AND MOVES ON TO ANOTHER USERS REQUEST.

THE PROCESS OF EXECUTING A USERS REQUEST MAY REQUIRE THAT ONE
PROCEDURE CALL ON A SECOND PROCEDURE, AND THAT THE SECOND PROCEDURE
CALL ON A THIRD PROCEDURE, ETC. SINCE ANY ONE OF THESE PROCEDURES
MAY BE REQUIRED TO WAIT FOR I/O OR FOR SOME INTERMEDIATE RESULT, IT
IS NOT FEASIBLE FOR THESE PROCEDURES TO CALL ON ONE ANOTHER
DIRECTLY, AS THE MOST RECENTLY CALLED PROCEDURE WOULD "FORGET" WHERE
IT WAS CALLED FROM.

THEREFORE, A STACK "REGISTER" (I.E., A VARIABLE USED TO INDICATE
THE CURRENT STACK WORD BEING USED FOR CONTROL PURPOSES) AND A STACK
"CONTROL WORD" (A WORD IN THE TOP OF THE STACK WHICH CONTAINS
INFORMATION NECESSARY TO RE-START THE PROGRAM, SHOULD IT BE
SUSPENDED) ARE USED TO KEEP TRACK OF THE STATUS OF A USERS REQUEST.

THE "REGISTER" USED TO "POINT" TO THE STACK CONTROL WORD IN USE IS
CALLED "PREVRCW". "PREVRCW" IS ASSIGNED A VALUE OF "LLWORDS" (THE
"TOP" OF THE STACK) WHENEVER THE USERS LINE IS IDLE, AND IS
DECREMENTED BY 1 EACH TIME THAT A NEW PROCEDURE IS GIVEN CONTROL.
WHENEVER A WAIT IS REQUIRED, THE VALUE OF "PREVRCW" IS SAVED IN THE
USERS STACK WORD "WAITINGSCW".

THEN, WHEN THE USERS LINE IS AGAIN READY TO CONTINUE, THE VALUE OF
"PREVRCW" IS OBTAINED FROM "WAITINGSCW", AND THE NECESSARY RE-START
INFORMATION IS OBTAINED FROM THE STACK CONTROL WORD POINTED TO BY
"PREVRCW".

CANDE INFORMATION TABLES

CANDE MAINTAINS SEVERAL INFORMATION TABLES, THE PRIMARY USE OF WHICH IS TO PROVIDE THE "SYNTAX" OF THE COMMAND AND EDIT LANGUAGE. THESE ARE THE "VERBTABLE", THE "TYPETABLE" AND THE "RESWRDTABLE". THE "VERBTABLE" IS MADE UP OF TWO WORD ENTRIES, THE FIRST WORD BEING THE "NAME" OF THE C & E VERB, AND THE SECOND WORD A NUMBER CODE FOR THAT VERB. THE LENGTH OF THE "VERBTABLE" IS SPECIFIED BY THE DEFINED VALUE "NUMOFVERBS" ([0:2 X NUMOFVERBS]).

THE FIRST WORD IN A "VERBTABLE" ENTRY, WHICH CONTAINS THE CHARACTER STRING CORRESPONDING TO THE "NAME" OF THE VERB, IS MADE UP OF TWO PARTS: THE MINIMUM LENGTH OF THE VERB (IN THE FIRST CHARACTER), AND THE FULL "NAME" OF THE VERB (LEFT JUSTIFIED, WITH TRAILING BLANKS IF FEWER THAN 7 CHARACTERS IN LENGTH). THE MINIMUM LENGTH IN THE FIRST CHARACTER SPECIFIES THE MINIMUM NUMBER OF CHARACTERS WHICH THE USER MUST INPUT TO THE SYSTEM TO IDENTIFY THAT COMMAND. IF THE USER CHOOSES TO USE MORE THAN THIS NUMBER OF CHARACTERS, HE MAY DO SO, BUT HE MUST THEN USE THE SAME SPELLING AS IS SHOWN IN THE "VERBTABLE" WORD.

THE SECOND WORD IN A "VERBTABLE" ENTRY IS MADE UP OF THREE PARTS. THE [33:15] FIELD OF THIS WORD CONTAINS THE OCTAL REPRESENTATION OF THE NUMBER (I.E., THE "CASE" NUMBER FOR THE CASE STATEMENT IN PROCEDURE "HANDLETHISLINE") OF THIS VERB. THE [9:9] FIELD IS A "LINK" REFERRING BACK TO THE LOCATION OF THIS VERB IN THE "VERBTABLE". THE LINK IS USED WHEN IT IS NECESSARY TO CONVERT BACK FROM THE OCTAL REPRESENTATION OF THE VERB, TO THE CHARACTER REPRESENTATION. THE [6:3] FIELD IS USED AS A FLAG FOR THE CANDE "SCANNER", AND MARKS THIS AS A VERB REQUIRING SPECIAL TREATMENT WHEN GREATER THAN ZERO. THE REMAINING FIELDS OF THIS WORD ARE NOT CURRENTLY USED.

THE "TYPETABLE" IS MADE UP OF SINGLE WORD ENTRIES, AND ITS LENGTH IS DETERMINED BY THE DEFINED VALUE "NUMOFTYPES". THE "TYPES" REFER TO THE VALID FILE TYPES, AND CANDE REFERENCES THESE TYPES BY THEIR POSITION IN THE TYPETABLE.

THE "RESWRDTABLE" ALSO CONSISTS OF ONE WORD ENTRIES, AND ITS LENGTH IS DETERMINED BY THE DEFINED VALUE "NUMOFRESWRDS". A USER IS NOT PERMITTED TO USE A RESERVED WORD FOR FILE NAMES, AND CANDE CHECKS THIS TABLE TO DETERMINE THE VALIDITY OF A USERS REQUEST. THE POSITION OF THE RESERVED WORDS IN THE "RESWRDTABLE" IS OF NO CONSEQUENCE SINCE CANDE ONLY CHECKS FOR A WORDS PRESENCE OR ABSENCE.

EVENTS

SINCE CANDE IS A NORMAL STATE PROGRAM, IT MUST GET INFORMATION ABOUT
THE STATUS OF JOBS AND LINES FROM THE TSSMCP. THIS IS DONE IN THE
FORM OF EVENTS, WHICH ARE QUEUED BY THE TSSMCP IN THE ORDER OF THEIR
OCCURRENCE AND ARE PASSED TO CANDE WHEN IT REQUESTS ONE VIA A
COMMUNICATE. AN EVENT IS NORMALLY A FIVE WORD ARRAY, BUT CAN BE
EXTENDED TO AS MANY AS 20 WORDS IN SPECIAL CASES, SUCH AS END-OF-JOB
EVENTS THAT CONTAIN "CHAIN" REQUESTS. THE FIRST WORD CONTAINS THE
STATION 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 BELOW.

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

    18:7   EVENT NUMBER
    25:8   STATION NUMBER

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

EVENT		WORD	CONTENTS
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 MESAGE/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 ADDRESSS 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	TSSMCP MESSAGE	1	MESSAGE TYPE (SEE BELOW)
		2-4	NOT USED
15	PSEUDO DATA EVENT	1	ZERO, IF THIS MESSAGE CONSISTS

```
        FOR ? MESSAGE                        SOLELY OF A QUESTION MARK.
                                             NON-ZERO,  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 VALUE 28
  (PROCEDURE "DISKREAD") IF IT WAS A READ, OR A 13, WHICH IS
  DISCARDED, IF IT WAS A WRITE.

THE TSSMCP 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
```

USERS INPUT TO THE TIME SHARING SYSTEM

THE TSSMCP 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 THE TSSMCP HAS ACCUMULATED AND SCANNED A COMPLETE RECORD OF
INPUT FOR CANDE, IT PLACES A CONTROL WORD IN FRONT OF THE RECORD.
THIS WORD CONTAINS THE ACTUAL CHARACTER COUNT (LENGTH OF THE INPUT
STRING) IN THE [33:15] FIELD, THE LOGICAL LINE NUMBER (THE NUMBER
OF THE LINE ORIGINATING THE INPUT) IN THE [10:8] FIELD, AND, IF THE
INPUT WAS A "SPECIAL" MESSAGE, I.E., A MESSAGE BEGINNING WITH A
QUESTION MARK, A "1" IN BIT [1:1].

THE MCP THEN DETERMINES WHETHER OR NOT THERE IS ENOUGH ROOM IN THE
CURRENT TANK SEGMENT TO STORE THIS INPUT AND, IF THERE IS ENOUGH
ROOM, PLACES THE INPUT ALONG WITH THE CONTROL WORD IN THE SEGMENT.
IF THERE IS INSUFFICIENT ROOM IN THE CURRENT TANK SEGMENT, THE MCP
THEN QUEUES THE CURRENT TANK SEGMENT, OBTAINS ADDITIONAL CORE SPACE
FOR A NEW SEGMENT, AND PLACES THE LAST INPUT IN THE NEW TANK SEGMENT.

CANDE USES THE SAVE ARRAY "TTYINPUT", THE VARIABLES "DATADISKLOC",
"DATAWORDS", "DATAOFFSETT", AND THE BOOLEAN VARIABLES "NODATA",
"NODATAREQUESTED" AND "WECANUSEDATA" TO OBTAIN INPUT DATA FROM THE
MCP.

WHEN THE CANDE PROGRAM IS FIRST STARTED UP, CERTAIN TABLES AND
VARIABLES ARE INITIALIZED. THEN, THE PROCEDURE "RUNCANDE" IS CALLED
AND RUNS CONTINUOUSLY UNTIL CANDE IS DS-ED. IT IS IN PROCEDURE
"RUNCANDE" THAT INPUT DATA IS REQUESTED FROM THE MCP.

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 FRUM 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 BOOLEAN VARIABLE "NODATA" IS SET TO A VALUE OF TRUE WHEN THE
CANDE PROGRAM IS FIRST STARTED UP, AND WHENEVER CANDE RUNS OUT OF
DATA. THE BOOLEAN VARIABLE "WECANUSEDATA" IS ALMOST ALWAYS SET TO A
VALUE OF TRUE - IT IS FALSE ONLY WHEN CANDE HAS QUEUED UP A BACK LOG
OF USERS REQUESTS, OR WHEN THE MCP TANK FILES HAVE BEEN FILLED.

WHEN THE BOOLEAN VARIABLES "NODATA" AND "WECANUSEDATA" ARE BOTH
TRUE, A REQUEST WILL BE MADE TO THE MCP FOR ADDITIONAL DATA (THIS
IS DONE IN PROCEDURE "RUNCANDE"). THE MCP THEN PASSES CANDE A
"DATA" EVENT, INFORMING CANDE OF THE LOCATION (DISK ADDRESS) OF
THE NEXT SEGMENT, THE OFFSET, I.E., STARTING POSITION OF THE NEXT
INPUT WITHIN THE SEGMENT, AND THE TOTAL LENGTH OF THE INPUT WITHIN
THE SEGMENT. THESE VALUES ARE ASSIGNED TO CANDE VARIABLES
"DATADISKLOC","DATAOFFSETT" AND "DATAWORDS" IN PROCEDURE "INPUTDATA".

THE BOOLEAN VARIABLE "NODATAREQUESTED" IS SET TO A VALUE OF FALSE AT
THE TIME THAT CANDE REQUESTS DATA FROM THE MCP, AND IS RESET TO A
VALUE OF TRUE WHEN THE MCP RESPONDS WITH A "DATA" EVENT. THIS
VARIABLE PREVENTS CANDE FROM MAKING ANOTHER REQUEST FOR DATA FROM
THE MCP BEFORE THE MCP HAS RESPONDED TO THE FIRST REQUEST.

AT THE TIME THAT THE MCP PASSES CANDE A "DATA" EVENT, THE CORE COPY
OF THE NEXT TANK SEGMENT IS PLACED IN CANDES "TTYINPUT" ARRAY (
ACTUALLY, THE MCP SIMPLY CHANGES THE CORE ADDRESS OF THIS ARRAY).
IF THE "DATA" EVENT RESULTS IN THE TRANSFER OF ADDITIONAL VALID DATA
TO CANDE, THE VARIABLE "NODATA" IS RESET TO A VALUE OF FALSE.

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.

THERE ARE MANY TYPES OF "EVENTS" WHICH CAN BE PASSED TO CANDE BY THE
MCP. SOME EVENTS, SUCH AS THE "DATA" EVENT DESCRIBED ABOVE, ARE
QUEUED UP IN RESPONSE TO A REQUEST BY CANDE. OTHER EVENTS, SUCH AS
A "DIAL-UP" EVENT (QUEUED BY THE MCP WHEN A USER DIALS INTO THE
TIME SHARING SYSTEM) ARE PASSED TO CANDE BY THE MCP AUTOMATICALLY.
EACH EVENT IS "TAGGED" WITH THE NUMBER OF THE LOGICAL LINE WITH
WHICH THE EVENT IS ASSOCIATED, AND WITH AN "EVENT NUMBER" INDICATING
THE NATURE OF THE COMMUNICATION.

EACH TIME THAT PROCEDURE "RUNCANDE" REGAINS CONTROL, THE MCP IS
INTERROGATED TO DETERMINE WHETHER ANY "EVENTS" HAVE BEEN QUEUED UP
FOR CANDE. THIS INTERROGATE FUNCTION IS PERFORMED BY PROCEDURE
"NEXTEVENT". IF AN EVENT HAS, IN FACT, BEEN QUEUED FOR CANDE, THE
MCP RETURNS A VALUE OF 1 TO PARAMETER "WAITBIT" AND RETURNS THE
EVENT TO ARRAY "EVENT". IF AN EVENT HAS NOT BEEN QUEUED FOR CANDE,
AND IF THERE IS DATA IN CANDES "TTYINPUT" ARRAY, THE PROCEDURE
"BUILDADATAEVENT" IS CALLED WHICH BUILDS A PSEUDO EVENT IN CANDES
ARRAY "EVENTS".

ONCE AN EVENT HAS BEEN OBTAINED, THE EVENT NUMBER AND THE LOGICAL
LINE ASSOCIATED WITH THE EVENT ARE OBTAINED AND ASSIGNED TO CANDE
VARIABLES "REASON" AND "LL". PROCEDURE "RUNCANDE" THEN CALLS
PROCEDURE "HANDLETHISLINE", PASSING THE ARRAY ROW LINE[LL,*], AND
LINE CONTROL WORD LLCONTROL[LL] AS PARAMETERS. PROCEDURE
"HANDLETHISLINE", IN CONJUNCTION WITH PROCEDURE "ANOTHEROPERATION",
WILL THEN TRANSFER CONTROL TO THE APPROPRIATE CANDE PROCEDURES,
DEPENDING UPON THE TYPE OF EVENT OBTAINED.

THE VALUE OF "REASON" (THE EVENT NUMBER) IS USED TO TRANSFER
CONTROL TO THE PROPER CANDE PROCEDURE. FOR EXAMPLE, A "DIAL-UP"
EVENT IS NUMBERED AS EVENT #1, AND CASE 1 IN PROCEDURE
HANDLETHISLINE CALLS ON CANDE PROCEDURE "HITHERE" (THE LOG-IN
ROUTINE). THE SELECTION OF A SUBROUTINE BASED ON AN EVENT NUMBER
PROVIDES THE MEANS BY WHICH CANDE CAN BE "INSTRUCTED" TO CARRY OUT A
USERS REQUEST.

THE CANDE "STACK" MECHANISM

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.

THE FIRST STEP IN ANALYSING A USERS REQUEST IS TO "SCAN" THE INPUT
CHARACTER STRING AS RECEIVED FROM THE MCP. THE INITIAL STEP IN THIS
PROCESS IS THE TRANSFER OF A MESSAGE FROM CANDES "TTYINPUT" ARRAY TO
CANDES "WORK" ARRAY. THIS IS ACCOMPLISHED IN PROCEDURE
"BUILDADATAEVENT". THE "TTYINPUT" ARRAY CONTAINS ONE OR MORE INPUTS
FROM VARIOUS USERS IN THE SYSTEM. THE MANNER IN WHICH CANDE OBTAINS
THIS SEGMENT FROM THE MCP WAS DISCUSSED ABOVE. WHEN PROCEDURE
"BUILDADATAEVENT" IS CALLED, CANDE FIRST LOOKS AT THE CONTROL WORD
PRECEDING THE ACTUAL USERS INPUT. THE POSITION OF THIS CONTROL WORD
IN THE "TTYINPUT" ARRAY IS DETERMINED FROM THE VALUE OF
"DATAOFFSETT". THE LOGICAL LINE (FIELD [10:8]), THE MESSAGE SIZE
(FIELD [33:15]), AND THE SPECIAL MESSAGE FLAG (FIELD [1:1]) ARE
OBTAINED FROM THE CONTROL WORD. THEN, THE ACTUAL MESSAGE IS
TRANSFERRED TO ARRAY "WORK", STARTING AT WORK[1], AND WORK[0] IS
ASSIGNED THE NUMBER OF CHARACTERS IN THE MESSAGE.

THE VALUE OF "DATAOFFSETT" IS INCREMENTED TO POINT TO THE NEXT INPUT
MESSAGE IN THE "TTYINPUT" ARRAY, AND THE VALUE OF "DATAWORDS" IS
DECREASED BY THE LENGTH OF THE MESSAGE JUST TRANSFERRED. IF THIS
RESULTS IN THE VALUE OF "DATAWORDS" (THE TOTAL LENGTH OF ALL
MESSAGES IN THE "TTYINPUT" ARRAY) BEING REDUCED TO ZERO, THE
BOOLEAN VARIABLE "NODATA" IS AGAIN SET TO A VALUE OF TRUE, AND WILL
CAUSE CANDE TO ASK THE MCP FOR ANOTHER DATA SEGMENT WHEN PROCEDURE
"RUNCANDE" AGAIN REGAINS CONTROL.

PROCEDURE "BUILDADATAEVENT" THEN BUILDS A PSEUDO EVENT BY ASSIGNING
A VALUE OF 26 (25 IF THE SPECIAL MESSAGE FLAG IS ON) TO "REASON",
AND ASSIGNS THE LOGICAL LINE FOR THIS INPUT TO "LL".

THE CANDE PROCEDURE "SCAN" IS THE PROCEDURE RESPONSIBLE FOR BREAKING
THE USERS INPUT MESSAGE INTO "TOKENS". THESE "TOKENS" ARE STORED IN
THE SCRATCH PAD AREA OF THE USERS STACK (THE AREA ABOVE "SBASE").
THE "SCAN" PROCEDURE IS PASSED FOUR PARAMETERS : A STARTING
"POINTER" (THE STARTING ADDRESS OF THE NEXT CHARACTER IN THE "WORK"
ARRAY TO BE SCANNED), A TERMINAL "POINTER" (WHICH WILL CONTAIN THE
ADDRESS OF THE NEXT CHARACTER TO BE SCANNED WHEN THE PROCEDURE IS
EXITED), THE STACK "WORD" IN WHICH WILL BE PLACED THE "TOKEN"
EXTRACTED BY THE "SCAN" ROUTINE, AND A COMPARISON VALUE WHICH IS
USED TO DECIDE HOW TO BREAK THE INPUT CHARACTER STRINGS INTO TOKENS.

THE "SCAN" ROUTINE RECOGNIZES THREE TYPES OF TOKENS:

    A) AN ALPHA STRING, 7 OR FEWER CHARACTERS IN LENGTH, WHICH
       BEGINS WITH A LETTER, AND IS COMPRISED ENTIRELY OF ALPHA
       CHARACTERS WHICH ARE NOT GREATER THAN THE COMPARISON VALUE
       (PASSED AS A PARAMETER TO THE PROCEDURE). WHEN A STRING
       OF THIS TYPE IS FOUND TO BE LONGER THAN 7 CHARACTERS, THE
       EXCESS CHARACTERS ARE THROWN AWAY. THE "TOKEN" THUS
       DEFINED IS TRANSFERRED TO THE OUTPUT STACK WORD. THE RIGHT
       MOST 7 CHARACTERS OF THE OUTPUT WORD ARE USED TO STORE THE
       "TOKEN" WHICH IS PLACED LEFT JUSTIFIED IN THIS FIELD WITH
       TRAILING BLANKS. THE LENGTH OF THE "TOKEN" IS PLACED IN
       THE [3:3] FIELD OF THE STACK WORD, AND THE [1:2] FIELD IS
       ASSIGNED A VALUE OF ZERO.

    B) A DIGIT STRING, 8 OR FEWER CHARACTERS IN LENGTH, COMPRISED
       ENTIRELY OF DIGITS. THE "SCAN" ROUTINE CONVERTS THIS
       DIGIT STRING TO AN OCTAL VALUE WHICH IS TRANSFERRED TO THE
       STACK WORD. IF THE STRING IS LONGER THAN 8 CHARACTERS,
       THE REMAINING CHARACTERS ARE **NOT** THROWN AWAY, BUT ARE
       LEFT IN ARRAY "WORK" TO BE SCANNED AGAIN WHEN THE "SCAN"
       ROUTINE IS AGAIN CALLED.

    C) A SPECIAL (NON-ALPHA) CHARACTER. THE "SCAN" ROUTINE PLACES
       THIS CHARACTER IN THE STACK WORD, RIGHT JUSTIFIED, WITH
       LEADING ZEROS, AND PLACES THE VALUE "1" IN THE [1:2] FIELD
       OF THE STACK WORD (THIS MARKS THE WORD AS CONTAINING A
       SPECIAL CHARACTER).

ANOTHER ROUTINE, "SCANONE", IS USED BY CANDE TO DETERMINE WHAT THE
NEXT CHARACTER IN THE "WORK" ARRAY IS. ITS MAIN FUNCTION IS TO
DETERMINE WHETHER THE NEXT "TOKEN" IS A GROUP MARK (END OF RECORD)
OR NOT.

CANDE PROCEDURE "SCANNER" CALLS ON PROCEDURES "SCAN" AND "SCANONE",
PLACING THE "TOKENS" THUS OBTAINED IN THE USERS "STACK", TAKING CARE
OF "-" AND "+" SIGNS, AND CHECKING SYNTAX (WHETHER THE INPUT STARTS
WITH A LEGITIMATE VERB [FROM THE "VERBTABLE"] OR A SEQUENCE NUMBER).
THE "SCANNER" ROUTINE PLACES "-" AND "+" SIGNS, WHICH PRECEDE A
DIGIT STRING, IN THE [1:2] FIELD OF THE STACK WORD CONTAINING THE
OCTAL VALUE OF THE DIGIT STRING.

THE NET RESULT OF THE COMBINED EFFORTS OF THE "SCAN" AND "SCANNER"
ROUTINES, IS THAT THE NATURE OF ANY STACK WORD CAN BE IDENTIFIED BY
OBSERVING THE [1:5] FIELD OF THAT WORD.

    [1:5] = 0 (AN UNSIGNED OCTAL NUMBER)
    [1:2] = 0 , [3:3] GTR 0 (A CHARACTER STRING, [3:3] LONG)
    [1:2] = 1 (A SPECIAL CHARACTER)
    [1:2] = 2 (A NEGATIVE OCTAL NUMBER)
    [1:2] = 3 (A "+" SIGNED OCTAL NUMBER)

THE "SCANNER" ROUTINE ASSEMBLES THESE "STACK" WORDS STARTING AT A

POSITION MARKED BY THE VALUE OF CANDES VARIABLE "BASE". THE STACK
[BASE] WORD IS MADE UP OF THREE PARTS: THE [33:15] FIELD CONTAINS A
NUMBER CORRESPONDING TO THE NUMBER OF STACK "WORDS" PLACED ABOVE THE
"BASE" BY THE "SCANNER", THE [18:15] FIELD CONTAINS THE NUMBER OF
"VERBS" WHICH THE "SCANNER" FOUND IN THE USERS INPUT STRING, AND THE
[3:3] FIELD CONTAINS A DIGIT CODE CORRESPONDING TO THE TYPE OF INPUT
CONTAINED IN THE STACK. THE CODES FOR THE [3:3] FIELD ARE:

        0 = COMMENT (IGNORED BY CANDE)
        1 = INPUT BEGINS WITH A VERB
        2 = INPUT BEGINS WITH A SEQUENCE NUMBER
        3 = INPUT BEGINS WITH A SPECIAL (NON-ALPHA) CHARACTER
        4 = INPUT IS AN "SS" MESSAGE

THE "SCANNER" PLACES A "VERBWORD" IN THE STACK WHICH IS ALSO
COMPRISED OF THREE PARTS: THE [33:15] FIELD CONTAINS THE INTERNAL
DIGIT CODE CORRESPONDING TO THE VERB, THE [18:15] FIELD CONTAINS THE
NUMBER OF PARAMETERS ASSOCIATED WITH THE VERB, AND THE [9:9] FIELD
CONTAINS THE POSITION OF THE VERB NAME IN THE "VERBTABLE". THE
PARAMETERS TO THE VERBS ARE PLACED IN THE STACK FOLLOWING THE VERB
WORD. THESE PARAMETERS MAY BE OCTAL NUMBERS OR CHARACTER STRINGS.
FIGURE I ILLUSTRATES THE CANDE STACK THAT WOULD BE GENERATED FROM
THE USERS INPUT: MAKE TEST ALGOL; SEQ 1000 + 50

FIGURE I

STACK GENERATED BY THE REQUEST: MAKE TEST ALGOL; SEQ 1000+50?

```
   BASE       BASE+1    BASE+2   BASE+3   BASE+4     BASE+5      BASE+6
    :           :          :        :        :          :           :
 XXXXXXXX    XXXXXXXX    4TEST 5ALGOL   XXXXXXXX    XXXXXXXX    XXXXXXXX
 .   .   .    .   .   .               .   .   .    .    .    .    .
 .   .   .    .   .   .               .   .   .    .    .    .    .
 .   .   .    .   .   .               .   .   .    .    .    .  ...[21:27]=50
 .   .   .    .   .   .               .   .   .    .    .    .
 .   .   .    .   .   .               .   .   .    .    .  ...[1:2]=3 (+ NO.)
 .   .   .    .   .   .               .   .   .    .    .
 .   .   .    .   .   .               .   .   .    .  ...[21:27]=1000
 .   .   .    .   .   .               .   .   .    .
 .   .   .    .   .   .               .   .   .  ...[1:2]=0 (UNSIGNED NO.)
 .   .   .    .   .   .               .   .   .
 .   .   .    .   .   .               .   .  ...[33:15]=53 (CODE FOR "SEQ")
 .   .   .    .   .   .               .   .
 .   .   .    .   .   .               .  ...[18:15]=2 (NO.OF PARAMS)
 .   .   .    .   .   .               .
 .   .   .    .   .   .             ...[9:9]=6 (LOCATION IN "VERBTABLE")
 .   .   .    .   .   .
 .   .   .    .   .  ...[33:15]=55 (CODE FOR "MAKE")
 .   .   .    .   .
 .   .   .    .  ...[18:15]=2 (NO.OF PARAMS TO "MAKE")
 .   .   .    .
 .   .   .  ...[9:9]=12 (LOCATION IN "VERBTABLE")
 .   .   .
 .   .  ...[33:15]=6 (NO.OF WORDS ABOVE "BASE")
 .   .
 .  ...[18:15]=2 (NO.OF "VERBS" IN STACK)
 .
...[3:3]=1 (TYPE OF STACK , 1="VERB" )
```

THIS REQUEST TO CREATE AN ALGOL FILE "TEST", AND TO ENTER SEQUENCE
MODE, WOULD BE PASSED TO CANDE BY THE MCP IN A "DATA" EVENT. CANDE
WOULD SUBSEQUENTLY CALL "BUILDADATAEVENT", GENERATING A PSEUDO TYPE
26 EVENT, AND PLACING THE USERS INPUT STRING IN ARRAY "WORK". CASE
26 (IN THE CASE STATEMENT IN PROCEDURE "HANDLETHISLINE") CALLS ON
PROCEDURE "INPUT", AND PROCEDURE "INPUT" CALLS ON PROCEDURE
"SCANNER" TO GENERATE THE "STACK" ILLUSTRATED ABOVE.

THE "TYPE" OF INPUT (VERB, WORKFILE RECORD, ETC.), WHICH IS
RETURNED IN STACK[BASE].[3:3] BY THE "SCANNER", INFORMS PROCEDURE
"INPUT" WHAT COURSE OF ACTION TO TAKE. IN THIS INSTANCE, CONTROL
WILL BE PASSED TO CASE 55, A CALL ON CANDE PROCEDURE "MAKENEWFILE".

"IAM", "WANT", AND "WAITFOR"

CANDE USES THREE VARIABLES - "IAM","WANT" AND "WAITFOR"- TO TELL
PROCEDURE "ANOTHEROPERATION" HOW TO CONTINUE PROCESSING A USERS
REQUEST. PROCEDURE "ANOTHEROPERATION" IS THE ROUTINE USED TO CREATE
AND/OR RETRIEVE THE "STACK CONTROL WORDS" THAT ARE USED TO
"REMEMBER" THE CALLING SEQUENCE OF CANDE PROCEDURES.

 IAM   INDICATES WHETHER OR NOT THE PROCEDURE WISHES TO REGAIN
     CONTROL.
 WANT   SPECIFIES WHICH OTHER PROCEDURE, IF ANY, SHOULD GET
     CONTROL NEXT.
 WAITFOR SPECIFIES WHAT EVENT, IF ANY, MUST OCCUR BEFORE CONTROL
     IS 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 BELOW.

IAM	WANT	WAITFOR	ACTION
=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

THE VARIABLE "IAM" ALWAYS REFERS TO THE PROCEDURE NUMBER (IN THE
CASE STATEMENT OF PROCEDURE "HANDLETHISLINE") WHICH IS CURRENTLY IN
CONTROL. THUS THE "MAKE" VERB, WHICH IS EVENT NUMBER 55, WOULD
RESULT IN THE VARIABLE "IAM" BEING ASSIGNED A VALUE OF 55. THIS IS
ACCOMPLISHED IN THE FOLLOWING MANNER:

  1) THE VALUE OF "IAM" IS TESTED AND, IF IT IS NOT ZERO,
   A "RETURN CONTROL WORD" IS STORED IN THE USERS STACK.
   THE CONTROL WORD IS STORED BY:

    A) DECREMENTING THE VALUE OF "PREVRCW" BY 1

    B) STORING ALL THE VALUES REQUIRED FOR RE-START IN
     STACK[PREVRCW]. THE VALUE OF "IAM" IS ONE OF THE
     VALUES STORED IN THIS CONTROL WORD.

  2) THE VALUE OF "WANT" IS THEN ASSIGNED TO VARIABLE "IAM".

SINCE THE VALUE OF "WANT" IS ASSIGNED TO "IAM" *AFTER* THE CURRENT VALUE OF "IAM" IS STORED, ANY NON-ZERO VALUE FOR "WANT" WILL ALWAYS CAUSE CONTROL TO BE TRANSFERRED FROM THE PROCEDURE SPECIFIED BY "IAM" TO THAT SPECIFIED BY "WANT".

3) THE VALUE OF "WAITFOR" IS TESTED, AND, IF NON-ZERO, ANOTHER CONTROL WORD IS STORED FOR THE CURRENT VALUE OF "IAM" (UNLESS THE VALUE OF "IAM" IS CURRENTLY ZERO, IN WHICH CASE THE CONTROL WORD IS *NOT* STORED). THE VALUES OF "PREVRCW" AND "WAITFOR" ARE THEN STORED IN THE LINES VARIABLE "WAITINGSCW", AND THE LINE IS PLACED IN A WAITING CONDITION.

WHEN THE LINE IS IN THIS "WAITING" CONDITION, EACH EVENT THAT OCCURS FOR THE LINE IS TESTED AGAINST THE VALUE OF "WAITFOR" THAT IS STORED IN VARIABLE "WAITINGSCW". WHEN THE NUMBER OF THE OCCURRING EVENT IS THE SAME AS THE NUMBER STORED, THE VALUE OF "PREVRCW" IS COPIED FROM "WAITINGSCW" AND THE USERS JOB IS RE-STARTED, BASED ON THE CONTROL INFORMATION STORED IN STACK[PREVRCW].

FIGURE II ILLUSTRATES THE MANNER IN WHICH PROCEDURE "ANOTHEROPERATION" FUNCTIONS.

FIGURE II.

OPERATION OF PROCEDURE "ANOTHEROPERATION".

```
     L1:
       .
       .
.............
.          .
.IAM NEQ 0........>.........
.          .              .
.............             .
       .            RCW := RETURN CONTROL WORD
       .            STORE RCW IN STACK
   DOING:=0              .
       .                 .
       ..............<.......:
BASE:=BASE+TEMP+PARAM
SREG:=BASE
IAM:=WANT
SCW := STACK CONTROL WORD
       .
       .
       .
.............
.          .
.WANT NEQ 0 ......>.........
.          .              .
.............             .
       .                 .
       .            STACK[BASE] := PARAM
       .            DOING:=WANT:=0
       .                 .
       ..............<........:
       .
TEMP:=PARAM:=0
       .
       .
................
.             .
.WAITFOR NEQ 0  ................>...............
.             .                               .
................                              .
       .                                      .
       .                                      .
ANOTHEROPERATION:=NOT ALLDONE                 .
       .                                      .
       .                                      .
```

2-72

```
      .                                    .
      .                                    .
      .                                    .
  .............................            .
  .NOT ALLDONE AND IAM=0 ...>.....         .
  .                            .           .
  ...........................  .           .
                            .  .           .
                            .  .           .
                            .  RECALL RCW FROM STK  .
                            .  SCW:=STACK CNTRL.WORD .
                            .  WAITINGSCW:=SCW       .
                            .  .           .
                            .  .      .............
                            .  .      .           .
                            .  .      .IAM NEQ 0  .... (TO L1)..
                            .  .      .           .
                            .  .      .............
                            .  .      .
                            .  .      .
                            .  .      ANOTHEROPERATION:=FALSE
                            .  .      WAITINGSCW := SCW
                            .  .           .
                            .  .           .
  .................<...................<......:
  .
  .
  .
END PROCEDURE ANOTHEROPERATON
```

THE "BASE" ALWAYS POINTS TO THE STACK LOCATION THAT IS CURRENTLY
BEING USED TO STORE THE PARAMETERS AND INTERMEDIATE RESULTS
ASSOCIATED WITH THE PROCESSING OF A USERS REQUEST. THE "BASE" IS
ALMOST ALWAYS INCREMENTED WHEN ONE PROCEDURE TRANSFERS CONTROL TO
ANOTHER PROCEDURE (BY ASSIGNING APPROPRIATE VALUES TO "WANT") SO
THAT THE INTERMEDIATE RESULTS OF THE FIRST PROCEDURE WILL NOT BE
DESTROYED BY THE SECOND PROCEDURE.

TWO VARIABLES, "TEMP" AND "PARAM", ARE USED TO SPECIFY THE NUMBER OF
WORDS THAT THE "BASE" WILL BE SHIFTED WHEN A NEW PROCEDURE GAINS
CONTROL. THE SHIFT IS EXECUTED IN PROCEDURE "ANOTHEROPERATION"
(BASE := BASE + TEMP + PARAM). "TEMP" IS USED EXTENSIVELY
THROUGHOUT THE CANDE PROGRAM TO SPECIFY THE NUMBER OF STACK WORDS
WHICH MUST BE ALLOWED TO REMAIN UNCHANGED IN THE STACK WHEN CONTROL
IS TRANSFERRED TO ANOTHER PROCEDURE.

IT IS FREQUENTLY NECESSARY FOR A SECOND PROCEDURE TO BE ABLE TO
"LOOK BACK" INTO THE STACK BELONGING TO A PREVIOUSLY EXECUTED
PROCEDURE. FOR EXAMPLE, IN A REQUEST SUCH AS "PRINT 500-700",
PROCEDURE "INPUT" SCANS THE USERS REQUEST AND PLACES THE RESULTS IN
THE STACK ABOVE THE "BASE" ASSOCIATED WITH THAT PROCEDURE.
PROCEDURE "INPUT" ALSO ASSIGNS TO VARIABLE "PARAM" THE NUMBER OF
WORDS WHICH WERE PLACED IN THE STACK ABOVE THE "BASE", AND ASSIGNS A
VALUE OF 1 TO "TEMP" SO THAT THE NEXT "BASE" WILL BE LOCATED AT BASE
+TEMP+PARAM.

WHEN PROCEDURE "INPUT" ASSIGNS A VALUE OF 56 TO "WANT" AND THEN
GIVES UP CONTROL, PROCEDURE "ANOTHEROPERATION" WILL INCREMENT THE
VALUE OF "BASE" BY "PARAM" + "TEMP" SO THAT THE SCRATCH AREA FOR THE
NEXT PROCEDURE TO BE CALLED WILL NOT INTERFERE WITH THE RESULTS
ALREADY PLACED IN THE STACK BY PROCEDURE "INPUT".

AFTER THE VALUE OF "BASE" IS INCREMENTED, PROCEDURE
"ANOTHEROPERATION" PLACES THE VALUE OF "PARAM" (THE NUMBER OF WORDS
IN THE STACK ABOVE THE PREVIOUS "BASE") IN WORD STACK[BASE]. THEN,
WHEN PROCEDURE "LISTIT" IS GIVEN CONTROL, THIS PROCEDURE MAY
DETERMINE, BY THE VALUE IN STACK[BASE], WHERE TO LOOK FOR THE
PARAMETERS TO THE "PRINT" REQUEST.

THE VALUES OF "TEMP" AND "PARAM" ARE STORED IN THE STACK CONTROL
WORDS CREATED AND/OR RETRIEVED BY PROCEDURE "ANOTHEROPERATION" IN
ORDER TO HAVE THESE VALUES AVAILABLE WHEN A PROCEDURE IS RE-STARTED
AFTER A WAIT.

ANOTHER CANDE VARIABLE, "SREG", IS USED AS A POINTER INTO THE
SCRATCH PAD AREA OF THE STACK. "SREG" IS ASSIGNED THE VALUE OF
"BASE" BY PROCEDURE "ANOTHEROPERATION" WHENEVER A NEW PROCEDURE IS
TO BE GIVEN CONTROL. "SREG" ALWAYS POINTS TO THE STACK WORD THAT IS
CURRENTLY BEING ACCESSED BY A PROCEDURE. THIS VARIABLE IS ALSO
STORED IN THE USERS STACK CONTROL WORD SO THAT IT CAN BE
"REMEMBERED" WHEN CONTROL IS TRANSFERRED TO ANOTHER PROCEDURE AND
THEN BACK AGAIN.

MANY CANDE PROCEDURES ARE BROKEN DOWN INTO "CASES", WITH EACH CASE
BEING A "SUBFUNCTION" OF THE PROCEDURE. THE CANDE VARIABLE "DOING"
IS USED TO SELECT THE CASE TO BE EXECUTED IN A PROCEDURE. SINCE THE
VALUE OF "DOING" IS STORED IN THE STACK CONTROL WORD BY PROCEDURE
"ANOTHEROPERATION", THE CASE NUMBER WHICH WAS IN EFFECT WHEN A
PROCEDURE LOST CONTROL CAN BE "REMEMBERED" WHEN THE PROCEDURE
REGAINS CONTROL AGAIN.

THE PROCEDURE "LISTIT", FOR EXAMPLE, CURRENTLY IS MADE UP OF FIVE
SUBFUNCTIONS. CASE 0, WHICH IS SELECTED WHEN THE VARIABLE "DOING"
IS ASSIGNED A VALUE OF ZERO, IS RESPONSIBLE FOR ANALYSING THE SYNTAX
OF THE USERS REQUEST, AND FOR CALLING OUT THE LIST/CANDE OR QUIKLST/
CANDE PROGRAM.

THE REMAINING CASES (1 THROUGH 4), ARE USED AS RE- START AND
TRANSFER POINTS WITHIN THE PROCEDURE, DEPENDING UPON THE SYNTAX OF
THE USERS REQUEST. SINCE THE VALUE OF "DOING" IS "REMEMBERED" IN
THE STACK CONTROL WORD, A PROCEDURE MAY SPECIFY THE SUBSECTION OF
ITSELF WHICH IS TO BE GIVEN CONTROL WHEN THE ROUTINE IS RE-STARTED.

THE MAJOR CONTROL VARIABLES, "PARAM", "TEMP", "BASE", "DOING",
"IAM", AND "SREG", ARE ALL STORED IN THE CONTROL WORD STACK
[PREVRCW], AND ARE THEREFORE "REMEMBERED" BY CANDE EVEN THOUGH
CONTROL IS TRANSFERRED FROM ONE PROCEDURE TO ANOTHER, OR FROM ONE
USER TO ANOTHER.

FORMAT OF CONTROL WORD "RCW"

...[40:8] = VALUE OF "IAM"

...[33:7] = VALUE OF "DOING"

...[25:8] = VALUE OF "BASE"

...[18:7] = VALUE OF "TEMP"

...[11:7] = VALUE OF "PARAM"

...[3:8] = VALUE OF "SREG"

...[1:2] = 2 (IDENTIFIES THIS AS A "RETURN CONTROL WORD")

AN EXAMPLE OF THE COMPLEXITY INVOLVED IN PROCESSING A RELATIVELY
SIMPLE COMMAND IS THE LIST COMMAND WHEN USED TO LIST THE WORK FILE.
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=56 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=27, FOR DISPATCH, AND WANT=35, 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=35 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 TSSMCP 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 TSSMCP-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.

WORK FILES.

CANDE CURRENTLY MAINTAINS THREE FILES (THE "WORK" FILES) FOR EACH
USER CONNECTED TO THE TIME SHARING SYSTEM. THESE ARE THE "1S", THE
"1P" AND "1T" FILES.

THE "1P" FILE

THE "1P" FILE, WHICH HAS THE CONFIGURATION "1PNNNBB" (THE "N S" ARE
A THREE DIGIT LOGICAL LINE NUMBER AND THE "B S" ARE BLANKS, HENCE,
THE USER CONNECTED TO LINE 12 WILL HAVE THE FILE "1P012 "/"USERCODE"
), IS USED TO STORE "POINTERS" TO THE MOST RECENT ADDITIONS TO THE
"WORK" FILE. THE FILE IS COMPRISED OF 30 WORD SEGMENTS, AND EACH
WORD IS MADE UP AS FOLLOWS:

```
.........................
.  .   .   .    .       .       "1P" POINTER WORD
.  .   .   .    .       .
.........................
   .  .   .    .      .
   .  .   .    .      ...[21:27] = SEQ. NO. OF RECORD
   .  .   .    .
   .  .   .    ...[16:5] = REL. ADRS. IN SEGMENT
   .  .   .
   .  .   ...[8:8] = REL. ADRS. OF SEGMENT IN TANK ROW
   .  .
   .  ...[4:4] = WD. NO. IN "CTRANDBASE" CONTAINING
   .               TANK ROW ADDRESS.
   .
   ...[1:2] = CODE :  0 = SEQ. NO. APPEARS IN RECORD
                      1 = DELETE CODE
                      2 = AUTO. SEQ. MODE OPERATION
                      3 = "FIX" COMMAND
```

PROCEDURE "INPUT" STORES EACH "1P" WORD IN THE LOWER PART OF THE
USERS STACK (IN STACK[TPENTRY]). WHENEVER TEN SUCH POINTER WORDS
HAVE BEEN PLACED IN THE STACK, OR WHEN THE USER HAS REQUESTED AN
"UPDATE", THE "1P" WORDS ARE TRANSFERRED TO THE DISK "1P" FILE. THE
POINTER WORD FOLLOWING THE LAST VALID DISK ENTRY IN THIS FILE HAS
THE VALUE 100000000. THIS IS USED AS A "CHECKPOINT" FOR THE CANDE
PROGRAMS, AND MARKS THE END-OF-FILE LOCATION WITHIN THE "1P" FILE.

THE "CTRANDBASE" ARRAY IS USED TO STORE THE DISK ADDRESSES OF THE
TANK FILE ROWS USED BY THE MCP FOR STORING INCOMING MESSAGES. THE
DISK ADDRESS OF THE SEGMENT CONTAINING THE MESSAGE REFERENCED BY A
"1P" WORD CAN BE CALCULATED BY ADDING THE [8:8] FIELD (THE RELATIVE
ADDRESS OF THE SEGMENT WITHIN THE ROW) TO THE ADDRESS OF THE ZEROTH
RECORD IN THE ROW. THIS ADDRESS IS IN THE CTRANDBASE WORD SPECIFIED
BY THE [4:4] FIELD OF THE "1P" WORD. THE START OF THE ACTUAL
MESSAGE WITHIN THE SEGMENT CAN BE DETERMINED FROM THE [16:5] FIELD

OF THE "1P" WORD (THE OFFSET WITHIN THE SEGMENT).

THE CANDE VARIABLE "ROWINUSE" POINTS TO THE CURRENT WORD IN
"CTRANDBASE" WHICH IS IN USE. "CTRANDBASE" IS A WRAP AROUND ARRAY,
AND THEREFORE THE FIRST WORD IN THIS ARRAY IS NOT NECESSARILY THE
FIRST ROW IN USE IN THE TANK FILE.

THE "1S" FILE

THE "1S" FILE, WHICH HAS THE CONFIGURATION "1SNNNBB" (THE "N S" ARE
A THREE DIGIT LOGICAL LINE NUMBER AND THE "B S" ARE BLANKS) IS USED
TO CONTAIN THE "WORKING" COPY OF THE WORK FILE. WHEN A USER FIRST
LOGS INTO THE TIME SHARING SYSTEM, CANDE CREATES THE FILE "1SNNNBB/
USERCODE" WITH THE END-OF-FILE POINTER (IN THE FILE HEADER) SET AT
-1. EACH TIME THAT THE PROGRAM "LIST/CANDE" IS EXECUTED, RECORDS
ARE TRANSFERRED FROM THE "SOURCEFILE" TO THE "1S" FILE, MERGING IN
NEW ADDITIONS AND EXECUTING "FIXES" AND "DELETIONS" DURING THE
PROCESS.

THE "SOURCEFILE" IS A WORD IN THE USERS STACK THAT IS USED TO
CONTAIN THE NAME OF THE FILE WHICH WAS MOST RECENTLY UPDATED. FOR
EXAMPLE, WHEN A USER "LOADS" A FILE FROM DISK, THE VARIABLE
"SOURCEFILE" WILL BE GIVEN THE NAME OF THE "LOADED" FILE SO THAT
"LIST/CANDE" WILL KNOW WHERE TO LOOK FOR THE USERS RECORDS. SINCE
"LIST/CANDE" ALWAYS WRITES THE MERGED RECORDS ON THE FILE "1SNNNBB/
USERCODE", THE NAME OF THE "SOURCEFILE" IS CHANGED FROM THE DISK
FILE NAME TO THE "1S" FILE NAME THE FIRST TIME THAT "LIST/CANDE" IS
EXECUTED.

THE SECOND TIME THAT "LIST/CANDE" IS RUN, THE PROGRAM WILL LOOK IN
THE FILE "1SNNNBB/USERCODE" FOR THE USERS RECORDS, SINCE THIS FILE
NOW CONTAINS THE MOST RECENTLY UPDATED RECORDS. SINCE THE OUTPUT OF
THE "LIST/ CANDE" PROGRAM IS ALWAYS THE "1S" FILE, THE PROGRAM WILL,
IN THIS INSTANCE, REPLACE THE "OLD 1S" FILE WITH A "NEW 1 S" FILE,
AND THE VALUE OF THE "SOURCEFILE" WILL REMAIN AT "1SNNNBB".

WHEN THE USER REQUESTS A "SAVE", THE MCP "REPLACES" THE "OLD FILE"
WITH THE MOST RECENTLY UPDATED "1S" FILE, TRANSFERRING THE FILE TYPE
AND FILE SECURITY INFORMATION FROM THE OLD HEADER TO THE NEW HEADER,
REMOVING THE OLD FILE FROM DISK, AND ENTERING THE FILE NAME (THIS
IS THE ACTUAL "NAME" OF THE FILE AS SPECIFIED BY THE USER, NOT THE
"1S" NAME) IN THE DIRECTORY. A NEW "1SNNNBB/ USERCODE" FILE IS
THEN CREATED (WITH EOF POINTER SET AT - 1).

WHEN A USER CREATES A NEW FILE (WITH THE CANDE "MAKE" VERB) THE
VALUE ASSIGNED TO VARIABLE "SOURCEFILE" WILL BE THE "1SNNNBB" FILE
NAME, SINCE NO DISK FILE EXISTS ON THE DISK AT THIS TIME. WHEN
"LIST/CANDE" ATTEMPTS TO READ THE "SOURCEFILE", THE EOF POINTER WILL
BE RECOGNIZED AS BEING -1, AND ONLY THE PATCH RECORDS WILL BE PLACED
IN THE "NEW 1S" FILE.

THE CANDE VARIABLE "FILENAME" IS USED TO STORE THE ACTUAL NAME OF

THE FILE AS SPECIFIED BY THE USER. A REQUEST TO "LOAD" OR TO "MAKE" A FILE WILL RESULT IN "FILENAME" BEING ASSIGNED THE FILE NAME SPECIFIED BY THE USER. WHEN NO WORK FILE EXISTS, AS IS THE CASE WHEN A USER FIRST LOGS INTO THE TIME SHARING SYSTEM, OR AFTER A "REMOVE" HAS BEEN EXECUTED, THE VALUE OF "FILENAME" IS ZERO.

TO SUMMARIZE, VARIABLE "FILENAME" ALWAYS CONTAINS THE NAME OF THE WORKFILE WHEN IT EXISTS, AND IS ZERO WHEN NO WORK FILE HAS BEEN CREATED. VARIABLE "SOURCEFILE" CONTAINS THE SAME NAME AS "FILENAME" AFTER A FILE HAS BEEN LOADED (BUT NOT UPDATED BY "LIST/CANDE"), AND AFTER A FILE HAS BEEN SAVED. "SOURCEFILE" CONTAINS THE "1S" FILE NAME WHEN A FILE IS BEING CREATED, AND AFTER AT LEAST ONE "UPDATE" HAS BEEN PERFORMED ON THE WORK FILE. "SOURCEFILE" HAS A VALUE OF ZERO WHEN NO WORK FILE EXISTS.

THE "1T" FILE

THE THIRD "WORK FILE" USED BY CANDE IS THE "1T" FILE, WHICH HAS THE CONFIGURATION "1TNNNBB". THE "1T" FILE IS THE "TAB", AND CONSISTS OF 30 WORD RECORDS, WITH EACH WORD RELATING THE PHYSICAL POSITION OF A FILE RECORD TO ITS SEQUENCE NUMBER.

THE TAB FILES ARE CURRENTLY USED ONLY BY PROGRAM "LIST/CANDE" FOR THE PURPOSE OF DETERMINING WHETHER A RECORD EXISTS IN THE WORK FILE, AND, IF SO, ITS PHYSICAL POSITION WITHIN THE FILE. SINCE ONLY ONE "FILE READ" IS REQUIRED TO OBTAIN 30 SEQUENCE NUMBERS WHEN READING THE TAB FILES, IT IS ADVANTAGEOUS TO USE THESE FILES TO DETERMINE THE CONTENTS OF THE WORK FILE, RATHER THAN READING EACH 10 WORD RECORD FROM THE WORK FILE.

THE TAB FILES ARE CREATED BY THE CANDE PROGRAMS WHICH OPERATE ON THE "WORK FILES", SUCH AS THE "LOAD/ CANDE" AND THE "COPY/CANDE" PROGRAMS. THE EXCEPTIONS ARE THE TYPE DATA FILES WHICH ARE NOT SEQUENCED AND, THEREFORE, DO NOT HAVE ANY TAB FILES ASSOCIATED WITH THEM.

THE TAB FILES, LIKE THE "1P" FILES, ARE TERMINATED BY A WORD CONTAINING THE OCTAL NUMBER 100000000.

USE OF "ESP" SEGMENTS.

CANDE OBTAINS TWO SEGMENTS OF "ESP" DISK FOR EACH USER THAT LOGS INTO THE TIME SHARING SYSTEM. THESE 30 WORD SEGMENTS PROVIDE THE MEANS BY WHICH CANDE COMMUNICATES WITH THE PERIPHERAL PROGRAMS (I.E., THE "LIST/CANDE", "COPY/CANDE", ETC., PROGRAMS).

EACH TIME THAT A USER ENTERS A REQUEST, CANDE CHECKS THE SYNTAX OF THE REQUEST AND PLACES THE APPROPRIATE VALUES IN THE "WORK" ARRAY. IF THE REQUEST REQUIRES THAT A PERIPHERAL PROGRAM BE CALLED OUT, THE CONTENTS OF THE "WORK" ARRAY ARE WRITTEN ON DISK AT A LOCATION CORRESPONDING TO THAT USERS "ESP" ADDRESS. THIS ADDRESS IS ALSO COMMUNICATED TO THE PERIPHERAL PROGRAM THROUGH "COMMON" AT CALL OUT TIME, SO THAT THE SEGMENT IS AVAILABLE TO THE PROGRAM.

FOR MOST CANDE PROGRAMS, A SINGLE 30 WORD "ESP" SEGMENT IS SUFFICIENT FOR COMMUNICATING ALL NECESSARY VALUES. A FEW PROGRAMS, SUCH AS THE "FIND/CANDE" AND "REPLACE/CANDE" ROUTINES, REQUIRE TWO SEGMENTS OF DISK. SINCE ONLY ONE ADDRESS MAY BE COMMUNICATED THROUGH "COMMON", THE CONVENTION ADOPTED IS THAT

> A) THE VALUE OF "ESP1" IS ALWAYS COMMUNICATED THROUGH COMMON
>
> B) THE VALUE OF "ESP2" IS LOCATED IN ESP1[0], WHERE REQUIRED.

ESP[21] THROUGH ESP[29] ARE THE WORD LOCATIONS RESERVED FOR THE SEQUENCE LIST ASSOCIATED WITH A CANDE REQUEST. EACH SEQUENCE NUMBER IS PLACED IN SUCCESSIVE LOCATIONS IN THIS PORTION OF THE ESP RECORD. WHEN A SEQUENCE RANGE IS SPECIFIED, THE UPPER LIMIT IS PLACED IN THE RECORD WITH A NEGATIVE SIGN. LOWER LIMITS AND INDIVIDUAL SEQUENCE NUMBERS ARE PLACED IN THE RECORD UNSIGNED.

ESP[2] THROUGH ESP[12] ARE USED BY THE CANDE PROGRAMS FOR COMMUNICATING FILE NAMES AND RESEQUENCE PARAMETERS. THE CURRENT FORMAT FOR THIS PORTION OF THE ESP RECORD IS AS FOLLOWS:

```
ESP[2] = "USERCODE"
ESP[3] = INPUT FILE FIRST NAME
ESP[4] = INPUT FILE SECOND NAME
ESP[5] = LOWER BOUND FOR RESEQ
ESP[6] = UPPER BOUND FOR RESEQ
ESP[7] = BASE FOR RESEQ
ESP[8] = INCREMENT FOR RESEQ
ESP[9] = OUTPUT FILE FIRST NAME
ESP[10]= NAME OF "WORKFILE" (SOURCEFILE)
ESP[11]= LOWER BOUND FOR DELETION (IN MERGE)
ESP[12]= UPPER BOUND FOR DELETION (IN MERGE)
```

THE "LIST/CANDE" PROGRAM USES ESP RECORD POSITIONS ESP[5] THROUGH ESP[20] FOR TANK FILE ROW ADDRESSES (THESE ARE THE ADDRESSES CONTAINED IN CANDES "CTRANDBASE" ARRAY).

ESP[1] IS USED TO COMMUNICATE THE PARAMETER COUNT, LINE NUMBER, "TAG" VALUE AND CERTAIN OPTIONS WHICH HAVE BEEN SET BY THE USER. THE FORMAT OF THIS WORD IS CURRENTLY AS FOLLOWS:

```
ESP[1].[01:01] = "CHANGES" OPTION SET
ESP[1].[02:01] = "SQUASHED" OPTION SET
ESP[1].[03:01] = TYPE DATA FILE
ESP[1].[04:01] = "NUMBERED" OPTION SET
ESP[1].[05:01] = REVERSE MERGE OPTION SET
ESP[1].[06:01] = OUTPUT FILE IS THE WORKFILE
ESP[1].[07:01] = "DISPLAY" OPTION SET
ESP[1].[08:01] = "CONCISE" OPTION SET
ESP[1].[27:06] = PARAMETER COUNT
ESP[1].[33:07] = "EVENT" NUMBER ASSOCIATED WITH JOB
ESP[1].[40:08] = "LINE" NUMBER ASSOCIATED WITH JOB
```

SINCE ANY PARTICULAR CANDE PERIPHERAL PROGRAM DOES NOT REQUIRE ALL OF THE ABOVE VALUES, CERTAIN OF THESE VALUES MAY BE OMITTED FROM THE ESP RECORD, DEPENDING UPON THE PROGRAM BEING CALLED.

THE NEGATIVE COMMUNICATES

TO FREE CANDE EVEN FURTHER FROM THE MENIAL CHORES OF FILE HANDLING, ALL OF ITS FILE OPERATIONS ARE DONE BY THE TSSMCP. USING COMMUNICATES, CANDE TELLS THE TSSMCP 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 TSSMCP 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 FOLLOWING IS A LIST OF THE NEGATIVE COMMUNICATES, WHICH HAVE BEEN ADDED TO THE TSSMCP FOR TIME SHARING. OF THESE, ONLY -11 AND -13 ARE NOT USED BY CANDE. FOR THOSE COMMUNICATES WHICH SIMPLY PROVIDE ACCESS TO TSSMCP PROCEDURES, THE PROCEDURE IDENTIFIER IS LISTED.

COMMUNICATE	USE
-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, STATUS, AND PAPER TAPE
-16	DISCONNECTS
-17	CARRIAGE SIZE LOGGING
-18	NOT USED
-19	JOB INITIATION FOR JOBS OTHER THAN CANDE
-20	NEW SCHEDULE FILE

TO PERFORM A COMMUNICATE, A PROCEDURE IN CANDE CALLS ANOTHER PROCEDURE CONSISTING ONLY OF THE COMMUNICATE STATEMENT, PASSING 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 LOGS A "CC SHORT" OR "CC LONG" MESSAGE. 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 STATION NUMBER AND A "REASON", I.E., THE NUMBER OF THE CANDE PROCEDURE DOING THE I/O. THIS ALLOWS THE TSSMCP 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 DEVIOUS 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    STATION 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 TSSMCP CONTROL CARDS. COMM9
SIMPLY FORKS CONTROLCARD AND GIVES IT THE INFORMATION.

A -15 COMMUNICATE IS USED BY CANDE FOR 14 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.
9) TO "GO TO RECORD" FOR A SCHEDULE FILE.
10) TO NOTIFY THE MCP THAT CANDE IS READY FOR INPUT FROM
    A SCHEDULE TASK.
11) TO OBTAIN THE STATUS OF A SCHEDULE TASK.
12) TO TERMINATE A SCHEDULE TASK.
13) TO DETERMINE THE STATUS OF A RUNNING JOB.

THE INFORMATION PASSED TO COMM15 INCLUDES A FLAG SPECIFYING WHICH
TYPE OF COMMUNICATE THIS IS, THE USER CODE, FILE-NAMES, THE STATION
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, I.E., 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, INDIANBOY 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 TSSMCP MESSAGE CODES 5, 6 AND 7 (SEE "EVENTS") 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 TSSMCP 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 (TSSMCP MESSAGE #5) OR A CHANGED EVENT (#6) WHEN HE IS FINISHED.

FOR TYPE 8, INDIANBOY INITIALIZES SEQARRAY, GETS 65 WORDS FOR THE CORE TANKS, INITIALIZES THEM AND SENDS THE OK MESSAGE AND AN X-ON.

THE ACTIONS TAKEN FOR TYPES 9 - 12 ARE DESCRIBED UNDER "SCHEDULE LINES".

FOR TYPE 13, INDIANBOY SEARCHES THE JAR FOR THE JOB SPECIFIER AND, IF IT IS FOUND, RETURNS THE PROCESS AND I/O TIME FOR THE JOB TO CANDE. IF THE JOB IS A COMPILATION, THE SEQUENCE NUMBER (IN PRT CELL @27) IS ALSO RETURNED.

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

SCHEDULE LINES AND TASKS

SCHEDULE TASKS CONSIST OF AN INPUT FILE, CONTAINING CANDE COMMANDS
AND JOB INPUT, AND AN OUTPUT FILE, INTO WHICH IS PUT EVERYTHING THAT
NORMALLY WOULD HAVE APPEARED AT THE REMOTE TERMINAL. THE NAME OF
THE OUTPUT FILE IS SPECIFIED BY THE USER. THE INPUT FILE, WHICH IS
REALLY A COPY OF THE FILE SPECIFIED IN THE SCHEDULE COMMAND, IS
NAMED "FILE<TASK NUMBER>/SCHEDUL", WHERE <TASK NUMBER> IS THE THREE
DIGIT NUMBER ASSIGNED TO THE TASK BY THE MCP, I.E., THE NUMBER USED
IN "XS" AND "ES" MESSAGES.

SCHEDULE TASKS ARE RUN ON SCHEDULE LINES, AS DECLARED IN "SYSTEM/
DISK". FOR THESE LINES, THE MCP DATACOM TABLES ARE USED
DIFFERENTLY, AS SHOWN BELOW.

LINETABLE

1:1		NOT USED
2:1	SCHEDBUSY	ON IF LINE IS SV-ED
3:3	LINEDISC	=7 FOR SCHEDULE LINES
6:42		NOT USED

STATABLE

1:1		NOT USED
2:8		SAME AS FOR REGULAR LINES
10:3	STATIONTYPE	=0, I.E., SCHEDULE LINES ARE TREATED AS TELETYPES
13:1		NOT USED
14:1	BREAK	IF ON, ALL OUTPUT IS DISCARDED
15:1	DIALEDUP	ON IF LINE IS IN USE
16:1		ON
17:21		NOT USED

SEQARRAY

1:1		CANDE INPUT READY FLAG
2:1	SCHEND	ON IF THE TASK ON THE LINE IS BEING TERMINATED
3:15		NOT USED
18:15		=1, IF A RESTART AFTER H/L IS BEGIN DONE, OTHERWISE, IT IS 0.
33:15		ADDRESS OF 80 WORD ARRAY FOR I/O BUFFERS

"TANKS", "INPUTANK", AND "TNAOG" ARE NOT USED WITH SCHEDULE LINES.

THE 80 WORD ARRAY POINTED TO BY "SEQARRAY" IS DIVIDED INTO TWO 40-WORD BUFFERS. WORDS 0 THROUGH 38 ARE FOR OUTPUT AND 39 THROUGH 80 ARE FOR INPUT. I/O-S ARE DONE ONE SEGMENT AT A TIME, SO THE FIRST 30 WORDS OF EACH AREA HOLD THE CURRENT SEGMENT AND THE REMAINING WORDS CONTAIN INFORMATION PERTINENT TO IT. THE FORMAT IS:

WORD	CONTENTS
0-29	OUTPUT BUFFER
30	NOT USED
31	I/O MASK, I.E., 19:1 IS TURNED ON WHEN DISK WRITE IS COMPLETE.
32	POINTER TO STARTING PLACE FOR NEXT RECORD IN THE BUFFER (0, 10, OR 20)
33	CURRENT SEGMENT NUMBER
34	ROW SIZE OF OUTPUT FILE
35	THIS PLUS CONTENTS OF WORD 33 GIVES CURRENT DISK ADDRESS
36	DISK ADDRESS OF HEADER (BINARY)
37	CURRENT RECORD NUMBER
38	MAXIMUM NUMBER OF RECORDS IN OUTPUT FILE
39	RESERVED FOR DISK ADDRESS FOR INPUT BUFFER I/O-S.
40-69	INPUT BUFFER
70	TASK NUMBER
71	I/O MASK
72	POINTER TO START OFF NEXT INPUT RECORD
73	CURRENT SEGMENT NUMBER
74	ROW SIZE OF INPUT FILE
75	THIS PLUS CONTENTS OF WORD 73 GIVES CURRENT DISK ADDRESS
76	DISK ADDRESS OF HEADER
77	CURRENT RECORD NUMBER
78	NUMBER OF RECORDS IN INPUT FILE
79	NOT USED

THE "SYSTEM/DISK" RECORD FOR A SCHEDULE LINE IS ALSO DIFFERENT FROM ITS COUNTERPART ON A REGULAR LINE. ITS CONTENTS ARE:

WORD		CONTENTS
0		SAME AS FOR REGULAR LINES
1		USERCODE (BCL)
2		"FILE<TASK NUMBER>" (BCL)
3	18:15	CURRENT INPUT RECORD
	33:15	CURRENT OUTPUT RECORD
4		SAME AS FOR REGULAR LINES

WORD 3 IS UPDATED BY "NSECOND" AND WHEN CANDE DOES A "GO TO" COMMUNICATEWORDS 0 AND 4 ARE USED TO SPECIFY THE LINE AS A SCHEDULE

LINE. THUS, WORD 0 IS ZERO EXCEPT POSSIBLY FOR THE DIALED-UP BIT
AND WORD 4 IS ZERO EXCEPT FOR A 7 IN THE 3:3 FIELD (THE LINE
DISCIPLINE).

THE QUEUE OF TASKS SCHEDULED BUT NOT YET RUN IS KEPT THROUGH THE
HEADERS OF THE INPUT FILES. FOR THIS PURPOSE, THE FOLLOWING
MODIFICATIONS HAVE BEEN MADE TO THE HEADER:

WORD		CONTENTS
5		NAME OF OUTPUT FILE (BCL)
6	1:1	ON IF TASK HAS BEEN XS-ED
	2:1	ON IF TASK IS BEING TERMINATED
	3:45	TIME AFTER WHICH TASK SHOULD BE RUN
27		CHARGE CODE
28		NUMBER OF PREVIOUS SCHEDULED TASKS IN THE QUEUE OR 0 IF THIS IS THE FIRST TASK.
29		NUMBER OF NEXT SCHEDULED TASK IN THE QUEUE OR 0 IF THIS IS THE LAST TASK.

THE QUEUE IS ORDERED ACCORDING TO THE TIME-AFTER. TASKS WITHOUT A
TIME-AFTER PRECEDE THOSE WITH A TIME-AFTER AND ARE QUEUED IN THE
ORDER IN WHICH THEY WERE SCHEDULED. THE ONLY EXCEPTION TO THIS IS
THAT TASKS SCHEDULED BETWEEN MIDNIGHT AND 8AM WITHOUT A TIME-AFTER
ARE PUT AT THE END OF THE QUEUE. THIS IS DONE IN AN ATTEMPT TO
INSURE THAT ALL TASKS SCHEDULED ONE DAY ARE RUN BY 8AM OF THE
FOLLOWING DAY.

THE QUEUE IS MAINTAINED OUT OF A WORD IN THE MCP-S PRT, SCHEDWRD,
WHICH IS ALSO KEPT IN THE 21ST WORD OF "DIRECTORYTOP" AND IS UPDATED
ON DISK WHENEVER IT IS CHANGED IN CORE. THE CONTENTS OF "SCHEDWRD"
ARE:

WORD		CONTENTS
1:1	SCHEDTOG	INTERLOCK. QUEUE IS LOCKED WHEN "SCHEDTOG" IS OFF
2:1		NOT USED
3:15	LSTSCHED	NUMBER OF THE LAST TASK IN THE QUEUE
18:15	FRSTSCHED	NUMBER OF THE FIRST TASK IN THE QUEUE
33:15	SCHEDNUM	NEXT AVAILABLE TASK NUMBER

IF THE QUEUE IS EMPTY, "FRSTSCHED" EQUALS "SCHEDNUM".

THE HEADER OF THE OUTPUT FILE HAS ALSO BEEN SLIGHTLY MODIFIED, AS
SHOWN BELOW.

WORD		CONTENTS
4	36:6	0 (FILE TYPE UNKNOWN) UNTIL TASK IS RUN

```
                              10 (FILE TYPE "INFO") WHEN TASK IS RUNNING
                              OR COMPLETED.
6         1:32                NON-0 IF TASK IS SCHEDULED OR RUNNING, 0
                              AFTER IT IS DONE
          33:15               NUMBER OF TASK.
```

OPERATION OF SCHEDULE TASKS AND LINES

SCHEDULE TASKS ARE RUN IN BACKGROUND MODE. THAT IS, JOBS INITIATED FROM SCHEDULE LINES ARE TREATED AS BACKGROUND JOBS AND CANDE IS GIVEN SCHEDULE INPUT ONLY WHEN THERE IS NO OTHER INPUT AVAILABLE. HOWEVER, IF THE "NOBATCH" OPTION IS SET, SCHEDULE LINES WILL RUN ALMOST IDENTICALLY TO REGULAR LINES, SINCE, IN PRACTICE, THE INPUT RESTRICTION HAS LITTLE EFFECT.

THE MCP PROCEDURES INVOLVED WITH THE OPERATION OF SCHEDULE LINES ARE:

INDIANBOY	HANDLES COMMUNICATES FROM CANDE
SCHEDLOOK	HANDLES SPO INPUTS AND ALSO IS USED TO REBUILD THE LINKS IN THE QUEUE OF SCHEDULED TASKS IF SOMETHING HAPPENS TO THEM.
SCHEDIO	HANDLES I/O FOR BOTH INPUT AND OUTPUT.
SCHEDIDLE	LINKS, DELINKS, INITIATES, AND TERMINATES SCHEDULE TASKS.

WHEN A USER SCHEDULES A TASK, CANDE SIMPLY EXECUTES THE "SCHEDUL/ CANDE" PROGRAM. IT COPIES THE SPECIFIED INPUT FILE INTO A NEW FILE CALLED 9S<LOGICAL LINE NUMBER>/<USER CODE>. IT THEN CREATES THE DESIRED OUTPUT FILE, PUTS "YOUR TASK HAS NOT YET BEEN RUN" INTO THE FIRST RECORD, AND DOES A -20 COMMUNICATE. THE -20 COMMUNICATE DELETES THE HEADER OF THE 9S<LOGICAL LINE NUMBER>/<USER CODE> FILE FROM THE DIRECTORY, PUTS THE CHARGE CODE, TIME AFTER AND OUTPUT FILE NAME INTO ITS CORE COPY OF THE HEADER AND FORKS "SCHEDIDLE", PASSING THE HEADER TO IT. "SCHEDIDLE" ASSIGNS THE TASKS A NUMBER (SCHEDNUM) AND LINKS IT INTO THE QUEUE, SEARCHING FROM "LSTSCHED" TOWARD THE FRONT OF THE QUEUE FOR THE CORRECT SPOT. IT THEN ENTERS THE INPUT FILE INTO THE DIRECTORY AS "FILE<TASK NUMBER>/SCHEDUL", AND INCREASES "SCHEDNUM". NEXT IT ATTEMPTS TO START THE TASK.

TO INITIATE A TASK, "SCHEDIDLE" FIRST LOOKS FOR AN AVAILABLE SCHEDULE LINE, I.E., ONE THAT IS NOT IN USE AND NOT "SV"-ED. IF THERE ARE NONE, "SCHEDIDLE" EXITS. IF IT FINDS ONE, "SCHEDIDLE" LOCKS THE TASKS INPUT FILE, BUILDS THE 80 WORD ARRAY, AND OPENS THE OUTPUT FILE AFTER CHANGING IT TO TYPE INFO. IT PUTS "YOUR TASK IS RUNNING" INTO THE FIRST RECORD OF THE OUTPUT BUFFER, SPOUTS THE LOG- ON MESSAGE, UPDATES SYSTEM DISK, AND INFORMS CANDE OF THE NEW TASK VIA A "RESTART" EVENT. "SCHEDIDLE" THEN GOES BACK TO SEE IF THERE ARE ANY OTHER TASKS IT CAN INITIATE. THIS IS DONE BECAUSE THE INITIATE CODE IS ALSO USED AFTER THE OPERATOR READIES A LINE OR LINES AND AFTER A TASK TERMINATION.

ONCE THE TASK IS RUNNING, "SCHEDIO" HANDLES THE READS AND WRITES. WHEN A JOB REQUESTS INPUT, COMM13 CALLS "SCHEDIO", PASSING IT THE LINE NUMBER, BUFFER ADDRESS AND NUMBER OF WORDS REQUIRED. "SCHEDIO" CALLS ITSELF TO WRITE THE NEXT INPUT RECORD INTO THE OUTPUT FILE,

THEN MOVE THE RECORD INTO THE BUFFER AND, IF NECESSARY, INITIATES A READ ON THE NEXT SEGMENT. IF THE INPUT FILE IS EXHAUSTED, I.E., THE CURRENT RECORD NUMBER EQUALS THE MAXIMUM NUMBER OF RECORDS, "SCHEDIO" DS-ES THE JOB. THIS CAUSES CANDE TO TERMINATE THE LINE WITH AN ERROR.

INPUT TO CANDE IS MORE COMPLICATED. WHEN CANDE HAS COMPLETELY FINISHED ONE INPUT FROM THE LINE, SHE DOES A TYPE 10 -15 COMMUNICATE. THIS CAUSES THE 1:1 BIT OF SEQARRAY TO BE SET. IF CANDE IS WAITING FOR INPUT (I.E., SHE DID A -5 COMMUNICATE WHEN THERE WAS NO INPUT AVAILABLE), "SCHEDIO" IS FORKED. AS BEFORE, "SCHEDIO" CALLS ITSELF TO PUT THE INPUT INTO THE OUTPUT FILE. THEN IT TAKES THE RECORD AND QUEUES IT TO "OLDWIERDHAROLD", WHO PROCESSES IT NORMALLY AND PASSES IT TO CANDE. IF CANDE IS NOT WAITING FOR INPUT, THE MCP WAITS UNTIL CANDE DOES A -5 COMMUNICATE WHEN THERE IS NO INPUT TO GIVE HER AND THEN FORKS "SCHEDIO" IF THERE IS A LINE WHICH HAS THE 1:1 BIT OF "SEQARRAY" SET. THUS, INPUT IS TAKEN FROM SCHEDULE LINES ONLY IF THERE IS NO INPUT AVAILABLE FROM REGULAR LINES.

IF THE INPUT FILE IS EMPTY WHEN CANDE REQUESTS INPUT, "SCHEDIO" PASSES HER A LEFT ARROW TO TAKE HER OUT OF SEQUENCE MODE IN CASE SHE WAS IN IT. CANDE THEN ASKS FOR MORE INPUT AND "SCHEDIO" GIVES HER A "BYE", THUS INITIATING NORMAL TERMINATION PROCEDURES.

FOR OUTPUT, "SCHEDIO" IS PASSED THE ADDRESS OF THE MESSAGE, THE NUMBER OF CHARACTERS AND THE LINE NUMBER. IT MOVES THE MESSAGE INTO THE BUFFER, WRITES IT TO DISK IF THE BUFFER IS FULL, AND GETS A NEW ROW FOR THE FILE IF NECESSARY. WHEN THE OUTPUT FILE IS WITHIN 10 RECORDS OF BEING FULL, "SCHEDIO" WRITES "**END OF SCHEDULE TANK" INTO IT. IF A JOB IS RUNNING IT IS DS-ED UNLESS IT IS MARKED "NDSABLE" IN WHICH CASE THE "BREAK" BIT IS SET IN "STATABLE". THEN THE INPUT FILE IS MARKED AS BEING AT END OF FILE. THIS CAUSES TERMINATION WHEN CANDE NEXT ATTEMPTS TO READ INPUT.

WHEN CANDE RECEIVES A "BYE" COMMAND FROM A SCHEDULE LINE, SHE GOES THROUGH THE NORMAL "BYE" PROCEDURE TO GENERATE THE LOG-OFF OUTPUT. THEN SHE CALLS "DISCONNECT", WHICH CLEANS UP HER TABLES AND DOES A -15 COMMUNICATE TO LOG THE USER OFF. FOR SCHEDULE LINES, "INDIANBOY" JUST SPOUTS THE LOG-OFF MESSAGE AND THEN FORKS "SCHEDIDLE". "SCHEDIDLE" TURNS ON "SCHEND", REMOVES THE INPUT FILE, UPDATES "SYSTEM/DISK", AND CLOSES THE OUTPUT FILE, FIXING THE HEADER, BLANKING THE FIRST RECORD, AND CRUNCHING THE FILE. IF OTHER TASKS ARE WAITING, "SCHEDIDLE" ATTEMPTS TO INITIATE THEM, OTHERWISE IT EXITS.

IF AN ERROR OCCURS DURING THE RUNNING OF A TASK, CANDE USES THE "GO TO" COMMUNICATE (-15, TYPE 9) TO SET THE INPUT RECORD COUNTER TO THE END OF THE FILE, THUS CAUSING A TERMINATE. THE "GO TO" CAN BE USED TO SET THE CURRENT RECORD TO ANY SPECIFIED RECORD IN THE INPUT FILE, BUT CANDE ONLY USES IT FOR END-OF-FILE, WHICH IS SPECIFIED BY PASSING -1 INSTEAD OF A RECORD NUMBER. "INDIANBOY" JUST UPDATES THE RECORD POINTERS IN WORD 3 OF "SYSTEM/DISK" AND THEN FORKS "SCHEDIDLE" TO DO THE DIRTY WORK OF ADJUSTING THE POINTERS SO THEY

ARE CONSISTENT WITH THE VALUES IN "SYSTEM/DISK". IF NECESSARY, A
NEW INPUT SEGMENT IS READ AND "SCHEDIO" IS FORKED IF CANDE IS
WAITING FOR INPUT FROM THE LINE.

IF A "STATUS" OR "STOP" IS DONE, CANDE USES THE TYPE 11 OR TYPE 12
-15 COMMUNICATE. THE RETURN EVENT HAS THE FOLLOWING CONTENTS.

WORD   CONTENTS
----   --------
0   STANDARD
1   =0 NO SUCH OUTPUT FILE
                            =1 FILE IS NOT A SCHEDULE FILE
                            =2 TASK IS SCHEDULED
                            =3 TASK IS RUNNING
                            =4 TASK IS COMPLETE
2   CURRENT INPUT RECORD IF FILE IS RUNNING

FOR A "STOP", IF THE TASK IS SCHEDULED, BIT 2:1 OF THE SIXTH WORD OF
THE INPUT FILE HEADER IS TURNED ON AND "SCHEDIDLE" FORKED AS IF AN
"ES" HAD BEEN DONE. IF THE TASK IS RUNNING, "INDIANBOY" FINDS OUT
WHICH LINE THE TASK IS ON, UPDATES THE NUMBER OF RECORDS IN THE
OUTPUT FILE HEADER (THUS ALLOWING IT TO BE LISTED AFTER A "STATUS"
IS DONE) AND, FOR A "STOP", IF A JOB IS RUNNING, TERMINATES IT,
FORCES AN END-OF-FILE ON THE INPUT FILE, AND CALLS "SCHEDIO" TO
WRITE "**TASK TERMINATED BY USER".

THE OPERATOR CONTROLS SCHEDULE LINES VIA THE "XS", "ES", "RY", "SV",
AND "CL" MESSAGES, THE FIRST FOUR OF WHICH ARE HANDLED BY
"SCHEDLOCK". "SV" AND "RY" SIMPLY INVOLVE SETTING AND RESETTING
"SCHEDBUSY" FOR THE LINE. FOR "XS" AND "ES", "SCHEDIDLE" IS FORKED.
IF IT IS AN "XS" THE TASK IS DELINKED FROM THE QUEUE AND THEN LINKED
BACK IN AT THE HEAD OF THE QUEUE. BIT 1:1 OF THE SIXTH WORD OF THE
HEADER IS TURNED ON, MAKING THE TIME-AFTER NEGATIVE AND MARKING THE
TASK AS AN "XS"-ED TASK. FOR AN "ES", AFTER DELINKING THE TASK,
"SCHEDIDLE" REMOVES THE INPUT FILE, WRITES A ONE RECORD EXPLANATION
INTO THE OUTPUT FILE AND FIXES UP ITS HEADER. "CL"-S ARE HANDLED BY
"LINECLEAR", WHICH DS-ES THE JOB IF THERE IS ONE, SETS THE INPUT
POINTER TO END-OF-FILE, AND CALLS "SCHEDIO" TO WRITE "**TASK
TERMINATED BY OPERATOR". FOR A "CL$", THE LINE IS ALSO SV-ED.

"SCHEDLOOK" IS ALSO CALLED FOR A "TS" AND DURING A HALT/LOAD. IN
THIS CASE, IT LINKS FORWARD THROUGH THE TASK QUEUE STARTING FROM
"FRSTSCHED" AND BUILDING BACK LINKS AS IT GOES. IT STOPS WHEN IT
REACHES THE END OF THE QUEUE OR IF AN INPUT FILE IS NOT THERE. IT
THEN UPDATES "LSTSCHED". FOR A "TS", A MESSAGE IS SPOUTED FOR EACH
TASK FOUND. FOR A HALT/LOAD, IT SPOUTS THE TOTAL NUMBER OF TASKS
FOUND, IF DIFFERENT FROM ZERO. THIS CODE IS ALSO CALLED FROM
"SCHEDIDLE" IF IT FINDS SOMETHING WRONG WITH THE QUEUE. FOR
INSTANCE, A MISSING INPUT FILE. OF COURSE, THIS SHOULD NEVER
HAPPEN, BUT IF IT DOES, THIS CODE ALLOWS THE QUEUE TO BE AT LEAST
PARTIALLY RECOVERED.

WHEN CANDE IS STARTED, "SCHEDIDLE" IS FORKED TO RESTART LINES ACTIVE BEFORE THE HALT/LOAD. "SCHEDIDLE" GETS THE TASK NUMBER FROM "SYSTEM/DISK", LOCKS THE INPUT FILE, SETS UP THE BUFFERS AND OPENS THE OUTPUT FILE. IF EITHER FILE IS MISSING, THE TASK IS DELETED FROM "SYSTEM/DISK". IT THEN SETS UP THE POINTERS, USING WORD 3 OF "SYSTEM/DISK", TELLS CANDE, AND SPOUTS AN "ON" MESSAGE. THE 1:2 FIELD OF THE SECOND WORD OF THE RESTART EVENT (THE USERCODE) IS USED AS FOLLOWS:

    0   REGULAR RESTART
    1   SCHEDULE LOG-ON
    2   SCHEDULE RESTART

CANDE PUTS THE TASK INTO HER TABLES, BUT THEN, SINCE SHE HAS NO WAY OF KNOWING IF THE LINE CAN BE RESTARTED (SINCE SOME JOBS CANNOT BE RERUN, FOR INSTANCE, THOSE UPDATING DISK FILES), SHE TERMINATES THE TASK WITH AN ERROR.

APPENDIX A

HOW TO ANALYZE A TIME SHARING SYSTEM MEMORY DUMP

IN ORDER TO BE ABLE TO UTILIZE THE FOLLOWING INFORMATION, IT IS NECESSARY TO HAVE THE SETTINGS OF THE C,S, AND F REGISTERS, A TSSMCP PRT LISTING, AND SEVERAL MANUALS FOR REFERENCE:

    1) THE B 5500 HANDBOOK.
    2) THE B 5500 SYSTEMS REFERENCE MANUAL.
    3) THE TIME SHARING SYSTEM REFERENCE MANUAL.

ADDITIONAL MATERIALS THAT WOULD BE HELPFUL ARE LISTINGS OF THE PRT ENTRIES AND A CROSS-REFERENCE AND PROSORT OF THE TSSMCP PRT LISTING.

THE SEQUENCE OF PROCEDURES INCLUDED WITHIN THIS DOCUMENT ARE SUGGESTED UNTIL THE USER DEVELOPS THE ABILITY TO SEARCH THE DUMP SELECTIVELY BASED UPON CUMULATIVE EXPERIENCE IN ANALYZING MEMORY DUMPS.

IF CORE WAS NOT TOO BADLY CLOBBERED BY THE HANG, MOST OF THE THINGS MENTIONED BELOW WILL BE AUTOMATICALLY DONE BY THE TSDUMP/ANALYZE PROGRAM. HOWEVER, ANYONE WHO EXPECTS TO BE REGULARLY CONFRONTED WITH TSSMCP DUMPS SHOULD BE THOROUGHLY FAMILIAR WITH THESE PROCEDURES IF HE HOPES TO HAVE ANY SUCCESS IN TRACKING DOWN THE CAUSE OF THE DUMP.

FIRST DIAGNOSIS

A FIRST DIAGNOSIS INVOLVES THE INSPECTION OF THE "PRT" AND "JAR" ARRAY DESCRIPTORS.

    PRT ARRAY DESCRIPTOR = 5000003600AAAAA
    JAR ARRAY DESCRIPTOR = 5000003600AAAAA

THE C-FIELD ([33:15]) OF EACH OF THE ABOVE POINTS TO THE ROW VECTOR THAT CONTAINS THE DESCRIPTOR POINTING TO THE ARRAY FOR EACH MIX NUMBER.

    PRT[MIX,*]   PRTROW[MIX]
    JAR[MIX,*]   JARROW[MIX]

BY CONVERTING THE FIRST TWO WORDS OF THE "JAR" ENTRY FOR EACH JOB, THE NAME OF THE JOB THAT IT REFERS TO WILL BE THE RESULT OF THIS OCTAL CONVERSION. THESE ARRAYS CONTAIN INFORMATION THAT WILL BE USED IN FURTHER ANALYSIS.

THERE SHOULD BE A ONE-TO-ONE CORRESPONDENCE BY MIX INDEX BETWEEN THE ENTRIES IN "PRTROW" AND "JARROW".

DUE TO SWAPPING, THE ENTRIES IN "PRTROW" AND "JARROW" DO NOT NECESSARILY POINT TO THE "PRT" AND "JAR" ENTRIES FOR A GIVEN JOB SINCE THAT JOB MAY BE SWAPPED OUT.

DETERMINING THE JOB RUNNING

NEXT, ONE CAN GO TO "PRT" CELL "P1MIX" AND FIND OUT WHAT JOB WAS RUNNING. YET, JUST BECAUSE A CERTAIN NUMBER IS IN "P1MIX" DOES NOT NECESSARILY IMPLY THAT THIS JOB WAS THE CAUSE OF THE HANG, OR THAT THE CODE PRESENTLY BEING EXECUTED WAS THE CODE OF THAT JOB. NEVERTHELESS, THE POSSIBILITY DOES EXIST AND MUST BE CONSIDERED. ALSO, IF THERE IS NO SECOND PROCESSOR, P2MIX=-1 (OCTAL 2000000000000001). IF P1MIX=0, THERE IS A VERY HIGH PROBABILITY THAT TSSMCP CODE WAS BEING EXECUTED AT THE TIME OF THE HANG.

PROCEEDING ON, ONE CAN TAKE THE C-REGISTER SETTING AND INQUIRE AS TO WHAT WAS EXECUTING BEFORE THE SYSTEM HANG. THIS C-REGISTER SETTING IS THE ADDRESS OF THE PROGRAM WORD (INSTRUCTION) THAT WAS EXECUTING. GIVEN THE ABILITY TO DIFFERENTIATE BETWEEN CODE AND DATA, IT SHOULD BE EVIDENT WHETHER THE C-REGISTER IS POINTING TO EXECUTABLE CODE OR NOT. IF SO, THE NEXT STEP WOULD BE TO FIND OUT TO WHAT ROUTINE THE CODE BELONGS.

IF THE CONTENT OF THE C-REGISTER IS LESS THAN THE C-FIELD OF PRT CELL ESPBIT, THEN THE CODE IS MAINLINE, OUTER-BLOCK TSSMCP CODE. ONE MERELY NEEDS TO CONVERT THIS OCTAL NUMBER IN THE C-REGISTER TO DECIMAL AND LOOK UP THIS ADDRESS AT THE END OF THE TSSMCP LISTING USING THE FAR-RIGHT COLUMN OF NUMBERS, IN THE LAST FIVE OR SIX PAGES OF THE LISTING.

IF THE C-REGISTER IS NOT POINTING TO OUTER-BLOCK TSSMCP CODE AND P1MIX=0 (TSSMCP EXECUTING), THERE ARE SEVERAL WAYS OF IDENTIFYING THE CODE BEING EXECUTED.

ONE METHOD INVOLVES THE C-REGISTER AND A SERIES OF COMPARISONS TO DETERMINE THE EXACT ROUTINE. FIND THE LARGEST C-FIELD IN ANY DESCRIPTOR OF THE TSSMCP PRT LESS THAN OR EQUAL TO THE C-REGISTER SETTING. THIS C-FIELD REFERENCES ANY TSSMCP PROCEDURE IN CORE. THE ADDRESS OF THIS DESCRIPTOR IS THE PRT SLOT NUMBER OF A ROUTINE. LOOKING THIS UP IN THE LISTING OF THE PRT CELLS WILL GIVE YOU THE NAME. THEN, TAKING THE C-FIELD OF THIS DESCRIPTOR AND SUBTRACTING IT FROM THE C-REGISTER SETTING AND CONVERTING IT TO A DECIMAL, WILL POINT TO THE EXACT WORD IN THE ABOVE DETERMINED ROUTINE THAT WAS EXECUTING.

A SECOND WAY TO FIND WHERE ONE WAS HUNG IS TO GO TO THE STACK AND GO TO THE RCW POINTED TO BY THE F-REGISTER. THE C-FIELD OF THE RCW GIVES THE ADDRESS OF THE SYLLABLE TO WHICH THE TSSMCP WILL RETURN CONTROL ONCE THE SUBROUTINE HAS EXITED. THEREFORE, BACKTRACKING IN THE CODE SPECIFIED BY THIS ADDRESS WILL EVENTUALLY RESULT IN AN OPERAND CALL NEAR A "0441", A MARK STACK COMMAND. DECIPHERING OBJECT CODE THUS:

```
  1     4     4     2
 001   100   100   010
```

DELETING THE TWO RIGHTMOST BITS WHICH SIGNIFY AN OPERAND CALL, WE

READJUST INTO OCTADES GETTING:

    OPDC   011 001 000   OR   OPDC 310

THIS NUMBER(OCTAL 310) POINTS TO A PRT ENTRY WHICH CONTAINS A
CONSTANT, A VARIABLE, OR DESCRIPTOR SIGNIFYING A ROUTINE. WHAT IS
DESIRED IS THE OPDC WHICH IS IN A CELL PAST BUT CLOSEST TO THE MARK
STACK COMMAND. HAVING THAT, USE ITS STARTING ADDRESS TO COMPUTE THE
LOCATION OF EXECUTION AS IN THE ABOVE CASES.

A WAY TO CHECK THE ABOVE CALCULATIONS IS TO CONVERT THE C-REGISTER
ADDRESS TO DECIMAL AND SUBTRACT FROM IT THE BASE ADDRESS AT THE END
OF THE LISTING. THEN SUBTRACT FROM THAT RESULT, THE BASE ADDRESS OF
THE DIAGNOSED PROCEDURE. THE ANSWER SHOULD POINT TO APPROXIMATELY
THE SAME LOCATION IN THE PROCEDURE AS THE ABOVE METHODS.

NON-MCP CODE.

GO TO THE APPROPRIATE "PRTROW" ENTRY FOR THE APPROPRIATE MIX. THIS
DESCRIPTOR POINTS TO THE PRT FOR THAT JOB. THEN GO TO THE C-FIELD
ADDRESS+4. THIS IS THE SEGMENT DICTIONARY ENTRY. THIS SEGMENT
DICTIONARY CONTAINS THE DESCRIPTORS THAT POINT TO THE AREAS OCCUPIED
BY THE SEGMENTS. THEN YOU COMPARE THE ADDRESS IN THE C-REGISTER
AGAINST THE C-FIELDS OF THE DESCRIPTORS UNTIL YOU HAVE THE LARGEST
C-FIELD LESS THAN THE C-REGISTER SETTING. THE SEQUENTIAL NUMBER OF
THIS ENTRY FROM THE STARTING ADDRESS TELLS WHICH SEGMENT THE C-
REGISTER WAS EXECUTING FROM. THEN SUBTRACTING THE ADDRESS IN THE C-
FIELD FROM THE C-REGISTER SETTING AND CONVERTING TO DECIMAL WILL
GIVE YOU THE POINT OF EXECUTION.

ANOTHER METHOD UTILIZING MEMORY LINKS IS DESCRIBED IN THE FOLLOWING
SECTION.

MEMORY LINK WORDS

MEMORY LINK WORDS ARE USED TO KEEP TRACK OF DATA AND PROGRAMS THAT
ARE ASSIGNED TO CORE MEMORY. THEY ARE VERY USEFUL IN DETERMINING
WHAT AN AREA IS WITH REFERENCE TO THE ADDRESSES GIVEN IN THE
ANALYSIS OF DUMPS. THERE IS AN ADEQUATE ILLUSTRATION OF THEM ON
PAGES 5-7 OF THE BURROUGHS B 5500 HANDBOOK. TO USE THEM, ONE MUST
FIND THE MEMORY LINK EITHER BY USING THE "MEMROW" CONSTRUCT OR BY
SCRUTINIZING THE AREA AROUND THE WORD FOR A MEMORY LINK. THE MEMORY
LINK THAT IS RELEVANT IS THE ONE THAT APPEARS IN A LOCATION CLOSEST
TO AND LESS THAN THAT OF THE ADDRESS BEING QUESTIONED. THIS ADDRESS
BEING QUESTIONED COULD COME FROM THE C-FIELD OF A RETURN CONTROL
WORD (RCW) OR THE C-REGISTER. THE MEMORY LINK PROVIDES INFORMATION
THAT TELLS YOU THE STATUS OF THE AREA, THE TYPE OF THE AREA, AND TO
WHOM THE AREA IS ASSIGNED, WHO IS USING THE AREA BY MIX INDEX. THE
WORD FOLLOWING THE MEMORY LINK, AS ILLUSTRATED IN THE HANDBOOK,
TELLS THE SIZE OF THE SEGMENT AND MORE INFORMATION AS TO WHO IS THE
OWNER OF THE SEGMENT, DEPENDING UPON ITS TYPE.

FOR EXAMPLE, AN ADDRESS IS OBTAINED FROM AN RCW. AFTER HAVING FOUND

THE FIRST LINK THAT PRECEDES THE ADDRESS, THE MIX INDEX IS INSPECTED. IF IT IS ZERO, THEN THE C-FIELD IN THE SECOND WORD IS THE PRT ENTRY NUMBER FOR THIS AREA. IF IT IS NON-ZERO, THEN THE HANDBOOK DEFINES THE REMAINING OPTIONS. IF [0:3] IS GREATER THAN ONE FOR THE FIRST WORD, THEN THIS AREA IS AVAILABLE. THIS MEANS THAT THIS AREA HAS BEEN DISCARDED AND IS NOT CONNECTED TO ANY JOB.

PRT CELL "MEMROW" IS A DESCRIPTOR THAT POINTS TO AN ARRAY THAT CONTAINS THE MEMORY LINKS FOR EACH JOB THAT HAS AN ENTRY IN THE "PRT" ARRAY. MEMROW[0]=0 SINCE CELL ZERO IS THE FIRST MEMORY LINK FOR MIX=0.

MOST OF THE PREVIOUS INFORMATION ALLOWS ONE TO PINPOINT THE LOCATION OF AN INSTRUCTION AND IN WHAT PROCEDURE IT EXISTS (SOURCE AND/OR OBJECT). ANOTHER IMPORTANT POINT TO BE CONSIDERED IS THE SEQUENCE OF CALLS THAT PRECEDED THE HANG AND THEIR PARAMETERS. THE MEANS FOR TRACING THIS SEQUENCE OF CALLS IS THE STACK. THE S- AND F-REGISTERS ARE THE POINTERS TO THE STACK OUT OF WHICH THE TSSMCP WAS EXECUTING AT THE TIME OF THE HANG. THE S-REGISTER POINTS TO THE TOP OF THE STACK AND THE F-REGISTER IS THE MOST RECENT RETURN CONTROL WORD (RCW) OR MARK STACK CONTROL WORD (MSCW). A GOOD DESCRIPTION OF THEIR USE IN RELATION TO ENTERING AND EXITING SUBROUTINES IS GIVEN ON PAGE 5-10 OF THE B 5500 SYSTEMS REFERENCE MANUAL. A COUPLE OF HELPFUL HINTS ARE:

IF THE USER PROGRAM IS EXECUTING AND THE TSSMCP COMES IN, THE MSCW LOOKS LIKE:

    60000010000000000

AN RCW BY ITS F-FIELD SETTING MAY BACK-LINK OUT OF THE STACK POINTED TO BY THE S-REGISTER.

A STACK OF A JOB IS BOUND BY WORDS OF:

    22222222 22222222 AT THE TOP AND
    33333333 33333333 AT THE BOTTOM.

BED

THE BED ARRAY CONTAINS INFORMATION ABOUT A JOB THAT HAS BEEN
SUSPENDED UNTIL A CERTAIN CONDITION EXISTS. THIS ARRAY HAS ENTRIES
FOR EVERY JOB THAT HAS BEEN PUT TO SLEEP. BY LINKING FORWARD
THROUGH THE PRT CELL "BED" OR LINKING BACKWARD THROUGH PRT CELL
"BED1", THE WHOLE BED CAN BE ACCESSED. IF NOTHING IS ASLEEP OR,
EQUIVALENTLY, NOTHING IS IN THE BED, THE THE MCP BED ENTRY POINTS TO
ITSELF AND THE MCP BED1 ENTRY POINTS TO BED. FURTHER INFORMATION
THAT ACCOMPANIES BED ARRAY ENTRIES IS PROVIDED IN "THE BED: SLEEPING
AND WAKING".

FORK QUEUE

THE PRT CELL "FORKQUE" IS AN ARRAY DESCRIPTOR THAT CAN BE ADDRESSED
AS THE ARRAY "FORKQUE". THIS ARRAY IS A QUEUE OF REQUESTS TO RUN
INDEPENDENT PROCESSES. THIS QUEUE IS FILLED BY CALLING THE "FORK"
ROUTINE. IT CONTAINS THE PRT ADDRESS OF THE PROCEDURE FORK-ED.
FURTHER INFORMATION AND BREAKDOWN IS PROVIDED IN "THE FORK QUEUE:
ITS STRUCTURE AND OPERATION".

READY

THE PRT CELL "READY" GIVES THE STATUS OF THE HARDWARE UNITS WHICH WERE INDICATED TO THE TSSMCP TO BE IN THE READY STATE. A BREAKDOWN OF THIS WORD IS LOCATED IN THE B 5500 HANDBOOK.

SAVERESULTS

IF THE "SAVERESULTS" OPTION IS SET WHEN THE TSSMCP IS COMPILED, THE FOLLOWING INFORMATION WILL BE STORED IN THE PRT ARRAY "RESULTHOLDER":

 1) UNIT[30] WHENEVER AN INTERROGATE REQUEST INTERRUPT IS RECEIVED BY THE TSSMCP.
 2) THE I/O DESCRIPTOR FOR ALL DATACOM I/O-S.
 3) THE RESULT DESCRIPTOR, STATABLE[ST], AND LINETABLE[ST] WHENEVER A DATACOMM I/O COMPLETE OCCURS.

THIS INFORMATION IS STORED CYCLICALLY WITH PRT CELL "LASTRESULT" POINTING TO THE MOST RECENT ENTRY.

FOR EXAMPLE, CONSIDER THE ITEMS STORED WHEN A USER TYPES A LEFT-ARROW (GROUP MARK) FROM A TELETYPE WITH NOTHING ELSE GOING ON:

 1) UNIT[30] FROM THE INTERROGATE REQUEST.
 2) I/O DESCRIPTOR FOR THE RESULTING INTERROGATE.
 3) RESULT DESCRIPTOR FROM THE INTERROGATE.
 4) STATABLE[ST].
 5) LINETABLE[ST].
 6) I/O DESCRIPTOR FOR THE READ.
 7) RESULT DESCRIPTOR FOR THE READ.
 8) STATABLE[ST].
 9) LINETABLE[ST].
 10) I/O DESCRIPTOR FOR WRITING CARRIAGE-RETURN AND LINE-FEED.
 11) RESULT DESCRIPTOR FOR THE WRITE.
 12) STATABLE[ST].
 13) LINETABLE[ST].
 14) UNIT[30] DUE TO THE INTERROGATE REQUEST CAUSED BY THE LINE GOING IDLE (I.E., THE DATA HAS BEEN SENT TO THE USER).
 15) I/O DESCRIPTOR FOR THE RESULTING INTERROGATE.
 16) RESULT DESCRIPTOR FROM THE INTERROGATE.
 17) STATABLE[ST].
 18) LINETABLE[ST].

THUS, THIS ARRAY CAN BE USED TO PINPOINT EXACTLY WHAT WAS BEING DONE ON THE DATACOM SUB-SYSTEM.

FINAL NOTE: THE LOCATION OF ANY PRT CELL WITHIN A PRT LISTING IS DEPENDENT UPON THE OPTIONS USED IN THE COMPILATION OF THE TSSMCP.