

COMPUTATION CENTER  
Massachusetts Institute of Technology  
Cambridge 39, Massachusetts

August 7, 1962

TO: Computation Center Staff

FROM: F. J. Corbató, M.M. Daggett, and R. C. Daley

TITLE: Abbreviated Instructions for the use of the Mod. 9 Time-Sharing System

### Introduction

This description of a time-sharing system is an abridgement of the paper, "An Experimental Time-Sharing System" by F. J. Corbató, M.M. Daggett and R. C. Daley, delivered at the Spring Joint Computer Conference in San Francisco, May 1962. In both descriptions, tapes are referred to since, at the present time, the system is dependent on tapes for file storage, system record storage, and temporary storage, system record storage, and temporary storage. This system will be revised for the replacement of tapes by a disk storage unit when the latter is installed.

### Description

In this Time-Sharing System, there are two types of users, those on-line at the typewriters in a foreground system and those of the passive user, the background system, running in the Restricted Fap-Mad-Madtran System.

The time-sharing foreground system of user-consoles is organized around commands that the user can give on his typewriter and the private user files which (for want of a disc) are kept on a magnetic tape. For convenience the format of the private tape files is such that they can be written or punched using the off-line equipment. (This also offers a crude form of large scale input-output.) Tape files are composed of binary mode card images, initiated by a Hollerith title card (cols. 1-6=title, cols. 7-12=class) and terminated by an end-of-file mark. (The first file should be of class "first", the last file of class "last".)

The commands are typed by the user to TSS (not to his own program) and thus can be initiated at any time regardless of the particular user program in memory. For similar coordination reasons, TSS handles all I/O of the foreground system typewriters.

Commands are composed of segments, containing a maximum of 6 characters, separated by vertical strokes: the first segment is the command name and the remaining segments are parameters pertinent to the command. Each segment consists of the last 6 characters typed (starting with an implicit 6 blanks) so that spacing is an easy way to correct a typing mistake. A carriage return is the signal which initiates action on the command. Whenever a command is received by the supervisor, "WAIT", is typed back followed by "READY" when the command is completed.

While typing, an incomplete command line may be ignored by the "quit" sequence of a code delete signal followed by a carriage return. Similarly after a command is initiated, it may be abandoned if a "quit" sequence is given. In addition, during unwanted command typeouts, the command and output may be terminated by pushing the special stop output button.

Commands

1. | $\alpha$

$\alpha$  = arbitrary text treated as a comment.

2. login | $\alpha$ | $\beta$

$\alpha$  = user problem number

$\beta$  = user programmer number

Should be given at beginning of each user's session. Rewinds user's private file tape; clears time accounting records.

3. logout

Should be given at end of each user's session. Rewinds user's private file tape; punches on-line time accounting cards.

4. marktp

The user's private file tape is rewound and initialized, i.e. the files called "first" and "last" are written. This command may be used at the first session or thereafter whenever there is no pertinent information on the tape.

marktp | $\alpha$ | $\beta$

Causes the file called "last" to be written after the file  $\alpha, \beta$ . This command effectively truncates a file tape after given file. As a precaution, the command marktp | $\alpha$ | $\beta$  should always be used if the following commands are terminated before completion by either of the quit sequences: input, file, mad, madtrn, fap, split, and combin.

5. input

Sets user in input mode and initiates automatic generation of line numbers. The user types a card image per line according to a format appropriate for the programming language. When in the automatic input mode, the manual mode may be entered by giving an initial carriage return. The appropriate line number is then typed followed by | and the line. This may be repeated for as many lines as desired. To reenter the automatic mode, an initial carriage return is given.

When in the automatic mode, the user should type only after a line number has been typed by TSS. The manual allows the user to overwrite previous lines and to insert lines. cf. File Command.

The manual mode must be entered before a new command can be given.

6. edit | $\alpha$ | $\beta$

$\alpha$  = title of file,  $\beta$  = class of file

The user is set in the automatic input mode with the designated file as initial input lines. The same conventions apply as to the input command.

7. file| $\alpha$ | $\beta$

$\alpha$  = title to be given to file.

$\beta$  = class of language used during input.

The created file will consist of the numbered input lines in sequence; in the case of duplicate line numbers, the last version will be used. The line numbers will be written as sequence numbers in the corresponding card images of the file.

For convenience the following editing conventions apply to input lines:

- a. an underline signifies the deletion of all previous characters of the line.
- b. a backspace signifies the deletion of the previous character in the field.

The following formats apply:

- a. FAP: symbol, tab, operation, tab, variable field and comment.
- b. MAD, MADTRN,: statement label, tab, statement. To place a character in the continuation column: statement label, tab, backspace, character, statement,
- c. DATA: cols. 1-72.

8. fap| $\alpha$

Causes file designated as  $\alpha$ ,fap to be translated by the FAP translator. Files  $\alpha$ ,syntb and  $\alpha$ ,bss are added to the user's private file tape giving the symbol table and the relocatable binary BSS form of the file.

9. mad| $\alpha$

Causes file  $\alpha$ ,mad to be translated by the MAD translator. File  $\alpha$ ,bss is created.

10. madtrn| $\alpha$

Causes file  $\alpha$ ,madtrn to be translated into an equivalent file  $\alpha$ ,mad(added to user's file) and then translation occurs as if the command mad| $\alpha$  had been given.

11. load| $\alpha_1$ | $\alpha_2$ |. . .| $\alpha_n$

Causes the consecutive loading of files  $\alpha_i$ ,bss ( $i=1,2,\dots,n$ ). An exception occurs if  $\alpha_1 = (\text{libe})$ , in which case file  $\alpha_{i+1}$ ,bss is searched as a library file for all subprograms still missing. (There can be further  $\alpha_i$  as well as further library files.)

12. use| $\alpha_1$ | $\alpha_2$ |. . .| $\alpha_n$

This command is used whenever a load or previous use command notifies the user of an incomplete set of subprograms. Same  $\alpha_i$  conventions as for load.

13. start| $\alpha$ | $\beta$

Starts the program setup by the load and use commands after first positioning the user private file tape in front of the title card for file  $\alpha, \beta$ . (If  $\beta$  is not given, a class of data is assumed; if both  $\alpha$  and  $\beta$  are not given, no tape movement occurs and the program is started.)

14. pm| $\alpha$

$\alpha$  = "lights", "stomap", or usual format of F2PM request: "subprogram name |loc<sub>1</sub> |loc<sub>2</sub>|mode|direction" where mode and direction are optional.

Produces post-mortem of user's dormant programs according to request specified by  $\alpha$ .

The post-mortem requests may be:

pm|lights

The machine conditions and the location of the stop are given.

pm|stomap

The BSS loading table is printed, giving the origin and the entry point of all subprograms loaded.

pm|stop

The location of the stop is given, and the contents of the two locations on either side of the stop are printed.

pm|auto

An automatic post-mortem is printed that corresponds to a "lights" plus "stop" pm.

pm|name

The four initial locations of subprogram name are printed.

pm|name|loc<sub>1</sub> |loc<sub>2</sub>|mode|direction

All locations from loc<sub>1</sub> through loc<sub>2</sub> of subprogram 'name' are printed in the given mode according to the direction specified.

"Name" is the name of the subprogram, (Main) for the main program.

"Loc<sub>1</sub>" is a number assumed to be decimal unless preceded by a slash, meaning octal.

"Mode" specifies the form of the printed output and may be: all, fix, flo, dec, oct, bcd.

"Direction" specifies whether the locations of the request are to be printed in ascending or descending order over the range of loc<sub>1</sub> through loc<sub>2</sub>. Direction may be: fwd or rev. If fwd is requested, the smaller of the pair (loc<sub>1</sub>, loc<sub>2</sub>) is printed first with subsequent locations in ascending order.

Mode and direction are optional; when they are not specified, the mode "all" and the direction "fwd" are assumed.

pm|loc<sub>1</sub>|loc<sub>2</sub>|mode|direction

An absolute post-mortem of the locations from loc<sub>1</sub> through loc<sub>2</sub> is given in the specified direction according to the requested mode. Mode and direction, if omitted, are assumed to be: all | fwd.

15. skippm

Used if a pm command is abandoned on output and the previous program stop is to be restarted by the start command.

16. listf

Types out a list of the names of all files on user's private file tape.

17. printf|α|β|γ

If γ is omitted, types out the whole file α,β. Whenever output buffer fills, the command program goes into I/O wait status allowing other users to time-share until buffer needs refilling. A third parameter may be given to indicate that the file is to be typed starting at line number γ. Even though the identification field in the card contains alphabetic characters, γ represents only numeric portion.

18. xdmp|α| β

Creates file α,β (if β omitted, xdmp assumed) on user's private file tape consisting of complete state of user's last dormant program.

19. xundmp|α|β

Inverse of XDMP in that it resets file α,β as user's program, starting it where it last left off.

20. split|a|c|b<sub>1</sub>|s<sub>1</sub>! . . .|s<sub>n-1</sub>! b<sub>n</sub>

This command splits the file with name a and type c into n new files with names b<sub>1</sub>, . . . , b<sub>n</sub> all of type c. The splitting of the original file is done after the record sequence numbers s<sub>1</sub>, . . . , s<sub>n-1</sub>. The n new files created are appended to the user's private file tape with no resequencing performed on them. If any b<sub>i</sub> is blank, the corresponding file is not appended to the user's file tape. This provides a convenient way to extract subfiles from long master files. If any s<sub>i</sub> is blank or cannot be matched with a sequence number in the remaining portion of the original file, this is an error. In this case the remainder of file a is included with the last b<sub>i</sub> processed, and an error comment is typed out. Matches in sequence numbers must be exact. If file a of type c cannot be found, the "need-use" convention as used with the LOAD command is followed.

21. combin|r|w| c|a<sub>1</sub> . . . |a<sub>n</sub>

This command combines the files with names a<sub>1</sub>, . . . , a<sub>n</sub>, all of types c, into a single file with name w and type c. The new file is resequenced starting at sequence number r and then is appended to the user's private file tape. If r (up to a five-digit decimal number) is omitted, there is no resequencing performed. If any of the a<sub>i</sub> cannot be found a complete list of those not found is typed for the user, and the "need-use" convention is followed. No new file is added to the user's tape unless the command is completed.

(Note: if a USE command specifies a blank file name during a COMBIN command, it is as if the original file name did not appear.)

22. octlk|α|β

Gives octal post mortem of β words starting at location α where α and β are given in octal.

23. octpat|α|β<sub>1</sub>|β<sub>2</sub>| . . . |β<sub>2n-1</sub>|β<sub>2n</sub>

Stores starting at octal location α the consecutive words formed by the β segment pairs where β<sub>1</sub> is a 6-digit right-justified octal left-half word, β<sub>2</sub> a right-half word, etc.

24. octtra|α

Starts immediately user's program at octal location α.

Preparing a Private File tape.

Make up a deck of cards as follows:

- File 1: Title card containing "FIRST" in cols. 7-12.  
Standard End-of-file card: 7+8 punches in column 1 and "End of File" in cols. 16-26.
- File 1: Title card containing file name in columns 1-6 and class name in 7-12,  
card deck  
Standard End-of-file card.

This deck should be left in the dispatching area when computer time is requested. The deck should be clearly identified with problem and programmer number.

Inspecting a Private file tape.

If it is desired to inspect a private file tape or reorganize it, leave a request with the dispatcher to punch the correct number of files on a given tape.

Example

Example of a user's first session where several files are input and translated onto his private file tape. (The symbol ↵ indicates carriage return and → indicates a tab.)

```
login|MI01|1900 ↵
WAIT,
MI01 1900 LOGGED IN 3/5 1542.6
READY.
marktp ↵
WAIT,
READY.
input ↵
10|→|ENTRY →|SVFL ↵
```

20 |OVFL→|SKD→|OVFL-2,4 ↓

.  
.

100 |→|END ↓

110 | ↓  
file|subri|fap ↓

WAIT,  
INPUT PROCESSING DONE.  
READY.

fap|subri ↓  
WAIT,

00150 IS THE FIRST LOCATION NOT USED BY THIS PROGRAM  
FAP DONE. ASSEMBLY OK.

READY.

input ↓

10 |→| ZETA = ALPHA + . ABS. (ZB-Q) + SQRT. (X) ↓

20 |→| INTEGER MAX, MIN, TIME ↓

.  
.

200 |→|END OF PROGRAM ↓

210 | ↓  
file|mainp|mad ↓

WAIT,  
INPUT PROCESSING DONE.  
READY.

mad|mainp ↓

At this point Mad prints out a list of the names and locations for the constants, variables, and statement labels used in the program.

READY.

load|mainp|subri ↓

WAIT,  
NEED FLEXPB .SETUP  
use|(libe)|libr ↓

WAIT,  
READY.

start ↓

WAIT,  
PROGRAM STARTED  
EXIT CALLED. RUN TERMINATED.  
READY.

logout ↓

WAIT,

MI01 1900 LOGGED OUT 3/5 1555.9

COMPUTER TIME USED 1.0 MIN.

READY.