

STANFORD ARTIFICIAL INTELLIGENCE LABORATORY  
OPERATING NOTE 50.3  
University of Illinois,  
Coordinated Science Lab

October 1970  
1971 addendum

November 1976

SON OF STOPGAP (SOS)

by

William Weiher and Stephen Savitzky  
(Stanford University)

Ted Hess  
(Digital Equipment Corporation)

Charles L. Hedrick  
(University of Illinois)

This manual reflects version 21C(75)-5  
of SOS as distributed by  
The Coordinated Science Laboratory  
University of Illinois at Urbana-Champaign  
Urbana, Ill. 61801

ABSTRACT

SOS is a line-number oriented editor for text files. It features intraline editing, string search and substitution, hyphenless text justification, and other glories.

The work reported here was supported in part by the Advanced Research Projects Agency of the Department of Defense under Contract SD-183 at Stanford.

## CREDITS

The original Stopgap text editor was designed and programmed by Bill Weiner. Steve Savitzky added text justification (JU, JC, JR, and JL commands) and more extensive string search features. Dan Swinehart subsequently added automatic file saving (SAVE and ISAVE). Case inversion commands (Vx commands and V alter command) were added by Rich Neely at Carnegie-Mellon University. Ted Hess translated the code from FAIL to MACRO-10 and adapted it to the environment of recent TOPS-10 monitors, adding a number of useful features in the process. Charles Hedrick rewrote much of the file handling code and did much debugging. This manual was rewritten by Les Earnest at Stanford, and updated by Charles Hedrick.

## TABLE OF CONTENTS

	SECTION	PAGE
I	INTRODUCTION . . . . .	1
	Operation . . . . .	1
	File Handling . . . . .	2
	Specifying Lines and Ranges . . . . .	4
	Correcting Mistyped Commands . . . . .	5
	Control-C Intercept . . . . .	6
	Transliteration . . . . .	7
	The Checkpoint System . . . . .	7
II	BASIC COMMANDS . . . . .	8
	Insert . . . . .	8
	Print . . . . .	9
	Delete . . . . .	10
	List . . . . .	11
	End . . . . .	11
III	INTERMEDIATE COMMANDS . . . . .	14
	save World . . . . .	14
	Go . . . . .	14
	Mark . . . . .	14
	Kill page . . . . .	14
	Number . . . . .	15
	Alter . . . . .	16
IV	OTHER USEFUL COMMANDS . . . . .	22
	Replace . . . . .	22
	next line . . . . .	22
	previous line . . . . .	22
	Copy . . . . .	22
	Transfer . . . . .	24
	eXtend . . . . .	24
	set current line . . . . .	24
	invert . . . . .	24
	Help . . . . .	24
	indirect commands . . . . .	25
	set . . . . .	25
	give information . . . . .	25
	parameters . . . . .	25
V	ADVANCED COMMANDS . . . . .	34
	Join . . . . .	34
	JUstify . . . . .	34
	Justify Left . . . . .	35
	Justify Right . . . . .	35
	Justify Center . . . . .	35
	Justify Words . . . . .	35
	Find . . . . .	36
	Substitute . . . . .	38
VI	LINE NUMBERS REVISITED . . . . .	42

VII	CHECKPOINTING . . . . .	43
A	CONVERSION & STANDARD TEXT FORMAT . . . . .	45
B	CHARACTER CODES . . . . .	46
C	SUMMARY OF COMMANDS . . . . .	47
D	SUMMARY OF PARAMETERS . . . . .	49
E	TERMINALS AND LINE PRINTERS . . . . .	51
F	FILE ENCRYPTION . . . . .	53

## I. INTRODUCTION

SOS provides the ability to insert, delete, modify, and print lines of text. While most commands are line-number oriented, string search and substitution commands are available. Commands are discussed below roughly in order of increasing complexity. It is suggested that you begin by reading sections I and II, then do some text editing. Successive sections describe more elegant functions and may be consumed one at a time.

A command to the editor consists of one or two characters followed by a list of arguments. The input format is free field, i.e., spaces are ignored except that they delimit numbers and identifiers. Tabs are treated as multiple spaces.

## OPERATION

To edit an existing text file named FOO, say "EDIT FOO" to the monitor. This will start SOS and look up FOO. When SOS is ready to accept a command, it will type \*. If FOO isn't found, SOS will give an error message "File not found". To create a new text file named FOO, say "CREATE FOO". SOS will then respond with 00100<tab>, representing the number of the first line, and expect you to type in some text. It will continue supplying line numbers and expecting you to type in more text. To get out of this mode, type an altmode. This will cause SOS to type \* and accept a command. Note that "CREATE <file>" will always create a new file. If a file of the same name exists, SOS will warn you of that fact when you exit from SOS (via the "E" command) and give you the option of either deleting the old version of the file or choosing a different name for the file you have just created.

If you wish to read the file FOO but not modify it, you may type "EDIT FOO/R", which puts you in read-only mode. In this case, any attempt to modify the file will give the message %Illegal when read-only. This mode is recommended if there is no need to alter the file, as it is more efficient than the regular mode.

The command "EDIT", i.e. with no file name given, will cause SOS to edit the file made by the last CREATE or EDIT command that did mention a file name. Similarly, the command "CREATE" will create a new file with the same file specifications as those used in the last CREATE or EDIT command that specified a file. If no previous command was typed, SOS will respond with "File:" and expect a file name.

For those installations where the EDIT and CREATE commands do not invoke SOS, there are two other ways to start SOS. In both methods, no distinction is made in the command between editing an existing file and creating a new one. If the file specified already exists, it is edited (like EDIT). If not, it is created (like CREATE). Some installations may have the monitor command "SOS". If so, "SOS FOO"

will either edit or create FOO (as just described). In any case, one can always say "R SOS". In this case, SOS will prompt you for a file name, typing "File:". At this point you should type the name of the file to be edited or created. For installations having the SOS command, it is also possible to get the effect of the CREATE and EDIT commands by using the characters "<" and ">". "SOS > FOO" is equivalent to "CREATE FOO", and "SOS < FOO" to "EDIT FOO".

The "EDIT <filename>" and "CREATE <filename>" are sufficient for most uses. Actually, considerably more complex commands are possible. An example of the full form is

```
CREATE GORP,F4(SECRET)<155>=DSKD:FOO[5,731,DOC]/NONUM/NOBAK
```

The name on the left side of the = is the name that will be given the output file, i.e. the edited version which will exist after you exit. The name on the right side of the = is the name of the input file, i.e. the file you are editing. (If only one file name is given, it is used for both the input and output file.) (a.) Device names and PPN's have the usual significance. If omitted, DSK: is assumed for the device, and the job's default PPN (your own PPN if you haven't done anything to change it) is used for the PPN. If the same file is used for input and output (and no explicit output device is given), the output will be put on the same disk structure where the input is found. (b.) The output file specification may include a file protection in angle brackets. If omitted, SOS gives the output file the same protection as the input, or for new files, the system default protection. (c.) Either specification may include an encryption key, enclosed in parentheses. See Appendix F on file encryption for a description of the significance. If omitted, a null encryption key (i.e. no encryption) is assumed. (d.) A list of switches preceeded by / may follow the file name. For a list of switches together with their meanings, see the end of Chapter IV.

Note that when two files are specified (e.g. "CREATE X=Y"), the CREATE command should always be used (or SOS if EDIT and CREATE are not being used). "EDIT X=Y" will not have the desired effect. (In order to make "EDIT" always work correctly, "EDIT X=Y" is exactly equivalent to "EDIT X", i.e. Y is ignored.)

When SOS is processing the file name, several error messages are possible. "? insufficient privileges" indicates that the system will not let you create the output file that you have specified. You are probably trying to change someone else's file. If so, try "EDIT <file>/R" if you just need to read it, or "CREATE =<file>" to put the output on your disk area with the same name. "? Monitor path error" indicates that the monitor has lost track of where the file came from. Try to specify its PPN explicitly.

#### FILE HANDLING

SOS works by recopying the text of your file with corrections and additions onto a file whose name it invents. The name of this file is of the form nnnSOS.TMP, where nnn is your current job number. When the edit is finished, SOS renames this temporary file to the

name specified by the user. Usually this is the same as the name of the original file. (It can be changed by specifying an output file name as described above, or as described under the E command.) In such a case the original version of your file will no longer be available under the same name. To provide some protection against errors, the original version of the file is renamed to a file name that differs from the original only in that the extension begins with a Q. Thus F00.F4 become F00.Q4. Of course any old backup file is deleted. If the extension of your original file already begins with a Q, the backup file is given an extension that begins with a Z. (This saving of the original file under a different name can be suppressed. See the description of the E command.)

If the user attempts to reference a line which occurs earlier in the file than the one last referenced, SOS may have to finish the current copy and start copying over again. This process is automatic but does take time, so there may be a delay in executing certain commands. SOS keeps a portion of the file in core during the editing process so that a certain amount of backup can be done without recopying. The =LOCATION command may be used to find the first line in the current core buffer. Notice that the form of editing used means that if the user exits to the monitor without closing the output file (by using ^C) or if the system crashes, his original text file will be unchanged, so any editing will have been lost. See Chapter VII for how to recover from such painful situations.

There are certain error messages which may occur at almost any time during an edit and which cause the editor to call EXIT. Note that this will cause the edit to be lost, but the original file will not be harmed. These errors are as follows:

?Disk not available	Something is very wrong, the disk is not available.
?Device output error	Output error. No recovery will be attempted.
?Device input error	Input error. No recovery will be attempted.
?Internal confusion	SOS has just discovered a bug in itself.
?Illegal UUO	You have been bitten by a bug.
?Bad BASIC file format	You are reading a file in /BASIC mode, and it turns out to have an incorrect format. The problem is probably a form feed or page mark, which are illegal in BASIC files. SOS cannot continue.

The following monitor error may occur at any time. It indicates that the disk on which your file is being written has no free space. Jump up and down until someone frees some space, then type "CONTINUE" to retry the operation.

?Quota or storage exhausted on <diskname>

In addition to the above, there are two messages which are primarily warning messages. The first of these is %Input file line too long. This means that some line in your text is too long (more than 500 characters). The line will be shortened to 500 characters and printed. The second message is %Input file out of order, which indicates that some line of your input has a number lower than the line before it. The line which is out of order will be printed. In both of the above cases, the page number on which the error occurs will also be printed. This condition may be corrected by renumbering the page containing the error.

Note that the two messages just described are printed by the disk read routines. This can cause some confusion with commands that can generate the error %Out of order, e.g. the Number command. A perfectly correct renumber operation may still result in %Input file out of order, if read routines see order problems with the original version of the file. A real order error in the command can be distinguished because the error message, %Out of order, does not mention "Input file". A similar distinction can be made between %Input file line too long, which indicates a problem with the original version of the file, and %Line too long, which is the result of an erroneous command.

The error %Illegal line format may occur at any time. This probably means that you have a line containing a <return> not followed by a <line feed> or a <line feed> not preceded by a <return>. The best cure is to replace the line completely (R command).

#### SPECIFYING LINES AND RANGES

The text file is organized in terms of pages which are subdivided into lines. The pages are numbered sequentially starting at 1 and continuing to the end of the file. The division into pages is determined by the user rather than by the editor. Page marks (which indicate the start of a page) may be inserted and deleted by the user. Page numbers are "floating"; that is if page mark 2 were deleted, the page which was formerly numbered 3 would now be page 2.

In contrast to this, each line on a given page has a line number which is "sticky". That is, a given line will retain the same number regardless of insertions or deletions. The lines are usually numbered by 100 or some other increment larger than 1 to allow room to insert new lines. Different pages may have lines of the same number.



Most SOS commands refer to either a single line or a range of successive lines. A single line is specified by giving both the line number and the page number in the form <line number>/<page number> as 100/3 for line 100 on page 3. A range is specified by giving the first and last lines of the range separated by a colon as 100/3:4702/6 for line 100 on page 3 through line 4702 on page 6. Alternatively, a range may be specified by a starting line and a number of lines. For example, 200/4:10 refers to the 10 lines beginning on line 200 of page 4. In either case, this construction is referred to as a range specifier.

Instead of a number, the symbol "." may be used. "." means either the current page or the current line depending on whether it appears after or before the "/". Thus 100/. means line 100 on the current 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. 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 as a line number to represent the first line on a page, and as a page number to represent page 1. Similarly, \* represents the last line on a page or the last page in the file.

The usual monitor editing characters can normally be used with SOS. Thus <rubout> or ^H will delete the last character typed and ^U will delete the entire line currently being typed in (assuming you haven't already typed the <return> at the end of the line). However, in a few cases this is not sufficient. For example, it is impossible to <rubout> or backspace over an <altmode>. Thus SOS has a special sequence to abort a command. Typing two <bell> characters (^G) in succession will abort any partially typed command. SOS will prompt again with a \*. This sequence should work any time SOS is waiting for input, except for a few cases where it is waiting for single character responses to questions.

### CONTROL-C INTERCEPT

SOS "intercepts" control-C. This means that typing ^C will not get you out of SOS immediately, as it would with most programs. Instead it gets you into a special mode that allows you to choose between several courses of action. You may type

- C to cause SOS to go back to whatever it was doing, as if you had not typed ^C.
- E to end the edit. SOS exits in exactly the same way that it would if you typed the E command after it had given you a \*.
- Q to quit. This is like an EQ command. That is, SOS will immediately exit, leaving all files in the state they were in before you began SOS (unless you have given a W command). This means that anything you typed during the current editing will be lost. Do this if you realize you made a horrendous goof and want to forget it all. SOS will take time to delete various random files it creates that contain information needed by it internally.
- M to get to the monitor immediately. This is similar to Q in its effect, but SOS does not delete the files mentioned above. If you exit this way, you may get odd messages the next time you run SOS. In general it should be used only if you need to get to the monitor to type TTY LC, or use the SEND command to reply to a message, but intend to CONTINUE your SOS job. (But be warned that most monitor commands wipe out your SOS job and so do not allow it to continue.)
- R to get back into SOS command mode (i.e. get a \*) as quickly as possible. Used if you typed a bad substitute command, and want to stop its execution. Note that in such cases, ".", the current line, may no longer be in the buffer, and a "recycle" may occur if you try to access it. You can use "=LOCATION" to find the smallest line that can be accessed without "recycling".

Note that certain functions are particularly sensitive, and cannot be interrupted. If a ^C is done during one of these functions, servicing it will be delayed until the operation is finished. The commands during which ^C's will be delayed are save World (W), copy (C), and transfer (T). ^C's are entirely ignored once the end command (E) has been given.

#### TRANSLITERATION

If the correct switches are set, it is possible to type in all 128 possible ASCII characters from any terminal. This includes lower case and all control characters. To accomplish this, the single quote is used as an "escape character" to generate otherwise untypable characters. See Appendix E for a further description of this feature. (Note that this feature is disabled by default, so the single quote does not have any unusual properties unless you have specifically enabled its special function.)

#### THE CHECKPOINT SYSTEM

SOS keeps a record of all commands issued during a session, using a "checkpoint" file. This file is written in such a way that a fairly recent version will survive a crash. When you log on after the crash it is possible to get to where you were in the edit immediately before the crash occurred. For details, see Chapter VII.

## II. BASIC COMMANDS

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<line>
I<line>,<increment>
I<line>;<increment>
I/<page>
```

## Insert

```
I(<line>)(,<increment> | ;<increment>)
or
I/<page>
```

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, unless another value is specified by means of the /INCR or /STEP switches.

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>. SOS will return to command mode. If there is text on the line before the <altmode>, it will be inserted. If the <altmode> is the only thing on the line, that line will not be inserted. When an <altmode> is given, the number of that line and the current increment are remembered. A later I command with no arguments will then continue that insert. (If an insert is terminated automatically, rather than by typing an <altmode>, it is not remembered in this way.)

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 (^U and <rubout> or <back space>).

## Example:

```
*I100
00100 NOW IS THE
```

```

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

```

If the line at which inserting is to start already exists, an insert will be done on line  $\langle \text{number} \rangle + \langle \text{current increment} \rangle$  unless there is a line with a number between  $\langle \text{number} \rangle$  and  $\langle \text{number} \rangle + \langle \text{current increment} \rangle$  in which case the line number will be halfway between  $\langle \text{number} \rangle$  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 on the specified page. If an attempt is made to insert a line containing more than 500 characters (not counting the  $\langle \text{return} \rangle$  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.

If a semicolon is used in place of the comma to specify the increment, the specified increment is used only for this command, and is not made the current increment. As a further convenience to the user, SOS can also calculate the correct increment to use. If you know you want to insert 5 lines beginning at 400, type I400;15. SOS will calculate an increment small enough to fit at least 5 lines in between 400 and the next line (if any). This increment will be chosen from a list of "round" numbers, so in some cases it will be a bit smaller than is actually required, and the insertion will have to be terminated by typing an altmode.

The command I/<page> puts a page mark at the end of the specified page. This leaves you with a new, blank page whose number is one greater than the one specified. The insert begins with line 100 on that new page. Thus I/5 creates a new page 6 and begins the insertion at 100/6. (Actually the first line generated is controlled by the switch /START. However its default value is 100.)

Print

P(<range>)(,S)(,W)(,E)(,F)(,N)

The print command accepts a range specifier as its argument. The lines specified will be printed on the terminal. 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>".

Various options may be specified by following the P command with letters separated by commas. (The order of the letters does not matter.) These options are designed to make it easier to get "clean copies" of final drafts. The available options are

- S (surpress) - Line numbers and page numbers will not be printed. All characters will be printed as is. (I.e. no " or ' will be used to make them visible.) Finally, the \* for the next command will be suppressed.
- W (wait) - SOS will wait at the end of each page, so the user can change paper, etc. To get the next page, type a carriage return. Typing G will also cause SOS to go on, but it will not pause at the end of future pages. Typing a Q causes it to quit printing. These characters are not echoed, so they will not mess up your copy.
- E (eject) - SOS will supply line feeds to get to the bottom of the paper at the end of each page. Even if W is not specified, SOS will wait before the first page. Note that the length of the paper is a parameter that can be set by /PAGESIZE=. The default is /PAGESIZE=66. The number of lines of text (including header line, if any) that will be put on a page is controlled by /LENGTH=. The default is /LENGTH=55. To use SOS to print a file that has been divided into pages (e.g. RUNOFF output), be sure to set LENGTH large enough that SOS doesn't supply extra pages.
- F (final) - This is equivalent to both W and E.
- N (numbers) - This is useful only when E is also set. It causes line numbers to be put near the bottom of each page. The numbers are in the format -<logical page>[,<subpage>]-. Logical page refers to the SOS page number. Subpage is used if more than one physical page is needed for an SOS page.

### 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).

If the specified range crosses a page mark, the page mark will be deleted. As a result it is possible that the last page will have 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. (See the N command in Chapter III.)

If an entire page or more is to be deleted, SOS will type "Massive delete" and ask the user to confirm that he really means it, unless he has set /EXPERT. After a delete (or replace) command, SOS will print a message indicating exactly which lines, and how many, were deleted. Again, this message will not be printed if /EXPERT is set.

### List

L(<range>)(,S)

List is like print in format and error messages, but the output goes to the line printer instead of the terminal. 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.

As explained in appendix E, the switches /L64 and /L96 control character mapping, and are equivalent to /M33 and /M37 for the P command. /L96 indicates a line printer with 96 characters, i.e. upper and lower case letters. /L64 indicates a line printer with 64 characters, i.e. upper case only. The initial setting of this switch is /L64, though this can be changed by the installation.

Output can be directed to a device other than LPT: by using the switch /LDEV:<device>. The installation may also set a device other than LPT: as being the default device. If the device uses file names, the name of the output file is used, with the extension changed to .LPT. If the device is a disk, the output generated by the command is appended to the end of any previous file.

End

E(:<file name>)

This command is used to terminate the edit. The newly edited version of the file will be given the output name specified when SOS was started, unless a /NAME command has intervened. What else happens at

this point depends somewhat upon whether SOS believes that the output file has been created by editing an old file of the same name. If SOS was started by "EDIT X" (and no /NAME has been done), this is obviously the case. On the other hand, if SOS was started by "CREATE X", there is assumed to be no old version of X. In other cases, such as "CREATE X=Y", SOS simply compares the input and output files to see whether they match. The output is assumed to have been made by editing an old file of the same name if the input and output files have the same name, and are on the same structure and path. A similar comparison of output and input file names is made if the output name has been changed with the /NAME command.

If SOS believes that the output file has come from an input of the same name, it will look for the old version and rename it to a "backup" name. This allows for recovery if it turns out that something went wrong with the edit. Any old backup file will be deleted. The name used for backup is the same as the original name, except that the first character of the extension is made a Q. If it already was a Q, it is made a Z.

In other cases, SOS assumes no backup is necessary. In the case of commands like "CREATE X=Y", the output file will be given the output name, here X, and the input file will remain undisturbed. Should a file by this name already exist, SOS will type the message "File already exists! Delete? (Y or N)" and wait on the same line for a response. If "Y" is typed, SOS will delete the old file and substitute the new one. If "N" is typed, SOS will respond with "FILE=" and wait for a new name to be typed; this will be assigned to the edited file. If "Y" and if the old file cannot be deleted, SOS will respond with "FILE WRITE PROTECTED, TRY ANOTHER NAME", and accept a new filename as above. It is possible to type "C" when SOS asks for a new name, but this is not recommended, as the edited version would be left with a temporary name such as 017SOS.TEM.

SOS will return to the monitor when it has finished.

The E command may optionally be followed by one the the following command modifiers:

- B Effective only when SOS believes the output file comes from an old version with the same name as the new one. The B modifier suppresses the creation of the backup file, so the old version is simply deleted. It also causes any old backup file to be deleted.
- D Delete the original input file (and don't create an output file either). SOS requires you to verify that you really mean it, in case of typos.
- Q Quit now; leaves your files as they were before you entered SOS (or as of the last W command, if you have done one). It also deletes any nnnCHK.TMP, nnnSOS.TMP or nnnsos.TEM files that SOS may have created during this edit



as working files.

- S Remove line numbers in the resulting output file, and turn page marks into form feeds.
- C Do not delete the checkpoint file nnnCHK.TMP, as is normally done when SOS exits. (It is mainly used to protect against a crash occurring while you are editing.)

Certain combinations of these letters can be used, as for example "ESB" or "EBS" to suppress creation of a backup file and remove line numbers. In readonly mode (e.g. if SOS was entered by "EDIT X/R"), only E and EQ are legal. They both cause an immediate exit to the monitor. No output files or temporary working files are created in a readonly edit.

## III. INTERMEDIATE COMMANDS

## save World

W(:&lt;file name&gt;)

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, or in case you do not like your subsequent edits. File names given to W commands are "sticky", and need not be given twice.

Note that this command is not necessary to protect against crashes unless the checkpointing system is disabled by large values of SAVE and ISAVE, or by the /NOCHK switch. When a W is executed, the checkpoint file is automatically deleted and begun again. Thus checkpoint restorations will always begin with the most recent version of the file.

## Go

G(:&lt;file name&gt;)

This command is the same as the E command except that it also causes the COMPILER program to re-execute the last system command of type COMPILER, LOAD, EXECUTE, or DEBUG. If it is desired to set up SOS to automatically execute some other program when G is typed, this can be done by using the switch /RUN:<program name>.

## Mark

M&lt;line&gt;

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 after the line of next lower 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" (or use "I/,").

## Kill page

K/<page>[:/page]

This command deletes the page mark before each of the specified pages. Thus "K/5" deletes the page mark between pages 4 and 5, with the result of combining the original text of both pages to make a new page 4.

Deleting a page mark may result in the creation of a page with line numbers that are out of sequence. If so, SOS will print the message "%OUT OF ORDER". In such a case, it is necessary to renumber the page immediately. If a range is specified, it is possible for the deletion of each page mark to create an order error. In such a case only one error message will be issued. (However if an order error existed previously the buffer handling subroutines may issue messages "%Input file out of order".)

There is no page mark before the first page, so "K/1" cannot be done. If the first page specified does not exist, the error message "%NO SUCH PAGE" will be issued. However the second page in a range specification need not exist. Thus "K/2:/99999" is perfectly legal, assuming there is a page 2. It would delete all the page marks in the file.

Note that only page numbers should be specified as arguments to this command. Thus "K/2:/5" deletes the page marks before pages 2 to 5. "K100/2:/5" would be illegal. It would also be nonsense, since the K command affects only page marks, never regular text lines.

#### 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. This command is useful when the "%OUT OF ORDER" message indicates that line numbers are out of sequence, or more generally, when so many insertions have been made that there is no more space between the lines.

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 preceding 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 lines 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.

If the first and third parameters are left out, they default to the values of /STEP and /START, which are usually 100. If the first parameter is specified but the third is not, its value is also used for the third.

The command "NP" will not reset the line numbers when crossing page boundaries.

The command "NA" will add the increment to the existing line numbers.

### 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 terminal, so that the line shown on the terminal 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. A negative number can also be used, in which case the pointer is moved backwards in the line. Negative S and K commands leave the pointer positioned to the left of the sought character. "-x" (where x is a command) is short for "-1x".

next==<space>

This command causes the character pointed to to be printed on the terminal 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 <^H>

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 if a rubout was used. If ^H is used it is assumed the user has a terminal which can backspace (^H is the backspace character), and the cursor is simply backspaced over the character.

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 is followed by another character. 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 terminal will be ignored. <return>, <line feed>, and <altmode> have the same effect as for the "I" command described below. Note that if a count is specified, this command replaced the specified number of letters with the same number of newly typed letters. If an <altmode> is typed this operation is stopped even if the count is not exhausted. In the latter case, the same number of characters will be replaced as were typed. That is, C never changes the total number of characters in the line. If a <line feed> is typed, a new line is begun, but no characters are deleted, and the line feed is not counted for purposes of the repeat count.

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. Note that if the user has set /EXPERT mode (See the description

of the set command near the end of Chapter IV.), the deleted characters will not be printed out with \\'s. In this case, there will be no output for this command. After the command, the pointer will be pointing to the character to the right of the one deleted.

### Insert=-I

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

<rubout> 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.

^H has the same effect as <rubout>, except that the cursor is moved back rather than having characters printed between \\'s.

If enough characters are inserted to make the total length of the line more than 500, 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> is typed, the insert will be terminated as if <rubout> was typed. Additionally, intraline edit will terminate as in the <return> command.

If a <line feed> is typed, 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. Note that the "R", "X", and "C" commands, whose descriptions refer to this section, allow the use of a repeat count before the command. Thus there is no way to specify a "provisional line number" for them.

**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 is started, otherwise SOS returns to command mode.

**End**--E

This has the same effect as the <return> key, except that the rest of the line is not printed on the terminal.

**Quit**--Q

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 started. It is equivalent to typing "Q" and then giving the alter command again.

**Skip**--S

This command accepts one subsequent character from the terminal without echoing it. (Note that ' escape characters are counted as single characters.) 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". The editor 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. Typing <altmode> as the search character will abort the command. Typing <return> as the search character will abort the command, and will have the additional effect of finishing editing of the line, just as if it had been typed as a command. <line feed> when used as a search character will match the end of the line (or the beginning if a negative repeat count is used).

**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 characters deleted will be printed, surrounded by \\'s, when not in /EXPERT mode. Note that K<line feed> is especially useful for deleting the rest of the line.

#### 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

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.

#### jump=-<tab>

A tab (^I) will cause SOS to do repeated nexts until the end of the line is reached. If the tab is preceeded by a minus sign, then repeated backs are done until the beginning of the line is reached.

#### 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 500 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.

#### invert=-V

This command causes all letters from the one you are on up to the first non-letter to get their cases inverted. If you are not on a letter, then this is a no-op. A numeric argument of <n> will cause the V command to terminate when a non-alphabetic



character is encountered or after <n> characters have been inverted, whichever comes first. The command IV is particularly useful to change the case of the next character.

#### Word==W

This command does repeated nexts to skip to the end of the next word to the right of the pointer, and any blanks or tabs following it. A word is defined as a series of alphanumeric characters surrounded by tabs and spaces. Note however that special characters are considered single words in themselves. Negative repeat counts are not allowed for this command.

#### rePlace word==X

This command deletes those characters which a W command would skip over. It then does an I command. Like the D command, the characters deleted are printed between \\'s unless /EXPERT has been set. Negative repeat counts are not allowed for this command.

Any other commands to intra-line edit mode are ignored, and will cause a bell to echo.

It may not be clear from the preceding description, but the following three special characters are intended to have reasonably consistent interpretations in whatever context they are typed:

<altmode> stops the current command. Of course this only works for commands that take input from the terminal (C, I, K, R, S, and X).

<return> functions like altmode, but additionally finishes up the altering of the line, as if <return> had been typed as a command.

<line feed> represents end of line. If inserted it causes a new line to be begun. For the search commands (K and S) it represents the end of the line (or the beginning if a negative repeat count is used).

## IV. OTHER USEFUL COMMANDS

## Replace

R<range>(,<increment> | ;<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, is 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. As with the I command, a semicolon before the increment specifies a temporary increment. ;!<number of lines> is also acceptable to this command.

## 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 | /R)),<source>(,<inc1>)(,<inc2>)

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.

Note however that this command is an exception to the rule that a specification of the form "/page" is equivalent to all the lines on that page. "/page" actually specifies all the lines on the page, together with the page mark occurring at the beginning of the page. Thus if the command "C100/2,/5" were executed, line 100/2 would be followed by a page mark, and then the text of the old page 5. The rest of the old page 2, if any, would of course remain after the inserted material (with no extra page mark). For the purposes of this command, there is assumed to be a "phantom page mark" before page 1. Note that the page mark is included only when the page number is given without line numbers. Thus "C100/2,-/5:\*/5" would not copy the page mark.

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 large blocks of text may result in the error message %Insufficient core available. 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 either of the errors %File not found or %Disk not available. The source file name may include a disk structure.

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 "C\*". 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>...). Using /R instead of /S causes the file to be read in /BASIC mode.

## Transfer

T&lt;dest&gt;,&lt;source&gt;(,&lt;inc1&gt;)(,&lt;inc2&gt;)

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) As with the Copy command, "/page" in the source specification means all of the lines of the page and the page mark before it. The page mark is both copied to the destination and deleted from the source.

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(&lt;range&gt;)(,S)

This command is like the Alter command except that on each line it automatically prints the line (unless ,S is given in the command), and leaves the pointer at the right end of the line in insert mode. It is useful for adding comments to lines.

## set current line

,&lt;line&gt;

The , command makes the line indicated the current line; i.e. it "moves" , to the <line>.

## inVert

Vx&lt;range&gt;

This command is used for controlling the case of letters in SOS text. VV<range> inverts the case of all letters in the range, i.e., "a" becomes "A" and "A" becomes "a". VU<range> makes all the letters upper case. VL<range> makes all the letters lower case. All three Vx commands type out the range as they are inVerting.

## Help

## H(S)

The "H" command causes SOS to type out the SOS help file, which contains a short description of all the commands and switches.

There is a variant, "HS", which causes SOS to list all only the switches present in the current version. This differs from the full H command in that the output is far more brief, and in the fact that the switches are taken from an internal table, and so do not depend upon the help file, which may be out of date, or even missing entirely.

## indirect commands

## @&lt;filename&gt;

If a particular sequence of commands is used frequently, it is possible to insert these commands in a file. Then SOS can be instructed to read commands from the file, rather than from the terminal. The command @<filename> will do this, provided the list of commands is found in the named file. Only the following will be printed on the user's terminal: (1) error messages, (2) printout from the P command, (3) items found by the F command, (4) substitutions made by the S command, and (5) lines altered by the A command. Note that every character to be read by SOS must be in the file. Thus the decide option of the S command is not very useful, since you would have to anticipate each response character and put it in the file.

When SOS has executed the last command in the "indirect command" file, the user will be returned to command mode with the message "Indirect EOF". (This message will be printed even if ^O has been typed to stop output on the terminal.)

Note that when making up an indirect file, it is sometimes necessary to enter such characters as <altmode> or <return> without a following <linefeed>. <altmode> can be typed as ^= if you have set /C128. (See Appendix E.)

## set

## /&lt;parameter&gt;(&lt;number&gt;)

The set command sets (or changes) parameters in SOS. For example, after SOS has been used for a while, the user may want shorter forms of error messages. By typing "/EXPERT" all error messages will be abbreviated to five characters or less. Also, certain information and warning messages will be omitted entirely. This is reversed by typing "/NOVICE". Note that SOS begins in /NOVICE mode. Some parameters require numerical values. Thus the left margin (for use in the justify commands) may be changed by "/LMAR:5". Note that parameters may be abbreviated, as long as the abbreviation is unique.

For example, /NOVICE can be abbreviated as /NOV, but not as /NO (since several other parameters begin with "NO").

Most Parameters can also be set by switches in the SOS monitor command. The format for these switches is /<parameter>(<value>). Thus one might use "SOS FOO/EXPERT/LMAR:5". This has exactly the same effect as if one typed "SOS FOO", and gave "/EXPERT" and "/LMAR:5" as soon as the \* prompt is given.

Furthermore, parameters can be set by default by an appropriate entry in SWITCH.INI. This facility is designed for users who usually want the same switches set. When SOS is started, the first thing it does (before even reading the file name) is to look on the user's default disk area (This is his own PPN unless he has reset it,) for a file named SWITCH.INI. If it is found, it is scanned for lines beginning with "SOS". The rest of the line is assumed to be a list of switches, which are immediately acted on. Thus the author has in his SWITCH.INI a line reading

```
SOS/EXPERT/C128/NOBAK/M37
```

should the user have certain sets of options he often uses together, it is possible to give them special names. Each set of options is defined by a separate line in SWITCH.INI, of the form

```
SOS:ODD/LENGTH:3/PLINES:5
```

Such a line is not read by the normal SOS initialization procedure. However, if the switch /OPTION:ODD is given after the file name, all of the switches on the line beginning SOS:ODD are set. (/OPTION:ODD could also be typed as an SOS command during the edit session and would have the effect of processing all the switches on the line at that point.

HELP is a special case. It is not really a parameter at all. However, to be consistent with a system-wide convention, /HELP:SWITCH prints a list of all the parameters. It is exactly equivalent to the HS command. SWITCH may be abbreviated, but any other value for the HELP "parameter" is illegal.

Note that there are a few parameters that are legal only when given with the file name when SOS is started, or in SWITCH.INI. These parameters have a \* after their name in Appendix D, and are noted in their descriptions below. If one of these parameters is set during an edit, the error "%Can't set now" is given. The parameter NAME is just the opposite. It can be set only during an edit, and is illegal with the initial file name or in SWITCH.INI.

#### give information

```
=<parameter>
```

Any time the user is in command mode, he may request the value of a parameter by using the give information command. Thus =LMAR would cause the value of the parameter LMAR to be typed out. Note that some parameters that are set individually are displayed as a group. For example, the "=CASE" command displays information that is set by

C64, C128, LOWER, M33, M37, etc. Parameters may be abbreviated with this command as with the set command.

#### PARAMETERS

The following sections contain descriptions of all currently defined SOS parameters. For an up to date list of the parameters defined in the current version, type the command "HS" in SOS. The list below is organized into sections describing related parameters. The effects of both / and = are given with each parameter if they are legal with that parameter. If either / or = is not mentioned, it is illegal with that parameter. For a list in alphabetical order, but with essentially no description, see Appendix D.

BAK  
NOBAK  
OLD

These parameters control the functioning of the "backup" mechanism. This mechanism is described in gory detail under the end ("E") command. The description of the end command assumed that /BAK was in effect. This is the default mode. If /NOBAK is in effect, the backup mechanism is turned off. Any subsequent "E" command will implicitly be "EB". This means that the old version of the file will simply be deleted, rather than renamed to xxxxxx.Qxx (or .Zxx). Also, any old backup file will be deleted.

In order to understand the effect of /OLD, one must realize what it is designed to be used for. Suppose that one intends to edit a file several times, and wishes to preserve the original version of the file. The normal backup mechanism is no help, since only the penultimate version is preserved by it. /OLD is designed to solve this problem. The first time the file is edited, the original version will be renamed to xxxxxx.Zxx. In successive edits, /OLD will act exactly as /BAK, causing the penultimate version to be preserved as xxxxxx.Qxx. The way this is implemented is as follows: If /OLD is in force, and circumstances are such that an old version is to be backed up, SOS looks for xxxxxx.Zxx. If it does not exist, the old version is renamed to xxxxxx.Zxx. If such a file already exists, the old version is renamed to xxxxxx.Qxx, and any old xxxxxx.Qxx is deleted. (This is exactly like the behavior under /BAK.) Notice that SOS will never delete the xxxxxx.Zxx file. This mechanism is of course useless for files whose extensions begin with Q or Z. In both of these cases /OLD is exactly equivalent to /BAK. When the original file is xxxxxx.Qxx, xxxxxx.Zxx is used as a regular backup file.

To examine the setting of these parameters, use =BAK.

## BASIC

/BASIC may only be specified with the file name when SOS is started (or in SWITCH.INI). If it is specified (e.g. "EDIT X/BAS") the file is assumed to have been produced by DEC10 BASIC. SOS will convert the BASIC line numbers into real SOS line numbers, and will edit in a special mode necessitated by the fact that BASIC files do not necessarily have tabs after each line number. If one ends an edit of a BASIC file with the ES command, the line numbers will be turned back into strings of ASCII digits and the tabs supplied by SOS at the beginning of each line will be turned into spaces. Note that page marks and form feeds are illegal in a BASIC file. If they are discovered in a file being read with /BASIC set, the edit is aborted.

## LOCATION

=, prints the line number that is represented by ",", i.e. the current line. =LOCATION prints the line number of the first line in the in-core buffer. (See the section "File handling" in the Introduction for an explanation of what this means.) Any access to a line before the value of LOCATION will cause a "recycle".

## BIG

=BIG prints the number of the last page in the file, i.e. the number that will be used for "/\*". If SOS has not yet looked at the whole file, it will have to go to the end of the file to find this out.

C64  
C128  
CASE  
DPY  
LOWER  
L64  
L96  
M33  
M37  
UPPER

These parameters are all involved in controlling the way characters are input and output by SOS. Except for CASE, they can be set only. E.g. /C128 sets C128 mode, but =C128 is illegal. The command =CASE tells which of these are set, except that /C128 and /M33 will not be mentioned if they are in effect. See Appendix E for a complete explanation of these parameters.

DECIDE  
NODECIDE



`/DECIDE` sets the "auto decide" mode. Any subsequent `S` command will implicitly have the `,D` (decide) switch assumed. `/NODECIDE` cancels this mode. Use `=DECIDE` to see which of these is set. `/NODECIDE` is default.

DISK  
DSK

`=DISK` and `=DSK` are exactly equivalent. Both display the number of free blocks left on the disk structure where the output file is going. They also give how many blocks remain in the user's logged in quota on that structure.

## ERROR

`=ERROR` repeats the full form of the most recent error message. This is useful mainly if you have set `/EXPERT` and don't recognize an abbreviated error message.

EXPERT  
NOVICE

If `/EXPERT` is in effect, SOS abbreviates most of its error messages to a maximum of 5 characters. It also turns off certain printouts that are useful mainly to novices (such as the number of lines deleted by a delete command and some output for the alter command), and avoids having SOS ask for confirmation of the `D` command. These various effects are documented under the commands affected. `/NOVICE` cancels this mode, and is the default. `=EXPERT` will indirectly give you the setting of this switch, as it is illegal. So `%ILC` indicates `/EXPERT` is set, and `%illegal` command the `/NOVICE` is set.

## INCR

This parameter is the increment that will be used by the insert, replace, justify, copy, and transfer commands. Its exact effect is documented with each command it affects. It may be set by `/INCR:<positive integer>` and displayed by `=INCR`. It may also be set by the insert and replace commands, q.v. It initially defaults to the value of `/STEP`, which is usually 100.

ISAVE  
SAVE

These parameters control the frequency with which the checkpoint file is updated. See Chapter VII for a description of its exact effect. Note that even if they are both set to zero, the checkpoint file will be updated when the internal buffer is filled. To disable checkpointing completely `/NOCHK` should be used. These parameters may be set, with `/SAVE:<non-negative integer>` (or `ISAVE`), and displayed with `=SAVE` or `=ISAVE`. Default values are `SAVE=1`, `ISAVE=5`.

## LDEVICE

This parameter allows output from the list command to be directed to a device other than the default device. If the device has filenames, the name of the output file is used, with the extension .LPT. If the device is a disk, the output produced by the L command is appended onto the end of any previous file. The parameter may be set by /LDEVICE:<device name> and examined by =DEVICE. The default in the distributed version is LPT, but may be changed by the installation.

LENGTH  
PAGE SIZE

These parameters are used by the "pretty-print" options of the print command. LENGTH is the number of lines of text that is to be put on one page of output. PAGE SIZE is the physical length of the paper in lines. They may both be set by / and examined by =. They should be positive integers. Default is LENGTH=55, PAGE SIZE=66.

LMAR  
MAXLN  
PMAR  
RMAR

These parameters are used by the justify commands. LMAR is the left margin and RMAR the right margin. PMAR is the column where the first line of each paragraph will be put (the "paragraph margin"). MAXLN is the number of lines to a page. A page mark will be inserted if JU or JW see more than MAXLN lines on a page. They may all be set by / and examined by =. RMAR should be between 2 and 500. LMAR and PMAR should be between 1 and RMAR. MAXLN should be between 1 and 99999. The defaults are LMAR=1, RMAR=69, PMAR=1, MAXLN=99999.

## NAME

=NAME displays the name of the output file that will be generated when the end command (or W command) is given. /NAME:<file spec> allows this to be changed. Note that device names are illegal in the file spec, since only RENAMEs are done at exit. /NAME may not be used with the initial file specification or in SWITCH.INI.

SEPARATOR  
NONSEPARATOR

/SEPARATORS causes %, &, and . to be considered separator characters for the find and substitute commands. That is, %: will match any of these characters. /NONSEPARATORS is the default mode, and causes these characters to be considered normal text characters. If /SEPARATORS has been set, that fact will show up in =CASE.

NONUMBERS  
NUMBERS

This manual assumes that /NUMBERS, the default, is in effect. If /NONUM has been set, no line numbers will be printed at the beginning of lines by print, alter, etc. The numbers will still exist, and can be displayed by =., =LOCATION, etc. Operation of SOS is otherwise unchanged by /NONUM. (This is a difference from DEC SOS.)

## OPTION

/OPTION:<option name> causes SWITCH.INI to be scanned, and lines beginning SOS:<option name> to be processed. See the description under the set command.

## PLINES

PLINES controls how many lines are printed by the print command when given without arguments. It may be set by / and displayed by =. The default is 16.

R  
ROONLY  
READONLY

/R, /ROONLY, and /READONLY all have exactly the same effect. They may be specified only with the filename during initialization, or in SWITCH.INI. They cause the edit to be readonly. In a readonly edit no output file is produced, and any command that would change the file is illegal. The only kinds of end command that are valid in readonly mode are E and EQ.

## RUN

This parameter is the name of the program which will be run if the G command is used. See the documentation of that command for details. It may be set by /RUN:<file spec> and examined by =RUN. If no device is specified, the default is SYS:. The default value is SYS:COMPIL. SYS:COMPIL is run with an offset of 1. If RUN is reset, an offset of 0 is used.

SEQUENCE  
UNSEQUENCE

If /UNSEQUENCE is in effect, any end command is implicitly an ES. See the description of the end command. /SEQUENCE is the default. =SEQUENCE displays which of these is set.

START  
STEP

These parameters control the generation of line numbers, when SOS must do this for itself. START is the first line on a new page. It is used by the number, mark, insert, copy, and transfer commands. STEP is the permanent increment. It is used by the number command. Also, these values control the line numbers which are generated by SOS for unnumbered files. For unnumbered files, line numbering is controlled by the parameter values in effect when the file is first opened. Thus for the main file, only values in the initial command or SWITCH.INI are relevant. For second-file copy operations, the values in effect when the copy command is given are used. The default value of STEP is 100. START defaults to STEP. (Also, INCREMENT also defaults to STEP.)

#### STRING

=STRING will display the current strings for the find and substitute commands, and for the line number syntax described in Chapter VI. The latter consists of three strings, one for each nesting level.

#### VERSION

VERSION is the version number to be used for the output file. This is put in location ,RBVER of the RIB, and is printed by the DIRECT command. (Beware that older versions of PIP remove this information when copying or renaming files. To preserve it extended LOOKUPS, ENTERS, and RENAMEs must be done.) It may be set by / and examined by =. The standard DEC10 version number syntax is used. (Indeed the routines implementing it are from SCAN.) If there is an input file, the default is the version number from the input file. If not, the default is no version number (i.e. zero in ,RBVER).

CHKPT  
NOCHKPT  
NOSAVECHKPT  
SAVECHKPT

These parameters control the checkpointing mechanism. /CHKPT and /NOCHKPT may be given only with the initial file specification, or in SWITCH.INI. The status of these parameters may be displayed by =CHKPT. See Chapter VII for a description of the exact effect of these parameters. The default is /CHKPT, /NOSAVECHKPT.

## V. ADVANCED COMMANDS

## Join

J&lt;line&gt;

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&lt;range&gt;

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  $RMAR(\text{right margin}) - LMAR(\text{left margin}) + 1$  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 to make the length of the line exactly  $RMAR - LMAR + 1$ .  $LMAR - 1$  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. Note: If the increment value is wrong it will cause an %OUT OF ORDER error. 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.

## NOTE

WARNING about the following three commands: If a line exceeds  $RMAR=LMAR+1$ , SOS will print the error message %LINE TOO LONG. The text, when printed, may not be as expected; the margins should be reset, and the command should be reissued.

## Justify Left

JL&lt;range&gt;

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&lt;range&gt;

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

## Justify Words

JW&lt;range&gt;

Similar to JU, but skips the step of adding blanks to bring the right edge of the text to the right margin. The text, therefore, appears as typewritten text without a justified right margin. Lines are reorganized to get them as nearly full as possible without going over the right margin, however.

For a summary of the parameters used in the Justify family of commands, their initial values, and how to change them, see the description at the end of Chapter IV.

### Find

```
F(<string><altmode>(<range>)(,A)(,K)(,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:

```
FFOOBARS/5
```

(where \$ is used here 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 "Foo" will find the same lines. The use of <return>, <line feed>, '?', '%', ')', ':', 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. FFOOBARS) 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:

```
FFOOBARS:/10
```

Giving the command F<return> will cause a previous search to be continued beginning with the line after the current one. This is a continuation of the previous command in the sense that the options ",A", ",K", ",N", ",E", and ",<number>" (see below) will still be in effect if they were before, and the search will stop at the end of the previously specified range. In contrast, the command F\$<return>, although it also uses the previous string, does not remember the options and always searches to the end of the file.

If no occurrence of the string is found in the given range, SOS will print %Search fails, then return to command mode for a new 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).

The following is a list of all the switches that control options for the F command. These switches may be typed in any order on the command line, so long as they follow the range specification, if any.

## 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: FFOOBARS/5,A or FBAZ\$,A (range omitted) 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.

## Kill switch (,K)

If ",K" is included in the specification, every line found will be deleted. The lines will still be printed as usual. Furthermore, unless /EXPERT has been set a message indicating which lines have been deleted will be printed when execution of the command is finished. This option is particularly useful in combination with ",99999". For example, "FFOOS,K,99999" will delete the first 99999 (presumably all) lines containing FOO.

## 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 terminals 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.

## Number of occurrences

It is possible to find more than just the first occurrence of a string. This may be done by including in the command string ",<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:

- ' : 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.
- '% Will cause the character following it to be matched by any character which it would not normally be matched by. Thus the string F'%AB will be matched by FBB, FCB, FDB, FSB, etc., but not by FAB or FaB. '%: will be matched by any letter, number, etc. '%/ will be matched only by the beginning or end of a line. Thus '%/FOO will find only those occurrences of FOO at the start of a line.
- '7 Is used to quote the next character. Thus '7% is used to search for the character '%' and '77 to search for '7. Note that '%7% will match any character but '%.
- ') Is used to mean "any number of" whatever follows it. Thus the string A')BC will be matched by AC, ABC, ABBC, etc. In case of ambiguity, the shortest such string will be found. Thus the string ')AB will find B rather than AB. Strings of the form ')%'7% are perfectly legal. The example will be matched by any number of characters which are not '%s.

Certain strings which can be formed with the above characters are considered illegal and may produce the message %Illegal search string. 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 '%, ') , or '7 when not followed by another character and the construct '%) , . . . 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.

```
S(<ostring><alt><nstring><alt><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, as long as a range is specified. (If no range is given, only the first occurrence is used, as with the F command.) 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:

```
SFOOsBAZs/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:

```
Ss/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 <ostring> (but not in <nstring>).

If the range is omitted, (e.g. SXsYS) then the range used will be from the line after the present one to the end of the file. However, for just this special case, the substitution will be made only in the first line found (unless the "<number>" option is used). If only the second half of the range is specified, the search will be from the line after the present one to the specified line, and substitutions will be made in all lines. S<return> is a continue in the same manner as F<return>. 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 excess <ostring>'s will be deleted (i.e. replaced by null <nstring>'s). Thus, to simultaneously substitute

ALPHA for BETA and DELTA for GAMMA on page 5 through page 7, use the command:

```
SBETA
GAMMASALPHA
DELTAS/5:/7
```

The errors %Too many strings and %String too long will occur under the same circumstances as for F.

The following switches may be specified in any order. However they must follow the range specifier, if any.

#### 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, include the switch ",D" in the command line. 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 users console. The possible responses are

<rubout> The indicated substitutions will not be made; the old version of the line will be retained. SOS will then print the next line and, when found, will wait for another "D" command character.

E SOS will return immediately to command mode without performing the substitution.

G SOS will use the modified line to replace the original line; Decide mode will be halted and further substitutions will be made automatically.

A SOS will enter intraline edit mode with the substitutions already made; the user then can issue one of the following:

\*U Restore the original line

<return> Leave intraline edit mode and continue substitution.

Q Abort the entire process by returning SOS to command mode.

<space> The new line will replace the old line and the substitution will continue.

No print(,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, include ",N" in the command. Note that the decide switch ("D") will cancel this option if both are present.

#### Exact compare (,E)

As with F, ",E" will cause upper and lower case letters to be treated separately in the first string.

#### Number of Occurrences

As with F, "<number>" will cause the substitution to be stopped when the specified number of lines have been changed, or the range is exhausted, whichever happens first.

#### Special Characters

All of the special characters permitted in the string of an F command ('), '/', ':', '%', 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 <string> 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':s'*1'*BAZ'*2'*$<range>
```

If multiple strings are being substituted, then the count of partially specified string starts at 1 for each target string.

" can be used instead of the '\*<number>\*' construction in certain simple cases. The first " represents '\*1\*', the second '\*2\*', etc.

To insert a '\*', a '"', or an '?', precede 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 '\*12\*' when there are only 2 partially specified strings.

## VI. 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:

```
LsFOOs
:sBAZs
,S
```

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

If the string between <altmode>s 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:

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

The <number> (call it k) specifies that the kth 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) option will, upon finding a line, cause the line (or just its number) to be printed. If the next character typed is a <rubout>, another line will be looked for. A "Q" will abort the search completely and return to command mode. 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.) The options can be typed in any order.

As with the F command, a completely null command is a "continue" from the last command. It uses the same string, the same setting of the switches, and continues the count if it is not exhausted. Such a null command consists of the beginning <altmode>, followed immediately by a carriage return, without the <altmode> specifying the end of the string.

Also note that the range used in this construction can be another search-type construction. This sort of thing can be nested to a depth of 3. Anyone who thinks he needs more depth is invited to consult a psychiatrist.

## VII. CHECKPOINTING

A "checkpoint" mechanism exists in SOS to facilitate recovery of files if a crash should occur during editing. After a crash the file being edited is in the state it was when editing began. A new file being created will not exist at all after a crash. Checkpointing does not change this. However, it does keep a record of everything you type on the terminal during the editing. Thus the @ command can be used to duplicate what you had done before the crash automatically. What you typed went into a file called nnnCHK.TMP (nnn is your job number). The file is written in such a way that a close is not needed. Thus it will exist even if a crash occurs.

To recover after a crash, you must first be sure you know the name of the nnnCHK.TMP file created before the crash, as well as the name of the file you were editing. You can find out the former by typing

```
.DIRECT ???CHK.TMP
```

If more than one exists, type them out to see which one you want. You should then type exactly the same SOS command you used before the crash. (If you get a message like "Overwrite existing checkpoint file?", you should reply "n". This is very important!! If you replied "y", the checkpoint file from which you are trying to do the restore would be deleted. See the note below.) Once SOS gives you a \*, just type the command

```
*@nnnCHK.TMP
```

where nnnCHK.TMP is the file name of your checkpoint file. (You don't type the \*, of course.) SOS will now do over again everything you did before the crash. When it is finished, it will type %Indirect EOF. You should probably exit at this point and begin editing again, since any changes you make to the file after restoring it from the checkpoint file are not protected by checkpointing.

If the file was being created by the original edit (i.e. it didn't exist before), SOS will begin by supplying you a line number (probably 00100). To do the restoration, you must get out of input mode by typing <esc>. Then you can give @nnnCHK.TMP. (SOS inserts an extra i100 command into checkpoint file in create mode to allow this sequence to work.)

There is an exception to the above procedure. If a W command had been given before the crash, checkpointing would have been reinitialized. Thus any parameter setting commands done before the last W command would not show up in the checkpoint file. So before typing the @ command, you should type any set commands that had been done before the most recent W before the crash.

Note that it is illegal to use the @ command in a command file. Thus in particular, @ commands appearing in checkpoint files will be ignored. So if you use command files for purposes other than restoration after a crash, be warned that there may be problems.

Also beware of the ^C-REENTER sequence with SOS. This (or the "R" response to the SOS ^C intercept) are useful to stop runaway searches or substitutes. However, there is no way for the checkpoint file to know that you ^C'ed. Thus in a checkpoint restore you will get whatever would have happened had you not ^C'ed. This may result in a different current line number, and any substitute you stopped will go on. The only way to guarantee results is to use the "W" command after your ^C-REENTER. Since this saves a complete copy of your current edit, it also starts checkpointing over from the new copy.

There are some switches that you can use to control checkpointing: the switches /CHKPT (default) and /NOCHKPT turn checkpointing on and off. These switches are valid only in the initial command. The switches /SAVECHKPT and /NOSAVECHKPT (default) control whether the checkpoint file is saved if a normal exit occurs. The effect of /SAVECHKPT can also be gotten by exiting with EC, or some combination of C and the old alternatives B, S, Q, and D. The command =CHKPT will describe the status of checkpointing. SAVE and ISAVE control how often the checkpoint file is written. (It is written when a disk block is full in any case.) They have defaults of 1 and 5 respectively. This means that the checkpoint file will be updated after every SOS command that changes the file (this is controlled by SAVE=1) and after every 5 lines are typed in insert mode (this is controlled by ISAVE=5). In order to facilitate restoration, commands with a decide switch (e.g. Sx\$Xs.::\*d) take the decision character from the current input source, whether indirect file or TTY:.

It is possible to get the message

Overwrite existing checkpoint file? (y or n)

at the beginning of an edit. This means that for one reason or another a checkpoint file already exists and SOS is about to delete it in order to write a new one. This should cause no problem UNLESS YOU ARE DOING A CHECKPOINT RESTORATION. If you are about to restore an edit lost due to a crash, you do not want the old checkpoint file to be overwritten, so you should reply "n". Replying "n" means that SOS will be unable to create a checkpoint file covering the current editing session, but if all you are doing is restoring an old edit from the checkpoint file, that is no problem.

## APPENDIX A

## CONVERSION &amp; STANDARD TEXT FORMAT

A few words about copying files and converting formats.

1. To copy files from disk to disk, disk to dectape, or dectape to disk, use the system COPY command.

2. To convert SOS files to TECO files (i.e., to remove line numbers, convert page marks to form feeds, and allow lines to cross block boundaries), simply read them with TECO with /SUPLSN switch or use the ES option of SOS.

3. To convert TECO files or other files without line numbers to SOS files simply read them with SOS. SOS will number them by 100 and insert a page mark if there are more than 999 lines on a page. Any form feed which is the first character on a line will be converted into a page mark. "Bare" <return>'s are deleted and "bare" <line feed>'s are changed to <return> <line feed>.

If you want a TECO file to be numbered by some increment other than 100, see the /START and /STEP switches.

The format of a standard text file is defined as follows:

A line number is a string of 5 ASCII digits, left justified in a single word, and having bit 35 a 1. A line consists of a line number followed by the text of the line in ASCII characters, left justified in words with bit 35 a 0. The first character of the text is a tab and the last two characters are <return> and <line feed>. The null character (0) may not appear in the text. The last word of the line is filled with nulls to make a complete word if necessary.

A page mark is a word containing 5 ASCII spaces (40), left justified in a word with bit 35 a 1, followed by a word containing the characters <return>, <form feed>, <null>, <null>, <null> and with bit 35 a 0.

Lines are placed into disk blocks starting with the first word. A line is never broken across block boundaries. No 0 words appear in a block except after the last line of that block. All unused words in a block are 0.



APPENDIX B  
CHARACTER CODES

This appendix supplied separately in hard copy form.

## APPENDIX C

## SUMMARY OF COMMANDS

I	Insert	[<position>,<increment>]
P	Print	[<range>,s] (,s for no line no's)
.	Move pointer	[<position>]
D	Delete	[<range>]
K	Kill page mark	[/<page number>]
R	Replace	[<range>,<increment>]
E	End (Exit to monitor)	
ES	Unsequence resulting output file	
EB	Suppress ".BAK" file creation	
EQ	Quit (while you're ahead)	
ED	Delete input file	
EC	save Checkpoint file	
G	Go (Exit and do last COMPILE class cmd)	
W	save World (same as E but leaves edit open)	
M	Mark	[<position>]
N	Number	[<increment>,<range>,<start>]
NA	Add <increment> to <range>	
NP	Preserve line numbers over page marks	
L	List (on LPT)	[<range>,s] (,s for no line no's)
J	Join	[<position>]
F	Find	[<string>\$<range>,<foptions>]
S	Substitute	[<old string>\$<new string>\$<range>,<soptions>]
C	Copy	[<dest-position>,<range>]
	or	[<dest-position>=<filespec>,<range>]
T	Transfer	[<dest-position>,<range>]
X	extend	[<range>,S] (,S suppresses typeout)
JW	fill (a la runoff)	[<range>]
JU	fill and Justify	[<range>]
JL	flush Left	[<range>]
JR	flush Right	[<range>]
JC	Center	[<range>]
VV	inVert case	[<range>]
VL	to Lower case	[<range>]
VU	to Upper case	[<range>]
H	Help (types this text)	
HS	Help Switches	
<LF>	Same as P,+1	
S	Same as P,-1	
@	Indirect command file	[<filespec>]
/	Set parameter	[<param>=<value>]
A	Alter	[<range>]
nC<ch>	Change n characters	
=nD	Delete next/last characters	
E	Finish with no printing	
nI<text>s	Insert with temp increment n	
J	Break line and stick rest at front of next	
=nK<ch>	Delete to n occruances of <ch>	
L	Print line and recycle	

P	Print line and recycle to current position
Q	Quit ALTER (restore original line)
=nR<text>\$	nD then 0I
=nS<ch>	Skip forward/backward n occurrences of <ch>
nW	Skip forward n words
X<text>\$	Delete word then 0I
nV	invert case for n char's or to end of word
=n<SP>	Space forward/backward n characters
<CR>	Finish and print rest of line
n<RUB>	Backspace n characters
n<BS>	Backspace n characters
=<HT>	Skip to end/start of line
*U	Quit and restart ALTER

## APPENDIX D

## SUMMARY OF PARAMETERS

name	set	print	default	what
.	n	y	---	current <position>
BAK	y	y	on	create backup file (,Qxx)
BASIC*	y	n	off	read BASIC files
BIG	n	y	---	largest page #
C64	y	CASE	on	64 character set
C128	y	CASE	off	128 character set
CASE	n	CASE	C64 M33 UPPER	give case information
CHKPT*	y	y	on	do checkpointing
DECIDE	y	y	off	auto ",D" for "S" command
DISK	n	y	---	current quota information
DSK	Same as DISK			
DPY	y	n	off	special VT05/06 mode
ERROR	n	y	---	print last error message
EXPERT	y	n	off	more steamlined messages
INCREMENT	y	y	STEP	default line increment
ISAVE	y	y	5	how often chkpt in insert
LDEVICE	y	y	LpT	output device for L comnd
LENGTH	y	y	55	lines of text on TTY page
LMAR	y	y	1	left margin (for JU etc.)
LOCATION	n	y	---	first line in edit buffer
LOWER	y	CASE	off	default lower case
L64	y	CASE	on	upper case only LPT
L96	y	CASE	off	upper/lower case LPT
M33	y	CASE	on	upper case only TTY
M37	y	CASE	off	upper/lower case TTY
MAXLN	y	y	99999	max line = (for JU etc.)
NAME	y	y	---	output file spec
NOBAK	y	n	off	no backup file
NOCHKPT*	y	CHKPT	off	no checkpointing done
NODECIDE	y	n	on	no auto ",D"
NONSEPARATOR	y	CASE	on	don't treat %,s,, as alpha
NONUMBERS	y	n	off	don't print line nos.
NOSAVECHKPT	y	CHKPT	on	delete chkpt file on exit
NOVICE	y	n	on	regular messages
NUMBERS	y	n	on	print line nos.
OLD	y	BAK	off	create backup file (,Zxx)
OPTION	y	n	---	read "SWITCH.INI" file
PAGESIZE	y	y	66	TTY physical page size
PLINES	y	y	16	used by "P<CR>"
PMAR	y	y	1	paragraph margin (JU etc)
R*	Same as RONLY			
RONLY*	y	n	off	read only file
READONLY*	Same as RONLY			
RMAR	y	y	69	right margin (for JU etc)
RUN	y	y	"SYS:COMPIL"	set run spec

SOS Editor  
D. Summary of Parameters

SAVE	y	y	1	how often chkpt commands
SAVECHKPT	y	CHKPT	off	don't delete checkpt. file
SEPARATOR	y	CASE	off	treat %,s,, as alpha
SEQUENCE	y	y	on	line nos. in output file
START	y	y	STEP	1st seq. # on new page
STEP	y	y	100	perm. seq. # incr.
STRING	n	y	---	current F & S strings
UNSEQUENCE	y	n	off	no line nos. in out. file
UPPER	y	CASE	on	default upper case
VERSION	y	y	from input	version number

## APPENDIX E

## TERMINALS AND LINE PRINTERS

This appendix explains in a bit more detail some of the switches that affect how your output looks on the terminal or the line printer. Most of these switches affect the terminal and the line printer in exactly the same way. However, /M33, /M37, /L64, and /L96 set the type of terminal and line printer respectively.

The most important choice of modes is made by the switches /C128 or /C64. In /C64 mode characters are inputted or outputted exactly as typed, except that the usual uparrow convention is used for control characters (e.g. ^A for control-A). The difficulty with this mode is that it is not possible to enter certain control characters (e.g. ^C) directly, since the monitor responds to them in special ways. Furthermore, if your terminal or line printer does not have lower case letters, there is no way to enter or display lower case letters. Thus a second mode, set by /C128, is provided. In this mode, otherwise nonprintable characters are represented by a single quote followed by some normal character. Thus ^C is represented as '^', and a lower case a as '^A. When you are in /C128 mode control characters are always printed with the single quote conversion. They may be inputted the same way, or with the control key (for those characters that the monitor hasn't defined to have special effects). Note that in this mode, a real single quote is represented as '^'. In /C64 mode, the single quote has no special properties.

Because the availability of upper and lower case depends upon the terminal type, what happens to them in /C128 mode is controlled by the switches /M33, /M37, /L64 and /L96. If /M37 (for Model 37 Teletype) is set, it is assumed that your terminal has both and upper and lower case. Thus all letters are printed exactly as is. If /M33 (for Model 33 Teletype) is set, it is assumed that your terminal has upper case only, and lower case letters print as the corresponding upper case letter prefixed by '^'. The same options are controlled for the line printer (i.e. the "L" command) by the switches /L64 and /L96. If /L96 is set, it is assumed that you have a 96 character line printer, with upper and lower case. Thus all letters are printed as is. If /L64 is set, lower case letters are printed as the corresponding upper case prefixed by '^'. Note that if /M33 is set, lower case letters may be typed in either with the '^', as they print, or as real lower case (in case you lied when you set the /M33 switch). Beware that these switches, /M33, /M37, /L64, and /L96, have no effect in /C64 mode, since in that mode the single quote has no special properties.

There is a remaining problem with the above scheme. Many times users are working with textual material that is mostly lower case. Thus a typical sentence would show up as follows if /C128 and /M33 were set (or /C128 and /L64 if printed by the "L" command):

```
00100 T'H'I'S 'I'S 'A 'T'Y'P'I'C'A'L 'L'I'N'E,
```

Thus there is a further mode setting that allows the single quote to be used to indicate upper case, and print lower case as upper case without a single quote. This mode is /LOWER. (The normal is /UPPER.) Note that setting /LOWER only affects output and input where the single quote is being used to show case. So it is irrelevant for /C64, or for /M37 with the "P" command and /L96 with the "L" command. If /C128, /M33 (or /L64), and /LOWER are all set, the above line would look as follows:

```
00100 'THIS IS A TYPICAL LINE,
```

Default settings for these various parameters are usually /C64, /UPPER, /L64, and /M33. However if you have issued the monitor command TTY LC, it is assumed that you have a terminal with lower case and /M37 is set for you automatically. At many installations, TTY LC is done automatically for lines that are known to have terminals with lower case connected to them.

Most users are advised to make the following settings:

/C128 vs. /C64

Use /C128 if you need the single quote for either control characters or lower case. Otherwise don't set any of these parameters (as the default is /C64).

/M33 vs. /M37

Let SOS set this up for you, but be sure TTY LC has been typed to the monitor if you have a lower case terminal and want to be able to type lower case.

/L64 vs. /L96

Set /L96 if your line printer has lower case. Otherwise leave this parameter alone, since it will default to /L64. (An installation all of whose line printers have lower case is well advised to change the default setting to /L96.)

/UPPER vs. /LOWER

If your terminal and line printer have lower case, don't worry about this one. If not, set it up according to what you are working on. If it is mostly lower case (e.g. text), set /LOWER. Otherwise leave it alone and it will default to /UPPER.

## APPENDIX F

## FILE ENCRYPTION

SOS can edit files which have been encrypted by the standard DEC encryption algorithm. Such files are unintelligible unless one knows the "key" that was used to do the encryption when the file was written. One may say

```
SOS <file>(<key>)
```

to create or edit a file with specified encryption key. The editing proceeds as usual, but SOS decrypts the file as it is reading it and encrypts it as it is writing it out. If an encrypted file is edited without specifying the encryption key (or with the wrong key), all SOS will see is garbage. The key, which is typed between parentheses following the file name, consists of up to 20 characters. These characters can be almost anything except carriage return, line feed, or the close parenthesis. All lower case letters in the key are treated as if they were upper case. Thus the case you type the key with has no effect. It is possible to change the key of a file by a command like

```
SOS <ofile>(<new key>)=<ifile>(<old key>)
```

Omitting the key specification, or specifying a null key with (), both cause the file to be read or written in normal (non-encrypted) form.

## NOTE

WARNING! NO ONE can recover a file whose encryption key you have forgotten. Keys are not stored anyplace, so no system programmer can help you. They must be typed in every time the file is to be edited.