MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

ARTIFICIAL INTELLIGENCE PROJECT — MEMORANDUM MAC-M-250

MEMO 81 — JULY 23, 1965

To: Project MAC Participants

From: Peter Samson

Subject: PDP-6 TECO (July 1965)

Table of Contents

General Description	Page	2
Loading TECO		3
Typing in Commands		3
Quantities		4
Use of Q-Registers		5
DECtape Commands		6
Input Commands		8
Basic Editing Commands		11
Search Commands		12
Search Summary		13
Macros, Iterations and Conditionals		14
Returning to MACDMP		15
Some Simple Examples of Editing		16
Index of Commands		17

I. General Description

TECO is a scope-keyboard text-editor. It uses an on-line command language (which permits macro-definitions, corditionals, etc.) to control input-output (paper and magnetic tapes, scope, etc.) as well as text operations. The macro language permits the most sophisticated search, match, and substitution operations as fwell as simple typographical corrections to text.

The original TECO (Tape Editor and Corrector) was a PDP-1 program written by Daniel L. Murphy. With the arrival of a PDP-6 at Project MAC, the need was seen for a similar program on the new machine, and PDP-6 TECO wasf written by Steward Nelson, Jack Holloway, and Richard Greenblatt. Although similar to the PDP-1 program in many respects, there are countless differences in detail; furthermore PDP-6 TECO commands have undergone several changes in their history. Only the 6 July, 1965 version of PDP-6 TECO is described herein. Differences between this description and current versions on the System Tape will be listed in the Console Book.

There are three kinds of storage areas within TECO: (1) the buffer; (2) the 36 Q-registers; designated 1,...9, A,B,...Z; (3) the command string area. The buffer at all times contains a (possibly null) character string, generally constituting a page of the text being edited. At certain times the command-string area contains a string of characters which TECO interprets as a series of commands. Each Q-register may: (a) be undefined; (b) contain a positive or negative integer; (c) contain a character string. The buffer is that area where text to be edited is examined and modified; the Q-registers are locations for remembering quantities, or strings of text, for later use; the command string is the area from which TECO takes its instructions.

Associated with the buffer is the "pointer": this is a movable flag which always sits between two adjacent characters in the buffer (or at either end of the buffer). The pointer has a numeric value, equal to the number of characters to its left in the buffer. If at the moment there are n characters in the buffer, the pointer's value may range from Ø to n.

A general description of the usage of TECO is as follows: the user gets TECO into the computer; types in commands to describe where the material to be edited (the "input") is (either on paper tape or on DECtape as named file.) By further commands, he reads the first page into the buffer; inspects it on the display and makes changes by typing in commands; outputs

the page onto paper tape of DECtape, and reads in the next page; etc.; and after out-putting the last page, if to DECtape he files it, giving the new file its name. This is by no means the only way one may proceed, but it is the most common.

II. Loading TECO

The most recent version exists on the DECtape labelled MACDMP SYSTEM as the file TECO.

To load it, put the MACDMP SYSTEM tape on a drive, set that drive unit to 1, press INSTRUCTION STOP, IO RESET; set TA to Ø; press READ IN. (This loads MACDMP.) When the on-line Teletype carriage-returns, type TECO followed by a carriage return. When the Teletype then line-feeds, TECO is in and running, waiting for commands.

III. Typing in Commands

TECO is always listening for the user to type in commands (except during the performance of the VW command; see below). Nearly any string of characters may be typed in. If carriage return is typed, TECO imagines that the user typed a line feed following it. All characters typed in are echoed back out by TECO (including the imagined line feed) unchanged except as follows; (1) ALT MODE is typed out as $; (2) characters which have no visible effect and are typed in via the CTRL shift key come out as ↑ followed by the appropriate un-CTRL character; (3) RUBOUT does not go in, but instead erases the last character that did go in. This echoes out the character that has just been erased. RUBOUT may be used any number of times in a row, the user does not pause an instant between RUBOUTs, TECO will erase all characters typed, and when the RUBOUTs cease will type carriage return--line feed. This feature may be used dependably by pressing RUBOUT and REPT. (4) Two successive ALT MODEs indicate the end of the string being typed in. When these are encountered, TECO echoes them as $s. type carriage return - -line feed, copies the entire string of characters typed (except those erased) into the command string area, and starts to interpret and perform these new commands. The format in which TECO expects these commands, and the actions they direct it to take, comprise the bulk of knowledge the user should have about TECO.

IV. Quantities

Many TECO commands take arguments with numeric values, and some commands take different action depending on whether arguments are given or not. The following characters may appear in command strings to develop numeric values. Furthermore, some commands, beside directing action on TECO's part, have a numeric value themselves and so may act as an argument to a subseqjent command. If a command may have a value, this will be stated in its description. In all command description, Ⓢ ;means ALT MODE and /α menas α with the CTRL key held.

Ø-9 digits form decimal integers.

B = Ø (i.e. returns the numeric value Ø).

Z = number of characters in the buffer.

. = number of characters to left of the pointer.

Qi (where i is a digit or letter) = numeric value most recently put into Q-register named i.

nA (where n is a numeric argument) = value in 7-bit ASCII of the character to the right of the pointer.

@ has the value of elapsed time, in 6Øths of a second, that TECO has been running.

m+n add } take 1 or 2 arguments
m-n subtract }

m*n multiply } take 2 arguments
m/n divide (truncates) }

m&n logical and: bitwise and of binary representations of m and n. Operators +, -, *, /, and & are performed from left to right. This may be overruled by use of

() numeric parenthesis.

n= causes the value of n to be typed out.

nUi puts quantity n in Q-register named i.

m,n if a command takes two numeric arguments, comma is used to separate them.

H an abbreviation for B,Z which is a frequently used pair of arguments for commands.

V. Use of Q-Registers

nUi (mentioned in IV.) Puts the numeric value n in Q-register named i (where i is a character Ø,1,...,9,A, B, ...,Z.)

Qi (mentioned in IV.) Has the value of the latest quantity put into Q-register i.

i Adds 1 to the quantity in Q-register i and stands for the new value.

m,mXi Copies a portion of the buffer into Q-register i. It sets Q-register i to duplicate of the (m+1)th through nth characters in the buffer. The buffer is unchanged. (for example: HXA puts a replica of the entire buffer into Q-register A.)

nXi Into Q-register i is copied the string of characters starting immediately to the right of the pointer and proceeding through the nth line feed.

Gi The text in Q-register i is inserted into the buffer at the current location of the pointer. The pointer is then put just to the right of the insertion. The Q-register is not changed.

It does not make sense to TECO to "G" a Q-register given a number by "U" nor to "Q: or "%" one given text by "X".

VI. DECtape Commands

TECO uses the MAC file structure on DECtape to put on one tape a large number of independent ASCII files. For details of this scheme the reader is referred to MAC-M-249, "Mac PDP-6 DECtape File Structure." A "file" is in many ways the equivalent of a phusical paper tape. Different files are distinguished on one DECtape by their names: a file name consists of one or two subnames. Each subname is a string of printing characters (ASCII 41_8 through 137_8) from one to six characters long; in the case of two subnames, the subnames are separated by a space. The first subname may be preceded by a space, Each file is in one of four modes. The mode of a file is set ;by DECDMP when it dumps, and by TECO and MIDAS when output is filed. Associated with each mode is an identifying character which appears with each mode file name in file-directory listings. This character is never input by by the user, and is only to provide him with information about the file.

PROGRAM	MODE OF OUTPUT	MODE NUMBER	IDENTIFYING CHARACTER
TECO	ASCII	∅	SPACE
MACDMP	DUMP	1	!
MIDAS (absolute format)	SBLK	2	"
MIDAS (relocatable format)	RELOC	3	

In the descriptions below, nm or qp signifies a file name of one or two subnames; u and v signify a DECtape unit number which must be given in the command; and o signifies an optional DECtape unit number---if not given, the most recent unit mentioned by explicit number or filed upon is assumed.

oERnm (S) — Prepare to read file nm from the tape on unit o.

oEI — Prepare to put output onto tape o. DECtape output since last EF is forgotten.

oEFnm (S) — Consider all DECtape output since last EF or EI on file, with the name nm. The argument o is ignored, and the file is written out on the tape having received the most recent EI. The EF command also copies onto the tape the results of previous EN and ED commands regarding that tape.

oEDnm (S) — Delete file nm from the file directory in memory for tape o. This must later be followed by EC or EF on the same unit to mark the deletion on the tape.

uEC — Copy out file directory from core onto tape. This is used to finalize previous ED and EN operations on that tape.

u,vEC — Write file directory for tape v onto tape u.

u, ∅EC — Write empty file directory onto tape u.

oENnm (S) qp (S) — Rename file nm on o with new name qp. An EF or EC must follow on same tape.

i, uENnm (S) qp (S) — Rename file nm to qp and set it to mode number i.

uET — Load file directory for tape u into core. This command is not essential, since all DECtape commands load in the file directory if it is not already in core. ET has no effect at all if the directory is in core.

oEL — List the file directory for unit o on the on-line Teletype. A number after the file name is its length in blocks.

uEK — Delete file directory for unit u from core. This has no effect on the tape.

o/U — Display file directory on scope. Upper half of display has space for each block on the DECtape: Block 1 in the upper left, block 2 to its right, etc. A blank space indicates a free (currently unused) block: otherwise the space will contain a decimal number, related to the file name by the chart in the lower part of the display. That chart gives for each file: file number, mode character, 1st subname (@ if only one subname), 2nd subname, length in blocks. The /U command should be the last in a command string; the file-directory display goes away when any new command string is put in.

VII. Input Commands

Upon loading TECO, paper tape input is initially selected.

oERnm ESC — (explained in VI.) Prepare to read file nm from DECtape unit o. ER turns on DECtape input and turns off paper tape input.

/Q — Prepare to read from DECtape.

/S — Prepare to read from paper tape. This command is needed only to counteract ER or /Q.

/C — Clear paper tape reader buffer. The i-o routines read more paper tape than the user asks for, sequestering the yet uncalled-for text in a "buffer" area which can be voided with /C.

Y — Render the buffer (not the "reader buffer") empty; then read into the buffer from selected input file (paper tape or file on DECtape) through the next FORM character. That FORM does not enter the buffer. The pointer is left at Ø (the beginning of the buffer). If there is no FORM character in the input, it is read until an end of file is encountered: this is the end of the physical paper tape, or the end of a DECtape file.

A	Append to the end of the buffer from the selected input, through the next FORM or end-of-file. The pointer is not moved by A.
Itext $	Insert, at the current pointer location, the text following the I up to but not including the first ALT MODE. The pointer is put to the right of the inserted material.
↑ Ijtextj	(this is ↑I, not /I.) Insert, at the current pointer position, the text surrounded by instances of the terminator j, which may be at the user's choice any character not appearing in the text. The pointer is put to the right of the insertion.
nI	Insert at the pointer a character whose 7-bit ASCII code is n (base 10). The pointer is moved to the right of the new character.

VIII. Output Commands

Upon loading TECO, output is selected for paper tape alone.

oEI	(Explained in VI.) Prepare tape o for output. This command must be given before any output is done to that tape. EI turns on DECtape output and turns off paper tape output.
/W	Turns on DECtape output. Obsolescent command.
/V	Turns off DECtape output.
/R	Turns on paper tape output. Notice that both paper and DECtape may simultaneously receive the same output. To select this, give the /R command after the EI.
/T	Turns off punched tape output.
/E	Turns on line printer output (that is, if you have a line printer). This is independent of paper tape and DECtape output.
/B	Turns off line printer output.
nV	Displays on the scope the n lines to the right of the pointer, i.e. from the pointer through the nth following line feed. If n is negative, n lines to the left of the pointer are displayed.
V	Same as IV.
i,jV	Display the (i+1)th through jth characters in the buffer.

Notes on the V command: the display ceases if the V is not the last command of the command string, and when a new command string is entered. Usually V is not needed, since whatever not processing commands TECO automatically displays 2Ø lines on each side of the pointer (except of the last command it saw was V or /U). In all displays of the buffer, the pointer is displayed as a flashing vertical bar, unless it is not in the range being displayed or is at the end of the buffer. Non-printing characters are displayed as follows: space, tab, carriage return, and line feed have their usual effect; BELL appears as a bell symbol; others appear as ↓ followed by a subscript which is their 7-bit code in octal.

n/N	Set number of lines automatically displayed on each side of pointer to n.
n/D	Set display size to n. Legal sizes are Ø, 1, 2, and 3: 1 is the usual size, 2 is appropriate to demonstrations to groups, Ø is too small and 3 is too big. /D affects V and VW commands, but not /U.
nVW VW i,jVW	Same as similar form of V except: when this command is encountered, the display is maintained and further commands in the string are not interpreted until the user types in a character. Then the display ceases, the VW is treated as a numeric quantity equal to the 7-bit ASCII code of the character just typed in, and command interpretation resumes. (See Example 6.)
nT	Type out the string of characters starting at the right of the pointer and continuing through the nth line feed encountered. If n is negative, n lines to the left of the pointer are typed.
T	Same as IT.
i,jT	Type out the (i+1)th through the jth character of the buffer.

Notice that T is identical to V, except that it uses the Teletype instead of the scope. Observe also why HT types out the entire buffer.

PW	Output the entire buffer, followed by a FORM character, to the selected output devices (paper tape punch, DECtape, line printer). Buffer is unchanged and pointer is unmoved.

P	Is identical to PWY.
nP	Is identical to PP...P (P performed n times).
i,jP	Outputs (i+1)th through jth characters of buffer. No FORM is put at the end. Buffer unchanged; pointer unmoved.
nF	The paper tape punch feeds n characters of blank tape.
oEFnm (§)	(Explained in VI.) File DECtape output with name nm. This command should be given after the data is comprise a DECtape file has been output.

IX. Basic Editing Commands

nJ	Move the pointer to the right of the jth character in the buffer. (I.e. give "." the value n.)
J	Same as ØJ.
nC	Same as .+nJ. Note that n may be negative
nR	Same as .-nJ .
nL	If n Ø: move pointer to the righ, stopping when it has passed over n line feeds. If n Ø: move pointer to the left; stop when it has passed over n+1 line feeds and then move it to the right of the last line feed passed over.
L	Same as 1L .
nD	Delete /n/ characters from the buffer: if n is positive, delete then just to the right of the pointer; if n is negative, delete then just to its left.
nK	Perform nL, but delete everything the pointer moves over.
m,nK	Delete the (m+1)th through nth character from the buffer. The pointer is then put where the deletion took place.
m,nXi	(Explained in V.) The string from character m+1 through character n of the buffer is copied into Q-register i.
Itext (§) ↑Ijtextj	(Explained in VII.) Inserts text in buffer at pointer; puts pointer to right of insertion.
nI	(Explained in VII.) Inserts character whose 7-bit ASCII code is n at pointer; puts pointer at right of new character.

Gi (Explained in V.) Inserts at the pointer a copy of the text in Q-register i. The pointer is set at the right of the insertion.

X. Search Commands

There are three search commands: S, N, and ←. Each of these may be preceded by either or both of the modifier characters; and ↑. Preceding the command letter, but following any modifiers, may be a numeric argument; if this is not supplied, 1 is assumed. Following the command letter is a text argument in the same format as for the I or ↑I command, depending on the absence or presence of the ↑ modifier. If the : modifier is not used, a search command has no numeric value. If : is used, the value is -1 if the sought material is found and Ø otherwise.

The search commands attempt to find in the buffer a match to their text argument. The search is begun at the location of the pointer just prior to the search, and proceeds to the right until either a match is found or the end of the buffer is reached. If the match is found, the pointer is moved to the right of the string found in the buffer and the command is finished. If the match has not been found, the action depends on whether the command was S, N, or ←. The S command in such a circumstance is said to fail; the N command performs a P, and then resumes searching; the ← command performs a Y and resumes the search. If N or ← reaches the end of the file without finding the match, they fail. If a search fails (S reaches the end of the buffer; or N or ← reaches the end of the file), the pointer is left at the end of the buffer. Then if the : modifier was not specified, TECO types out a question mark and ignores the remainder of the command string.

Should a numeric argument precede the command letter, this is equivalent to writing out the entire search command a number of times equal to the value of the argument. In other words, if the; number is n it searches for the nth following instance of the text.

SEARCH SUMMARY

All searches begin at the current location of the pointer. They may include a numeric argument n saying to search for the nth occurrence of the text. The pointer is left to the right of the string found if the search is successful; otherwise at the end of the buffer. Use of the modifier determines the form of the text argument which follows the command:

if ↑	jtextj	where j is any character not in text
if no ↑	text (§)	where text contains no ALT MODE

COMMAND	Action at end of buffer	Action at end of file	Value when succeeds	Value when fails	Type ? when fails
S	fails	can't reach it	none	none	yes
:S	fails	can't reach it	-1	∅	no
N	P, resume search	fail	none	none	yes
:N	P, resume search	fail	-1	∅	no
←	Y, resume search	fail	none	none	yes
:←	Y, resume search	fail	-1	∅	no

XI. Macros, Iterations and Conditionals

Mi	Perform now the text in Q-register i as a series of commands.
< >	Iteration brackets. When the > is encountered, command interpretation is sent back to the < .
n;	Iteration test. If the value of n is negative, this command has no effect. If n is Ø or greater, command interpretation is sent just past the matching >to the right.
:	If the most recent search failed, send command to first unmatched >to the right. Otherwise, no effect.
!tag!	Tag definition. The tag is a name for the location it appears in a macro, iteration, or command string.
Otag$	Go to the tag named tag. The tag must appear in the current macro or command string.
n"G	Has no effect if n>Ø. Otherwise send command interpretation to next matching'. The " and ' match similar to (and).
n"L	Send command to matching ' unless n<Ø.
n"N	Send command to matching ' unless n≠Ø.
n"E	Send command to matching ' unless n=Ø.
n"C	Send command to matching ' unless the value of n as a u-bit ASCII character is a letter, number, . $ or %.

Examples:

1) To read and perform a command string from paper tape (assuming paper tape input is selected):

```
YHXZMZ
```

2) To count line feeds in the buffer:

```
JØUN< :S
      (§) ;%N >QN=
```

3) To replace all instances of JUMPA with JRST

```
J<:SJUMPA (§); - 4DIRST (§) >
```

4) To change, throughout the buffer, A to B wherever the A is followed by a digit (ASCII code 48_{10} through 57_{10}):

```
J<:SA (§); 1A-47"GIA-58"L-1DIB (§)' ' >
```

5) To move the pointer to the left of the first character that is not a space (i.e. to space it over a row of spaces):

!BK!1A-32"E1COBK (§)'

With complicated loops and iterations it is generally wise to perform them as macros, so that their text is in Q-registers for debugging and editing.

6) To move on to a page which one would recognize but can not describe in sufficient detail to use N :

<HVW-127;P>

As each page appears on the scope, type in any character except RUBOUT ($=127_{10}$) to proceed to the next page, or if this is a page to be edited type RUBOUT to leave the iteration.

XII. ERRORS

It is conceded that TECO's error messages are not overly informative. Such typed-out comments as TAPE FULL and FILE NOT FOUND are clear enough, and are part of the 1-o routines rather than TECO proper. For all illegal or meaningless commands TECO types out ? and ignores the remainder of the command string, returning to the idel state. At this point the user may type ? back in, and TECO would then respond by typing out the command string, ending with the bad command, Search commands are "bab" if they fail and the : modifier was not used.

There are two occurrences that cause TECO to panic: a) PDL OV (pushdown list overflow), TECO types out one bell, gives up the command string and returns to idel state; b) NON EX MEM (non-existant memory addressed), TECO types out many bells, goes back to idle.

XIII. Returning to MACDMP

The command /G (BELL) will, when performed, return to MACDMP, but it will have waited for all of TECO's output to be done. MACDMP will very possibly still have the file directory of the tape it last referred to---presumably the one off which was loaded TECO.

XIV. Some Simple Examples of Editing

1. To duplicate an ASCII paper tape, making no changes in it:

   Y1ØØØ P (§)(§)

   The P command is terminated by reading an end-of-file; 1ØØØ was chosen as being larger than any reasonable number of pages.

2. To copy the file PGM ENG1 on DECtape 3, putting the new version on tape 2 with the file name PGM ENG2:

   3ER PGM ENG1 (§) ZEIY1ØØØPEF PGM ENG2 (§)(§)

3. Same as (2), but in the new copy change the title line from TITLE PGM1 J. SMITH to TITLE PGM2 W. JONES.

   2EI3ER PGM ENG1 (§) YNTITLE PGM (§) K12 W. JONES
   (§)(§) (Here check the correction by looking at the scope.)
   1ØØØ PEF PGM ENG2 (§) /U (§)(§)
   (Check the file directory on the scope to see that PGM ENG2 has been added.)

XV. Index of Commands

A	4,8	PW	10	\B	9
B	4	Q	4	\C	8
C	11	R	11	\D	10
D	11	S	12	\E	9
EC	7	T	10	\G	15
ED	7	U	5	\N	10
EF	7	V	9	\Q	8
EI	7	VW	10	\R	9
EL	7	X	5	\S	8
EK	7	Y	8	\T	9
EN	7	Z	4	\U	7
ER	7	ø-9	4	\V	9
ET	7	+	5	\W	9
F	11	-	5	"C	14
G	6	*	5	"E	14
H	5	/	5	"G	14
I	8,9	( )	5	"L	14
J	11	.	4	"N	14
K	11	←	12	'	14
L	11	↑	9, 12	\@	5
M	14	%	5	<>	14
N	12	&	5	:	12
O	14	'	5	;	14
P	11	=	5	!	14
				?	15