INSTRUCTION MANUAL FOR CREATING DATABASES WITH THE SPIRIT SYSTEM

PREPARED FOR NASA KSC PUBLIC AFFAIRS OFFICE

DECEMBER 1989

Written by Mark A. Clark

CONTENTS:

I. Instructions for IBM/PC Software Modules.

A. Introduction:

Several software modules have been created to aid in producing the required marked database to be indexed by the SPIRIT system. The modules were written in Turbo C and Turbo Pascal on an IBM/XT. They will run on any IBM/XT, AT, or any compatible computer. The Turbo C or Turbo Pascal compilers are not needed, as the modules are in an executable form. Even the Turbo Pascal Database Toolbox modules are included since they are used in some of the Pascal programs. All files are contained on the 5 1/4" disk labeled NASAPROJECT. A directory of this disk appears in figure 14 at the end of this instructional manual. All that is necessary to use the modules is to make the drive that NASAPROJECT is in the current drive. Then enter the name of the module desired. For example, to use the CONVERT2 program, simply type in "convert2" and hit enter at the dos prompt. Each module will prompt the user for certain input. An explanation for each module is written below.

With regard to SPIRIT, the modules most often used in database preparation are CONVERT (or CONVERT2), and FILEVIEW.

B. Explanations of modules:

(c) means written in Turbo C
(pas) means written in Turbo Pascal

CONVERT(pas)-Used for marking documents for the SPIRIT system. Same as CONVERT2 but not as many features (can't override the automatic marking process). Follow the prompts. See CONVERT2.

CONVERT2(pas)-This program takes a list of files and formats them for the SPIRIT database system. It will remove garbage control characters, and is intended to be used on documents created by WordPerfect or Wordstar THAT ARE SAVED AS AN ASCII TEXT FILE. It puts in the special document markers between each paragraph, and formats a line to a maximum of LINELEN characters. One can specify whether or not right justification is desired. In the input file, which is a list of filenames, there should be only one filename per line (drive and pathnames are allowed to proceed the filename in the usual dos manner). Each line should end with a carriage return, NOT a linefeed. A filename proceeded by a # (like #a:myfile.doc) will be treated as a single document. This is allowed for special parts of the document created where it is desired to override the automatic marking system. If the # character is not present, each paragraph (if more than 4 lines) will be treated as a document in the output. All files in the list of filenames will be concatenated into one long file for output. The input file (list of filenames) can be created with

any simple text editor. The Turbo editors are fine. An example of such an input file can be found in the file SHUTTLE.LST on this disk. One can also create the input file as the program is running, and it will save it for later use. The LINELEN constant has been set to 76, but can be changed by changing its value in the code and recompiling. This was done as a safety feature since SPIRIT restricts its lines to no more than 78 characters. CONVERT2 expects paragraphs to end with a linefeed or carriage return. The user might want to look at his file with FILEVIEW to make sure this is the case. Otherwise CONVERT2 will most likely not work as intended. When giving the name of the output file, make sure there is enough storage room available for the output to be saved. Follow the prompts for what to enter.

REMARK: With regard to the marked documents SPIRIT expects, CONVERT2 creates documents with two fields, an identifier field and a text field in every document (see section II-C3, "Document markers and document preparation").

FARHEAP(c)-Utility that returns the number of bytes left in the farheap.

FILEVIEW(pas)-Use this to look at text files created by wordprocessors to see what special characters are in the file. The three most common are 10 (carriage return), 13 (linefeed), and 26 (end of file). Special characters are set out between vertical bars so they can be "seen" in the document. Consult an ASCII table if necessary. Use this utility to look at the file BEFORE trying to use CONVERT or CONVERT2.

FREQCNT(c)-Produces frequency count of words in a single text file. Creates "outfile.doc" (alphabetic list) and "outfile.frq" (by order of frequency) in the current directory. Follow prompts.

MAKEALPH(c)-Creates an alphabetic list of words and their frequencies from many documents. Enter input files as prompted. Enter quit when there are no more files. Output stored in "b:outfile.doc".

MAKEFREQ(c)-Makes a frequency list from an alphabetic list. Each line must have just "integer, string". Usage of the file created by MAKEALPH or MARK is an option.

MANYDOCS(c)-Same as FREQCNT, but it works on many documents at once. Enter documents as prompted, quit if no more documents to input. Output saved in "outfile.doc" (alphabetic list) and "outfile.frq" (frequency list) in the current directory.

MARK(pas)-Program takes a list of files and indexes all the files in that list. It then creates an alphabetic list of all words that appear in all the files with their frequency count. This program contains a list of 100 most common words that are not included in the list created. It uses the Turbo Pascal Database

Toolbox B+ trees for efficiency. Enter index to be created, input file (list of files to examine), and then the name of the alphabetic list to be made.

MERGETWO(c)-Merges two alphabetic lists into one list. Each input file must have "integer, string" per line (such as the files created by MARK or MAKEALPH). Enter first source file, second source file, and destination file at the prompts.

QUICKFREQ(pas)-Program takes alphabetic list (such as one created by MARK) and converts it to a list ordered by frequency counts. Again, each line of the input file should have "integer, string". Uses quicksort routines from the Turbo Pascal Database Toolbox. This is the recommended module to use to create a frequency list of words, but first use MARK to create the alphabetic list.

SHUTTLE.LST(Not a module)-An example of the type of input file that is expected for the module CONVERT2.

TRUN(c)-Utility can be used to truncate a long list after a printer jam and one wants to continue printing, but does not want to start over.

II. Instructions for using SPIRIT.

A. Introduction:

Most documents created at the NASA Kennedy Space Center are created on IBM-PC compatible computers such as the Compac Deskpro 386 using Wordstar 2000. However, the shuttle manual was created by Rockwell using WordPerfect. Both of these wordprocessors have the ability to save a document as an ASCII text file which is virtually free from the headers and control characters that make text processing difficult.

It will be assumed that one already has created all the documents that one is going to place in the database before starting this task of getting the database into SPIRIT. SPIRIT (Syntactic and Probabilistic Indexation and Retrieval of Information in Texts) is a commercial software product that runs on an IBM mainframe in a VM/CMS operating environment. Since the NASA KSC Public Affairs Office already does its wordprocessing on the PC, the decision was made to do all the text processing needed for SPIRIT on the PC. Afterwards, the database would be sent from the PC to the mainframe ready for SPIRIT to index.

In this manual, it is assumed there is a communication link between the PC (in this case it is connected to a LAN) and the

IBM mainframe (in this case an IBM4381). If this communication link is not present, the user will need to find another method for getting the file to the mainframe.

First, the steps taken on the PC are described, then the steps taken on the mainframe. As the steps taken on the mainframe are described, the necessary features of SPIRIT used to prepare a database are mentioned. Finally, some other auxiliary features of SPIRIT are described that are not needed when creating the database.

B. Steps to create a database.

1. Examine documents/create list of files for the marking system

To examine the documents, use the DOS command TYPE, such as "type myfile.doc". However, this method has the drawback that control and special characters are not revealed. Use the FILEVIEW program to see what special characters are being used. Any editing must be done with an editor that will save the document as an ASCII text file.

Creating the list of files can be done with this same editor, or with one of the Turbo editors. It is at this point that one should decide what documents should be automatically marked and what documents should be kept in its entirety. One should also experiment with the CONVERT programs to see if they will work as the user expects. Remember, CONVERT is expecting linefeeds or carriage returns to mark the end of a paragraph, with no other special characters used for this purpose.

2. Run CONVERT or CONVERT2 to create the marked database.

Once the user is satisfied with the way the documents will be separated and marked, it is time to run either of the CONVERT programs to create the database.

If the document is prepared in another way, please see section II-C3, "Document markers and document preparation". The user must prepare and mark the documents properly before trying to index them. The database can be prepared from a variety of sources, even the vanilla editor in SPIRIT, but must be laid out with the proper markers for SPIRIT to use.

3. Send the marked database from the pc to the mainframe.

The explanations in this step are geared towards the equipment at the University of Central Florida (UCF). It is

expected this procedure will vary at other sites. If document creation and marking is done on the mainframe, then this step is unnecessary.

Use the Local Area Network (lan) terminal in the faculty I and R lab, or one of two lan terminals in CCII-106 that allow one to log on the lan and mainframe simultaneously. Make sure there is adequate room on a mainframe disk to store the base. Make drive "a" the current drive on the lan. Then enter:

a>SEND <lanfilename> <mainframefilename> FIIN A (ASCII CRLF

Case of letters is not important. The filename is arbitrary, but SPIRIT expects a filetype of FIIN. "A" is just the drive specification. At this point, the base should be on the mainframe and one can log off the lan.

SPIRIT expects the base file (which will often be referred to as the FIIN file, because the index file SPIRIT creates has a filetype of BASE) to be of fixed length records of 80 characters. SEND has created a variable record length file which should now be changed to fixed length. To change the file, enter:

COPYFILE <mainframename> FIIN A (RECFM F LRECL 80

Do a directory to make sure the FIIN file is now F80. This will be indicated by F and 80 in two of the columns of the directory. For those not familiar with CMS commands, one way to get a directory of files on disk D is to enter "fl * * d" from CMS. When done hit pf3 to quit. See a book of CMS commands for more details.

Now an important remark about base names used in SPIRIT. The filename given to the FIIN file becomes the (data)base name throughout SPIRIT, unless later changed. When SPIRIT asks for the name of the base, it means the filename before "FIIN <filemode>". If the input file is SHUTTLE FIIN D, then the base name would be SHUTTLE.

4. Enter SPIRIT and index the database.

A profile called PROFI should be executed to set up disks A, B, C, D, and G for the SPIRIT system. Enter "PROFI" to run it. This profile also moves the old A disk to E.

Move the FIIN file to drive D using "COPY <filename> FIIN E <filename> FIIN D". If this is done before running PROFI, replace E with filemode A. This will give the user two copies of the FIIN file. If only one copy of this file is desired, run the profile PROFI first before sending the file to the mainframe. Then just send the file to disk D instead of disk A.

Now enter SPIRIT by entering "$PROFX". The main menu will appear (fig. 1). Select CREATE A DATABASE by entering "2". A table will appear with some words in French (fig. 3). For most purposes, only two fields need be declared, identifier and text. The field type (type du champ) must be FACT for the document identifier, and TEXT for the text field. Leave the video field (champ video) as N. Leave the identifier field (champ ident) as N, EXCEPT the identifier field, which must have the letter "o" or "O". The "LG DE LA QUESTION" field is for the query grid. Enter any number such as 80.

When the table has been completed, move the cursor to the bottom center of the screen, and enter "s". For some more details on filling in this table, see section II-C3, "Document markers and document preparation". The example in figure 3 will suffice for most purely textual databases.

Next, one will see SPIRIT creating messages as it creates the base (fig. 4). Success at each step is shown by a return code of 0 (RC=0). There will be about two screens of processing messages before the base is created. When SPIRIT is finished indexing, a message appears as a reminder to give the user access to the base. Hit enter to return to the main menu.

5. Authorize the user to access the new database.

To authorize the user to access the database, choose MANAGEMENT UTILITIES by entering "g" at the main menu. Then from the UTILITIES menu (fig. 2), choose 1, PROFILE MANAGER. The user will be asked several questions that are still in French (hopefully, this will be corrected in later versions). Enter these answers to the questions (user answers will be in capital letters):

```
STEPS TO ADD A DATABASE NAME TO THE LIST OF BASES:
Choix (Tulib, Dpb,.....)    :T (fig. 5)
Commande                    :M (for modify or update)
Type du profil              :U (for user)
Profil a modifier           :SPIRIT (name of the profile that
                             contains information for the bases)
Voulez-vous voir.....       :F
Nro de variable ....        :12 (the choice to modify the list of
                             bases)
Ajout,....                  :<enter base name> (must be same name
                             that the FIIN file uses, see NOTE#1)
* more than one base name can be added to the list at this point
by separating base names with a semicolon (;).
Ajout (suite)               :<enter>
Suppression                 :<enter>
Voulez-vous voir la liste..:<enter>
```


25

To end and save the changes, enter "F" at the prompt "Nro de variable a modifier" (fig. 6). Continue to enter "F" until being returned to the UTILITIES menu (a total of four times). "F" is for "fin", which is French for "end". Enter "F" again to return to the main menu. The database is now ready to query.

Once the database name has been put into the list of databases to access, it will stay there unless the user removes it. Therefore, if the base is reindexed, this step does not have to be repeated. The name of the base can even be in the list when the base no longer exists. In this case, if the base that does not exist is requested (fig. 8), an error message will result. If another base is not desired when figure 8 appears, enter "fin" to quit.

To remove a database name from the list of bases, follow the same steps as above in "STEPS TO ADD A DATABASE NAME TO THE LIST OF BASES", EXCEPT for the following:

Ajout,.... :<enter> (unless of course there is another database name to be added)
Ajout (suite) :<enter>
Suppression :<enter a base name to delete>
* note: more than one base name can be deleted at a time by separating names by a semicolon (;).

All other steps are the same. End and save changes in the same manner as shown above.

C. Features of SPIRIT.

1. Vanilla editor.

The main menu choice one (SAVING AND STORING DOCUMENTS) takes the user to the vanilla editor which is described in more details below.

The most commonly used function keys are defined as follows:
pf1=help menu
pf3=file(save and quit)
pf4=tab key
pf7=back a page
pf8=forward a page
pf11=split/join
pf12=cursor home (toggles cursor between command line and text).

For a complete list, enter "q pf" on the command line.

The following is a list of the more commonly used features of this editor.

ADD LINES

Move cursor to numeric field in left column of line above where additional lines are desired. Enter a number followed by "a" and "enter" (or just "a" and "enter" for a single blank line) to enter as many blank lines as desired. Example: to insert 3 blank lines between lines 7 and 8, go to numeric field of line 7, type 3a, and hit enter.

DELETE LINES

Move the cursor in the numeric field on the left column of the screen to the line where deletion of some lines is desired. Enter a number followed by "d" and "enter" (or just "d" and "enter" to delete a single line) to delete as many lines as desired. Example: to delete lines 2,3, and 4, go to numeric field of line 2, type 3d, and hit enter.

DELETE A BLOCK (a block is two or more adjacent lines)

Block the text with "dd" in the numeric field of the first line of the block, and a "dd" in the numeric field of the last line of the block. Hit enter and the block will be deleted.

MOVE A BLOCK

Block the text with "mm" in the numeric field of the first line of the block, and a "mm" in the numeric field of the last line of the block. Move the cursor to the line just above where the text is to be moved, and put "f" (for following) in the numeric field of that line. Hit enter and the block will be moved. To move a single line, put "m" beside the line to be moved, and move the cursor to the line just above where the line is to be inserted. Then put "f" in the numeric field of that line, and hit enter.

COPY A BLOCK

Follow the same instructions for MOVE A BLOCK. Note however that "mm" should be replaced with "cc", and "m" with "c" when copying a single line.

STRING SEARCHES

To obtain all the lines of the file that contain at least one occurrence of a string, enter "ALL /<string>/" on the command line. As usual, hit the enter key when entering commands on the command line.

QUIT AND SAVE

Hit pf3, or enter "FILE" on the command line.

QUIT WITHOUT SAVING CHANGES

Enter "QQUIT" on the command line.

CURSOR HOME

Hitting "enter" when the cursor is in the text area will home the cursor to the command line. Also, pf12 toggles the cursor between the command line and the text.

SHIFT SCREEN LEFT OR RIGHT

For text that does not fit completely in the current window, there is the ability to shift the screen left or right. Enter "LEFT <number>" or "RIGHT <number>" on the command line to shift the screen left or right <number> characters.

REMARKS

The backspace and delete keys operate in the text as one would expect. To see less frequently used commands not mentioned here, hit pf1 to obtain help. To get specific help on a known command, enter "HELP <commandname>" on the command line.

The SPIRIT vanilla editor is useful for making minor changes in the documents, or for deleting documents. However, it is not recommended for making major changes or updates to the database due to its lack of features. Major changes can be made in a wordprocessor on the microcomputers and the database can be marked there for SPIRIT to use. However, one advantage of this mainframe editor is that it can handle larger files than most microcomputer based text editors. The user might need to experiment to see which editor they are most comfortable with. Regardless of which editor is used, SPIRIT must reindex the database for the changes to go in the base.

2. Natural language queries and accessing the documents.

To query the database, choose "S" (QUERY THE DATABASE) from the main menu. At the next screen (fig. 7), hit enter. Erase "***" and enter "?" for help, or "fin" to return to the main menu, but this is rarely needed.

First, the user will be given a list of bases to choose from (fig. 8). Enter the name of the base to query. This list is all the base names that one can now choose from. The BASE file must be present to access the base, otherwise this operation will not work. Another situation is that the BASE file has been created,

but the name of the base is not present in this list. This will also create a situation in which the user will not be able to access the database. See section II-B5, "Authorize the user to access the new database".

At the next step a screen telling the number of documents in the base and the date of the last update appears (fig. 10). Hit "enter" to continue.

Finally, the principal menu of the query system appears (fig. 11). Below are the descriptions of the principal menu choices.

a) QUERY or Q

This is for natural language queries on textual fields. Queries must be posed as questions (they should end with "?"). The case of the letters is not important (see fig. 11). Hit enter after typing in the question, and lists of empty and key words will appear. Hit enter to continue (generally anytime "***" appears in SPIRIT, it means hit "enter" to continue).

Next a list of classes of documents found appears with the number of documents in each class (fig. 12). Keywords are also listed for each class. SPIRIT ranks the classes by the number of hits (keywords matched) and the proximity of the keywords to each other. Keywords roughly next to each other will be separated by a hyphen (-). Keywords appearing in the same document but some distance apart, will be separated by commas. Select the classes to be shown by entering a list of numbers separated by commas.

For "What do you want to display?" the options are:
<<.....first page of the document.
< or pf7.....previous page.
doc or "d".....display another document.
end or "e" or pf3.....end this query.
stop or "s" or pf4.....stop session, return to main menu.

b) AFQUERY

This feature is for forming a natural language query on all the fields, not just the textual fields. It works the same as QUERY, just the search domain is larger.

c) CONTQ

Used to continue the previous question. Allows one to ask more than one question simultaneously, or to change the previous question slightly. The screens will look the same as in QUERY.

d) BOOL

Used for boolean queries which create a list of documents that satisfy a logical expression containing the logical operators AND, OR, and EXCEPT. The submenu options are:

doc or "d"...browse documents.
end or "e"...return to principal menu.
help.....short tutorial of boolean queries.
histo.....browse previously asked questions.
list or "l"..display document numbers that satisfy the most recent query.
listfld.....list the fields in the current base.
display.....show documents found in the current list.

Examples of boolean queries can be:
<1>: <singleword>...retrieves documents that contain <singleword>
<2>: <word1> AND <word2>...retrieves documents that contain <word1> and <word2>
<3>: <word1> OR <word2>...retrieves documents that contain <word1> or <word2>
<4>: <word1> EXCEPT <word2>...retrieves documents that contain <word1> but not <word2>
<5>: 2 AND (date:=810101)...represents result of query #2 in which the date is January 1, 1981 (words or expressions can be substituted with an integer which represents the result of that numbered query).
<6>: OR (date:=810101)...represent documents that satisfy the last query or ones in which the date is January 1, 1981 (if the question begins with a logical operator, the result of the last question is the first operand).

An operand is any logical expression between parentheses. Comparison operators on structured fields are <, >, =, <=, or >= preceded by a ":" (i.e. (date:<810101), (date:>= 810101), etc.).

An operand can also be the result of any previous query.

<7>: 1 AND 4...yields documents satisfying queries #1 and #4.
<8>: shuttle OR 5...yields documents having the word shuttle or satisfying query #5.

Order of evaluation is from right to left. OR has priority over AND and EXCEPT. Parentheses can be used to change this order of evaluation. To clarify this, "<word1> AND <word2> OR <word3> EXCEPT <word4>" means "<word1> AND ((<word2> OR <word3>) EXCEPT <word4>)" due to the default order of operations. It should also be noted that the operator AND can be left out of the expression. Any two operands not separated by an operator are assumed to be separated by AND ("cabin pressure" means "cabin AND pressure"). The boolean expression can get as complicated as one can imagine, but it is suggested that one modify the query in small steps.

```
<9>: ((shuttle (landing OR launch) site) EXCEPT night) OR
     (shuttle launch site EXCEPT (kennedy space center OR
     florida))
<10>: shuttle landing OR launch site (remember OR evaluated
                                       before AND, so no
                                       parentheses are needed)
     610 documents are selected (this message from SPIRIT)
<11>: EXCEPT night
     392 documents are selected
<12>: shuttle launch site EXCEPT (kennedy space center OR
      florida)
     126 documents are selected
<13>: OR 11
     3457 only one document is selected
```

e) DOCQ

Use to query with a document. It uses all the keywords in one document to find all related documents. SPIRIT takes all the keywords in this chosen document and finds the other documents in the base that contain at least one of these keywords. Documents are ranked by the number of keywords matched. Enter the document number with which to query. One can select the classes of documents to look at in the usual fashion. This command can be used to simulate hypertext linkages between documents.

f) ANSWER

Shows the list of documents that satisfied the last query.

g) DOC or D

Use to browse documents in the base. Enter the document number of the document to view.

h) BASE

Use to change the base the user queries (fig. 8).

i) STOP or S

Ends SPIRIT session and returns to main menu.

j) PRINT

Use to print documents if printer is set up for printing.

k) G(rid queries)

Use for queries on multiple fields. (not documented here)

l) HISTO

Use to see a list of previous questions in this session.

m) ?

Use to see a shorter summary of these commands

3. Document markers and document preparation.

An example of a marked document is figure 13. It consists of three documents (lines 1-7, lines 8-15, and lines 16-19) of two fields each.

Each document must begin with a document identifier field, and can have any additional number of fields following the identifier field. Each field begins with "$$#" on a line by itself where # is a positive integer indicating the field number. A recommended identifier field might be "DOC<docnumber>" on a line by itself. The documents follow one another without a particular separator except the identifier field. The FIIN file is therefore a sequence of field numbers followed by their content. The field numbers should be in increasing order for each document, and some of the fields can be skipped in a given document. If they are not in increasing order (say they go from 16 to 3), then this will be considered the beginning of a new document. Again, field number one must be the document identifier field.

The text line is limited to 78 characters. The file format must be fixed record length and record length of 80 (F80). Two characters are reserved by SPIRIT for the display mode.

The input file name is normalized as "<basename> FIIN <filemode>" where <filemode> is usually D.

The first part of a marked database might look like:

```
$$1                                          ----+
content of identifier field one                  |
$$2                                              |
content of field 2                               |
$$3                                              +--document 1
content of field 3                               |
$$10                                             |
content of field 10 (ok to skip some fields)     |
$$25                                             |
content of field 25                          ----+
```

```
$$1  (start of a new document)                  ----+
content of identifier field one                     |
(identifier field must be in every document)        |
$$7                                                 +--document 2
content of field 7                                  |
$$12                                                |
content of field 12                             ----+
```

For certain documents, the marker "$$" can be inconvenient. To change it, change the variable "&CSEP" of the "$SPIFORM" exec to another two character sequence.

The following are some explanations of the field definition screen which one sees when creating a database. An example is:

Field name	Abbreviation	Field Type	Video Field	Identifier Field	Length of Question
identifier	id	fact	N	o (letter)	80
date	date	date	N	N	30
last name	lnam	fact	N	N	50
first name	fnam	fact	N	N	50
address	addr	fact	N	N	100
background	bkgd	text	N	N	80

DOUBLE CHECK ANSWERS, THEN ENTER "S" TO EXIT:__

Possible choices for:
FIELD TYPE: FACT, TEXT, or DATE. (use FACT for identifier field)
VIDEO FIELD: most purposes N. O (the letter) indicates the field contains the address of a video image.
IDENTIFIER FIELD: all should be N, except the first which is O.
LENGTH OF QUESTION: length of the query criteria which is associated with the field (used to define the query grid).

4. Database updates.

Main menu choice 3 (UPDATE A DATABASE) can be used when a document is to be added to a database (inserted or appended to the end). Also, it can be used to update current documents of a database if the identifier for the database already exists. A document is created if the identifier for it does not exist in the base. The word update is used in the usual sense (i.e. add lines, delete lines, make any changes on current lines).

It is also possible to use this procedure to delete the content of a document. This can be done by deleting all the fields for that document in the input (FIIN) file, EXCEPT the identifier field (this will be the only field left in that

document). This will cause the document to be empty giving the appearance of being deleted.

When using this procedure it does not matter what files have been left on disk G, unlike when reindexing using the alternate method in section II-C6 below. It is recommended that disk G be completely erased before logging off if it is not a temp disk (see section II-C6, topic "THE WORKSPACE ON DISK G").

The procedure is as follows. Use the vanilla editor (or any editor) to make the changes to the input (FIIN) file (see section II-C1 above on the vanilla editor). Choose "3" from the main menu. Enter the base name and a letter to indicate the language of the base ("e" or "f") as prompted. The reindexing process now begins. If the base name is not yet in the list of databases to access, put it in the list before trying to use the base (see section II-B5, "Authorize the user to access the new database"). If this has already been done (probably was done after creating the base the first time), then the base should be ready to use.

If for some reason the base created has erroneous results (documents in the wrong order, document content not with proper document identifier, updates not put in, etc.), then the user MUST start over again creating the base with main menu choice 2. Certain things must be done BEFORE reindexing with this method. The reader should read "ALTERNATIVE METHOD FOR REINDEXING A DATABASE" in section II-C6 below first.

5. Database deletion.

The entire database can be deleted at once by choosing main menu choice 4 (DELETE SOME DOCUMENTS). Enter the name of the base, and a letter "e" or "f" to indicate the language of the base when prompted.

Actually what happens is that all the documents that appear in the FIIN file at that instance are deleted from the base. If no changes to the FIIN file have occurred since the last indexing, the entire collection of documents (i.e. the entire database) will be removed. If only a few documents are deleted from the FIIN file (say documents 2 and 3) before choosing 4, then the result of choosing 4 is all documents will be deleted from the base EXCEPT documents 2 and 3. Other strange occurrences result when documents are added to the FIIN file before choosing 4.

For these reasons, no changes should have been made to the FIIN file since the last indexing before choosing this option.

6. Miscellaneous features of SPIRIT.

ENTERING VM COMMANDS FROM INSIDE SPIRIT

Choose main menu choice V to do this. Then enter a VM/CMS command in the usual fashion.

EXITING SPIRIT

Choose main menu choice F to exit. Generally entering "F" at any level of SPIRIT causes an exit or quit. Also "S" and "E" are sometimes used to quit at certain levels, however "F" usually takes the user out to the greatest extent possible at that point.

ALTERNATIVE METHOD FOR REINDEXING A DATABASE

Recall that the original text of a database is stored in a file on disk D of filetype FIIN. Indexing creates four new files on D with filetypes DPB, GRIL, NBPAB, and BASE. To query the database, only the BASE file is needed. However, grid queries are not supported with just this file. Several files are created on the temp disk G during the indexing process.

If there are some problems reindexing a database after updating it, (i.e. problems using SPIRIT main menu choice 3, "UPDATE A DATABASE"), then here is another recommended technique to reindex a base:

a) Make sure disk G is completely empty. If the file CLEANG EXEC A is on the A disk, from outside SPIRIT one can enter "CLEANG" to empty the G disk. Also, one can enter "ERASE * * G1" from anywhere one can enter a VM command (see ENTERING A VM COMMAND FROM INSIDE SPIRIT in this section).

b) For the particular database name, erase the files beginning with that name and having the filetypes DPB, GRIL, NBPAB, and BASE from disk D. DO NOT ERASE THE FIIN FILE (the result of doing parts a) and b) will be like starting anew).

c) Index the database as done previously using main menu choice 2 (CREATE A DATABASE).

d) If the database name is not yet in the list of base names, add it now (see section II-B5, "Authorize the user to access the new database"). If this has already been done (which it should have if the base is being reindexed), then it does not have to be done again. Once the base name is in the list of bases, it will not leave the list unless it is deleted.

REMARK: The only operation main menu choice 3 (UPDATE A DATABASE) saves is filling in the field table again, and a few steps in the syntactic and linguistic analysis of the database.

THE WORKSPACE ON DISK G

Normally, disk G will be a temp disk that will erase itself upon logging off. IF THE DISK G IS NOT A TEMP DISK IT IS IMPORTANT THAT ONE ERASE ALL FILES ON G BEFORE INDEXING (OR BEFORE REINDEXING ANY DATABASE when not using main menu option 3). If G is not empty, SPIRIT might abort the indexing process. It is a good habit to erase G before logging out, and before indexing (or reindexing if not using option 3), PROVIDED DISK G IS A PERMANENT DISK AND NOT A TEMP DISK.

An exec called CLEANG can be used to do this FROM OUTSIDE THE SPIRIT ENVIRONMENT. Execute it by entering "CLEANG" from OUTSIDE SPIRIT (check to see if CLEANG EXEC A is on disk A). If G is a temp disk, this is not necessary WHEN LOGGING OUT, as all files on G are erased upon logging out. Disk G might still need to be cleaned if it is a temp disk AND it has some files on it AND the user is getting ready to index again. Remember, DISK G SHOULD BE EMPTY BEFORE INDEXING ANY DATABASE for the first time to be safe.

ERROR MESSAGES AND RUNTIME PROGNOSIS

A prognosis of the indexing process, along with any error messages, is found in files with the filetypes IMP1 or FERREUR on disk G. Examples are $SPIMAJQ IMP1 G for messages during creating or updating a base, and $SPISUPQ IMP1 G for messages during deletion of documents.

Another source of possible error messages is main menu choice 5 (DETECTION OF ERRORS). After choosing 5, enter the database name and language as prompted. A few messages might appear while running. Then the file $SPIGEN1 IMP1 G will come up in the vanilla editor. This file contains some of the runtime results of the last index. Also, it contains some possible spelling errors it has found, or possible words not in the SPIRIT dictionary. Hit pf3 to quit.

Since all error message files are placed on disk G, it is important that the user not erase disk G until they are sure these files are not needed. Once erased, they can only be obtained again by reindexing.

THE STATE OF A DATABASE

From the UTILITIES menu, choose 5 (LIST THE STATE OF A DATABASE). Enter the name of the base (i.e. "<basename> BASE <filemode>"). Then enter the number of records in the file, and the length of a single record as prompted. This information comes by doing a directory (i.e. the VM command "FL * * D") BEFORE getting to this point. A sample directory for one base is:

```
SAMPLE FIIN  D   F     80    10   1
SAMPLE GRIL  D   F     80     8   1
SAMPLE DPB   D   F     80     6   1
SAMPLE BASE  D   F  23472    33 190
SAMPLE NBPAB D   F     80     1   1
```

The first numeric column is the record length, the second numeric field is the number of records, and the last numeric column is the number of blocks in the file.

After answering the questions, a table will appear giving details of the fields of the base. As of this writing this is still in French. The user can page up and down with pf7 and pf8. Enter pf3 to quit from the command line.

7. A note about changes made to SPIRIT at UCF.

Four modifications have been made to SPIRIT at UCF since installing it in July of 1989.

First, some of the menus and runtime messages have been translated to English, particularly the main menu and the utility menu. Also, the runtime messages during the indexing which can be found in the exec $SPIMAJQ EXEC C, and during deletion of documents in the exec $SPISUPQ EXEC C. Other translating has been done in some of the PL1 code, but the code has not been recompiled as of this writing.

Secondly, the sort that is called while indexing databases has been replaced with SyncSort, a more efficient commercial sort routine and can handle a much larger number of records. The original sort routine could not handle the large number of records required to be sorted in the shuttle manual database (close to half a million at one point) because it did sorting in virtual memory only. SyncSort uses disk space as well as virtual memory during sorting. The use of SyncSort has decreased indexing time from between two thirds to one third of the previous indexing times.

Thirdly, for a short time disk G was changed from a temp disk to a permanent disk. This was done before indexing the shuttle manual since it was anticipated that SPIRIT would need many cylinders for the G disk. There was a limit to how many cylinders the system could obtain at any given time for a temp disk. Therefore, it was decided not to risk an abort of indexing, and G was made a permanent disk to provide more cylinders. SPIRIT does not care what type of disk G is, as long as it is available. However, it is important the user remember to clean off disk G before doing any indexing, as some of the old

files might cause the indexing process to abort (see THE WORKSPACE ON DISK G in section II-C6, "Miscellaneous features of SPIRIT").

Finally, some lines were changed in a few execs that had errors as discovered by Christian Fluhr and Tran, the developers of SPIRIT in France. Those changes will not be detailed here.

III. Database Backup and File Transfers.

A tape backup of disk D can be done by entering the following CMS commands:

1) Enter "SM IRSERV TAPE"....gets a tape drive number. Then have the operator mount the tape before doing the next command.
2) Enter "TAPE REW"....causes tape rewind.
3) Enter "TAPE DUMP * * D".
4) Enter "TAPE WTM"....writes a tape mark to separate dumps. (repeat steps 3 and 4 as needed for other drives)
5) Enter "TAPE WTM"....writes second tape mark to indicate end of tape (could use single command TAPE WTM2 after last drive or file)
6) Enter "TAPE RUN"....causes tape rewind and unload.
7) Have the operator dismount the tape.
8) Mount another tape and go to 2, or continue to step 9.
9) Enter "DET 181"....disconnects tape drive, and ends session.

To backup single files at a time, replace "* * D" in the command in step three with a specific filename, filetype, and filemode (disk).

To transfer files between mainframe accounts, enter the CMS command "SENDFILE <fn> <ft> <fm> TO <userid>" (do this from the account that the file is being sent FROM). Then log on the account that the file was sent TO and enter the CMS command "RECEIVE".

IV. Time Estimates for Creating a Database.

A. Introduction:

Most of the following time estimates were based on the database created from the September 1988 edition of the shuttle manual. It is therefore important that some facts are understood about the size of this document.

The shuttle manual after being marked contained 4902 documents, most of which were paragraphs, with a few appendices treated as an entire document. After sending it to the mainframe and converting it to a fixed record format of 80 characters, the size was 55,119 lines or 1077 blocks (4K per block). After indexing, the shuttle base file that was created took 3525 blocks of 4K each. Any reference to the time that the shuttle manual database took to create will be noted.

B. Time Estimates:

EXAMINE DOCUMENTS / CREATE A LIST OF FILES FOR THE MARKING SYSTEM

Depends on the number of documents. Approximately 15 minutes to two (or more) hours could be used, depending on the condition of the documents.

RUN CONVERT OR CONVERT2 TO CREATE THE MARKED DATABASE

Depends entirely on the number of documents and the speed of the computer. About one hour, 35 minutes was taken for the shuttle manual on an IBM/XT PC, using hard drives for all I/O. An AT or compatible should be used for faster processing, along with hard drives for retrieval of the documents and storage of the output file.

SEND THE MARKED DATABASE FROM THE PC TO THE MAINFRAME

Depends on the size of the database and the baud rate of the communication link. Once the software is set up, this task should only take about five to 30 minutes.

ENTER SPIRIT AND INDEX THE DATABASE

Again, the time for this task is very dependent upon the size of the database, and possibly on the amount of users on the mainframe. The entire shuttle manual took about three hours, 30 minutes to completely index during a period when the mainframe had moderately low use.

AUTHORIZE THE USER TO ACCESS THE NEW DATABASE

This task takes about three to five minutes, regardless of the database.

Remark: It is estimated that the actual number of hours required to get the shuttle manual database created was about seven hours, 30 minutes.

```
S P I R I T    V.67 R 0.0   - VM / CMS - MENU 0

                1. SAVING/STORING DOCUMENTS
                2. CREATE A DATABASE
                3. UPDATE A DATABASE
                4. DELETE SOME DOCUMENTS
                5. DETECTION OF ERRORS

                S. QUERY THE DATABASE

                G. MANAGEMENT UTILITIES
                P. REDEFINE DISK PARAMETERS
                V. EXECUTE A VM COMMAND
                F. END

     BY DEFAULT (ENTER ) ==> QUERY THE DATABASE
     BEFORE CREATING, MODIFYING, OR DELETING   DOCUMENTS,
     THEY MUST BE SAVED/STORED BY OPTION 1.

ENTER YOUR CHOICE  :

                                                VM READ   UCF1VM
4B                                                          0-001
```

Figure #1 - SPIRIT's Main Menu

```
S P I R I T    V.67 R 0.0   - VM / CMS - UTILITIES

                1. PROFILE MANAGER
                2. MESSAGE MANAGER
                3. UPDATE THE DICTIONARY
                4. COMMAND MANAGER
                5. LIST STATE OF THE DATABASE

                F. END (RETURN TO MENU 0)
                X. EXIT SPIRIT

     BY DEFAULT (ENTER ) ==> PROFILE MANAGER

ENTER YOUR CHOICE  :

                                                VM READ   UCF1VM
4B                                                          0-001
```

Figure #2 - SPIRIT's Utilities Menu

```
GENERATION DES DONNEES POUR LA DEFINITION DES CHAMPS SPIRIT

-------------------------------------------------------------------------------
    NOM DU CHAMP                NOM        TYPE DU   CHAMP     CHAMP     LG DE LA
                                COMPACT    CHAMP     VIDEO     IDENT     QUESTION
-------------------------------------------------------------------------------

 1. identifier_____________     id__       FACT      N         o         80_
 2. text___________________     txt_       text      N         N         80_
 3. _______________________     ____       FACT      N         N         ___
 4. _______________________     ____       FACT      N         N         ___
 5. _______________________     ____       FACT      N         N         ___
 6. _______________________     ____       FACT      N         N         ___
 7. _______________________     ____       FACT      N         N         ___
 8. _______________________     ____       FACT      N         N         ___
 9. _______________________     ____       FACT      N         N         ___
10. _______________________     ____       FACT      N         N         ___

-------------------------------------------------------------------------------

(TYPE DU CHAMP : 'TEXT' POUR TEXTUEL OU 'FACT' POUR FACTUEL OU 'DATE' POUR DATE
 CHAMP IDENT   : CHAMP D'IDENTIFICATION DU DOCUMENT)

VERIFIEZ ENCORE PUIS TAPEZ 'S' POUR SORTIR : s
4B                                                                    O-001
```

Figure #3 - Table for defining database fields in SPIRIT

```
********** FORMATTING SPIRIT RC=0
********** DOCS TO DELETE RC=0
STATE $SPIMAJQ FDOC G
+++ E(28) +++
********** BREAKING UP OF TEXT RC=0
WER226A END SYNCSORT, RECORD= 60, INCORE
********** SORT BEFORE ANALYSIS RC=0
********** MORPHOLOGICAL ANALYSIS RC=0
WER226A END SYNCSORT, RECORD= 68, INCORE
********** SORT BEFORE SYNTACTIC ANALYSIS RC=0
********** SYNTACTIC ANALYSIS RC=0
WER226A END SYNCSORT, RECORD= 44, INCORE
********** SORT BEFORE INDEXING RC=0
********** CREATING INDEX RC=0
WER226A END SYNCSORT, RECORD= 44, INCORE
********** SORT BEFORE WEIGHTS RC=0
********** CALCULATING WEIGHTS RC=0
WER226A END SYNCSORT, RECORD= 45, INCORE
********** SORT BEFORE MAIN BASE RC=0
********** CONSTRUCTION OF BASE RC=0
1 occurrence(s) changed on 1 line(s)
1 occurrence(s) changed on 1 line(s)

                                                        RUNNING    UCF1VM
4B                                                                 O-001
```

Figure #4 - Typical run-time messages created when indexing a database

41

```
   IL EST 17.16.07 - LE 18.08.89




--------------------------------------------------------------------------------
T,TULIB ->PROFIL DS LING     D,DPB ->PROFIL DS FIC BASE(GRILLE+DBDS+DCS)
CHOIX (TULIB,DPB,TERM,F OU ?) :

4B                                                                      O-001
```

Figure #5 - First screen seen after choosing the Profile Manager from the Utilities Menu

```
 **** PROFIL TYPE U : SPIRIT
 1. NRO DE L'UTILISATEUR  :  55
 2. DATE CREATION         :  860106
 3. DERNIERE MAJ          :  890817
 4. TYPE TERMINAL ASSOCIE :  2
 5. NRO IMPRIMANTE ASSOCIE:  2
 6. MESSAGES COMPLETS     :  0
 7. MODE D'AFFICHAGE (X)  :  FF
 8. TYPE LANGUE ASSOCIEE  :  1
 9. CONFIDENTIALITE BASE  :  0
10. CONFIDENTIALITE DOC.  :  0
11. NB BASES ACCESSIBLES  :  6
12. LISTE DES BASES       :




--------------------------------------------------------------------------------
NRO DE VARIABLE A MODIFIER : f

4B                                                                      O-001
```

Figure #6 - Screen typically seen when inside the Profile Manager to modify the list of available databases

```
                IL EST 17 H 21 MN, LE 18 AOUT 1989 : BONJOUR .

                    * * * * * * * * * * * * * * * * *
                    *                               *
                    *          S Y S T E M E        *
                    *            S P I R I T        *
                    *               V 2.1           *
                    *                               *
                    * * * * * * * * * * * * * * * * *

                    REGLES GENERALES D'UTILISATION :
                    ?   : POUR OBTENIR  DES  EXPLICATIONS
                    FIN : POUR REVENIR AU MENU  PRECEDENT
                    RC  : POUR LAISSER LE SYSTEME CHOISIR
                    *** : TOURNER LA PAGE  EN  FAISANT RC

***


4B                                                                    O-001
```

Figure #7 - Screen seen after choosing "Query the Database" from SPIRIT's Main Menu

```
                                  LISTE DES BASES

CEEA                HALFSHUT              NEW                 SHUTTLE
TATA                TOTO


--------------------------------------------------------------------------------
QUELLE BASE VOULEZ-VOUS INTERROGER ? : shuttle

4B                                                                    O-001
```

Figure #8 - Screen showing current databases available

```
  SSSSSSSSS  HH          HH UU          UU TTTTTTTTTTTT TTTTTTTTTTTT LL
 SSSSSSSSSSS HH          HH UU          UU TTTTTTTTTTTT TTTTTTTTTTTT LL
 SS          SS HH       HH UU          UU       TT           TT      LL
 SS             HH       HH UU          UU       TT           TT      LL
 SSS            HH       HH UU          UU       TT           TT      LL
  SSSSSSSSS     HHHHHHHHHHHH UU          UU       TT           TT      LL
   SSSSSSSSS    HHHHHHHHHHHH UU          UU       TT           TT      LL
          SSS   HH       HH UU          UU       TT           TT      LL
           SS   HH       HH UU          UU       TT           TT      LL
 SS        SS   HH       HH UU          UU       TT           TT      LL
 SSSSSSSSSSS   HH       HH UUUUUUUUUUUU         TT           TT      LLLLLLLLLLLL
  SSSSSSSSS    HH       HH  UUUUUUUUUU          TT           TT      LLLLLLLLLLLL


                    NUMBER OF DOCUMENTS :   4902

                    LAST UPDATE         : AUGUST 17TH 1989
***


4B                                                                  O-001
```

Figure #9 - Screen showing the number of documents in a database

```
                    IT IS 17 H 26 MN, ON AUGUST 18TH 1989


                  * * * * * * * * * * * * * * * * *
                  *                               *
                  *         S P I R I T           *
                  *         S Y S T E M           *
                  *            R 2.1              *
                  *                               *
                  * * * * * * * * * * * * * * * * *


                    P R I N C I P A L   M E N U




--------------------------------------------------------------------------------
MENU: (QUERY,AFQUERY,CONTQ,BOOL,DOCQ,ANSWER,DOC,BASE,STOP,PRINT,G,?): q


4B                                                                  O-001
```

Figure #10 - Principal Menu in the database query area of SPIRIT

```
                    NATURAL LANGUAGE QUERY ON THE SHUTTLE BASE

<1> : where are the launch sites for the shuttle located?
EMPTY WORDS : where, are, the, for, the.
KEY WORDS   : launch, sites, shuttle, located.




***


4B                                                              O-001
```

Figure #11 - Typical response after a Natural Language Query has been entered into SPIRIT

```
CLASSES  NB DOCS  KEY-WORDS
    1         3   launch-sites,located.
    2        17   launch-sites,shuttle.
    3         2   launch,sites,located.
    4         2   sites,shuttle,located.
    5         9   launch,sites,shuttle.
    6        48   launch-sites.
    7         1   launch,shuttle,located.
    8         6   sites,located.
    9         5   launch,sites.
   10        18   sites,shuttle.
   11         4   launch,located.
   12         5   shuttle,located.
   13        54   sites.
   14        71   located.
BOTTOM OF LIST



--------------------------------------------------------------------------------
LIST OF CLASSES TO BE DISPLAYED (?) :

4B                                                              O-001
```

Figure #12 - Listing of possible relevant documents for the given query ranked in order of decreasing probabilities

```
TATA      FIIN      D1  F 80  Trunc=80 Size=19 Line=0 Col=1 Alt=2
===>

0000 * * * Top of File * * *
0001 $$1
0002 DOC001
0003 $$2
0004 This is a sample document after the markers for SPIRIT have been placed
0005 between the documents (in this case a document is a paragraph).  It is
0006 not required that a document be a paragraph.  Actually, it can be of any
0007 length, and can contain any number of paragraphs.
0008 $$1
0009 DOC002
0010 $$2
0011 The only requirement is that the fields in a document be separated by
0012 markers.  "$$1" on a line, followed by a document identifier is the first
0013 marker and field.  "$$2" on a line, followed by some text (a document) is
0014 the second marker and field.  One may use as many of these fields as
0015 needed.  For this example, only two fields per document are used.
0016 $$1
0017 DOC003
0018 $$2
0019 See the manual for further information.  This is the last document.
0020 * * * End of File * * *
```

Figure #13 - A small sample database showing the correct placement of document markers (three documents are present, each with two fields)

```
 Volume in drive B is NASAPROJECT
 Directory of  B:\

MANYDOCS C        3348   5-24-89   4:54p
QUICKFRE EXE      9744   6-14-89   1:40p
INKEY    TPU      4608   6-11-89   2:31p
WINDOWS  TPU     25136   6-05-89   6:14p
DEBUGA   TPU      1008   6-05-89   6:14p
STARTSCR TPU       960   6-05-89   6:14p
FILEVIEW PAS      2816   7-18-89   3:21p
MARK     PAS      6179   6-15-89   4:52p
CONSTANT TPU      1536   6-05-89   6:14p
CONVERT  EXE      9536   7-18-89   4:08p
MARK     EXE     21344   8-18-89   3:33p
LSORT    TPU      6608  12-31-87   4:00a
LSORT    PAS     16869  12-31-87   4:00a
SORT     TPU      5536  12-31-87   4:00a
SORT     PAS     16861  12-31-87   4:00a
STRINGS  TPU      2240   6-11-89   2:30p
CONVERT  PAS      6356   7-18-89   4:14p
SHUTTLE  LST       759   7-20-89   5:30p
TRUN     EXE     10846   6-06-89  10:33p
TURBO    TP       1206   7-06-89   4:41p
FREQCNT  C        2674   5-19-89   3:41p
TACCESS  TPU     22768   6-11-89   2:24p
TURBO    PCK      1201   8-18-89   3:41p
MERGETWO EXE     11956   6-06-89   1:38p
TRUN     OBJ      1044   6-06-89  10:32p
SCR      CFG      4000   6-09-89   2:26p
TRUN     C         733   6-06-89   4:03p
MAKEFREQ EXE     11522   6-06-89   4:20p
MERGETWO OBJ      2929   6-06-89   1:38p
MAKEALPH EXE     11540   6-05-89   4:33p
MAKEFREQ OBJ      1514   6-06-89   4:20p
MERGETWO C        2611   6-06-89   1:40p
MAKEALPH OBJ      2706   6-05-89   4:33p
FARHEAP  EXE      7488   5-31-89   8:51p
FREQCNT  EXE      9644   5-19-89   3:40p
MAKEALPH C        4425   6-05-89   4:32p
FILEVIEW EXE      6432   7-18-89   3:19p
QUICKFRE PAS      1414   6-14-89   1:40p
MAKEFREQ C        1302   6-06-89   4:19p
FREQCNT  OBJ      1925   5-19-89   3:40p
CONVERT2 EXE     10736   7-27-89   5:35p
CONVERT2 PAS      8462   7-27-89   5:35p
TEST     NDX      1461   8-18-89   3:37p
MANYDOCS OBJ      2451   8-18-89   3:45p
MANYDOCS EXE     10002   8-18-89   3:45p
```

Figure #14 - Directory of files on the NASAPROJECT disk created in the Fall of 1989 by Mark A. Clark

Figure #15a - Listing of the program CONVERT2.PAS written in Turbo Pascal

```
{This program takes a list of files and formats them for the SPIRIT database
system.  It puts in the special document markers between each paragraph, and
formats a line to a maximum of LINELEN characters.  One can specify if right
justification is wanted.  In the file that is a list of filenames, there should
only be one filename per line (drive and pathnames are allowed to preceed the
filename.  A filename preceeded by a # (like #a:myfile.doc) will be treated
as a single document by this program.  This is allowed for special parts of
the document created to override the automatic marking system.  If the # sign
is not present, each paragraph (if greater than 4 lines) will be treated as
a document in the output.  All files listed in the list of filenames will be
concatenated into one long file for SPIRIT to use.)

     { Created for NASA Public Affairs by Mark A. Clark.}

uses strings, crt;
type
bufftype=string[100];

var
inputfile, inputlist, outputfile: text;
readstr : string[50];
buff : bufftype;
testchar, testrightjust, inchar, holdchar: char;
oddchar, i, j: integer;

const
linelen: integer=76;   {MAXIMUM LINELEN IS 80}
linecnt: integer=0; buffcnt: integer=0;
lastblank: integer=0;
doccnt: longint=1;
holding: boolean=false;
specialfile: boolean=false;
label redolist, redooutfile, restart;

{procedure strips out any trash bits in ascii text by wordprocessors}
procedure wordprocscreen(var ch: char);
var
bh: byte;
begin
bh:=ord(ch); oddchar:=0;
bh:=bh and $7f;
if ((bh in [10,13,26]) or ((bh>31) and (bh<127)))
   then ch:=chr(bh)
   else
      begin
      ch:=' '; oddchar:=1;
      end;
end;

procedure makelist(myfilestr: string);
var
filestr: string[50]; outputlist: text;
begin
writeln('Limit filenames to 50 characters, please.  Hit RETURN to end list.');
assign(outputlist, myfilestr);
{$I-}
reset(outputlist);
{$I+}
if ioresult=0 then
   begin
```

Figure #15b - Continuation of the listing for CONVERT2.PAS

```
   close(outputlist); erase(outputlist);
   end;
rewrite(outputlist);
write('Enter a filename: '); readln(filestr);
while filestr<>'' do
begin
   writeln(outputlist, filestr);
   write('Enter another filename: '); readln(filestr);
end;
writeln('Finished making file list.....');
close(outputlist);
end;

procedure rightjust(line: bufftype; len:integer);
var
blanks, shifts: array[1..25] of integer;
numblanks, curblank, pad, rtshift: integer;
newline: string[80];
tempstr: bufftype;
begin
numblanks:=0;
tempstr:=line;
for i:=1 to 25 do shifts[i]:=0;
for i:=1 to linelen do newline[i]:=' ';
for i:=2 to len-1 do
   if (tempstr[i]=' ') and (tempstr[i-1]<>' ') and (tempstr[i+1]<>' ') then
      begin
      inc(numblanks); blanks[numblanks]:=i;
      end;
pad:=linelen-len;
while pad>0 do
   begin
   if pad<=numblanks then
      begin
      for i:=1 to pad do
         inc(shifts[i]);
      pad:=0;
      end
   else
      begin
      for i:=1 to numblanks do
         inc(shifts[i]);
      pad:=pad-numblanks;
      end;
   end;
rtshift:=0; curblank:=1;
for i:=1 to len do
   begin
   newline[i+rtshift]:=tempstr[i];
   if i=blanks[curblank] then
      begin
      rtshift:=rtshift+shifts[curblank]; inc(curblank);
      end;
   end;
newline[0]:=chr(linelen);
writeln(outputfile, newline); writeln(newline);
end;


(MAIN PROGRAM)
```

Figure #15c - Continuation of the listing for CONVERT2.PAS

```
begin
clrscr;
writeln('Program takes a file created by ascii wordprocessors and formats the');
writeln('text for use in the indexing system of the SPIRIT database.  Each');
writeln('line is formatted to a specific line length and ends with CR.  Garbage'
writeln('control characters are removed from the file.  Segmentation of the');
writeln('file into separate documents is based on finding a carriage return,');
writeln('hence the end of a paragraph.'); writeln;
redolist:
writeln('Enter the name of the input file (list of files) to convert.');
readln(readstr);
write('Do you want to create the list of input files now? (Y/N)');
testchar:=readkey; writeln;
if upcase(testchar)='Y' then makelist(readstr);
assign(inputlist, readstr);
{$I-}
reset(inputlist);
{$I+}
if (upcase(testchar)<>'Y') and (ioresult<>0) then
   begin
   writeln; writeln('*** the list of input files does not exist ***');
   writeln('*** please re-enter the input file (list of files) again ***');
   writeln; goto redolist;
   end;
write('Do you want the output right justified? (slower to process) (Y/N)');
testrightjust:=readkey; writeln;

redooutfile: writeln;
writeln('Enter the name of the file CONVERT produces for the SPIRIT system.');
readln(readstr);
assign(outputfile, readstr);
{$I-}
reset(outputfile);
{$I+}
if ioresult=0 then
   begin
   writeln; writeln('*** This output file already exists ***');
   write('Do you want to over write it? (Y/N).');
   testchar:=readkey; writeln;
   if upcase(testchar)<>'Y' then goto redooutfile;
   end;
rewrite(outputfile);

writeln;
writeln('================================================================');
writeln('*** TEXT PROCESSING BEGINNING ***');
writeln('================================================================');

repeat
   readln(inputlist, readstr);
   {test to see if filename begins with # to indicate specialfile}
   if readstr[1]='#' then
      begin
      specialfile:=true;
      move(readstr[2], readstr[1], length(readstr)-1 );
      readstr[length(readstr)]:=' ';
      readstr[0]:=chr(length(readstr)-1);
      end;
   assign(inputfile, readstr);
   {$I-}
```

Figure #15d - Continuation of the listing for CONVERT2.PAS

```
reset(inputfile);
{$I+}
if ioresult<>0 then
   begin
   writeln; writeln('*** file ',readstr,' does not exist ***');
   writeln('*** you must restart the program ***');
   close(inputlist); close(outputfile); exit;
   end;
holding:=false;
writeln(outputfile, '$$1'); writeln('$$1');
writeln(outputfile, 'doc', doccnt:6); writeln('doc', doccnt:6);
writeln(outputfile, '$$2'); writeln('$$2');
inc(doccnt); linecnt:=0;
repeat
begin
   restart:
   if holding then
      begin
      holding:=false; inchar:=holdchar;
      end
   else
      read(inputfile, inchar);
   if ord(inchar)=9 {A TAB} then
      begin
      repeat
         inc(buffcnt); buff[buffcnt]:=' ';
      until (buffcnt mod 8=0);
      goto restart;
      end;

   wordprocscreen(inchar);
   if oddchar=1 then goto restart;
   inc(buffcnt);
   if inchar=' ' then lastblank:=buffcnt;
   if (not (ord(inchar) in [10, 13, 26])) then
      if buffcnt<=linelen then
         buff[buffcnt]:=inchar
      else
         begin
         buff[buffcnt]:=inchar;
         buff[0]:=chr(lastblank-1);
         if upcase(testrightjust)='Y' then
            rightjust(buff, lastblank-1)
         else
            begin
            writeln(outputfile, buff); writeln(buff);
            end;
         inc(linecnt);
         for i:=1 to buffcnt-lastblank do
            buff[i]:=buff[lastblank+i];
         buffcnt:=buffcnt-lastblank;
         lastblank:=0
         end
   else {then we have a carriage return(10), line feed(13), or EOF(26).}
      begin
      buff[0]:=chr(buffcnt-1);
      writeln(outputfile, buff); writeln(buff);
      buffcnt:=0; lastblank:=0;
      inc(linecnt);
      read(inputfile, holdchar);
```

Figure #15e - Continuation of the listing for CONVERT2.PAS

```
          while (ord(holdchar) in [10,13]) do
             begin
             inchar:=holdchar; read(inputfile, holdchar);
             writeln(outputfile); writeln;
             end;
          holding:=true;
          {
writeln('inchar= ',ord(inchar),' linecnt= ',linecnt);
writeln(outputfile,'***inchar*',ord(inchar),'*linecnt*',linecnt);
          }
          if not specialfile then
             if ((not (inchar=#26)) and (linecnt>2) and (holdchar<>#26)) then
                (don't want paragraphs with less than 6 lines)
                begin
                writeln(outputfile, '$$1'); writeln('$$1');
                writeln(outputfile, 'doc', doccnt:6); writeln('doc', doccnt:6);
                writeln(outputfile, '$$2'); writeln('$$2');
                linecnt:=0; inc(doccnt);
                end;
          end;


   end; until (inchar=#26);
   close(inputfile);
   if specialfile=true then specialfile:=false;
until eof(inputlist);
close(inputlist); close(outputfile);
writeln('Program finished.....SPIRIT text file created.');
end.
```