

WASHINGTON UNIVERSITY



ST. LOUIS, MISSOURI 63110

COMPUTER SYSTEMS LABORATORY
724 SOUTH EUCLID AVENUE

TELEPHONE: AREA CODE 314
FO 1 - 7356

June 9, 1967

Memo to: Holders of LINC Manufacturing Description, Volume 6

From: Wesley A. Clark, Director, Computer Systems Laboratory

Enclosed is section 3 of Volume 16 of the LINC documentation -
Programming and Use.

A handwritten signature in cursive, appearing to read 'WAC'.

LINC Volume 16
Programming and Use I

Section 3

LAP6 HANDBOOK

Mary Allen Wilkes

Computer Research Laboratory
Washington University
724 South Euclid Avenue
St. Louis, Missouri

May, 1967

This work was supported by the Division of Research Facilities and Resources of the National Institutes of Health under grant FR-218. Requests to reproduce this document, in whole or in part, should be directed to the author.

History and Acknowledgments

Until 1965 the standard LINC had only 1024 words of memory. Its assembly programs include LAP3, written in 1963, which could handle the fundamentals of program preparation and storage but lacked the automatic filing capabilities of the widely used LINC Utility System written by McDonald, Davisson, and Cox in 1965.

Doubling a 1024-word memory produces another small memory, but at this level the increment is dramatic; it is not possible to handle the file copying and editing techniques of LAP6 on the smaller LINC.

Discussion of the specifications for a new assembly program began in January, 1965. Visits to several biomedical research laboratories, where the LINC is in application, were made in an effort to establish some consensus of users' needs. All of us who use LAP6 are indebted to the many, especially to participants in the LINC Evaluation Program, whose suggestions and LINC experience greatly influenced its formulation.

The technique of handling the LINC tape to accomplish the editing facility is the contribution of my colleagues Mishell J. Stucki and Severo M. Ornstein, and was proposed in discussions during March of 1965. Their careful thought has saved LINC users hours of time.

Colleagues at Washington University have not only made valuable suggestions, but fearlessly have faced unproven versions and forced the flaws to surface. Any "guarantee" of reliability is largely due to them.

Any resemblance between LAP6 and an errant program known sometimes as "LAP5" is purely intentional. The name has been changed to protect the innocent.

To promise the System is a serious thing.

Søren Kierkegaard, *Philosophical Fragments*

Abstract

LAP6 is an on-line program for the 2048-word LINC which uses the LINC keyboard and scope for communication with the user, and the magnetic tapes for storage and working area. It may be used for preparation and editing of any character string (manuscript) or specifically for LINC program preparation. For general manuscript preparation not all of section V and none of section VI will be of interest.

LAP6 handles the manuscript display in such a way that any portion of the manuscript can be displayed at any time and edited directly by simply adding or deleting lines. Changes are shown integrated with the manuscript display as the user types.

Meta commands provide automatic filing of manuscripts and programs on LINC tapes and handle the conversion and memory loading of LINC binary programs. Debugging aids include displays of symbol tables and errors, and repeatable access to the manuscript display for editing and reconverting. Meta commands may be added by the user to suit his needs.

The Handbook describes the "Standard" configuration of LAP6 when referring to specific tape block numbers or block areas (manuscript working area, file area, etc.). LAP6 may, however, have other configurations as described in the "Notes."

LAP6 HANDBOOK

Contents

I.	Operating Procedure.....	1
II.	Display Format.....	1
III.	Keyboard Input.....	2
IV.	Files.....	5
V.	Meta Commands.....	7
	SAVE MANUSCRIPT.....	7
	ADD MANUSCRIPT.....	8
	DISPLAY INDEX.....	8
	COPY MANUSCRIPT.....	10
	COPY BINARY.....	10
	COPY FILE.....	10
	CONVERT.....	11
	DISPLAY SYMBOLS.....	12
	SAVE BINARY.....	12
	EXIT.....	13
	LOAD BINARY.....	13
	FREE.....	14
	COPY.....	15
	PRINT INDEX.....	16
	PRINT MANUSCRIPT.....	16
	LIST.....	16
VI.	Conversion Conventions.....	19
	Bibliography.....	26
	Chart I. Standard LAP6 Tape Organization.....	27
	Chart II. LAP6 Keyboard Code.....	28
	Chart III. LAP6 Instruction Mnemonics and Codes.....	29
	Chart IV. Reasons for NO.....	30
	LAP6 Operation Summary.....	31
Notes		
	Efficient Handling of LAP6.....	1
	Conversion Procedure.....	1
	Exiting.....	1
	Working Area Length.....	1
	LAP6 Manuscript Structure.....	2
	Copying LAP6 Tapes.....	2
	Filing Limitations.....	3
	Index Structure.....	4
	LAP6 Configuration.....	5
	Various Failures.....	6
	Tape Stopping.....	6
	Bad Block.....	7
	Tape Missing.....	7
	Damaged or Lost Manuscript.....	8
	Damaged Index.....	8
	Damaged File Area.....	9

LAP6 HANDBOOK

I. Operating Procedure

A. LAP6 occupies blocks 300-347 of a LINC tape. To operate:

1. Mount a LAP6 tape on unit 0.
2. Read blocks 300 through 307 into quarters 0 through 7:
Set LEFT SWITCHES: RCG (code 701)
Set RIGHT SWITCHES: 7300
Raise the DO TOG INSTR lever.
3. When the tape stops, push START 20.

This is the only console start or restart procedure. LAP6 is ready to accept keyboard input.

B. At the keyboard you may:

1. Ask LAP6 to "do something" by stating a meta command.
2. Add or delete a manuscript line.
3. Change the display to look at different manuscript lines with a locate request.
4. Rescind any of the above while they are being typed by striking the delete key, marked **del** on the keyboard.

C. At the console you may push START 20 to erase the current manuscript.

II. Display Format

All manuscript lines and meta command statements are displayed as they are typed. All keyboard characters (see Chart II) are displayed except EOL, **del**, and CASE.

- A. Manuscript lines are automatically centered around the middle of the scope and are displayed with "line numbers" assigned by LAP6. The manuscript lines appear at the right of the corresponding line numbers.
1. Six manuscript lines are normally displayed. This may be varied between 1 and 15 lines by rotating knob 7 on the display module.
 2. Lines which begin with either a number sign (#) or a left bracket ([) are automatically positioned one space to the right of the line number. All other manuscript lines are positioned 5 spaces to the right.
 3. A marker, **_**, appears with the current line (q.v.) on the right of the scope and marks the last usable slot on the scope line. Manuscript lines which are too long to fit on one scope line (more than 20 characters) are automatically carried over to the next scope line. Breaking words between scope lines can be avoided by spacing over to the marker when the next word is too long to fit in the remaining space.

- B. Meta command statements, preceded by the **META** arrow, appear automatically at the bottom of the scope without a line number.

III. Keyboard Input

- A. Some keys on the keyboard have been assigned both upper and lower cases. The characters in the middle of the keys are lower case (see Chart II), and LAP6 normally interprets the keyboard as lower case. To select upper case, strike **CASE**, and then strike the upper case character. The shift is not permanent; it is good for one character only. LAP6 returns to lower case automatically.

B. META COMMANDS

1. A meta command is one of 17 special directives to LAP6. It is executed by LAP6 at the time it is stated and erased from the display at that time.
 - a. The upper case character **META** produces an arrow at the bottom of the scope. To state a command, type the **META** arrow and the command, followed by any arguments the command may require. Terminate the statement by striking the "End of Line" key, **EOL**.
 - 1) The command is executed when the terminating **EOL** is struck.
 - 2) Illegal command statements are deleted automatically. Try again.
 - 3) A meta command must be stated on a line by itself, i.e., a line beginning with the **META** arrow. It cannot be stated while entering a manuscript line.

C. MANUSCRIPT

1. A manuscript is a collection of manuscript lines retained by LAP6 as a permanent record of keyboard input. A manuscript can be edited, saved in a file, copied from file to file, or otherwise manipulated.
 - a. One quarter of the LINC memory is used to collect manuscript. As the quarter is filled, it is saved on the LAP6 tape in the "manuscript working area" beginning in block 350 (see Chart I). 512 keyboard characters fill one block.
 - b. LAP6 will accept up to 45₁₀ blocks of manuscript. Additional manuscript is ignored until the size of the manuscript is reduced.
 - c. The manuscript in the working area, accessible to the scope and keyboard, is called the current manuscript.

Keyboard Input

2. A manuscript line is any combination of keyboard characters (Chart II) excluding **del** which does not constitute a meta command or locate request, and which is terminated with **EOL**. An **EOL** by itself does not constitute a manuscript line and will be ignored by LAP6.
 - a. A manuscript line may not exceed 512 struck characters. This includes spaces and the terminator. It does not include **CASE**. Lines are terminated with **EOL** by LAP6 automatically after the 511th character is struck.
 - b. LAP6 assigns a line number to every line entered. The numbers are sequential, beginning with 1, and octal.
 - c. LAP6 will accept no more than 7775₈ manuscript lines. Additional lines are ignored until the size of the manuscript is reduced.

56	ADD 9J	
57	#6K STC p+5	
60	JMP 3Y-2	
61	LDA i	
62	MTB	
63	STA	-

Example 1. Manuscript Display.

3. Current Line
 - a. The last line number on the scope (e.g., 63 above) is called the current line number. It identifies the manuscript line currently being, or about to be, added to the manuscript. At no time during the manuscript display does the manuscript not have a current line number.
 - b. The number 1 appears as the current line number whenever the current manuscript is erased by pushing START 20. After that the next current line number appears every time **EOL** is struck in terminating a manuscript line.

In Example 1 when the current line 63 is terminated with **EOL**, all the lines will move up one space, line 56 will disappear and 64 will appear as the new current line number.

Keyboard Input

D. LOCATE REQUESTS

1. The manuscript is said to be "located" at the current line. For reading or editing, it may be relocated, establishing a new line number as the current line number, at any time. There are two kinds of locate requests:
 - a. Meta command: the manuscript may be relocated by typing the **META** arrow, the line number of the last line you want to read, and **EOL**.

E.g., to locate at line 105, type

→104EOL

which will cause LAP6 to display lines through 104, and to display 105 as the current line number (Example 2).

- b. The manuscript may also be located by using the following undisplayed key combinations:

CASE 0 (zero): Forward one frame (begin display with current line)

CASE 1: Forward one line

CASE Q: Backward one frame (make first line on scope the current line)

CASE W: Backward one line

E.g., to change Example 2 so that 102 is the current line number, type either →101EOL or strike **CASE W** three times.

Like the meta commands these key combinations cannot be given while a manuscript line is being entered.

- c. Since you will seldom know the exact line number of the line(s) you would like to see, →*Line Number*EOL can be used to locate the general area, and the above key combinations to 'zero in.'

101	-----EOL
102	-----EOL
103	-----EOL
104	-----EOL
105	-----

Example 2. Locating at line 105.

2. To locate at the end of the manuscript, request any line number (octal) larger than the last line number. To locate at the beginning of the manuscript, type →0EOL. (→777EOL is equivalent to →0EOL.)

E. EDITING

1. Manuscript lines may be added or deleted wherever the manuscript is located. You may add or delete, in any combination, as many lines as you like.
 - a. By definition, a new line of manuscript is added on, and only on, the current line. It is retained exactly at the point at which it is added as a permanent part of the manuscript.

In Example 2, lines added will become lines 105 and following.

- b. Striking **del** will delete the information on the current line. If no information has been entered on the current line, the previous line will be deleted. In either case the line number of the deleted line is retained as the current line number.

In Example 1, striking **del** will delete "STA" on line 63, leaving 63 as the current line number. Striking **del** again will delete "MTB" on line 62; all the lines will move down one space, line 55 will appear as the first line displayed, and 62 will be the new current line number.

In Example 2, deletions will begin with line 104.

- 1) A deleted line is permanently deleted from the manuscript.
 - 2) It is not necessary after deleting to strike **EOL** before continuing with the next operation.
2. Manuscript lines following the point at which changes are made are automatically renumbered by **LAP6**.
3. The **LAP6** tape will move frequently, but briefly, while you are locating or editing. It will however, move at unpredictable times, perhaps even when you are entering a manuscript line or meta command. Try to ignore it.

IV. Files

Any tape on any unit may be used as a **LAP6** file, and several of the meta commands described in the next section use the tapes in this way.

A tape need not have **LAP6** on it to be a file tape. (In which case the **LAP6** area may be used for anything you like. See Chart I.)

An entry in a **LAP6** file may be either: a **LAP6** manuscript or a **LINC** binary program. A file may contain only binary programs, or only manuscripts, or both.

Files

A. File Index

When a tape is used as a file, blocks 426 - 427 are automatically reserved for an Index in which information about the file entries is recorded by LAP6.

1. Entry Names

- a. A file entry must be given a name at the time it is filed. In the Index one name describes one manuscript or one binary program, or both. A full Index, 63 names, therefore describes a maximum of 126 entries.

(1) A binary program given the same name as a manuscript is not necessarily the same program.

- b. An entry name can be any combination of displayable keyboard characters (except the comma), so long as at least one character is not a number between 0 and 7.
- c. Names can be no more than 8 characters long. Spaces at the beginning of a name do not count; spaces in the middle or at the end do count.

B. File Entry Placement

1. When a tape is used as a file, blocks 430 through 777 and blocks 0 through 267 are reserved by LAP6 for file entries (Chart I).¹ Within the reserved area a file entry is automatically saved as close as possible to the Index at 426 - 427. Thus, blocks 430 through about 570, being closer to the Index, will be filled with file entries before any entry is saved below block 270.

The order of entries on the tape can thus be controlled by the order in which they are filed, - if you care. E.g., the first entry filed in a new file will always be put at block 430. If it is, say, five blocks long, the second entry filed will be put at block 435, etc., until the free area below block 270 is closer to the Index than the free area above it. The rule is simply that the unused file space nearest the Index, on either side, will be the next space used.

2. A file entry is always saved in contiguous blocks. Block 777 is not contiguous to block 0.
3. Since the assignment of tape blocks within the reserved file area is entirely controlled by LAP6, care should be taken to distinguish file tapes from tapes you want to structure yourself.

¹Blocks 270-277 are never used by LAP6; they are yours.

V. Meta Commands

The following meta commands all use the tape(s) or scope, or both, for their execution. Most return automatically to the manuscript display when the operation is finished. They do not, however, necessarily always return with the manuscript located at the line which was current when the command was given.

- A. The meta commands, with the exception of ADD MANUSCRIPT, do not change the current manuscript in any way. (Except perhaps to relocate it.)
- B. Meta commands may be given at any time during the manuscript display, i.e., no matter where the current manuscript is located or what sort of editing has been done. They may be given in any order.
- C. If LAP6 finds that a legally stated command cannot be executed, NO will appear on the scope. Strike EOL to return to the manuscript display. The current manuscript and all files are still intact. (See Chart IV. to explain the NO.)
- D. Except for FREE, the following meta commands must be two letters, followed sometimes by arguments such as LN, LN, NAME, UNIT in that order.
 1. LN means a manuscript line number. NAME means the name of a file entry. UNIT means the tape unit, either 0 or 1, holding the file.¹
 2. When there are two or more arguments, they must be separated by a comma.
 3. Spaces are permitted almost anywhere.
 4. In the following, parentheses indicate optional arguments.

E. Commands

1. SAVE MANUSCRIPT

Format: → SM (LN, (LN,)) NAME, UNIT¹ EOL

- a. When no line numbers are given, the current manuscript is saved by NAME in the selected file, as close as possible to the Index.
- b. When line numbers, LN, are specified, LAP6 saves part of the current manuscript as a separate, new manuscript in the file by NAME. The line numbers, inclusive, indicate the part to save. When only one line number is given, the second line number is assumed to be the last line number of the current manuscript.

eg → SM 720, 1325, NAME = 210

no command button
SM + 15000

¹ UNIT may also be specified as 4 or 5 if there is a second tape transport whose instructions are coded 740 (MIT) and 750 (MIT u).

Meta Commands

AM

DX

When part of a manuscript is saved, LAP6 must isolate that part as a separate manuscript; to do this LAP6 uses the working area on the other unit of the tape transport. I.e., to save part of a manuscript you must have a tape on the other unit, and you must temporarily not care about the contents of the working area blocks (350 and following). This is the only command which automatically uses the other unit; it does so only when line numbers are included in the arguments.

- c. If a manuscript of the given *NAME* is already present in the selected file, LAP6 will display **REPLACE?**. Strike **R** to replace the file entry. Strike **EOL** not to.
- d. Strike **EOL** to interrupt the command and return to the manuscript display. This has no effect once LAP6 has updated the Index.

2. ADD MANUSCRIPT

Format: →AM *BN,UNIT*EOL
or →AM *NAME,UNIT*EOL

Any LAP6 manuscript may be added to the current manuscript.

- a. The manuscript to be added is identified either by its first block number, *BN*, on any tape *UNIT*, or by *NAME* in a file.
- b. The incoming manuscript is added to the current manuscript at the current line. This is the only time that the location of the current manuscript is relevant to the execution of a meta command. ~~The current manuscript may be added to itself anywhere (→AM 350,0EOL).~~ ↗ mD
- c. After the command the new current line will be at the end of the added manuscript (just as though the incoming manuscript had been added at the keyboard one line at a time). Manuscript lines which follow the added manuscript are automatically renumbered.
- d. If there is no current manuscript, the added manuscript will be the entire manuscript.
- e. If the arguments are inadvertently omitted, LAP6 will try to add a manuscript from block 0, unit 0. It is best not to interrupt it.

3. DISPLAY INDEX

Format: →DX *UNIT*EOL

This command displays the contents of the Index of the specified LAP6 file, and permits unwanted entries to be deleted. An example of the display might be:

NAME	BN	#BLKS
GEORGE	M 430	12
	B 262	2
12345678	B 442	3
DP-MAR20	M 445	27
	B 474	6
DP2 APR2	M 264	4
DISPLAY	B 502	3
MNEMONIC	M 505	11

Example 3. Index Display.

I.e., the manuscript (M) named GEORGE starts at block number (BN) 430, and is 12 (octal) blocks long (#BLKS). The binary program (B) named GEORGE is in blocks 262-263. Etc.

- a. 8 entries are displayed per frame.
- b. A manuscript and a binary program with the same name (e.g., GEORGE, or DP-MAR20) always appear together, manuscript first, in the Index display. Thus, by inference, this file contains no manuscript named 12345678 or DISPLAY, and no binary program named DP2 APR2. (No inference can be made about a binary program named MNEMONIC without displaying at least one more entry.)
- c. The following key combinations relocate the display:
 - CASE O: Forward one frame
 - CASE I: Forward one entry
 - CASE Q: Backward one frame
 - CASE W: Backward one entry
- d. Strike EOL to return to the manuscript display.
- e. Striking del will delete the last entry displayed (e.g., the manuscript named MNEMONIC in Example 3). Whatever is deleted from the Index display is effectively deleted from the file. The file space is made available for later use by LAP6.
 - (1) Strike R to restore the Index if you delete the wrong thing.
 - (2) Strike the number sign, #, to make the deletions permanent and return to the manuscript display. (Returning with EOL effectively ignores any deletions.)

Meta Commands

CM

CB

CF

4. COPY MANUSCRIPT Format: →CM NAME,UNIT EOL
 5. COPY BINARY Format: →CB NAME,UNIT EOL

An entry may be moved from one file to the file on the other unit of the tape transport¹ with a COPY MANUSCRIPT or a COPY BINARY command. The commands are identical except for the kind of file entry copied.

- a. The arguments identify the entry to be copied; i.e., *UNIT* is the unit where the entry of that *NAME* is to be found.
 - b. The entry is automatically given the same *NAME* in the other file, and saved as close as possible to the Index of the other file. The original entry is not affected.
 - c. If an entry of the given name is already present in the other file, LAP6 will display REPLACE?. Strike R to replace the entry. Strike EOL not to.
 - d. Strike EOL to interrupt these commands and return to the manuscript display. This has no effect once LAP6 has updated the Index.
 - e. These commands can be used to reorder file entries. If, for example, the three binary programs A, B, and C are to be next to each other and in that order in a file, copy them (→CB) one at a time in that order into a new file.
6. COPY FILE Format: →CF UNIT EOL

The COPY FILE command copies all the entries from the specified file into the file on the other unit of the tape transport.¹ It does not, however, disturb, or replace, any entries which are already in the other file. Thus, a file may be reorganized by combinations of the commands CM or CB, to move specific entries, and CF to move all the remaining entries.

COPY FILE puts the entries as close as possible to the Index in the other file. It uses any possible gaps created by former deletions and thus "packs" the file. The original file is not affected.

If COPY FILE is used to merge two files, NO may be displayed if the entries from the original file do not all fit into the other file or Index. LAP6 has, however, copied as many entries as it could; inspection of the two indices will show which entries were not copied.

Strike EOL to interrupt the command and return to the manuscript display. This has no effect once LAP6 has updated the Index.

¹From 0 to 1, 1 to 0, 4 to 5, or 5 to 4.

7. CONVERT

Format: →CV EOL

The current manuscript is converted to a LINC binary program, and three displays of information about the program are presented. Conventions which should be followed to prepare a LINC program manuscript are described in section VI.

- a. The binary program will be in blocks 340-347 of the LAP6 tape after conversion. The block numbers correspond to memory quarters 0-7 respectively; e.g., a program written to occupy memory registers in quarters 0 and 2 will be found in tape blocks 340 and 342. Locations not occupied by the binary program are cleared (set to +0).
- b. Strike **EOL** to interrupt the conversion process and return to the manuscript display. This leaves no binary program in blocks 340-347.
- c. Displays:

ERRORS. There are two kinds: undefined symbols appear with one manuscript line number; multiply defined symbols appear with at least two line numbers. Ten errors are displayed per frame. More than 45 errors will not be displayed, and the display is automatically omitted if there were no errors.

```

ERRORS
2A 1072 6015
1X 75

```

E.g., 2A is defined at both line 1072 and line 6015.
 1X is referred to on line 75, but never defined.

Keyboard Control: CASE O: Forward one frame
 CASE Q: Backward one frame
 R: Return to the manuscript display
 EOL: Get next display

MEMORY ALLOCATION. This display shows the memory locations, inclusive, required by the binary program. If the manuscript contains more than 10 "location statements" (see section VI), only 10 allocations will be displayed.

```

MEMORY ALLOCATION
20- 401
3670- 40

```

Meta Commands

DS

SB

E.g., the binary program requires memory registers 20 through 401 plus registers 3670 through 3777, and 0 through 40. Reading this display is the simplest way to tell whether a program exceeds a certain number of memory quarters, or whether perhaps certain portions overlap.

Keyboard Control: R: Return to the manuscript display
EOL: Get next display

SYMBOLS. All tags and equalities defined in the manuscript appear in this display. The latest definition is used for multiply defined symbols. Ten symbols are displayed per frame.

SYMBOLS		
VALUE	LINE	
1A 3777	52	
2B 1073	412	

E.g., all references to 1A in the manuscript have been assigned the value 3777 in the binary program, and 1A is defined on manuscript line 52.

Keyboard Control:
CASE O: Forward one frame
CASE Q: Backward one frame
R: Return to the manuscript display
EOL: Save this symbol table for access by the DISPLAY SYMBOLS command, and return to the manuscript display.

8. DISPLAY SYMBOLS

Format: →DS EOL

The symbols most recently saved by an EOL return from the SYMBOLS display may be redisplayed at any time as an on-line debugging aid.

Keyboard Control: CASE O: Forward one frame
CASE Q: Backward one frame
EOL or R: Return to the manuscript display

9. SAVE BINARY

Format: →SB NAME,UNIT EOL

The binary program most recently converted with the CV command by this copy of LAP6 is saved by NAME in the specified file, as close as possible to the Index. Only programs which have been converted with CV can be saved with SB.

- a. Only relevant, but inclusive, quarters are saved. E.g., if the program was written to occupy quarters 1 and 4, quarters 1 through 4 will be filed in four successive tape blocks. (The second and third blocks will contain all zeros.)
- b. If a binary program of the given name is already present in the specified file, IAP6 will display REPLACE?. Strike R to replace the file entry with this entry. Strike EOL not to.
- c. Strike EOL to interrupt the command and return to the manuscript display. This has no effect once IAP6 has updated the Index.

10. EXIT

Format: →EX EOL

The EXIT command makes it possible to leave IAP6, or leave the computer, without losing the current manuscript. The next time this copy of IAP6 is used, the manuscript will still be accessible to the scope and keyboard.

- a. After EXIT, IAP6 halts. Raise RESUME to rewind the IAP6 tape. (Raising it a second time reenters IAP6.)
- b. IAP6 may be reentered by executing the regular start procedure manually at the console (blocks 300-307 into quarters 0-7, START 20), or under program control as described below. In either case, EXIT is good for one reentry only. (Subsequent pushes of START 20 simply erase the current manuscript.)
- c. There is never any reason not to EXIT from IAP6 when leaving the computer.

11. LOAD BINARY

Format: →LO (NAME, UNIT) EOL

If no name and unit are given, the last binary program converted with the CV command by this copy of IAP6 is read into the memory. If a NAME and UNIT are specified, the binary program of that name is read into the memory from the selected file.

- a. Memory registers not occupied by the binary program are cleared (set to +0). Exception: if the binary program itself does not occupy any registers in quarter 3, registers 1770-1777 will not be cleared.
- b. IO uses register 0 after the binary program is read into the memory. Therefore, if the program requires an initial value in register 0, it must be reset.
- c. IO starts the binary program at register 1. You can ignore this feature by simply assigning nothing to register 1, in which case the computer will halt at 1. The program must then be started from the console. Should you wish to use this feature, the following hints may help:

Meta Commands

F

- (1) If the first instruction, or a JMP to the first instruction is put in register 1 (i.e., in the program manuscript following a **EI** statement), the program will start automatically as soon as it is read in.
 - (2) If nothing is put in register 1 and the first instruction, or a JMP to the first instruction is put in register 2, the computer will halt at register 1, but the program can be started by raising the RESUME lever. This is a helpful procedure when there are perhaps switches to be set or tapes to change before the program can be run.
- d. LO leaves the first block number of the program's tape location in the left 9 bits of the Z register. It leaves the corresponding unit number in the right 3 bits of Z (to make it easier to "layer" programs which have no fixed tape location).
 - e. LO "EXITS" from LAP6 before reading the binary program. I.e., the current manuscript may be recovered by executing the regular console start procedure, or by reentering LAP6 under program control as described below. Combinations of manuscript editing, CV, DS, and LO can therefore be used for fairly efficient on-line debugging.

12. FREE

Format: →F EOL

The FREE meta command makes it possible to add features to LAP6. When the command is given, LAP6 reads block 317 into quarter 0 and JMPs to 20. You may put any program you like into block 317. FREE is probably best used for programs which work closely with LAP6, such as manuscript manipulators, or for programs which have frequent use in the local environment. It is, of course, possible to have different FREE programs with different copies of LAP6.

FREE may be used as a dispatcher to any one of several separate programs. To facilitate this, LAP6 leaves the code (Chart II) for any character typed after the F in the right 6 bits of the Accumulator. E.g., the commands →F_{EOL}, →F_{WEOL}, and →F_{:EOL} will leave the codes for EOL, W, and :, respectively, in the Accumulator.

If nothing is put in block 317, the computer simply halts at 20.

LAP6 may be reentered at any time under program control as described below. (Starting LAP6 by executing the regular console start procedure is always legal, but, except following EX or LO, the current manuscript will be erased.)

N.b.: the rest of the memory, which contains the LAP6 tape instructions, is not cleared when FREE is executed. Putting undebugged programs into the FREE block is not recommended.

SEE SPECTOR WRITE-UP.

Programmed Return from EXIT, FREE, or LOAD BINARY

These three commands do not return automatically to IAP6. Since they essentially provide the ability to leave IAP6 gracefully in order to run other programs, what happens after one of these commands is executed is up to you.

If you wish, however, after executing one of these commands, to re-enter IAP6 other than manually, you may do so in the memory under control of your own program. Put

```
15) RCG = 701
16) 7|300
```

in registers 15 and 16 and **JMP 15**. (Executing RCG, 7|300 from the console switches and starting at 17 is equivalent.) IAP6 and the current manuscript will be as they were before the command was executed.

There is no other way to enter IAP6 under program control. Do not change the instruction, and do not move it to some other register. Do not enter IAP6 this way unless you left via EXIT, FREE, or LOAD BINARY.

13. COPY

Format: →CP EOL

This command copies the contents of any number of consecutive tape blocks on any unit to any place on any unit.¹ It should not be confused with putting entries in a IAP6 file, and has no effect on a file Index.

When the command is given, the following appears on the scope:

COPY	
? BLOCKS	←Number of blocks to move, octal
FROM BLOCK ?	} Present location
UNIT ?,	
TO BLOCK ?	} Requested location
UNIT ?	

- a. Fill in the question marks as indicated: type the number of blocks to be copied. This will replace the ? on the second line. Terminate with EOL and type the first block number of the blocks to be copied. This will replace the ? on the third line. Terminate with EOL. Continue answering the questions, terminating each answer with EOL. The command will be executed when the last EOL is struck after the sixth line.

¹Unit = 0, 1, 4, or 5.

Meta Commands

PX

PM

LI

- b. Striking **del** will delete the answers and restore the question marks, one at a time. Answer the question again. (Do any **del**s before the sixth **EOL**.) If **LAP6** doesn't like what you typed, the question mark(s) will reappear automatically.
- c. Strike **CASE** to interrupt the display and return to the manuscript display.
- d. The blocks being copied are not affected.
- e. To copy an entire tape, copy **1000** blocks from block **0**.
- f. To copy **LAP6** itself, copy **50** blocks from block **300** on unit **0**. (See Notes, "Copying **LAP6** Tapes.")
- g. Since **LAP6** can move only 7 blocks at a time (because of memory limitations), care must be taken not to overlap the block numbers on the same unit when **COPY**ing more than 7 blocks. Example: requesting **LAP6** to copy 12 blocks from block 550 to block 552 on the same unit will not work, because **LAP6** must write the first 7 blocks in blocks 552-560 before it can read the last 3 blocks from blocks 557-561. Requesting it to copy 7 or fewer blocks, however, from block 550 to block 552 on the same unit will work. (Obviously, if the units are different, the **COPY** will be successful.)

14. **PRINT INDEX**Format: **→PX UNIT EOL**

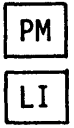
The contents of the specified Index are printed on a teletype which is connected to the **LINC** through bit 0 of the Relay Register, and which is turned on in the "line" mode. Printing time is 1 to 2 minutes.

- a. Strike **EOL** to interrupt the printing and return to the manuscript display.

15. **PRINT MANUSCRIPT**Format: **→PM (LN, (LN,)) (NAME, UNIT) EOL**16. **LIST**Format: **→LI (LN, (LN,)) (NAME, UNIT) EOL**

These two commands both use a teletype to **PRINT** or **LIST** **LAP6** manuscripts, and have several operating features in common:

- (1) The teletype must be connected to the **LINC** through bit 0 of the Relay Register and turned on in the "line" mode.
- (2) Arguments - all optional:
 - (a) If no name and unit are given, the current manuscript will be printed or listed. If **NAME** and **UNIT** are given, **LAP6** will print or list the manuscript of that **NAME** from the file on that **UNIT**.



- (b) If no line numbers are given, the entire manuscript will be printed or listed. When line numbers, *LN*, are specified, part of the manuscript will be printed or listed. The line numbers, inclusive, indicate the part. When only one line number is given, the second line number is assumed to be the last line number of the manuscript.
- (3) Approximately 70₈ manuscript lines are printed or listed per teletype page. The pages are approximately 11 inches long if cut at the marker which appears between pages.
- (4) The *NAME*, if any, and the page number in octal appear at the top left corner of each page. The line number of the first manuscript line which begins on that page appears at the top of the page in the format "LN= ".
- (5) Strike **EOL** to interrupt the printing or listing at any time and to return to the manuscript display. Interrupt with **EOL** if you forget to turn on the teletype.
- (6) Manuscript lines which are too long to fit on one teletype line are automatically carried over to the next teletype line.
- (7) The following characters differ between LINC keyboard (Chart II) and teletype print:

LINC Keyboard	Scope	Teletype
CASE CASE	—	^
CASE A	"	↑
CASE B	"	↓

- (8) When the printing or listing is finished, you must strike **EOL** to return to the manuscript display. (This prevents the manuscript display from burning the scope if you have left the teletype.)
- a. Print Manuscript

Any LAP6 manuscript can be printed with PM. PRINT MANUSCRIPT will not split words between teletype lines. Printing time is approximately 1 minute per page for LINC program manuscripts.

b. List

Any LAP6 manuscript which LAP6 can convert to a LINC program can be listed with LI. In addition to the manuscript lines, LIST prints the symbols associated with the manuscript, and the memory address and memory contents of each register of LINC program generated by the manuscript.

The symbols are printed first, in as many as 4 columns if necessary, but never more than one page. The manuscript is then listed one line at a time with the memory address (p), and the memory contents (CONT), in octal in the left two columns preceding the associated manuscript line:

Meta Commands

LI

```

NAME, PAGE

VALUE LINE    VALUE LINE
1A 0030 17    3T 0426 441
1B 0067 57    4B 1301 1354
1F 1737 2051  . . .
. . .
. . .
. . .

```

```

NAME, PAGE LN=1

p CONT
      [LIST SAMPLE
      EOL
15 0077 #4D SET i 17
16 6063     JMP 1B-4
17 0704     WRC
20 6040     6 |40
21 7301     JMP 4B
. . .
. . .
. . .

```

Example 4. LIST Format.

- (1) When line numbers are specified, the symbols are not printed. Thus, `→LI 1 (,NAME,UNIT) EOL`, lists the entire manuscript without the symbols.
- (2) Do not try to LIST a manuscript which IAP6 cannot convert; use PRINT MANUSCRIPT instead. LIST does not, however, affect the program last converted with CV, the operation of either SAVE BINARY or LOAD BINARY, and there is no need to convert a manuscript before you list it.
- (3) If the manuscript contains "text" (see section VI), the following should be noted:
 - (a) The "p" and "CONT" columns of converted text are listed before the corresponding manuscript line of text. (The manuscript line is not broken; it appears on one line in the listing as it does on the scope.)

Conversion Conventions

- (b) Occasionally (because of the placement of EOLs within the text) "p" and "CONT" will be repeated for a single register. The second listing for the register is the correct one. } Note

- (4) Listing time is approximately 2 minutes per page.

Hint: the fastest way to produce printed copy of program manuscripts on the teletype is to LIST the manuscript only through the symbol table. Interrupt by striking EOL. Then PRINT the MANUSCRIPT. The symbol table, providing both the value and the line number of all the symbols in the manuscript, contains all the information needed to find your way around the printed manuscript.

VI. Conversion Conventions

To facilitate program writing we generally prefer to use symbols instead of the binary, or even octal, numbers which we understand the symbols to represent. We prefer to write "ADD," not "2000" or, even worse, "010 000 000 000," making a note that when the program is put into the memory, 010 000 000 000 must be substituted for the mnemonically easier "ADD."

The CONVERT command, by making the appropriate binary substitutions for you, permits a LINC program to be written in a symbolic form as a LAMP6 manuscript (Example 5). To prepare the manuscript so that it can be converted correctly, certain conventions must be observed. The symbols used generally follow those in the programming examples in "Programming the LINC,"¹ with amplifications.

```

1 [LINC PROGRAM SAMPLE
2   B200
3 #9R LDA i
4   7B [VARIABLE
5   ADA i
6   p+4
7 #6X JMP 6X-10
10  JMP 9R
11  7B=ADD [EQUALITY
12  -4000
13  9R-6X
```

Example 5. LAMP6 Program Manuscript.

¹Wilkes, M.A. and W. A. Clark, LINC, Vol. 16, Programming and Use-I, Section 2, Washington University, St. Louis, June 1965.

Conversion Conventions

For example, a single register instruction (line 7 or 10) must occupy one manuscript line; double register instructions (lines 3-4, and 5-6) occupy two lines. In addition to these instruction lines, some special lines may also appear in program manuscripts, such as comments (line 1), location statements (line 2), and equality statements (line 11).

In the following, converted values are represented in octal (not binary).

A. Location Statements - **B**

To convert the manuscript to a binary program, LAP6 must know what registers you intended the program to occupy. A location statement (line 2 above) tells LAP6 to put the instructions (or perhaps data tables) which follow it into sequential registers beginning with the specified location. In Example 5, the **LDA i** instruction on line 3 will therefore be put in register 200, the **ADA i** instruction on line 5 into register 202, etc.

LAP6 continues to use sequential registers (2000 following 1777, and 0 following 3777) until a new location statement is encountered. There may be as many location statements in a manuscript as you like. If portions of the binary program overlap as a result, the location statement which appears latest in the manuscript has precedence; the instructions (or tables) which follow it will be the ones which appear in the binary program. This is sometimes useful when programs must be written to run with different segments using the same memory locations at different times.

1. A location statement must begin with the location symbol, **B**, followed by octal constants which represent a memory address.
2. Manuscripts containing no initial location statement will be located at 20.
3. A location statement must occupy a manuscript line by itself, except that it may include a comment. It is not, however, a line of program and does not appear in the binary program after conversion.

B. Tags - **#**

To eliminate the need to know the actual memory address of a given instruction or data word, any program line (i.e., any manuscript line except location statements and equality statements) may be "tagged." That is, it may be identified by symbols which represent its memory address, and it may be referred to by its symbolic, rather than its actual, address.

In Example 5, the tag **#9R** on line 3 represents the actual memory address, 200, of the **LDA i** instruction. This symbolic address is then used on line 10 where the code for "JMP 200" is represented symbolically by **JMP 9R**. Similarly the tag **#6X** on line 7 represents the actual memory address 204. "9R" is said to be "defined" on line 3; "6X" is defined on line 7.

Conversion Conventions

1. A tagged line must begin with the tag symbol, #, followed by a *Number Letter* combination.

C. Equality Statements - =

The equality symbol, =, can be used to assign any value you like to a *Number Letter* combination. Equalities should not be confused with tags, whose values are assigned by LAP6 relative to location statements.

Equality statements are generally used when a program value must be treated as a variable. E.g., in Example 5, 7B is defined by an equality symbol on line 11 to represent the code for ADD, or 2000. The equality statement tells LAP6 to convert all references to 7B, such as the one on line 4, to the value 2000. The value can be changed later simply by redefining 7B, i.e., setting it equal to a different value, and converting the program again. "7B" is "defined" on line 11.

1. An equality statement is a *Number Letter* combination followed by the = sign, followed by the definition of the *Number Letter* combination.
2. The definition may be any combination of characters which LAP6 can convert except "text" (q.v.). If, however, the definition itself contains *Number Letter* combinations, they must have been defined earlier in the manuscript.

E.g., putting 7B=6X on line 11 in Example 5 would properly assign the value 204 to 7B, because 6X is itself defined on line 7. Were 6X not defined earlier, it would appear as an undefined symbol in the ERROR display following conversion.

3. An equality statement must occupy a line by itself, except that it may include a comment. It is not, however, a line of program and does not appear in the binary program after conversion.

D. *Number Letter* Combinations

1. The *Number* must precede the *Letter*. The *Numbers* are 1 through 9; *Letters* are capitals A through Z.
2. A *Number Letter* combination may be defined once as a tag if it is not also defined as an equality. If it is defined more than once, it will appear as a multiple definition in the ERROR display following conversion. The latest definition in the manuscript will be the one used.

E.g., the following will be converted as shown:

Manuscript	Conversion
1 B100	p CONT
2 #2F ADD 3	100 2003
3 JMP 2F+2	101 60 <u>42</u> (Not 6102)
4 2F=40	

2F is defined on both line 2 and line 4.

Conversion Conventions

2. A *Number Letter* combination which is used, but not defined, is assigned the value zero, and appears as undefined in the ERROR display following conversion.

E.g., `JMP 3X`, when 3X is not defined, is converted to 6000.

E. Comments - [

The manuscript may be annotated anywhere with "comments" which LAP6 ignores during conversion.

1. Comments must begin with the comment symbol, [. Any displayable character may follow the comment symbol.
2. A comment may be either on a line by itself (line 1, Example 5), or included on any other line so long as it is the last thing on the line (lines 4 and 11, Example 5).

F. Symbolic Operation Mnemonics

1. Three letter mnemonics may be used to represent instruction codes. The mnemonics which LAP6 recognizes, and the codes which they represent, are defined on Chart III.

G. Present Location - p

The small p is used as a symbol for the memory address of the line itself on which it appears. E.g., the information on line 6 in Example 5 will be converted for register 203 (counting from the `B200` statement). The p on that line, therefore, represents the address 203. (`p+4` will then be converted to `203 + 4`, or 207.)

A tag reference can certainly be used instead of p (on line 6, `6X+3` would be equivalent to `p+4`), but p is often simpler.

H. Special Symbols

1. The "i" bit: A small i on a line of program or in the definition of an equality will be converted to 20.
2. The "u" bit: A small u on a line of program or in the definition of an equality will be converted to 10.
3. The vertical bar, |, can be used to separate QN and BN in the second line of tape instructions or to separate the h-bit in half word addressing:

E.g., `4|2777` will be converted to 6777.
`3|7E`, where 7E has the value, say, 45, will be converted to 3045.

Conversion Conventions

The value on the left of the bar is multiplied by 1000 and added to the value on the right of the bar.

You may have any combination of symbols which LAP6 can convert on either side of the bar.

I. Plus and Minus - + and -

These symbols are simply used to combine other symbols arithmetically, as in:

1. Relative addressing: $p + 4$ and $6X - 10$ in Example 5 will be converted respectively to 207 and $\overline{174}$ ($6X$ represents 204).
2. Signed numbers: -4000 will be converted to 3777. $3000 - 4000$ will be converted to 6777 (-1000).
3. Computing differences in memory addresses: $9R - 6X$ on line 13, Example 5, will be converted to 7773 (-4). This is sometimes useful for determining the lengths of tables.
4. Odd combinations: Any combination of convertible symbols may be manipulated arithmetically in any order.

E.g., $i + i - u - 6X + p - LAM + RDC \dots$,
while perhaps not meaningful, is perfectly legal.

LAP6 does all calculations sequentially in ones' complement, end-around carry arithmetic. Thus, $6 - 6$ (or $ADD - ADD$) will be converted to 7777 (-0).

J. 8's and 9's

All numerical information on a program line must be in octal ($6X - 10$, -4000). LAP6 will usually convert meaninglessly any numbers containing an 8 or 9.

K. Text - ""

LAP6 will retain in the binary program the 6-bit keyboard codes of any characters in the manuscript which are set off as "text" by the upper and lower quotation marks. E.g., The following will be converted as shown:

E1000	p	CONT
SAE i	1000	1460
"1R"	1001	0145

(01 is the code for 1, 45 for R; see Chart II.) This facility simply eliminates the need to refer to a keyboard coding sheet, and provides more readable manuscript.

Conversion Conventions

Clearly SAE i
0145 would accomplish the same thing. The text feature, however, is useful for coding characters for, say, display programs.

The "rule" for using text is simply: every character between quotation marks is retained as keyboard code in consecutive half words in the binary program.

1. The text initiator is the upper quotation mark (CASE A). The terminator is the lower quotation mark (CASE B). The quotation marks are not retained.
 - a. There may be any keyboard characters between the quotation marks except the lower quotation mark (which terminates the text) and **del** (which deletes the line). This includes the upper quotation mark itself.

and EOL?

1	E2000	EOL
2	[LABELS	EOL
3	#3R "GRAPH #?"	EOL
4	#4X "DATA: ?"	EOL
5	RUN ?.	[COMMENT EOL

Example 6. Text.

2. A line of text may be of any length. IAP6 will assign 2 character codes to a register, beginning in the left half of, in Example 6, register 2000:

p	CONT	
2000	3245	(GR)
2001	2443	(AP)
2002	3314	(H)
2003	2260	(#?)

Since there are a total of 8 characters between the quotation marks on line 3, four registers will be required. The next register used by IAP6 will then be 2004.

- a. When the number of characters between the quotation marks is odd, the character for SPACE (14) is automatically put in the right half of the last word:

Conversion Conventions

	Conversion
SHD i	1420
"A"	<u>2414</u>

(This will then act as a "blank" if the characters are displayed.)

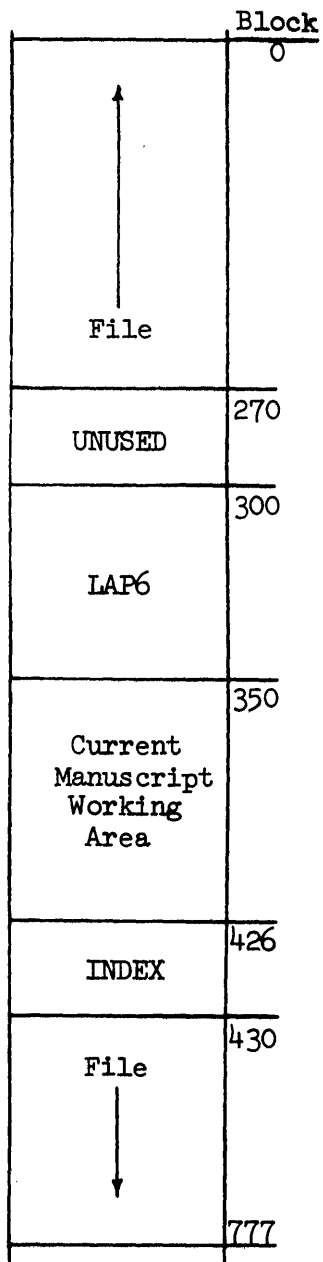
3. A line of text may occupy several manuscript lines when the text includes EOLs. In Example 6 a text line is begun on manuscript line 4 and terminated on manuscript line 5. The rule is the same, however, - every character between the quotation marks is retained, regardless of how many manuscript lines are involved.
4. A line of text may be tagged. Tag text before the upper quotation mark. The tag will represent the memory address of the first register used by the text. Thus, the tag #3R on line 3 of Example 6 will have the value 2000, and the tag #4X on line 4 will have the value 2004. (The instruction ~~SET~~ i α
3R-4X can then be used to supply an index register with the length of the character table generated by the text on line 3.)
5. A manuscript line which contains a line, or part of a line, of text may not contain any non-text information (program instruction, equality, etc.) except that it may be tagged before an upper quotation mark, and may include a comment after a lower quotation mark (line 5, Example 6).

L. Spaces and Line Format

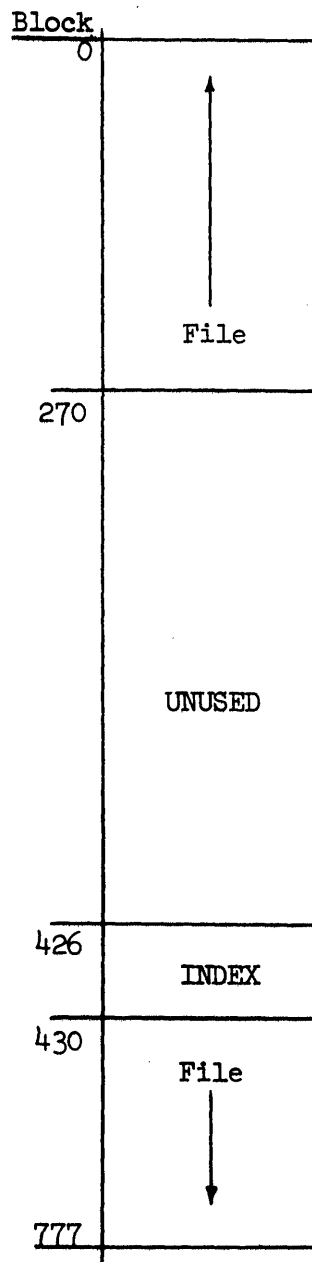
1. Spaces are permitted anywhere in a manuscript, except between the number and the letter of a *Number Letter* combination.
E.g., A D D , p - 1 0 , 3 | 7 C , 6 X = R D C , B 2 0 , # 2 A , etc.
2. Spaces are not required anywhere, except between a number and a letter which do not represent a *Number Letter* combination.
E.g., SAEi5 ≡ SAE5i ≡ 5 SAEi , etc.
3. The elements of a program line may appear in any order on the line so long as tags come first.
E.g., #3K ADD p - 10 ≡ #3KpADD-10 ≡ #3K-10+pADD , etc.
#4RRDC i u ≡ #4R u i RDC ≡ #4R u RDC i , etc.

Bibliography

- Clark, W. A. and C. E. Molnar, "A Description of the LINC," Computers in Biomedical Research, Vol. 2, 1965, Academic Press Inc., New York, 35-66.
- Convocation on the Mississippi, Proc. of the Final LINC Evaluation Program Meeting, March 18-19, 1965, Washington University, St. Louis.
- Marlowe, L., LINC Command System, Nov. 3, 1965, Brown University, (informal report).
- McDonald, M. D., S. R. Davisson, and J. R. Cox, Jr., A LINC Utility System, Technical Report No. 1, Mar. 19, 1965, Biomedical Computer Laboratory, Washington University, St. Louis.
- Moore, R. K., An Operating System for the LINC Computer, Technical Report No. IRL-1038, Nov. 1, 1965, Stanford University School of Medicine, Dept. of Genetics, Palo Alto.
- Wilkes, M. A., LAP3 Users' Manual, Aug. 1963, Center Development Office, decd., Massachusetts Institute of Technology, (informal report).
- Wilkes, M. A., "LAP5: LINC Assembly Program," Proc. of the DECUS Spring Symposium, May 1966, Boston, 43-50.
- Wilkes, M. A. and W. A. Clark, "Programming the LINC," LINC Vol. 16, Programming and Use-I, Section 2, June 1965, Computer Research Laboratory, Washington University, St. Louis.

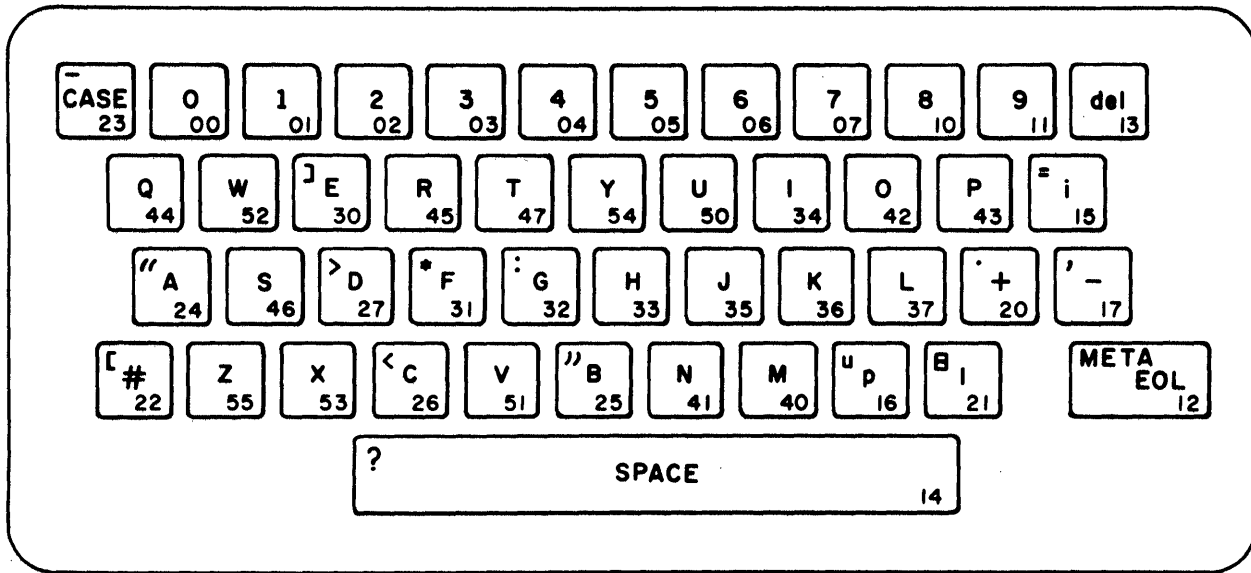


Standard LAP6 (with file)



Standard LAP6 File

Chart I. Standard LAP6 Tape Organization



		Lower Case				Upper Case	
0	00	+	20	M	40	(META)	→ 56
1	01		21	N	41	(CASE SPACE)	? 60
2	02	#	22	O	42	=	61
3	03	CASE	23	P	43	u	62
4	04	A	24	Q	44	.	63
5	05	B	25	R	45	.	64
6	06	C	26	S	46	B	65
7	07	D	27	T	47	[66
8	10	E	30	U	50	(CASE CASE)	- 67
9	11	F	31	V	51	(CASE A)	" 70
EOL	12	G	32	W	52	(CASE B)	" 71
del	13	H	33	X	53	(CASE C)	< 72
SPACE	14	I	34	Y	54	(CASE D)	> 73
i	15	J	35	Z	55	(CASE E)] 74
p	16	K	36			(CASE F)	* 75
-	17	L	37			(CASE G)	: 76

Handwritten note: case/letter

Chart II. LAP6 Keyboard Code

Numerical

0	HLT	517	LSW
0	MSC	600	LMB
5	ZTA	640	UMB
10	ENI	700	MTP
11	CLR	700	RDC
13		701	RCG
14	ATR	702	RDE
15	RTA	703	MTB
16	NOP	704	WRC
17	COM	705	WCG
40	SET	706	WRI
100	SAM	707	CHK
140	DIS	740	MTT
200	XSK	1000	LDA
240	ROL	1040	STA
300	ROR	1100	ADA
340	SCR	1140	ADM
400	SXL	1200	LAM
415	KST	1240	MUL
440	SKP	1300	LDH
440	SNS	1340	STH
446	PIN	1400	SHD
450	AZE	1440	SAE
451	APO	1500	SRO
452	LZE	1540	BCL
453	IBZ	1600	BSE
454	OVF	1640	BCO
455	ZZZ	1740	DSC
500	OPR	2000	ADD
515	KBD	4000	STC
516	RSW	6000	JMP

Alphabetical

		13	MUL	1240
ADA	1100		NOP	16
ADD	2000		OPR	500
ADM	1140		OVF	454
APO	451		PIN	446
ATR	14		RCG	701
AZE	450		RDC	700
BCL	1540		RDE	702
BCO	1640		ROL	240
BSE	1600		ROR	300
CHK	707		RSW	516
CLR	11		RTA	15
COM	17		SAE	1440
DIS	140		SAM	100
DSC	1740		SCR	340
ENI	10		SET	40
HLT	0		SHD	1400
IBZ	453		SKP	440
JMP	6000		SNS	440
KBD	515		SRO	1500
KST	415		STA	1040
LAM	1200		STC	4000
LDA	1000		STH	1340
LDH	1300		SXL	400
LMB	600		UMB	640
LSW	517		WCG	705
LZE	452		WRC	704
MSC	0		WRI	706
MTB	703		XSK	200
MTP	700		ZTA	5
MTT	740		ZZZ	455

Chart III. IAP6 Instruction Mnemonics and Codes

Command	Arguments	NO Appears on Scope
SM	(LN, LN,) NAME, UNIT	If no current manuscript; if Index or file on UNIT is full.
AM	BN, UNIT or NAME, UNIT	If no manuscript is found as specified.
DX	UNIT	If no Index is found on UNIT.
CM	NAME, UNIT	If no entry is found as specified; if other Index or file is full.
CB		
CF	UNIT	If no Index on UNIT; if other Index or file is full.
CV	None	Never
DS		
SB	NAME, UNIT	If no binary on LAP6 tape; if Index or file on UNIT is full.
EX	None	Never
LO	(NAME, UNIT)	If no binary is found as specified.
F	None	Never
CP		
PX	UNIT	If no Index is found on UNIT.
PM	(LN, LN,) (NAME, UNIT)	If no manuscript is found as specified.
LI		

Chart IV. Reasons for NO

CSP
RCG (701)
7300
START 20

PGR
15) RCG
16) 7300
(JMP 15)

LAP6 tape: 300-337 LAP6
340-347 Binary working area
350-425 Current ms area
File tape (with or without LAP6):
426-427 File Index
0-267 } File areas
430-777 }

Only after
EX, LO, or F

MD

MANUSCRIPT DISPLAY. EOL terminates ms line. →LN^{EOL} locates ms at LN. Knob 7 changes no. of lines displayed. START 20 erases current ms. c

META COMMANDS. State during MD. Begin with META (→). Separate two or more arguments* by commas. Terminate with EOL. If NO on scope, EOL to MD.

→SM +■	Save current, or part of current, ms in file on UNIT. If LN given, then temporary use of blocks 350-425 on other unit during execution.	d
→AM U	Add ms to current ms at current line. NAME or BN must precede UNIT.	
→DX U	Display Index of file on UNIT. R: restore Index after wrong deletions. #: make deletions permanent, return to MD.	c, e
→CM ■	Copy ms from file on UNIT to other unit.	d
→CB ■	Copy binary from file on UNIT to other unit.	d
→CF U	Copy all non-duplicate file entries from UNIT to other unit.	e
→CV	Convert current ms. Binary to blocks 340-347, LAP6 tape. 3 displays: ERRORS. Multiply defined, two LNs. Undefined, one LN. MEMORY ALLOCATION. Inclusive locations required by binary program. SYMBOLS. Value, LN for all symbols. EOL: save SYMBOLS, return to MD.	e a, b a b
→DS	Display SYMBOLS saved during CV.	b, e
→SB ■	Save relevant quarters of binary program from last CV in file on UNIT.	d
→EX	Exit from LAP6. Halt. RESUME rewinds LAP6 tape. CSP or PGR.	
→LO □	Load Binary into memory. BN to Z ₃ -Z ₁₁ . Unit no. to Z ₀ -Z ₂ . Uses reg.0. Starts binary at reg. 1. CSP or PGR.	
→F	Free meta command. Block 317 to quarter 0. Code of char. following F to Accumulator. Starts reg. 20. PGR.	
→CP	Copy blocks. Octal keyboard answers to displayed questions. EOL: terminate answers. del: delete answers. CASE: interrupt display, return to MD	
→PX U	Print Index on UNIT on teletype (bit 0, Relay Register).	e
→PM +□	Print ms on teletype (bit 0, Relay Register).	e
→LI +□	List ms with symbols, memory address, and memory contents on teletype (bit 0, Relay Register). LN = 1 lists without symbols.	e

* ■: NAME, UNIT required. U: UNIT required.
□: NAME, UNIT optional. If none, uses current ms or binary from 340-347.
+: LN or LN, LN optional (before NAME, UNIT if any). Indicates part of ms.

BN: first Block Number
CSP: Console Start Procedure, to MD
current ms: ms displayed during MD
LN: ms line number (max. 7775 octal)
ms: manuscript (max. no. blocks = 45;
max. no. chars./line = 512)
NAME: name of file entry, max. 8 chars.
PGR: Programmed Return to MD
UNIT: tape unit number (0, 1, 4, 5)

a		EOL: get next display
a	b	R: return to MD
b	c	CASE 0: fwd 1 frame
b	c	CASE Q: bkwd 1 frame
c		CASE 1: fwd 1 line
c		CASE W: bkwd 1 line
c		del: delete last scope line
d		R: replace duplicate entry
d	e	EOL: interrupt; return to MD

Notes

Efficient Handling of LAP6

Editing. Although lines may be edited in any order, LAP6 is occasionally more efficient (less tape shuffling) if low numbered lines are edited before high numbered lines. If you are simply reading the manuscript (i.e., not changing it), the tape efficiency is not affected.

The abilities to SAVE part of a manuscript and to ADD one manuscript in the middle of another manuscript can be used to reconfigure large manuscripts and to substitute for a REMOVE command. Saving part of a manuscript not only puts the new manuscript in a file, but also leaves it in the working area of the other tape. It can be brought into the current working area with the command
→AM 350,1EOL.

Adding a manuscript is much faster if the manuscript is not on the LAP6 tape. Move it with CM or CP to another tape before executing the AM command.

Conversion is faster if location statements which refer to addresses in one pair of memory quarters (pairs: 0-1, 2-3, 4-5, 6-7) are not interspersed with location statements which refer to addresses in different pairs. This technique will save some tape shuffling during conversion.

Conversion Procedure

The converted values of discrete elements of a program line are always added to the word being assembled. The complemented value of an element preceded by a minus sign is added. Thus, SET OPR, which is converted to 540, is equivalent to SET+OPR; $i\ i$, which is converted to 40, is equivalent to $i+i$.

Exiting

If either EX or LO is executed when 1 is the current line number, the manuscript display will suggest that there is no current manuscript, which may or may not be the case. Therefore, if LAP6, when restarted, displays only the line number 1, do not assume that there is no current manuscript without trying to locate forward (to a higher line number) once.

Working Area Length

LAP6 checks for various boundary crossings. It will not, for example, permit a manuscript to go beyond the working area into the Index.

If the manuscript-Index boundary is encountered, LAP6 ignores additional manuscript input. It will, however, continue to honor all the meta commands. You can thus SAVE the MANUSCRIPT, equip yourself with a configuration of LAP6 which has a larger working area, and continue.

Notes

LAP6 Manuscript StructureCoding Rules:

A LAP6 manuscript is a single string of 6-bit character codes. On a LINC tape the codes are stored in sequential half-words in sequential, contiguous blocks, in the order in which they appear on the scope.

1. The first word of the first block contains the number 2065. The second word contains 5712. The last half-word of the manuscript contains 77. There is no other control information associated with a LAP6 manuscript.
2. The characters are coded as on Chart II. The codes for del (13), CASE (23), META (56 and 57), and 77 (except last) do not appear in a LAP6 manuscript. One EOL (12) does not appear next to another EOL.
3. There are no more than 511_{10} codes between two EOLs.

Generating Manuscript:

A LAP6 manuscript may be generated other than with LAP6, and used with LAP6. However, it must:

1. Conform to the above coding rules.
2. Be put in contiguous blocks on a LINC tape.
3. Not be put in the manuscript working area (blocks 350-425) of a LAP6 tape.
4. Be added to LAP6 with ADD MANUSCRIPT, \rightarrow AM BN,UNIT EOL.

If the above procedure is followed, the generated manuscript may then be treated as any other LAP6 manuscript.

Manipulating Manuscript:

LAP6 may be used to generate a manuscript (e.g., a programming language, a bibliography, etc.) which is to be the data source for some other program. You may assume that the current manuscript on the LAP6 tape (blocks 350 ff.) conforms to the above coding rules, and is therefore accessible, following the commands EX, F, CP, CV, IO, and SM. You may read the manuscript directly from the working area. Do not, however, write back into the working area. Following these commands you may ADD the MANUSCRIPT from the current working area into the working area of another copy of LAP6.

Copying LAP6 Tapes

To copy a LAP6 file use CF. To copy the current manuscript use CP or AM (see "Manipulating Manuscript"). To copy LAP6 use CP and note the following:

Since LAP6 writes over parts of itself in blocks 300-347 while running, the probability that any two copies of LAP6 are ever exactly alike is extremely remote. Copying only some of the blocks between 300 and 347, i.e., to patch up a damaged LAP6, will therefore generally result in a mess. You must copy all 508 blocks at once.

Avoid copying a version in which either EX or LO was the last command executed. Such a version assumes that there is a particular manuscript in its working area. Using this version on a different tape may result in another mess unless the working area is also copied.

One procedure guaranteed to work is for the version which is to be copied to do the copying and to COPY itself. The last command executed will, of course, have been CP when the version is started from another tape, and there will be no confusion about the current manuscript.

Filing Limitations

The filing limitations of LAP6 are obvious. The SAVE BINARY command will file only a program which LAP6 has converted, and which therefore is no more than 8 blocks long. Furthermore, no command will permit an entry to be filed in specific blocks. If LAP6 files it, LAP6 decides where to put it.

There are various ways around these limitations. In this connection note that the COPY BINARY, COPY FILE, and LOAD BINARY commands are not bothered by binary entries longer than 8 blocks. The first two simply copy entries as they are described in the Index, no matter how long they are, and LOAD BINARY loads as many blocks as it can, through quarter 7. It does not load around the end of the memory. Thus, if a binary entry is, say, 12 blocks long and its first block is for quarter 3, LO will load only the first five blocks of the entry, into quarters 3 through 7.

To file binary information which LAP6 has not converted, you can create a pseudo file entry and then COPY into the entry's file space the binary information you want in the file. The easiest way to do this, if 8 or fewer blocks are needed, is to make up the three line manuscript:

```

1      B3777
2      0
3      0

```

Convert it (CV). Save the resulting 8 block binary "program" in a file (SB). This provides a named binary entry in the Index, and 8 contiguous blocks (of zeros) in the file. Display the Index to find out where the 8 blocks are, and then copy (CP) whatever you want into the 8 blocks. LOAD BINARY will, in this case, load all 8 blocks.

You may also, of course, COPY anything into a manuscript file space, which is thus not limited to 8 blocks. Neither ADD MANUSCRIPT nor LOAD BINARY, however, will be any use in retrieving it.

All the filing limitations can be avoided by creating a file Index yourself. You can prepare a LAP6 manuscript of Index entries which when converted is identical to a LAP6 Index, and can be copied (CP) into the two Index blocks on a tape. This permits you to make entries of any length, or to claim specific blocks in a file before LAP6 has a chance to assign the blocks to something else. The Index thus created can then be used by any LAP6 filing commands, which will simply file around your entries. So long as the fixed entries are

Notes

not deleted from the Index, their file space will never be used by IAP6.

If file entries are to keep fixed tape locations when being copied to another file, they must, of course, be moved with CP (not CM or CB). To copy a whole file which contains some fixed location entries, COPY the Index first. Delete from the "new" Index all the entries which do not need a fixed location. COPY FILE from the old file into the new one to move those entries, and COPY the entries which have a fixed location.

Index Structure

You may have occasion to make up a file Index as a IAP6 manuscript, or to write programs which scan an Index for specific entries. The Index structure must be well understood.

A IAP6 Index is always two blocks long and is always in tape blocks whose block numbers end in 6 and 7 (regardless of configuration). The two blocks are divided into 100_8 segments of 10_8 words each. The first segment is the Index identifier; the other 77_8 segments are for Index entries. An Index always has the correct identifier and at least one entry.

Identifier: if the two blocks are an Index, this segment has 5757 in all 8 words. If the two blocks are not an Index, these 8 words contain anything but 5757.

Entry Segments: an entry segment always contains the entry name in the first four words. If the segment contains a name, it must describe either a manuscript in the fifth and sixth words, or a binary program in the seventh and eighth words, or both. A name does not appear without at least one of these entries. A manuscript and a binary program of the same name must be described in the same entry segment.

An unused entry segment has 5757 in all 8 words.

Name: first four words. The entry name is stored in 8 half-words beginning in the left half of the first word of the segment. Unused half-words in the name contain 77. The name conforms to the rules for entry names described under "Files" in the Handbook, and does not begin with a space. The characters are coded as on Chart II.

Manuscript Entry: the first block number of the manuscript's location in the file is stored in the fifth word. The length of the manuscript (number of blocks) is in the sixth word. If there is no manuscript entry, these two words contain 5757.

Binary Entry: the first block number of the binary program's location in the file is stored in the seventh word, with the number of the first quarter into which it is to be loaded in the left three bits. The length of the binary program (number of blocks) is in the eighth word. If there is no binary entry, these two words contain 5757.

Example:

Name	3245	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">NAME</th> <th style="text-align: left;">BN</th> <th style="text-align: left;">#BLKS</th> </tr> </thead> <tbody> <tr> <td>GRAPH</td> <td>M 100</td> <td>33</td> </tr> <tr> <td></td> <td>B 230</td> <td>16</td> </tr> </tbody> </table>	NAME	BN	#BLKS	GRAPH	M 100	33		B 230	16
NAME	BN		#BLKS								
GRAPH	M 100		33								
	B 230		16								
	2443										
	3377										
	7777										
Manuscript Entry	0100										
	0033										
Binary Entry	4230										
	0016										

Coded Entry Segment

Corresponding Index Display

Used segments are not necessarily "packed" in the Index. They may be interspersed with unused segments.

Since all unused segments, or unused 2 word entries within a segment, contain 5757, this number can be used for scanning the Index to locate specific entries. If a segment is not used, it will contain 5757 in the first word of the name. If it is used, and describes a manuscript, the left three bits of the fifth and sixth words will contain 0. If the segment does not describe a manuscript, these bits will contain a 5. If the segment describes a binary entry, the left three bits of the eighth word will contain 0; otherwise, 5. (The seventh word, which may contain quarter number 5, should be ignored.)

LAP6 Configuration

As has been suggested, LAP6 need not have the tape configuration described in the Handbook, and a different configuration may be more appropriate in certain cases. Other configurations operate exactly as the standard LAP6, but do so with reference to different tape blocks and block areas. Thus if you are using another configuration, you must note the different block numbers and make the appropriate adjustment when reading the Handbook.

Regardless of what configuration LAP6 has, it assumes that any other tape which it may use during operation has the same configuration. You cannot, for example, use a configuration of LAP6 whose Index is at 526 to file entries in a file whose Index is at 426.

Any, or all, of the following may differ from the standard configuration:

Tape Unit: LAP6 may operate from another tape unit. If, for example, the configuration is for unit 1, it must be mounted on unit 1, and read with RCG u, etc. SAVE MANUSCRIPT with line number arguments will then use the unit 0 work-area during execution.

Lower File: (standard: 0-267) Begins at block 0 and may end anywhere below LAP6. I.e., may be of any length, creating more or less UNUSED blocks between the lower file and LAP6.

Notes

LAP6: (standard: 300-347) May be anywhere, but is always 50₈ blocks long. The configuration of blocks within LAP6 does not change. The FREE meta block (standard: 317) and the binary working area (standard: 340-347) are always in the same position relative to the first block of LAP6.

Working Area: (standard: 350-425) May begin anywhere above LAP6, creating UN-USED blocks between LAP6 and the working area. It may be of any length, but always ends at the block before the Index.

Index: (standard: 426-427) May be anywhere above the beginning of the working area, but is always 2 blocks long and terminates the working area.

Upper File: (standard: 430-777) May begin anywhere above the Index, creating UNUSED blocks between the Index and the upper file, but always ends in block 777.

The relative positions of the Index and the two file areas will affect the filing behavior of LAP6. Regardless of configuration, however, the filing algorithm does not change. LAP6 always files within the reserved file area only, as close as possible to the Index. If, for example, there is no lower file at all, all entries will be filed sequentially in the upper file area only.

If you change from configuration A on tape A to configuration B whose Index B is in a different place, you can move the Index on tape A to the Index blocks that configuration B requires with CP. Although LAP6 will file entries only in the file area reserved for the new configuration, it makes no assumptions about the information already in a file Index.

If, in order to work with a new configuration, an Index is moved, say, from blocks 426-427 to blocks 576-577, and that Index describes a 10 block manuscript named CHARLES in the old file area at block 430, it still describes such a manuscript. The manuscript CHARLES will be properly found by the new configuration.

If, however, the Index is copied into the old file area on top of a file entry, the record of the now destroyed entry must be deleted from the Index. It, of course, cannot be properly found by the new configuration.

Various Failures

LAP6 makes fairly heavy demands on the tapes and requires well tuned units. Trouble may appear which a machine has not previously exhibited. Although any sort of trouble should be fixed immediately, LAP6 is written to maximize the possibility of recovery.

LAP6, the current manuscript, the Index, and the file entries can all be treated as separate elements should you ever have to patch things up. If one element is damaged, the others are probably all right and can be salvaged. In addition, the general structure and identifying coding of manuscripts and file indices have been kept as simple as possible.

Tape Stopping

LAP6 uses the IBZ instruction frequently to control the tape motion. Occasionally, however, a tape will suddenly stop in the middle of otherwise very busy looking activity, and LAP6 will loop:

Notes

	p-1	→ IBZ	(code 453)
	p	JMP p-1	(code 6000 + p-1)
(START RS here)	p+1	-----	

Since the tape has stopped, the program will never skip on the IBZ test.

Recovery: if you suspect this problem, you simply have to take LAP6 out of the loop:

1. Push INSTR BY INSTR. (This is always safe.) The INSTRUCTION lights will show the code for either IBZ or JMP p-1.
2. STEP a few times to verify that the program is indeed in the 2 instruction IBZ loop. If not, raise RESUME.
3. Put "p+1" (the location after the JMP instruction) in the RIGHT SWITCHES.
4. Push START RS. LAP6 will continue unharmed.

Cure: if this happens frequently, you should find out why the tape is stopping before an IMARK appears. Possibilities:

1. The tape heads need cleaning. Frequently this is all that is required.
2. The tape reel is slipping on the hub. The hub needs another rubber band.
3. The tape timing channel is noisy; spurious END MARKS are appearing.¹
4. A spurious Unit change is causing a motion stop.¹

Bad Block

If the execution of a meta command ^{other than AM} reveals a bad block not in the LAP6 or current manuscript area, e.g., on another tape or in a file, you can recover:

1. STOP the computer.
2. Read LAP6 in again (RCG, 7300).
3. Put 17 in the RIGHT SWITCHES and push START RS.

EXIT from LAP6 and, if LAP6 was executing a filing command at the time, read the section on "Damaged File Area."

Tape Missing

If you forget to mount a tape on another unit, whose hub frustratingly starts to spin because LAP6 is trying to execute a meta command ^{other than AM} which needs that unit, use the 3-step recovery procedure described under "Bad Block." Then mount the needed tape and restate the meta command.

¹For standard LINC's see dwg. 1025, LINC, Vol. 12, Logic Drawings and Timing Diagrams.

Notes

Damaged or Lost Manuscript

There is no need to erase the current manuscript if you suspect that part of it has been damaged. Manuscript lines are self-contained; therefore any line which looks strange can simply be deleted and replaced.

If LAP6 loses track of the current manuscript (START 20 pushed too often, or LAP6 replaced and restarted), execute `→AM 350,0EOL`. The manuscript will be all right if the last thing LAP6 did was EX, F, CP, CV, LO, or SM. If you aren't sure what it did, proofread the manuscript. In any case, executing the AM command will do no harm.

If LAP6 refuses to add a manuscript (AM) to the working area, the first two coding words are damaged. Put 2065 and 5712 into the first two words of the first block of the manuscript. Add the manuscript again and proofread it.

If the final 77 is damaged, LAP6 will, while locating forward, or adding it with AM, keep adding manuscript until it finds a 77 or stops because of a boundary condition. In either case no harm will be done. The garbage will probably be immediately obvious on the scope and can be deleted.

Between the first two coding words and the final 77, the only thing that can be wrong results from rare trouble of the bit dropping (or picking up) variety which changes one character code into another. Thus it is possible that character codes which "do not appear" in a LAP6 manuscript may do just that. The spurious codes will do no harm in the manuscript. A 23 (CASE) however, will give the CONVERT and LIST commands indigestion.

To make them easier to spot and delete, these codes are all displayed:

1. 13 (delete) and 23 (CASE) appear as ?.
2. 56 (META) in the middle of a line unfortunately appears as a dash, -, but usually in an unlikely position in the line. At the beginning of a line it appears as a meta command.
3. 57 (META) appears as the arrowhead.
4. 77 appears as an unrecognizable "squiggle." The line should be deleted immediately since the 77 will truncate the manuscript.

Damaged Index

The Index of a file tape has no back-up. If damage to these two blocks is such that they cannot be read even with RDE, there is nothing you can do.

If the tape is all right, however, the only reason LAP6 will refuse to display the Index (displays NO) is that the Index identifier is incorrectly coded. Read the first block (426) of the Index into the memory and put 5757 in the first eight words only. Write the block back in 426. Execute the DX command again, and delete any damaged entries.

If one block, but not both, of an Index cannot be read, the other block is worth salvaging. On another tape make an Index you don't care about by saving something (SM or SB). Then try to COPY (CP) separately each of the two Index blocks on the bad tape into the Index blocks on the good tape. If at least one block copies successfully, that half of the Index is probably all right. Display

or print the good Index to find out where the file entries are on the bad tape, and move them with CP to the good tape.

To salvage the information in an Index block which can be read with RDE, but which does not check, read the damaged block into the memory and decode it segment by segment at the console (see "Index Structure").

Damaged File Area

If there is tape trouble while LAP6 is reading or writing a file entry, you can salvage the rest of the file by deleting from the Index the entry in the bad blocks, and then copying the file (CF) from the bad tape onto a good tape. Since the entry is not described in the Index, LAP6 will not try to read the bad blocks.

LAP6 always writes the Index before it writes the corresponding entry in the file. Thus, if trouble occurs inbetween these two operations, the Index will describe an entry which is not there. The entry should be deleted from the Index before you try to recover.

Notes

Notes

Notes

LAP6 Master Tape

This tape contains several configurations (see Handbook) of LAP6 plus a few special programs. It should be viewed as a back-up tape, not to be used, from which the things you want to use must be copied onto another tape. Most of the copies of LAP6, in particular, are not in the correct blocks for their configurations, and cannot be run from this tape. The programs and LAP6 configurations are described in detail on subsequent pages.

The format of the Master Tape is as follows:

<u>Special Programs</u>	<u>Block</u>	<u># of Blocks (octal)</u>
COPYTAPE	0	1 (Binary Program)
TRANSLAT	430	1 (Binary Program)
MARK	431	3 (Binary Program)
INDEX MS	434	4 (Manuscript)
LAP6GRID	440	2 (Manuscript)
<u>LAP6 Configurations</u>		
STANDARD LAP6	300	50
Standard BACK-UP	350	50
Configuration 1	10	50
Configuration 2	60	50
Configuration 3	130	50
Configuration 4	200	50
Configuration 5	600	50

Any deviation from the Standard configuration is given a configuration number. It is strongly recommended that the reel of any LAP6 tape, which has a non-Standard configuration, be clearly marked with its configuration number.

Index

The 2 block INDEX on this tape in blocks 426-427, which can be displayed by any Standard LAP6, describes the above tape contents.

M.A. Wilkes
Computer Research Laboratory
Washington University
724 S. Euclid Ave.
St. Louis, Mo. 63110

COPYTAPE

Block 0 1 block

Author: Robert Abbott

This program copies the contents of the entire tape on Unit 0 onto the tape on Unit 1. On the LAP6 Master Tape it is supplied as a leader in Block 0 for making back-ups of the Master Tape. It can, however, operate from any block on any unit.

To operate:

1. Read COPYTAPE into Quarter 0.
2. Mount the tape to be copied on Unit 0.
3. Mount a marked tape on Unit 1.
4. Push START 20.

All the blocks will be read from Unit 0 and written on Unit 1. All the blocks on Unit 1 will then be checked. A block will be rewritten if it does not check. When all have checked, "DONE" will be displayed on the scope, and the tape on Unit 1 will be rewound.

START 20 may be pushed again to copy a second tape, or to restart if there is trouble during operation. There is no need to read COPYTAPE in again before restarting it.

LAP6 manuscripts are not compatible with manuscripts prepared by other LINC Assembly Programs. TRANSLAT translates LAP3 or LAP4 manuscripts into LAP6 manuscripts. The LAP3 or LAP4 manuscripts must be on a tape on Unit 1. They will be filed by name in a LAP6 file on Unit 0.

TRANSLAT operates as a FREE meta command for the Standard configuration of LAP6.

To use:

1. Copy TRANSLAT from block 430 on the master tape into block 317 (the FREE block) of a Standard configuration of LAP6.
2. Using the same Standard configuration, start LAP6 and prepare a LAP6 manuscript describing the manuscripts to be translated:
 - a. On manuscript line 1 type the block number of the first block of a LAP3 or LAP4 manuscript which can be found on Unit 1.
 - b. On manuscript line 2 type the NAME the manuscript (from step a) is to have in the LAP6 file.
 - c. Repeat steps a-b for each manuscript to be translated, describing as many manuscripts as necessary. Block numbers should appear on odd numbered lines; corresponding Names on even numbered lines. Do not put anything except block numbers and legal LAP6 file Names in the manuscript.

Example:

```

1      201
2      TAPETEST
3      206
4      DATA RUN

```

3. When all the Unit 1 manuscripts which are to be translated have been described in the LAP6 manuscript, state the FREE meta command (\rightarrow^F_{EOL}).

TRANSLAT will move the manuscripts from Unit 1 into a Standard LAP6 file on Unit 0. The Unit 0 tape may not contain a needed LAP6 file or Index; TRANSLAT creates its own Index for the LAP6 tape, and uses whatever file space it needs. It does not, however, disturb the information on the Unit 1 tape.

It ruins any previous index on unit 0. Use a scratch tape.

The burden is on the user to prepare the manuscript for TRANSLAT correctly. LAP6 file Names may be no more than 8 characters long, not all numbers. Two manuscripts in the same file may not have the same Name (see "Files" in the Handbook).

TRANSLAT returns automatically to LAP6. Display the Index on Unit 0 (\rightarrow^{DX}_{EOL}) to make sure that all the manuscripts described were translated. A manuscript will be omitted if it was not found as a legal LAP3 or LAP4 manuscript on Unit 1, or if the Index or file on Unit 0 fills up before all have been translated. Move the manuscripts just translated off the Unit 0 tape with COPY FILE (\rightarrow^{CF}_{EOL}) before running the TRANSLAT program again.

MARK

Block 431

3 blocks

Author: Severo M. Ornstein

The MARK program marks virgin tape for use by the LINC. After marking, the tape will begin with a short end zone and end with a long end zone. There are two extra check marks at the end of each tape block.

To use:

1. Check the mark clock. It should be set as precisely as possible at 10 us.
2. Degauss the tape. *(remove your wristwatch!)*
3. If the tape is new, run it manually across the head a few times to align it on the reels.
4. Clean the tape heads.
5. Read the 3 blocks of the MARK program into quarters 0-2 and push START 20, or mount a Standard LAP6 on Unit 0 and LOAD the program (\rightarrow LO MARK, Unit_{EOL}). The following will be displayed:

ALLOW UNIT 0
TO REWIND, THEN
MOUNT VIRGIN TAPE
ON UNIT 1,
AND PRESS MARK BUTTON

6. Allow the tape on Unit 0 to rewind and mount the tape to be marked on Unit 1. Wind about 6" of the tape onto the lefthand hub (about one turn of the reel).
7. Press the "MARK" button.
8. When the program halts, manually rewind the tape onto the righthand hub to get the tape beyond the righthand end zone. This is accomplished by pressing the righthand pushbutton for about five seconds.
9. Raise the "RESUME" lever.

When the tape has been marked, written upon, and checked properly, the following will be displayed:

GOOD TAPE
EOL FOR LAP

If the MARK program was LOAded by LAP6, you can remount the LAP6 tape on Unit 0 and strike EOL to return to LAP6. If, however, you wish to mark another tape, repeat steps 2 through 4 and 6 through 9 above.

If there has been a failure in the marking process, "CHECK FAILURE" will be displayed. Start over again at step 1.

Author: Mary Allen Clark

INDEX MS is a LAP6 manuscript which when converted supplies the 5757 code for unused slots in a LAP6 Index. It is simply an aid to creating indices of your own.

To use:

1. Start LAP6, and erase any current manuscript by pushing START 20.
2. Add INDEX MS to the working area (\rightarrow AM INDEX MS,Unit_{EOL}). The current line number will be 1007.
3. Starting at line 1007, add as many manuscript lines as necessary to describe the entries you want in the Index. (Study the "Index Structure" described in the Handbook.) For example, the following nine manuscript lines, when converted with INDEX MS, will describe the two entries on p. 5 of the Handbook:

1007	⊠10	} Entry Name, 4 lines
1010	3245	
1011	2443	
1012	3377	
1013	7777	
1014	100	} Manuscript Entry, 2 lines
1015	33	
1016	4230	} Binary Entry, 2 lines
1017	16	

- a. Locate each entry segment (line 1007 above) at multiples of 10g (i.e., ⊠20, ⊠30, etc.), in the range 10 through 770.
 - b. Be sure that when the Index Manuscript is converted, each entry segment added will account for 8 words. (Name, 4; manuscript entry, 2; binary entry, 2.)
 - (1) Remember to put 77 in each unused half-word of the four Name words, and 5757 in the two words not used by a manuscript or binary entry.
 - (2) If you use "text" for the Name, remember that a SPACE code (14) will be filled in the last half-word if the number of characters in the Name is odd. The SPACE will then be part of the Name.
4. When as many entry segments as needed have been added to INDEX MS, convert the manuscript (\rightarrow CV_{EOL}). The Index created will be in blocks 340-341 (if Standard configuration). Copy these two blocks (\rightarrow CP_{EOL}) into the two Index blocks (blocks 426-427, Standard) on a tape.
 5. Display the new Index (\rightarrow DX Unit_{EOL}). If it looks strange, return to the manuscript display and correct the Index Manuscript. It has been incorrectly formatted.
 6. In creating the Index, be sure to describe entries which LAP6 can handle. I.e., do not duplicate Names, or describe entries which "go around the end" of the tape, etc.

LAP6GRID

Block 440 2 blocks

Author: Mary Allen Clark

LAP6GRID is a manuscript of the grid patterns used by LAP6 to generate the characters described on Chart II of the Handbook. It may be added to any other LAP6 manuscript and converted with it. LAP6GRID contains tags #6A, #6B, #6C, #6F, #6G, #6L, #6Q, and #6V. It contains no location statements, and occupies 176₈ memory registers when converted.

[ADD 44₈ for upper case in LAP6]

Block	
↑ File	0
UNUSED	270
LAP6	300
Current Manuscript Working Area	350
Index	426
File ↓	430
	777

STANDARD
Unit 0

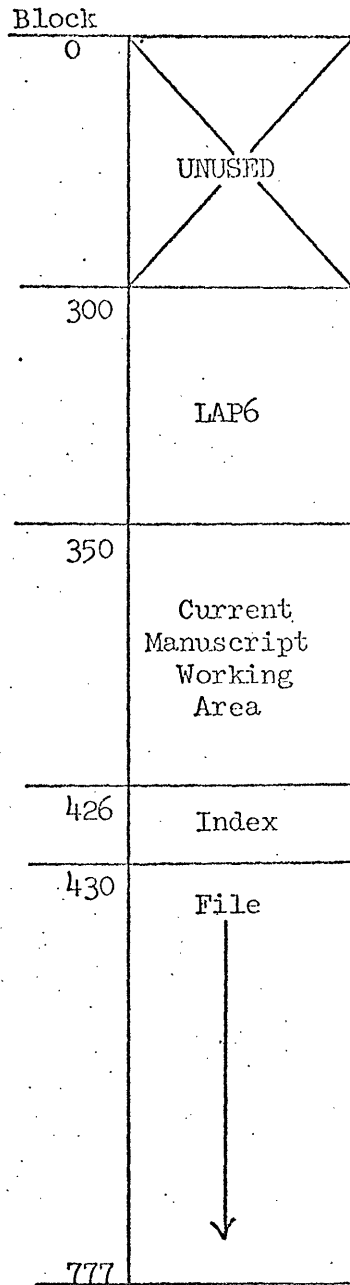
(Copy 50 blocks from block 300, or 350, master tape, to block 300. Run from Unit 0.)

Block	
↑ File	0
UNUSED	270
LAP6	300
Current Manuscript Working Area	350
Index	426
File ↓	430
	777

Configuration 1
Unit 1

Configuration 1 is identical to the Standard configuration, except that it must be run from Unit 1.

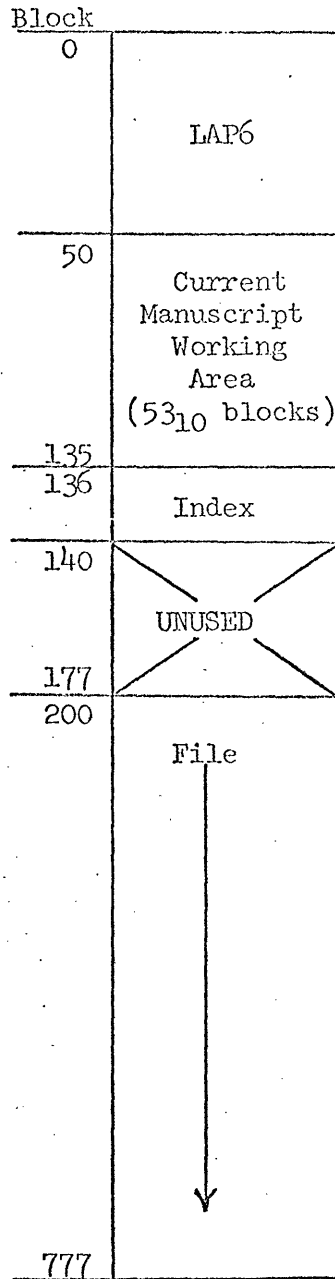
(Copy 50 blocks from block 10, master tape, to block 300. Run from Unit 1.)



Configuration 2
Unit 0

Configuration 2 is identical to the Standard configuration except that it has no reserved file area below LAP6.

(Copy 50 blocks from block 60, master tape, to block 300. Run from Unit 0.)



Configuration 3
Unit 1

Configuration 3 differs from the Standard configuration in all respects. Since it will file only sequentially, and is faster to load, this configuration may be useful for constructing files of binary programs or data.

(Copy 50 blocks from block 130, master tape, to block 0. Run from Unit 1.)

Block	
↑ File	0
UNUSED	270
LAP6	300
Current Manuscript Working Area	350
Index	426
UNUSED	430
File ↓	527 530
	777

Configuration 4
Unit 0

Configuration 4 is identical to the Standard configuration, except that blocks 430-527 are UNUSED.

(Copy 50 blocks from block 200, master tape, to block 300. Run from Unit 0.)

Block	
↑ File	0
270	UNUSED
300	LAP6
350	Current Manuscript Working Area
426	Index
430	UNUSED
527 530	File ↓
777	

Configuration 5
Unit 1

Configuration 5 is identical to configuration 4, except that it must be run from Unit 1.

(Copy 50 blocks from block 600, master tape, to block 300. Run from Unit 1.)

WASHINGTON  UNIVERSITY

ST. LOUIS, MISSOURI 63110

COMPUTER SYSTEMS LABORATORY
724 SOUTH EUCLID AVENUE

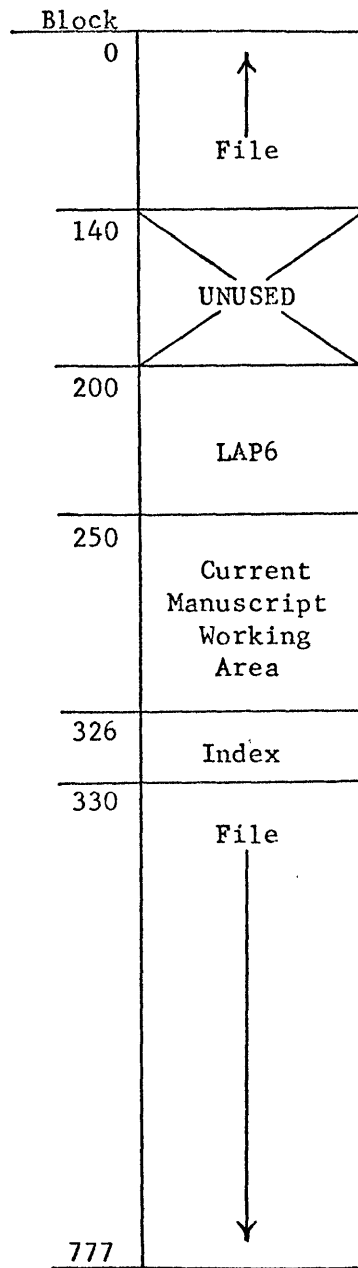
TELEPHONE: AREA CODE 314
FO 1 - 7356

June 1968

LAP6 Configurations 6-10 have been added to the configuration library, and are now available. If interested, please send a marked tape, attention M. A. Clark.

MAC/lz

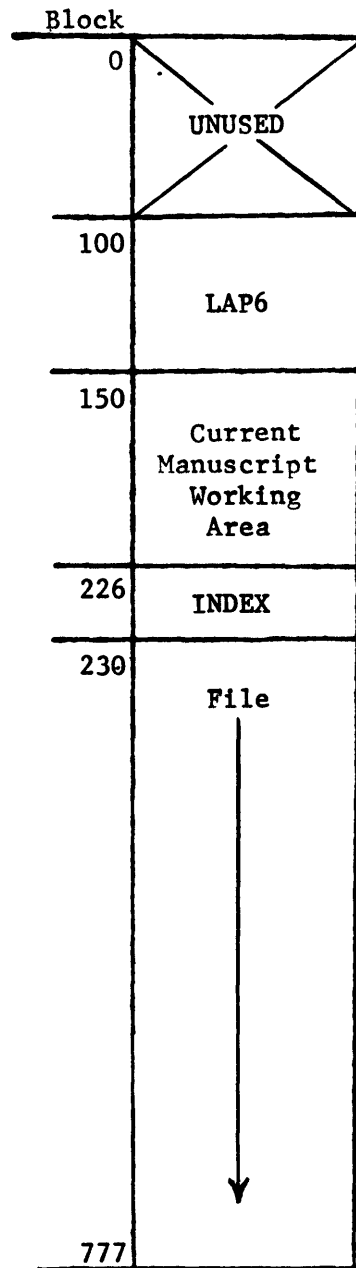
LAP6



Configuration 6
Unit 0

Configuration 6 differs from the Standard configuration of LAP6 in all respects, and has 40₈ UNUSED blocks below LAP6. Run configuration 6 from Unit 0.

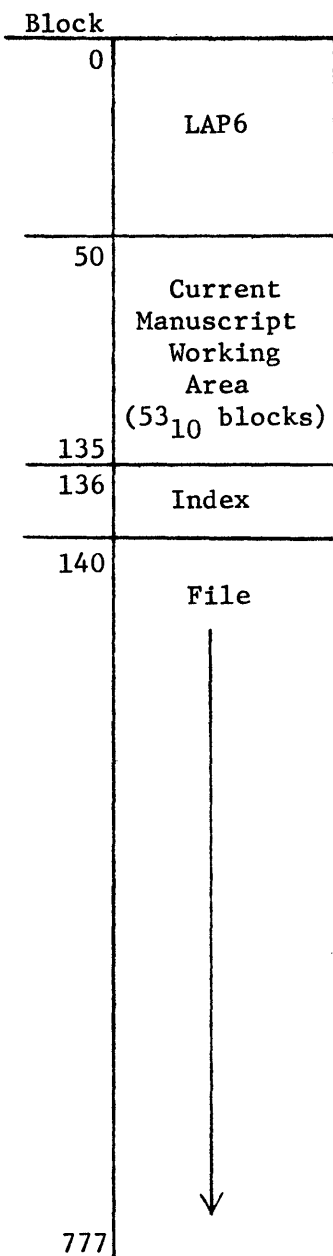
LAP6



Configuration 7
Unit 0

Configuration 7 differs from the Standard configuration in all respects, and has 100_8 UNUSED blocks below LAP6. Run configuration 7 from Unit 0.

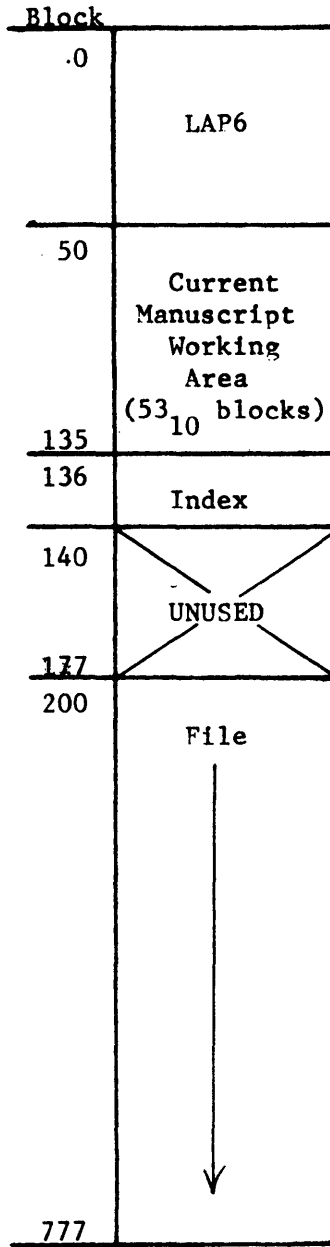
LAP6



Configuration 8
Unit 0

Configuration 8 is identical to Configuration 3 except that it has no UNUSED area and runs from Unit 0. Since it will file only sequentially, and is faster to load, this configuration may be useful for constructing files of binary programs or data.

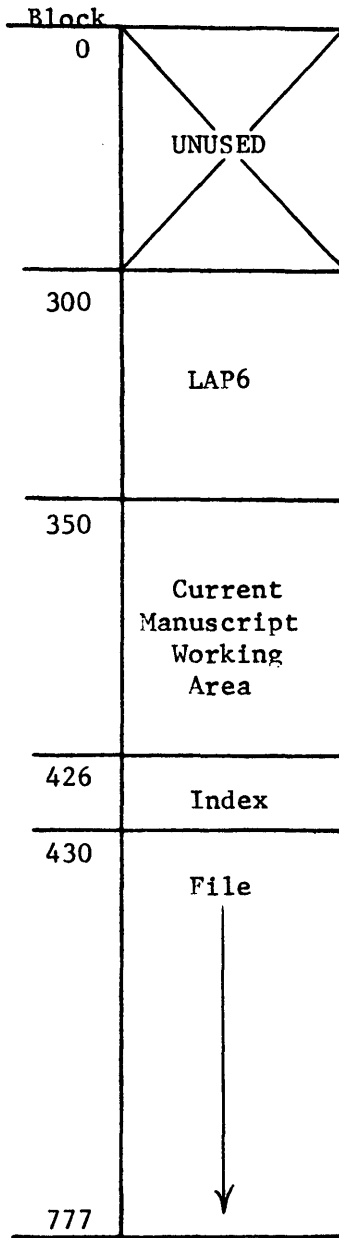
LAP6



Configuration 9
Unit 0

Configuration 9 is identical to Configuration 3 except that it is run from Unit 0.

LAP6



Configuration 10
Unit 1

Configuration 10 is identical to Configuration 2 except that it is run from Unit 1.