

page and  $./3$  means the current line on page 3 (the line on page 3 which has the same line number as the current line). The current line and page are determined by the last command which was executed (see the individual commands for further details).

For ease of use, some of the specifications may be omitted. If the page number of the specification is omitted, it is assumed to be the current page (in this case the slash is also omitted). Thus  $400/.$  and  $400$  specify the same line. If the page number for the end of a range is omitted, it is assumed to be the same as that for the start of the range. Thus  $400/7:3120/7$  and  $400/7:3120$  specify the same range. So do  $400/.:700/.$  and  $400:700.$

If just a page number is given with no line number, it means all the lines on that page. Thus  $/3$  means all the lines on page 3 (except for the delete command). Omitting the first line number of a range means the first line of that page, while omitting the second line number means the last line on that page. Thus  $/3:/5$  specifies all the lines on pages 3 through 5. It is not legal to omit both the line and page number (thus  $:100/6$  is not legal).

Relative page and line numbers may be used instead of absolute ones. Thus  $100+3/.$   $.+27/4$  and  $.-5/.+6$  are all legal specifications. For pages this has the obvious meaning (if you are on page 5, then  $.-4$  is page 1). For line numbers however,  $.+n$  means the nth line after the current one. Thus if a file has lines numbered 100, 103, 106, 109, 111, 142, and 200 and if the current line is 100, then  $.+3$  is line 109 not line 103. The start and end of the page act as boundaries for relative line numbers as follows: If a page has lines numbered 100, 200, 300, 400, 500, and 600, and if 300 is the current line, then  $.+3,$   $.+4,$  etc. are all line 600. Similarly  $.-2,$   $.-3,$  etc. are all line 100.

The symbol "\*" may be used to specify the last line on a page. "\*" may not be used to specify a page (use =BIG to find the last page explicitly). Expressions such as " $*-4$ " are permitted.

If you are not used to line-numbers, and/or would prefer not to see them, and/or are used to TECO or another string editor, you may suppress the appearance of line-numbering by the 'Suppress' switch (see the Set command).

#### 1.4 TRANSLITERATION

SOS is capable of using the full 128 character character set through Teletype keyboards. To do this it uses ? to give each character on the teletype a second meaning. Thus, typing ?2 on the teletype causes ^W to be entered into the file (octal code 027). Similarly a ^W in the file will type out as ?2 on the teletype. To enter a ? one must type ?. To enter lower case characters through an upper-case-only keyboard, precede each letter with a "?", or use the shift commands (\_LOWER and \_UPPER, see Section 4).

Below is a list of the alternate meanings of the various non-alphabetic characters:

Ascii	?x	<sup>^</sup> x	Stanford name	Ascii	?x	<sup>^</sup> x	Stanford name
1	?!	<sup>^</sup> A	down-arrow	24	?/	<sup>^</sup> T	for-all
2	?"	<sup>^</sup> B	alpha	25	?0	<sup>^</sup> U	there-exists
3	?#	<sup>^</sup> C	beta	26	?1	<sup>^</sup> V	circlex
4	?\$	<sup>^</sup> D	and	27	?2	<sup>^</sup> W	iff
5	?%	<sup>^</sup> E	not	30	?9	<sup>^</sup> X	underline
6	?&	<sup>^</sup> F	member, epsilon	31	?6	<sup>^</sup> Y	right-arrow
7	?'	<sup>^</sup> G	pi, bell	32	?4	<sup>^</sup> Z	tilde
10	?(	<sup>^</sup> H	lambda	33	?=	<sup>^</sup> [	not-equals
11		<sup>^</sup> I	tab	34	?<	<sup>^</sup> \	<=
				35	?>	<sup>^</sup> ]	>=
16	?)	<sup>^</sup> N	infinity	36	?7	<sup>^</sup> ~	equivalence
17	?*	<sup>^</sup> O	del	37	?8	<sup>^</sup> -	or (see App.A)
20	?+	<sup>^</sup> P	horseshoe-right	140	?@		accent breve
21	?,	<sup>^</sup> Q	horseshoe-left	173	?[		left-bracket
22	?-	<sup>^</sup> R	horseshoe-down	174	?:		vertical stroke
23	?.	<sup>^</sup> S	horseshoe-up	175	?]		right-bracket

All other characters have the same meaning whether preceded by a question mark or not. SOS will also do this sort of conversion on Teletype output. Vertical tabs and formfeeds are ignored.

If you are on a display, no question-mark conversion will be done, because the displays have the full character set. If you are on a TTY Model 37 or an ARDS display, use the "M37" command (one of the Set commands) to see lowercase and the curly-brackets.

## 2. BASIC COMMANDS

Whenever SOS types a '\*' in column 1 (except after a 'P,S' command), it is in command-mode, ready for input, and waiting for you to enter a command. You can edit any command with ^A (or rubout), ^R, and ^X. Except in the case of giving a filename, ^X will always abort the current command. When typing in a filename (Copy, End commands, etc.), ^X may only clear the current partial filename and expect another; you can tell by whether a '\*' appears.

As at Exec-level, you may preface a command-line with a ';' and the line will be ignored as a comment. This is handy if someone links to you. Similarly, a '?' will inform you of all possible SOS commands, though not the details of individual syntax. You may also type Help, and SOS will interact and try to explain what you don't understand.

The syntax of each command below is shown with optional arguments enclosed in parentheses. Where two or more arguments are shown with "|" between them, it means that any one (but only one) may be used. Thus the I command, below, may take any of the following forms:

```
I
I,<increment>
I<line>
I<line>,<increment>
```

Insert--I(<line>),<increment>)

The insert command is used to insert new lines into the file. Insert accepts a single line specifier as its argument and begins inserting at that line. Each time you complete a line, SOS will add the current increment to the number of the line just inserted and try to insert that line. The current increment may be set by giving a second argument to the insert command. Thus I100,30 will start inserting at line 100 and set the increment to 30. The increment is set to 100 at the start of editing.

When inserting, SOS will type out the number of the next line to be inserted. The user should then type the desired text of that line followed by a return. SOS will then either print the line number of the next line to be inserted or will return to command mode and print a "\*".

To stop inserting, type an ALTMODE. This causes the line in which it appears not to be inserted and returns SOS to command mode. When an altnode is given, the number of that line is remembered and will be used if the next "I" command is given with no arguments.

If the first line to be inserted specifies a page which does not exist, SOS will respond with \*NO SUCH PAGE\*. While inserting, you may correct typing errors using the system editing commands (i.e.,

<sup>A</sup> or RubOut, <sup>R</sup>, and <sup>X</sup>).

Example:

```
*I100
00100  NOW IS THE
00200  TIME FOR ALL
00300  <altmode>
*p100
00100  NOW IS THE
*I
00300  GOOD MEN
00400  ...
```

If the line at which inserting is to start already exists, an insert will be done on line <number> + <current increment> unless there is a line with a number between <number> and <number> + <current increment> in which case the line number will be halfway between <number> and the number of the next line in the file. If, however, any subsequent line which is to be inserted already exists, or if a line with number between two consecutive lines of the insert exists, then the insert will terminate and SOS will return to command mode. As an example, suppose a file has lines numbered 100, 200, 300, 400, and 500. The command I120,40 would allow lines 120 and 160 to be inserted and then would automatically return to command mode. The command I120,20 would allow lines 120, 140, 160, and 180 to be inserted and then would return to command mode.

The current line and page are set to the last line actually inserted (not the one terminated with altmode) on the specified page. If an attempt is made to insert a line containing more than 147 characters (not counting the return at the end) the error \*LINE TOO LONG\* will be given, the line will not be inserted, and SOS will return to command mode. If the next line to be inserted would have a number greater than 99999, SOS will stop inserting and return to command mode.

#### Delete--D<range>

The delete command will accept a range specifier as its only argument and will delete all of the lines specified. If there were no lines in the range specified, SOS will respond with \*NO SUCH LINE(S)\*. There are two exceptions to the normal manner of specifying lines. The delete command will not allow a page specification for the second line number of a range. Thus the command D100/5:200/6 is illegal and will result in the error message \*ILLEGAL COMMAND\*.

The second exception is that the command D/5 will not delete all of page 5, but will instead delete page mark 5. This may result in the error message \*OUT OF ORDER\*, indicating that the deletion results in

a page on which there are some sequence numbers not in proper order (e.g. 100, 200, 300, 150, 200). Note that the deletion has been made. The correct procedure at this point is to renumber all of the lines on the appropriate page (4 in the above example). See the Number command.

All of the lines on page 5 may be deleted by the command D0/5:99999. An attempt to remove page mark 1 (D/1) or some page mark which does not exist will result in the error message \*NO SUCH PAGE\*. The current line and page are set to the last line deleted on the specified page. Note that after a deletion, the command P. will give \*NO SUCH LINE(S)\*.

Print--P(<range>)(,S)  
Output--O(<range>)

The print command accepts a range specifier as its argument. The lines specified will be printed on the teletype. The current line and page will be set to the last line actually printed. If the range specified has no lines in it, the error \*NO SUCH LINE(S)\* will be given. If the range of printing includes the boundary between two pages, "PAGE n" will be printed to indicate the presence of the page mark. The command P<return> is the same as P.!16<return>.

If a second argument of ,S is given, line numbers will be suppressed in the printout. This is useful for clean copies on a TTY model 37. In addition, page numbers will not be printed, and the \* for the next command will be suppressed, so that copies so generated will be absolutely clean. The O command is equivalent to P,S .

When the 'Suppress' switch is on, then the sense of 'P,S' and 'O' are reversed: the line-numbering IS shown for the given range.

List--L(<range>)(,S)

The List command is not yet applicable at Utah, since we lack an online printer. Use RQLIST.

List is like Print in format and error messages, but the output goes to the line printer instead of the teletype. Page headings will be printed at the top of each line printer page: the name of the file, the time and date of printing, and the page number. Page numbers are given in the form M-N. where M is the actual SOS page number on which this text can be found and N denotes the N-th page of line printer paper required for logical page M. The current line and page are set to the last line printed. If no range specification is given, the entire file will be listed.

The S option for suppressing line numbers and headings also applies to the List command. Unlike the print command, however, the \* for the next command will print, as there is no reason for it not to.

End--E(<file name>) (,S)

This command is used to terminate the edit. If no arguments are given, the old copy of the file being edited will be deleted and the new copy will be renamed to the old name, with a higher version number. If an argument is given it will be taken as a file name (it should be of the form foo. or foo.baz) and the new copy of the file will be given this name.

If the original source file (assuming you're not creating) had a non-standard protection code, this will be propagated to the new output.

You may, if pertinent, give the old filename with a ':0' suffix. In this case, the old file is deleted (irredeemably by l0X) and the new one is substituted.

If you wish the file to be pristine, without line-numbers, then add the ',S' tail. The SOS file will be saved and then stripped, just as if by the CUSP Pippy (see Appendix A -- Data Conversion).

SOS will return to the monitor when it has finished.

To abort the E or W command, use rubout (and ^X after a comma).

No spaces are permitted between the E and the filename or comma; E. is not correct.  
Efilename (when the input had an extension) will produce a file with that extension --- as is the Tenex usual convention. To avoid this, use a period after the filename.

### 3. INTERMEDIATE COMMANDS

save World--W(<file name>)

The W command is the same as the E command except that it leaves you editing in the same place. This command is useful for saving the current version of the file in case the system should die. File names given to W commands are "sticky", and need not be given twice.

In case the system dies and you have not done a W recently, try the following procedure: type the system command 'DIR \*.TMP'; a temporary file with your previous job number may exist. If one does, it should be the one SOS was using when the system died. Edit the file to see if it is. If it is not, delete it and try again. If this fails, you are authorized to tear your hair (and ask yourself

why you didn't do a W).

Every now and then, say "DEL \*.TMP" to the system to get rid of accumulated garbage temporary files, which are caused by system crashes or by saying ^C during editing.

The SAVE and ISAVE parameters may be set to cause automatic "W" operations at regular intervals. See the Section 4 for parameter-setting details.

Go--G(<file name>)

This command is the same as the E command, at Utah. It ignores the ',S' if any, however.

Mark--M<line>

This command is used to insert a page mark into the text. It accepts a single line specifier as argument and places the page mark immediately before the line specified. Note that this will increase the page numbers of all following pages by 1. The current line is set to 0 and the current page is set to the new page. Thus if the command M4720/5 is given the current page will be set to 6.

If the line specified does not exist, the page mark will be inserted immediately before the line of next higher number on that page. If the page specified does not exist, the error \*NO SUCH PAGE\* will be given and no page mark inserted.

Hint: When inserting new text, to insert a page mark at the end of the current page use M99999.

Number--N(<increment>(<range>(<starting number>)))

This command is used to alter the numbers of currently existing lines. It takes 0, 1, 2 or 3 arguments. The command "N" with no arguments causes the entire file to be renumbered with an increment of 100.

The first argument is the increment to use in the renumbering. The first line renumbered will be given this number (unless there is a third argument) and each succeeding line will be given a number which is the sum of this argument and the number given the last line. If the renumbering crosses a page boundary, the first line on the new page will be given this number again. The current line and page will be set to the new number of the last line renumbered. Thus if page 3 has numbers 107, 254, 500 and page 4 has numbers 27, 39, 108, and the command N20,/3:/4 is given, the new numbers on page 3 will be 20, 40, 60 and on page 4 will be 20, 40, 60 and the "current position" will be line 60 on page 4.

If there is no second argument, the entire file will be renumbered. If the second argument specifies only a single line, only that line will be renumbered. If there are no lines in the range specified, the error \*NO SUCH LINE(S)\* will be given.

Note that if only portions of a page are renumbered, a situation can be created in which sequence numbers are out of order. If this happens the error message \*OUT OF ORDER\* will be given, however the renumbering has already been done. The best way to correct this situation is to renumber the entire page on which the error occurs.

If the third argument is present, it is used as the number for the first line renumbered. Thus if page 3 has lines 400, 700, 905, 1233 and the command N100,/3,47 is given, page 3 will now have numbers 47, 147, 247, 347. This feature is useful in renumbering a page before deleting a page mark in order to avoid an \*OUT OF ORDER\* error.

If the renumber increment is 0, the error \*ILLEGAL COMMAND\* will be given. If the increment is too large, i.e. some of the line would have numbers greater than 99999, the high order digits of the large numbers will be lost and the error \*WRAP AROUND\* along with the page on which the error occurs will be printed. Note that this leaves the page with line numbers out of order so it should be renumbered with a smaller increment.

#### Display Alter-- Z(<range>)

This command is valid only if you are using a Stanford display console or an Imlac. Each line in the range is sent back to the time-sharing system for alteration. Initially, the text-editing cursor will be set at the first character in the line. All editing features which are operative while inserting text (using the "I" command) are available for altering the line.

#### Alter--A<range>

This command is used to make changes within a line without having to retype the entire line. It accepts a range specifier indicating the lines to be altered. For each of the lines in the range, it prints the line number and then enters a special intra-line editing mode which has its own commands. These commands are not echoed on the teletype, so that the line shown on the teletype at the end of the intra-line edit is nearly the same as the line which will appear in the text.

The intra-line editor maintains a pointer within the line being changed. This pointer points to the character which the next command will effect. The pointer is initially placed pointing to the first character of the line. In general, any command in this mode may be preceded by a number which will cause it to be repeated that number

of times. For example, the command 10D causes 10 characters to be deleted.

next--<space>

This command causes the character pointed to to be printed on the teletype and the pointer to be moved right one character. If the pointer is already at the extreme right of the line, the command is ignored.

back--<rubout or ^A>

Moves the pointer to the left one character. If the pointer is already at the extreme left of the line, a <return><line feed> is done and the number of the line is printed again. The pointer then points to the first character of the line. The characters moved over are printed surrounded by \'s.

Thus, if the line being edited is "How now ..." and the intra-line commands 7<space>, 3<rubout>, <space> are given, the printed line will say "How now\won\n" and the pointer will be on the "o".

Change--C

This command causes a character to be accepted from the teletype. This character is printed and replaces the character pointed to. If the pointer is at the extreme right of the line, the command will be ignored. <rubout> from the teletype will be ignored, but ^A, ^R, or ^X will be used as legitimate replacements. <line feed>, <return>, or <altnode> will cause the remainder of the C command to be aborted. This is useful if a number was used which proves to be too large.

Delete--D

This command deletes the character pointed to. The deleted characters will be printed surrounded by \\'s. If the pointer is at the extreme right of the line, the command is ignored. After the command, the pointer will be pointing to the character to the right of the one deleted. If this command is preceded by a number, only the last 3 characters deleted will be printed, surrounded by \\'s.

Insert--I

This command causes characters from the teletype to be printed and inserted into the line just ahead of the pointer until an <altnode> is seen. The pointer is left pointing at the character to the right of those inserted.

<rubout or ^A> causes the character to the left of the pointer to be deleted. This character will usually be the last character inserted, but it is possible to delete more characters than were inserted. Typing <rubout> when at the left end of the line has no effect. The

characters deleted are printed surrounded by \'s.

<sup>^</sup>R will perform as if '<altmode>PI' were done.

<sup>^</sup>X will abort the line alteration as if '<altmode>Q' were done.

If enough characters are inserted to make the total length of the line more than 147, the error \*LINE TOO LONG\* is given and SOS returns to command mode without having made any changes in the line being altered.

If a <return> or <line feed> is seen, a <return><line feed> will be inserted at that point. This will create a new line whose contents are that part of the line to the right of the present pointer position. The number of this new line will be determined as follows: If the I command was preceded by a number, this number will be added to the current line number to produce the "provisional line number". If there was no number preceding the I command, the "provisional line number" will be created by adding the current insert increment to the current line number. If the "provisional line number" can be used without producing an order or wrap around error, it will be used. If not, the new line will be given a number which is halfway between that of the current line and the number of the next line. If the number of the next line is only one more than the current line number, an \*OUT OF ORDER\* error will be given, SOS will retype the line number and the contents of the line to the left of the pointer. SOS will then be ready to accept more characters to be inserted.

If no errors occur the pointer will be left pointing at the first character of the new line and the current line will be set to the new line created.

finish--<line feed> or <return>

Causes the part of the line to the right of the pointer to be printed and the intra-line edit to be finished. If any lines remain in the range specified for the alter command, the intra-line edit of the next line is started, otherwise SOS returns to command mode.

Quit--Q (or <sup>^</sup>X)

Causes intra-line editing to be terminated without having made any changes in the line being altered. SOS returns to command mode. This command is useful if you discover that a mistake is being made since it restores the line being altered to its original state.

Start over--<control U>

This command causes SOS to start the intra-line edit of this line over. The line is restored to its original state and intra-line editing is re-started. It is equivalent to typing <altmode> and then giving the alter command again.

Skip--S

This command accepts one subsequent character from the teletype without echoing it (note that on a teletype ?1 is counted as a single character) and moves the pointer to the right until it points to the next occurrence of that character. For example to go to the next M type SM; to get to the third m type 3Sm. It will print all of the characters that it passes over. The character currently pointed to is printed but not compared. If there are no further occurrences of the specified character the pointer will be moved to the extreme right end of the line. ^A, ^R, and ^X are legitimate.

Kill--K

This command is the same as S except that it deletes all of the characters it passes over instead of printing them. If there are no further occurrences of the specified character on the line, the command will be ignored instead of deleting the remainder of the line. As with the D command, the last 3 characters deleted will be printed, surrounded by \\'s.

Replace--R

<number>R is exactly equivalent to <number>DI.

Line--L

This command prints the remainder of the line to the right of the pointer, then returns and prints the line number and leaves the pointer on the first character of the line in Alter mode.

Print--P (or ^R)

This command prints the remainder of the line to the right of the pointer, then returns and prints the line up to the position it was in when the command was given.

Justify--J

This command inserts a <return> <line feed> at the place the pointer is currently pointing and then concatenates the portion of the line to the right of the pointer onto the start of the next line. The pointer is left positioned at the start of the next line. The current line will be set to the new line number.

This command is intended to be useful principally in hand justifying text. The error \*LINE TOO LONG\* will be given if the new line created is longer than 147 characters. The error \*NO NEXT LINE\* will be given if this line is the last one on this page. Either of these errors will cause the J command to be ignored and the line number and portion of the line to the left of the pointer to be typed out.

Any other commands to intra-line edit mode will be ignored.

#### 4. OTHER USEFUL COMMANDS

Replace--R<range>(,<increment>)

This command is the same as a delete command followed by an insert command. It accepts a range specifier and an optional second argument (separated from the first by a comma) which if present will be used to set the increment. It performs a D command using the range specifier given and then an I command with the first line specified by the range specifier.

There are some slight differences between the R and D commands. Whereas D/3 will delete page mark 3, R/3 will replace all of the lines on page 3. In addition, the R command will never give \*NO SUCH LINE(S)\* error messages.

Beginning--B

This command simply repositions you to the start of your file. It is chiefly useful in avoiding a range specifier to Find or Substitute commands, i.e., always saying 1/1:....

next line--<line feed>

This command causes the next line of the file to be printed. If the current line is the last of the file, the error NO SUCH LINE(S) will be given. If the current line is the last one on the current page, PAGE n will be printed where n is the number of the next page. If there are several blank pages the following typeout may result:

PAGE 10  
PAGE 11  
PAGE 12  
PAGE 13  
00100 This is the first line on page 13.

previous line--<altmode>

This command prints the line before the current line. If the current line is the first line of the file, the error NO SUCH LINE(S) will be given. Page numbers may be printed as in the <line feed> command.

Copy--C<dest>(\_<file>(,S)),<source range>(,<incl>)(,<inc2>)

TUG  
SOS Manual

The Copy command will insert a copy of a given piece of text in a given location. The source for the text may be on the file being edited or on some other file. The basic form of the Copy command is:

C<destination>,<source range >,<increment>

The Copy command acts as if an I<destination>,<increment> had been done and then all of the lines specified by the <source range > had been typed in. The current line is set to the last line entered.

If the <increment> is large enough that it would cause an \*ORDER\* or a \*WRAP AROUND\* error, the Copy command will pick a smaller increment. The message "INC1=<number>" will be printed to show what increment was chosen. If it is impossible to choose a small enough increment, either "INC1=ORDER" or "INC1=WAR" messages will appear and the given increment will be used.

Since all of the text to be copied must be contained in core at one time, copying huge blocks of text may result in the error message \*INSUFFICIENT CORE AVAILABLE\*. This should never happen, of course. The only possible solution is to copy several smaller blocks.

If the source lines contain page marks, the renumbering of lines will cease when the first page mark is reached. Lines between the first and last page marks will be inserted with their original numbers. Lines after the last page mark will be inserted with their original line numbers unless a second increment is given. This increment should appear immediately after the first increment and be separated from it by a comma. If the second increment is so large that \*ORDER\* or \*WRAP AROUND\* errors would occur, or if no second increment is given and an \*ORDER\* error would occur if no renumbering were done, SOS will choose an increment to use. SOS will print "INC2=<number>" to indicate the increment chosen. If there is no suitable increment, SOS will print "INC2=ORDER" and use the specified increment or the original line numbers if no second increment is given.

If the source lines are to be on some file other than the one being edited, the Copy command is given as:

C<destination>\_<source file name>,<source range specifier>...  
This command may give the error \*FILE NOT FOUND\*.

A special form of the Copy command is available for copying lines from another file when the line numbers are not known. The command:

C<destination>\_<source file name>,S

will cause SOS to respond with a \*\*. The file indicated is now being edited in read-only mode. The P,L, and F commands may be used to find the desired lines. After the lines have been found, say "E". SOS will respond with "SOURCE LINES=". At this point, type the remainder of the C command string (<source range specifier>...).

Transfer--T<dest>,<source>(<incl>)(<inc2>)

The Transfer command moves a set of lines from one place on a file to another. It acts like a Copy command followed by a Delete command. It has only two differences from the Copy command. The first difference is that the source lines must be on the file being edited. The second difference is that the error message \*ILLEGAL TRANSFER DESTINATION\* may be given. This error will occur when the destination is inside the source range. (i.e. T400/5,/2:/9)

If the deletion would produce an order error because of the removal of a page mark, SOS will reinsert one page mark and type

PAGE MARK INSERTED TO PREVENT ORDER ERROR

If the destination specified is on a page which does not exist, SOS will insert the text at the end of the file and type

TEXT INSERTED AT END OF FILE

eXtend--X<range>

This command is like the Alter command except that on each line it automatically puts the pointer at the right end of the line in insert mode. It is useful for adding comments to lines.

display extend-- Q(<range>)

Ignore this command at Utah, unless you are an Imlac.

This command is like the eXtend command, except that the system line editor is used. The pointer is positioned to the end of the line; you are therefore in a mode ready to append characters.

set--\_<parameter> (= <number>)

The set commands enable you to change certain modes and parameters that control the operation of SOS. For the various mode commands below, the first one given represents the initial condition of the editor, while the parameter commands show the initial values.

case shift-- \_UPPER \_LOWER

When operating on a Model 33 Teletype, it is often convenient to invert alphabetic case shifts. This makes it easier to edit text which is largely lower case. The command \_LOWER will cause the editor to enter the characters A through Z as their lower case counterparts and ?A through ?Z as upper case. The command \_UPPER reverses this so that A through Z are taken as upper case letters and ?A through ?Z are taken as the lower case.

Note that this may result in confusion if you are using a model 37 where it is possible to type in lower case letters directly. If this is the case, saying \_LOWER will cause the state of shift to be

inverted for letters. This case inversion also happens in a similar fashion on output.

character set-- DPY (for displays)  
M33 (for teletypes) M37

If you are running on a Model 37 Teletype, you should say M37, which will cause both upper and lower case letters as well as "|", "{", and "~~" to be printed directly rather than in the form ?A, ?:, etc. To set to "not a 37" use M33.

The DPY lets you get back into the right mode when running on a display, even though you may have been so foolish as to get into one of the other modes.

transliteration-- C128 C64

If you are operating on a teletype and would like to be able to type a single ? for ? instead of ??, a mode is provided which turns off the special properties of ? on input. Note that ? will still print as ?? unless you have said DPY. To enter this mode type C64. To leave it use C128.

messages-- NOVICE EXPERT

After you have used SOS for some time, you may want to have messages printed in a shorter form. Saying EXPERT will cause all error messages to be abbreviated to three characters (thus \*ILC\* for \*ILLEGAL COMMAND\*). To go back to full printout of error messages say NOVICE.

line numbers-- SHOW SUPPRESS

The initial mode of SOS is to always show line-numbers on Teletype output. This mode may be reversed by SUPPRESS, which also reverses the sense of the commands 'P,S' and 'O'. The line-numbers will still appear in the output file unless you choose to do an 'E,S'.

justification-- PMAR=1 LMAR=1 RMAR=69 MAXLN=99999

You may change the parameters used by the justification commands (JU, JL, JR, JC). PMAR is the beginning column for the first line of each paragraph, LMAR is the left margin for all other lines, RMAR is the right margin, and MAXLN is the maximum line number. Thus, to indent the first line of each paragraph 4 spaces (to column 5), say PMAR=5.

line spacing-- INC=100

The line increment used by the I command and others may be set directly.

file saving-- \_SAVE=34359738367 \_ISAVE=34359738367

You can cause the W command to be executed automatically at certain intervals, thus causing your edit file to be saved. ISAVE controls the number of lines of text which will be inserted (using the "I" or "R" command) between "W" operations. When SOS decides to save your file, it types "SAVING" on a new line. When the "W" operation is complete, SOS proceeds with the I or R operation. You may wait quietly while all this is happening, or continue to type -- SOS will catch up with you when it is through saving your file.

SAVE indicates the number of file-altering commands (A, X, D, I, R, etc) SOS will accept between automatic "W" operations. The actual saving operation is similar to that described for ISAVE, except that the editor is in command mode, not insert mode, after the operation. Note that the initial values of SAVE and ISAVE, given above, will cause automatic file saving to occur rather infrequently.

give information---<parameter>

This command is used to determine any of several pieces of information which are not otherwise available, such as the modes set by the \_ command. The following variants of this command are available:

- = Prints a summary of this list of things you may inquire about.
- =\_ Prints a short list of the parameters you may set with \_.
- =. Prints the current line and current page (e.g. "500/3").
- =BIG Print the page number of the largest page in the file.
- =CASE Prints out either UPPER or LOWER to indicate the mode set by the \_ command. If the user has done a \_M37 command, SOS will print either MODEL 37 UPPER or MODEL 37 LOWER. If the user is on a display, SOS will print DPY in front of the UPPER or LOWER. If a \_C64 has been done, C64 will also be printed.
- =PMAR Prints the current left margin for the first line of a paragraph not begun with a tab, as used by JU.
- =LMAR Prints the current left margin used by JU, JL, and JC.
- =RMAR Prints the current right margin used by JU, JR, and JC.
- =MAXLN Prints the maximum line number currently allowed by JU.
- =INC Prints the current increment (for I or R commands).

=ISAVE Gives the current ISAVE figure.  
=SAVE Gives the current SAVE figure.  
=ERROR Prints out the last error message given. Printout will always be done in full regardless of any \_EXPERT commands which may have been given.  
=FILE Prints the current name and version of the file you're editing, or the fact that you deferred the naming.  
=STRING Prints out the three strings used by the Find and Substitute commands. (See next section for further details). The strings are printed with the titles SEARCH, SUBSTITUTE, and FOR. The titles are indented while the strings start at the left margin.

## 5. ADVANCED COMMANDS

Join--J<line>

The join command is used to join two successive lines into one. Its argument is the first of the pair of lines to be joined. The new line formed will be given the number of the first of the pair. The error message \*LINE TOO LONG\* may be given, in which case, the lines will be unchanged. If the line given is the last line on a page, the error message \*NO NEXT LINE\* will be given. The current line will be set to the line created if there are no errors.

JUstify--JU<range>

This command takes a range as its argument. Note that there are two letters in the command. It justifies the text in the range by ignoring all line numbers, carriage returns and line feeds in the range, and inserting its own in such a way that adding an extra word to a line would cause its length to exceed RMAR(right margin) - LMAR(left margin)+1 characters. (See the " " command, above, for a list of initial values for parameters.) A word is taken to be anything between blanks. The end of a line is considered a blank. If a word ends in ".", "?", or "!", two blanks are permitted after it. Otherwise, only one blank is permitted, and others are ignored.

Next, extra blanks are inserted between words, starting from the left and right on alternate lines, to make the length of the line exactly RMAR-LMAR+1. Then LMAR-1 blanks are inserted in front of the new line, and it is given a line number which is the same as if the new, justified text had been numbered with a N<INC>,<range> command. A pagemark is automatically generated if the line number for the next line would exceed MAXLN. You will be told if this occurs.

There are exceptions to the above procedure, all having to do with paragraphing. Any of the following conditions are treated as the end of a paragraph: TAB in first column (note that tabs in other places do not start paragraphs), BLANK LINE, PAGE MARK, BEGINNING or END of RANGE.

When one of the above conditions is encountered (except of course beginning of range), the immediately preceding line is not expanded. It is, however, moved out to LMAR by the insertion of leading blanks, if necessary.

If the new paragraph begins with a TAB, the tab is merely inserted into the text. If not, the first line of the paragraph is made to start at PMAR (paragraph margin) rather than LMAR.

#### Justify Left--JL<range>

Lines in the range are left justified by removal of leading blanks, and the insertion of LMAR-1 leading blanks to move them out to the left margin. Paragraphs have their first lines treated as in the JU command. No chopping, filling, renumbering, etc. is done.

#### Justify Right--JR<range>

Like JL, only enough blanks are inserted to move the line out to the right margin. The required blanks are inserted to the right of the rightmost tab in the line.

#### Justify Center--JC<range>

Like JR, only half as many blanks are inserted, so that the line ends up centered between LMAR and RMAR.

For a summary of the parameters used in the Justify family of commands, their initial values, meanings, and how to change them, see the Set(\_) and Give(=) commands.

#### Find--F(<string>)<altmode>(<range>)(,A|,N)(,E)(,<number>)

The Find command is used for locating occurrences of given strings of text. The basic form of the Find command is:

F<string><altmode><range>

The first occurrence of the specified string within the specified range will be found and the line containing that string will be printed. If the range includes more than one page and the line found is not on the first page of the range, PAGE n will be printed where n is the number of the page on which the line occurs. For example, to find the first occurrence of the string "FOOBAR" on page 5, use the

command:

FFOOBAR\$/5

where \$ is used to indicate an altmode. To use the same string used by the last F command, simply omit the string (but not the <altmode>). For example, after the above search, to find the first occurrence of the string "FOOBAR" on page 14, use the command:

F\$/14

Note that it is possible to determine what string will be used in such a case by using the =STRING command. If no previous F command was done, the error message \*NO STRING GIVEN\* will be printed if the string is omitted. Upper and lower case letters will be considered the same inside <string>. Thus the strings "FOO" and "?F?O?O" will find the same lines. The use of <return>, <line feed>, ^~, ^E, ^N, ?:, and ?/ in search strings should be avoided until you learn how to use them. (See the section below on Special Characters).

If the range is omitted (e.g. FFOOBAR\$) then the range searched will be from the line after the present one (essentially .+1) to the end of the file. To search from the present position to some location, give only the second half of the range. Thus to search from .+1 to the end of page 10, use:

FFOOBAR\$/:10

Giving the command F<return> will cause the search to continue from the present point. This differs from the command F\$<return>, which searches to the end of the range specified in the previous search command instead of continuing to the end of the file.

If no occurrence of the string is found in the given range, SOS will simply print a \* and wait for the next command. The current line will be set to the last line found. If no line is found, the value of . will be unchanged.

#### Multiple Strings

To search for more than one string at the same time, separate the strings by a <return>. Thus to find the first occurrence of either "FOO" or "BAZ" on page 5 use the command:

FFOO

BAZ\$/5

If too many strings are specified in this manner, the error message \*TOO MANY STRINGS\* will be given. The current limit on number of strings is 6. The error message \*STRING TOO LONG\* will be given if the total length of all strings being searched for is greater than the table space available (currently 200 characters).

#### Alter switch (,A)

It is possible to cause SOS to enter intra-line edit mode automatically when a string is found. To do this, append ",A" after the range when giving the F command as:

FFOOBAR\$/5,A

When a match is found while using this feature, SOS will enter intra-line edit mode (A command) and move the intra-line edit pointer to point to the first character of the string found (using the <space> command). If the F<return> command is used after editing of that line is finished, the ",A" will remain in effect. Thus F<return> is really a continue command even as far as special modes are concerned. This effect of the F<return> command also holds for the N and E modes explained below.

Line Numbers only (,N)

Occasionally it is sufficient to know just the line numbers on which a given string occurs. This is especially true on teletypes where printing takes a great deal of time. For this reason, SOS allows ",N" to be added to an F command immediately after the range. This will cause only the line number to be printed when a line is found.

Exact compare (,E)

If it is undesirable to have upper and lower case letters treated as being identical, a ",E" may be included in the command string. It should occur immediately after the ",A" or ",N" if either is present, or after the range if both are absent.

### Number of occurrences

It is possible to find more than just the first occurrence of a string. This may be done by ending the command string with ",<number>" where <number> is the number of strings to be found (99999 will almost certainly find all of the strings). This has the effect of giving the F command and then a series of F<return> commands until either the count is exhausted or the end of the given range is reached.

## Special characters

Certain special characters may be included in the string to be searched for. Instead of being matched by themselves, they indicate a class of characters which may occur at that point in the string. These characters are as follows (in the form for Tenex teletypists):

- ? : Will be matched by any "separator". A separator is any character which is not a number, a letter, a ., a %, or a \$. (i.e. a character which cannot be part of a symbol in MACRO or FAIL.)
  - ?/ Will be matched by any character.

- ^E Will cause the character following it to be matched by any character which it would not normally be matched by. Thus the string F^EAB will be matched by FBB, FCB, FDB, F\$B, etc., but not by FAB or FaB. ^E?: will be matched by any letter, number, etc. ^E?/ will be matched only by the begining or end of a line. Thus ^E?/FOO will find only those occurrences of foo at the start of a line.
- ^^ Is used to quote the next character. Thus ^^^E is used to search for the character ^E and ^^^ to search for ^^. Note that ^E^^E (or equivalently, ??7?%) will match any character but ^E (%).
- ^N Is used to mean "any number of" whatever follows it. Thus the string A^NBC will be matched by AC, ABC, ABBC, etc. In case of ambiguity, the shortest such string will be found. Thus the string ^NAB will find B rather than AB. Strings of the form ^N^E^^E are perfectly legal. The example will be matched by any number of characters which are not ^E's.

Certain strings which can be formed with the above characters are considered illegal and may give an \*ILLEGAL SEARCH STRING\* message. The strings are not checked before use, so the message will only be given when an attempt is made to check for a match with that particular part of the string. The illegal conditions are ^E, ^N, or ^^ when not followed by another character and the construct ^E^N... Due to the fact that some of these special searches involve recursion and others require the use of table space inside the editor, it is possible to get the error message \*SEARCH STRING TOO COMPLEX\*. If this happens, try a simpler string.

```
Substitute--S((<ostring><altmode><nstring>)<altmode>
              (<range>)(,D|,N)(,E)(,<number>))
```

This command is used to substitute one string for all occurrences of another string. The basic form of the Substitute command is:

S<ostring><altmode><nstring><altmode><range>  
<nstring> will be substituted for all occurrences of <ostring> in the given range. Note that while the F command finds the first occurrence, the S command substitutes for all occurrences. The Substitute command will print all lines on which substitutions have been made. The line will be printed after all substitutions on that line have been made. As with the F command, "PAGE n" will be printed if the first line printed is not on the first page of the given range or if a subsequent line is not on the same page as a previous line.

For example, to change all occurrences of FOO to BAZ on page 17, use the command:

SFOO\$BAZ\$/17

To use the same strings as were used by the last S command, simply omit both strings and one of the <altmode>'s. Thus if it were now

desired to change all FOO's to BAZ's on page 33, one could use the command:

S\$/33

Note that as with F, it is possible to determine which strings will be used by using the =STRING command. If the strings are omitted and no previous S command has been given, the error message \*NO STRING GIVEN\* will be printed. Again as with F, upper and lower case characters will be considered the same in the first of the two strings (but not in <nstring>).

The effect of omitting the range or of specifying only the last half of the range is the same as for the F command. S<return> is a continue in the same manner as F<return> but is rarely needed since the S command affects all lines in the given range. The current line is set to the last line changed. If no substitutions are made, the value of "." is unchanged.

#### Multiple Substitution

As with the F command it is possible to do several substitutions at the same time. Several strings to be searched for, separated by <return>'s, may be given for <ostring> followed by an <altmode>, then several strings to replace them, again separated by <return>'s, are given for <nstring> followed by another <altmode>. The first string given for <nstring> will be substituted for the first given for <ostring>, the second for the second, etc. If more <ostring>'s than <nstring>'s are given, the last <nstring> will be used to substitute for the excess <ostring>'s. Thus to simultaneously substitute ALPHA for BETA and DELTA for GAMMA on page 5 through page 7 use the command:

SBETA  
GAMMA\$ALPHA  
DELTA\$/5:/7

The errors \*TOO MANY STRINGS\* and \*STRING TOO LONG\* will occur under the same circumstances as for F.

#### Decide switch (,D)

A special mode of the S command is provided in which the user has a chance to look at each line after substitutions have been made in it and to decide whether he wants the new line or the old one. To use S in this mode, put ",D" after the range in the command string. For each line in which a substitution is made, the line will be printed after all substitutions in it have been made. SOS will then wait for a single character to be typed on the user's console. If this character is <rubout>(or <BS>), the indicated substitutions will not be made and the old copy of the line retained. SOS will then proceed to look for the next line and repeat the process. If the character is E (or e), SOS will immediately return to command mode without having made the substitution. Any other character will cause the

line as printed to become the new line and substitution to continue.

Numbers only (,N)

If the user is very sure of himself, he may suppress printing of those lines in which a substitution has been made. To do this put ",N" after the range in the command string.

Exact compare (,E)

As with F, ",E" will cause upper and lower case letters to be treated separately in the first string. This should come after the ",D" or ",N" if present and otherwise after the range.

Special Characters

All of the special characters permitted in the string of an F command (^N, ?, :, ^E, and ^^) may be used in the first string of the S command. This may create a problem, however. Suppose it were desired to change all occurrences of FOO to BAZ but there were strings present containing FOO such as AFOO and FOOBAR. This can be circumvented by giving "?:FOO?:" as <ostring> but leaves the problem of replacing the separators found by themselves. All strings which match one of these special constructs are called partially specified strings. If the construct ?\*<number>?\* occurs in <nstring>, it is replaced by the <number>th partially specified string found by <ostring>. Thus the above problem can be solved by the command:

S?:FOO??:\$?\*1?\*BAZ?\*2?\*\$<range>

To insert a ?\* or an ^^, preceed it by an ^^. If a ?\* is not followed by a number followed by a ?\*, or if an ^^ is not followed by another character, the error \*ILLEGAL REPLACEMENT STRING\* will be printed. This same message will be given if a partially specified string which does not exist is specified as ?\*8?\* when there are only 2 partially specified strings.

BREAK-OUT--<control B>

There eventually comes a time when you make a mistake in the range given for a substitution or justification, and you want to recover from it immediately with as little actually changed as possible. Or you have started a long Find or Print command, but you decide to abort and want the current line/page pointer to be close to where you started (so the file doesn't have to recopy itself).

Rather than panic and pound ^C madly, you may simply hit ^B. ^B is always ready and able to stop cleanly any command which (explicitly or implicitly) involves a range specificifier. ^B will tell you at what point SOS gracefully screeched to a halt, whereas ^O lets a command run to completion (though hiding the output: a binary switch).

If you ^B a substitution, all changes to that point have been made in the text buffer, but only to that point. If you reacted too late, you may need to go back to the source file (do a FILSTAT to see what file you're reading from). To help as much as possible, ^B does not turn off Teletype output...so that you can see the extent of the substitutions made, or whatever. You may do a ^O concurrently.

If you are in the process of copying from a second file, the breakout is deferred until things are stable again.

## 6. LINE NUMBERS REVISITED

It is possible to address lines by content rather than by number. This is done by replacing a <line-number> field of any command by

<altmode><string><altmode><carriage-return>  
The rest of the command string is continued on the next line. Thus, to list everything between a line containing "FOO" and a line containing "BAZ", suppressing line numbers, use the command

```
L$FOO$  
:$BAZ$  
,S
```

where \$ denotes an altmode. Note that if the ",S" had not been included, two carriage returns would be required after the ":\$BAZ\$"; one to terminate the search, and one to terminate the L command.

If the string between altmodes is null, then the string last used in this context will be used again. Note that this is not the same as the string used in the F command.

The search will be from "." to the end of the file, unless otherwise specified. It is possible to otherwise specify: the full construction looks much like an F command, with altmode replacing the F:

```
<altmode>(<string>)<altmode>(<range>)(,D|,N)(,E)(,<number>)<return>
```

The <number> (call it N) specifies that the Nth occurrence of the string is to be used as the designated line. The E option, as in the F command, specifies that upper- and lower-case letters are not the same things.

The D or N options will, upon finding a line, cause the line or its number respectively to be printed. If the next character typed is a rubout or backspace, another line will be looked for. Anything else will cause the line just found to be used as the line you were looking for. this process will be repeated at most N times. You cannot reject the last line (the Nth line), and so N should be other than 1. (99999 will suffice in most cases, and is the largest number SOS can understand on input.)

Also note that the range used in this construction can be another search-type construction. This sort of thing can be nested to a