CGTM Number 194
June 1978

Revised:
November 1980
May 1982
September 1985

IDENTIFYING MISSPELLED WORDS IN COMPUTER READABLE DOCUMENTS

ROBERT C. BEACH
COMPUTATION RESEARCH GROUP
STANFORD LINEAR ACCELERATOR CENTER
STANFORD, CALIFORNIA 94305

TABLE OF CONTENTS

SECTION 1: INTRODUCTION

This document describes a program, running on an IBM computer, that may be used to check a computer readable document for spelling errors and a few other types of errors. The proofreading of a document for spelling errors is a very difficult thing to do; the eye invariably sees what the mind wants to see. This makes it very difficult for a person to proofread his own writings. However, a computer can be used to relieve people of part of this very tedious task.

The Word Scanning Program described here is similar to an earlier program prepared by John Steffani of the SLAC Computation Research Group. Unfortunately, that program was never documented and was completely destroyed in a disastrous head crash on a disk. A somewhat similar set of programs have been described by Scowen [Sco77]. There have also been examples of programs which attempt to correct spelling errors [Hen79, ISC79]. However, until I see a demonstration that proves otherwise, I cannot believe that these attempts can be very successful. A simple description of the Word Scanning Program is given in the remainder of this section and a complete description is given in Section 2.

This document also describes a second program, the Word List Utility Program which is used to maintain files containing lists of correctly (I hope) spelled words. The Word List Utility Program is described in detail in Section 3.

The word lists themselves are described in Section 4 and Section 5 describes some implementation problems.

SECTION 1.1: AN INTRODUCTION TO THE WORD SCANNING PROGRAM

The basic operation of the Word Scanning Program is quite simple. It reads a document and breaks it down into individual words. The words are translated so that all characters are upper case. Each word is then matched against an internal list of the 100 most common words in the English language. If a match is found the word is deleted. This will usually eliminate 40 to 50 percent of the words. The remaining words are sorted and compared with other external word lists. Matched words are again deleted. A cross-reference of all of the remaining words is then printed. At present, there are six external word lists available although only the first five of them are usually used. These word lists are:

1. A general vocabulary with more than 78,800 words.
2. A mathematical word list with more than 2800 words.
3. A computer science word list with more that 1900 words.
4. A physics word list with more than 2500 words.
5. A geographic word list with more than 1600 words.

6. An extended vocabulary with more than 7300 words.

Another thing the Word Scanning Program can do is identify "stutters". A stutter occurs when a word is repeated more than once when it really should appear only once. This error can be very difficult to find, especially when the word is small and the the first occurrence of the word is at the end of one line and the second occurrence is at the beginning of the next line [1].

It must be emphasized that this program is not a substitute for proofreading, it is only an aid. The program cannot find punctuation errors [2], grammatical errors [3], substitutions of one word for another [4], missing words [5], or any number of other kinds of errors.

SECTION 1.2: USING THE WORD SCANNING PROGRAM FROM VM/SP

An EXEC Program is available which allows a user to run the Word Scanning Program at a terminal. The EXEC Program is invoked by the commands:

```
GIME RCB 194
WORDSCAN
```

The WORDSCAN EXEC Program will ask a number of questions. The answers to these questions will be used to issue FILEDEF commands and load and execute the Word Scanning Program. The output of this program is written to a file on disk. This file should usually have a type of LISTING.

There are two things a user must be aware of when the WORDSCAN EXEC Program is invoked:

1. The format of the document to be scanned must be known. That is, the user will be asked to specify the columns within the records that contain the text to be scanned. Things are especially simple if the text is in columns 1 through 72 with a sequence field in columns 73 through 80.
2. A disk must be available to hold some temporary files. The amount of space needed on the disk is comparable to the amount of space occupied by the document to be scanned.

For a complete understanding of the questions being asked by the EXEC Program, the user should be familiar with most of the material in Section 2. The material in Section 2.2 is especially useful.

SECTION 2: THE WORD SCANNING PROGRAM

This program may be used to examine a document, break the document down into individual words, match these words with the words in a number of files, and finally, print a cross-reference or frequency count of the unmatched words. An optional stutter list, that is, a list of consecutive repeated words, may also be generated. The input may be in a number of different forms. It may, for example, be either the input or output from text processing programs like FORMAT, SCRIPT, or TEX. The input may also consist of FORTRAN, PL/1, etc. programs, in which case the executable code will be eliminated and only the comments will be retained.

The operation of the program is as follows:

1. The parameter list is scanned. The items in the parameter list can alter the normal operation of the program.
2. Each record on the input file is read and is usually processed as follows:
   A. If the input is of a special type, FORMAT input or a FORTRAN program for example, it is processed to eliminate the unwanted parts.
   B. The characters are translated so that all lower case alphabetic characters are changed to upper case and all non-alphabetic characters are changed to blanks.
   C. If stutter checking is activated, the words are examined to see if any stutters occur.
   D. The words are compared to an internal list of the 100 most common words in the English language and these words are eliminated. Words whose length is less than a minimum number of characters may also be eliminated.
   E. The remaining words are written to a temporary file along with an identification which will appear in the cross-reference listing. The identification may be the record count or a field within the record.
3. The words in the temporary file are sorted.
4. The sorted file is matched with the words in as many as eight files containing lists of words. Words which occur in any of these files are eliminated.
5. The remaining words are printed in a cross-reference or frequency listing. The cross-reference list shows the identification of all of the records containing the word.

At present, the program can only handle words up to 32 characters in length. If a word longer than this is encountered, it is truncated to 32 characters and the last character is changed to an asterisk when it is printed in the cross-reference or frequency listing.

The stutter listing should be used with care; many stutters in the English language are grammatically correct and make good sense [6]. This program can, however, pick out errors that have escaped numerous proofreadings.

SECTION 2.1: THE INPUT AND OUTPUT FILES

There is one required and eight optional input files, and one output file.

The input files are:

SYSDOC	A file containing the document or program to be examined. This file is, of course, required.
SYSWDL1	An optional word list.
SYSWDL2	An optional word list.
SYSWDL3	An optional word list.
SYSWDL4	An optional word list.
SYSWDL5	An optional word list.
SYSWDL6	An optional word list.
SYSWDL7	An optional word list.
SYSWDL8	An optional word list.

The output file is:

SYSPRINT	A print file which will contain the stutter list, if requested, and either a cross-reference listing or a frequency listing of the words.

The following temporary files are also needed:

SORTIN
SORTOUT

The LRECL value for these temporary files must be 48.

SECTION 2.2: THE PARAMETER LIST

The parameter list that may be passed to this program can contain a number of items which affect the operation of the program.

The parameter items controlling the assumptions to be made about the SYSDOC file are:

FORMAT	The input is assumed to be input to the FORMAT text processing program [Ber69, Ehr71]. FORMAT control cards and control items are eliminated from further processing.
SCRIPT	The input is assumed to be input to the SCRIPT text processing program [Wat78]. Most of the control items are eliminated from further processing. SCRIPT input is very difficult to analyze completely and this program only attempts to do a partial job; some control items may not be eliminated.
TEX	The input is assumed to be input to the TEX text processing program [Knu79]. Most of the control items are eliminated from further processing. TEX input is very difficult to analyze completely and this program only attempts to do a partial job; some control items may not be eliminated.
FORTCOM	The input is assumed to be a series of FORTRAN

programs [IBM65a and ANS78]. The executable code is eliminated and only the comments are retained.

PL1COM The input is assumed to be a series of PL/1 programs [IBM66]. The executable code is eliminated and only the comments are retained.

COBCOM The input is assumed to be a series of COBOL programs [IBM65b]. The executable code is eliminated and only the comments are retained.

ASMCOM The input is assumed to be a series of IBM 360/370 Assembler Language programs [IBM64]. The executable code is eliminated and only the comments are retained.

MORTCOM The input is assumed to be a series of MORTRAN programs [Coo75]. The executable code is eliminated and only the comments are retained.

At most, one of the above items should be given. The default, if none of the above is given, is to process all of the words in the input. The user should be careful that only syntactically correct input is supplied to this program. If, for example, the input is a PL/1 program with unbalanced comment or string delimiters, the output may be totally meaningless and no error message will be generated.

The parameter items controlling the part of the SYSDOC records to be used are:

FIELD=(M,N) This indicates that the field begins with the M-th character and ends with the N-th character. If M or N is negative, it means that the count is to be taken from the end of the record.

CARDFIELD This item is equivalent to FIELD=(1,72).

At most, one of the above items should be given. The default, if none of the above is given, is FIELD=(1,-1), that is, the entire record is used.

The parameters controlling the continuation of records from the SYSDOC file are:

CONTHYPHEN If the last non-blank character in a record is "-", then the hyphen is eliminated and the last word is concatenated with the first non-blank word in the next record. This item is useful when processing formatted documents that consist of one column and contain hyphenated words. If a document has been formatted into a multi-column format with hyphenation, this program will have difficulty with it. In that case, preprocessing with a text editor will probably be necessary.

CONTNOBLK If the last character of a record is not blank and the first character of the next record is also not blank, then the words are concatenated. This item is useful for processing FORMAT input in its compressed form.

At most, one of the above should be given. The default, if none of the above are given, is to not continue any records.

The parameters controlling the cross-reference identification are:

IDFIELD=(M,N) This indicates that the field begins with the M-th character and ends with the N-th character. If M or N is negative, it means that the count is to be taken from the end of the record. If more than eight characters are specified, only the first eight are used.

CARDID This item is equivalent to IDFIELD=(73,80).

At most, one of the above items should be given. The default, if none of the above is given, is to use the record count as the identification in the cross-reference listing.

The parameters controlling the translation of the records in the SYSDOC file are:

TRANSL=(K,L,M,N) K controls the translation of special characters, L the numerals, M the lower case letters, and N the upper case letters. Zero means no translation, one means translate the characters to blanks, M=2 means translate lower case to upper case, and N=2 means translate upper case to lower case. Unprintable characters are always translated to blanks.

KEEPNUMB This is an alternate to the TRANSL parameter. It indicates that numbers are not to be eliminated and is equivalent to TRANSL=(1,0,2,0).

The default, if neither of these items are given, is TRANSL=(1,1,2,0). This default value forces alphabetic characters into upper case and changes all other characters to blanks.

Parameter items controlling common and short word elimination are:

KEEPCOMN This indicates that the 100 most common English words are not to be eliminated. The default is to eliminate these words.

MINWORD=N This specifies the minimum word size to be processed. All words less than N characters are eliminated. The default is MINWORD=1, that is, no words are eliminated because of their length.

Some additional parameter items are:

FREQUENCY This causes the frequency count listing to be produced instead of the cross-reference listing.

STUTTER This causes a stutter listing to be produced. The default it to not produce the stutter listing.

STDCARD This indicates that the input is in standard card format and is equivalent to both FIELD=(1,72) and IDFIELD=(73,80).

NOIDCOMP This indicates that blanks are not to be compressed out of the identification field in the cross-reference listing. The default is to eliminate such blanks.

If this program detects any error in the parameter list, it indicates the nature of the problem in the printed listing and terminates. The punctuation is not critical but no error is tolerated in the keywords or numbers.

SECTION 2.3: THE WORDSCAN EXEC PROGRAM

A few additional notes may be helpful in using the WORDSCAN EXEC Program. The EXEC Program will lead you through a series of interactions and will construct the necessary FILEDEF commands and the parameter list. Any prompt may be answered with a question mark to obtain more information. The first few times that you use the EXEC Program you will probably want to get this extra information. After a few uses, you should rarely need this extra information.

As stated earlier, the most convenient way to have the document is in the "standard card" format. However, this is not always possible and the EXEC Program will ask for both the text field and the identification field if your document is not in the standard card format. For example, if your document is a print image file with carriage control characters in column 1, you could respond "2,-1" to the prompt requesting the scan field. This results in the item:

```
  FIELD=(2,-1)
```

being added to the parameter list. When you are asked for the identification field, you may reply with a simple carriage return to indicate that the record count is to be used as the identification.

Sometimes you may not want numerals translated to blanks. For example, when this document is checked, it is useful to retain PL1COM as one word and not two separate words or to obtain a full cross-reference of the SYSWDL1...SYSWDL8 file names. This can be done by supplying the additional parameter:

```
  TRANSL=(1,0,2,0)
```

or its equivalent:

```
  KEEPNUMB
```

when prompted.

If you want a cross-reference of all of the words in a document, you can add the additional parameter:

```
  KEEPCOMN
```

to the list and also delete all of the word lists when prompted.

SECTION 3: THE WORD LIST UTILITY PROGRAM

This program is a utility program that may be used to maintain word lists for the Word Scanning Program. This program can create a new word list, add words to an existing word list, or delete words from an existing word list. It may also generate a listing of a word list or convert a word list to card image format. A word list is a RECFM=VB file with each record consisting of a single word. The words are arranged alphabetically and each word normally consists only of upper case alphabetic characters. A data compression scheme is used whereby each word starts with a one byte count of the number of characters at the start of the word which are the same as the preceding word. Since many words are the same as the preceding word except for a suffix of "s", "ed", or "ing", this results in a substantial compression ratio. In addition, the first record of the word list is a title record which describes the word list. The title record is printed by the Word Scanning Program on its first page of output to identify each of the word lists being used.

The operation of the program is as follows:

1. The parameter list is scanned. The items in the parameter list can alter the normal operation of the program.
2. The deletion data, if it is given, is read and sorted.
3. The old word list is read and the words identified by the deletion data are eliminated. The remaining words are written to a temporary file.
4. The new words are read and translated so that all lower case alphabetic characters are changed to upper case and all non-alphabetic characters are changed to blanks. The resulting words are added to the temporary file.
5. The temporary file is sorted.
6. The sorted file is read, duplicate words are eliminated and the remaining words are written to the output files. Each word in the printed list is identified by its index number.

If the files containing the deletion data and the new words are both omitted, then steps 2 through 5 are all omitted, and in step 6, the words are read directly from the old word list.

At present, the program can only handle words up to 32 characters in length [7]. If the input contains a word longer than this, it is ignored.

SECTION 3.1: THE INPUT AND OUTPUT FILES

There are three input files and three output files. Any of these files, except the printed output, are optional; the program uses those files that are made available to it.

The input files are:

SYSOLD An existing word list.

SYSDEL A file containing the index numbers of the words in SYSOLD that are to be deleted.

SYSWDS A file containing words to be added to the new word list.

While all of these files are optional, it does not make much sense to try to run the program with both SYSOLD and SYSWDS missing.

The output files are:

SYSNEW A new word list.

SYSPRINT A print file that will contain a listing of the word list as well as other information. The listing gives the index number of each word in the word list. If this index number is followed by an asterisk, it means that the word came from SYSWDS. If the index number is followed by a dollar sign, it means that the word was duplicated in the input files.

SYSPCH A card image file that will contain the words in the new word list. The words will be packed into the first 72 characters of each record. The LRECL value for this file must be 80.

If either SYSDEL or SYSWDS is given, then the following temporary files are also needed:

SORTIN

SORTOUT

The LRECL value for these temporary files must be 36.

SECTION 3.2: THE PARAMETER LIST

The parameter list that may be passed to this program can contain a number of items which affect the operation of the program.

The parameter item controlling the title of the SYSNEW word list is:

TITLE="title" This item will be the identification of the SYSNEW file. The title, enclosed in quotation marks, should not contain any special characters and should be at most 30 characters long.

The default, if this item is not given, is TITLE="UNIDENTIFIED WORD LIST". Note that this identification must be supplied whenever a SYSNEW word list is created; the identification is not transferred from SYSOLD to SYSNEW and it is not put into SYSPCH.

The parameter items controlling the part of the SYSWDS records to be used are:

FIELD=(M,N) This indicates that the field begins with the M-th character and ends with the N-th character. If M or

N is negative, it means that the count is to be taken from the end of the record.

CARDFIELD This item is equivalent to FIELD=(1,72).

At most, one of the above items should be given. The default, if none of the above is given, is FIELD=(1,-1), that is, the entire record is used.

The parameters controlling the part of the SYSDEL records to be used are:

DELFIELD=(M,N) This item works the same as FIELD=(M,N) except that it applies to SYSDEL instead of SYSWDS.

DELCARDFIELD This is equivalent to DELFIELD=(1,72).

At most, one of the above should be given. The default is DELFIELD=(1,-1).

A parameter controlling the translation of the records in the SYSWDS file is:

TRANSL=(K,L,M,N) K controls the translation of special characters, L the numerals, M the lower case letters, and N the upper case letters. Zero means no translation, one means translate the characters to blanks, M=2 means translate lower case to upper case, and N=2 means translate upper case to lower case. Unprintable characters are always translated to blanks.

The default, if this item is not given, is TRANSL=(1,1,2,0). This default value forces alphabetic characters into upper case and changes all other characters to blanks. The records from SYSDEL are also translated but this translate table is not under the control of the user. The effect of the translate table is (1,0,1,1) which translates everything except numbers to blanks.

The parameters which control the output listing are:

NOPRINT This item suppresses the printing of the new word list. It can be useful when this program is being used to simply move a word list from one disk to another.

NOOLDPRINT This item suppresses the printing of the words in the new word list that came from SYSOLD. It can be useful when checking out some changes that are to be made to a word list.

NODELPRINT This item suppresses the printing of the words that were deleted from the old word list.

The default, if the above are not given, is to print both the deleted words and the new word list in their entirety.

If this program detects any error in the parameter list, it indicates the nature of the problem in the printed listing and terminates. The punctuation is not critical but no error is tolerated in the keywords or numbers.

SECTION 3.3: THE WORDUTIL EXEC PROGRAM

An EXEC Program, named WORDUTIL, is available to run the Word List Utility Program. It works in much the same way that the WORDSCAN EXEC Program works; it prompts the user for information, issues FILEDEF commands, constructs a parameter list, and executes the Word List Utility Program.

The first interaction the user has with the WORDUTIL EXEC Program is to enter the operation to be performed. The four available operations are General, Check, Run, and Move. The last three operations are special ones used to maintain the standard word lists and should not be used by the general user. The first operation may be used by anyone to maintain their own specialized word lists.

When trying to update a word list, it is better to proceed in two steps. The first step is to create the new word list with a different name. The output listing can then be examined to see that everything went correctly. If the new word list is not correct, it can be deleted and the creation job run again. When the new word list is correct, the second step consists of deleting the original word list and renaming the new one. If you just wish to verify that some new words and deletions are correct, a convenient way to do this is to add the parameter:

NOOLDPRINT

to the parameter list and run the Word List Utility Program without the SYSNEW output file. The result of such a job will be a list of all changes but no new file. This check job will run faster than the original job because of the reduced printing and the elimination of the work done in generating the new file.

Finally, it is worth noting that a listing of an existing word list can be easily obtained. If the only input is an old word list and the only output is the print file, then the program simply generates a listing of the input file. Be careful about generating listings of the general vocabulary however; the listing of that file takes nearly 500 pages.

SECTION 4: THE WORD LISTS

The word lists are of two distinct types. One list consists of the 100 most common words in the English language and is built into the Word Scanning Program. The other word lists are files which reside on disk and are maintained with the Word List Utility Program. One of the disk resident word lists contains a general vocabulary while the others contain technical terms from various fields. There is also an extended vocabulary that contains unusual or seldom used words.

Users of this program will undoubtedly find many words missing from the word lists. I would appreciate it if users would generate lists of missing words and communicate them to me. I will add these words to the word lists so that future uses of the programs will be more effective.

In general, I have tried to keep buzz words of obscure meaning and acronyms out of these word lists unless their use has become common and relatively stable. Thus the computer science word list contains the word "software" but does not contain the word "nibble" (half a byte). Proper names have also been excluded from the word lists unless something has been named after the individual or place. Thus the mathematical word list contains the word "bessel" because Bessel functions are well known. Finally, the user must remember that these word lists are American-English and not English-English; the general vocabulary contains the word "center" but not the word "centre".

There is a certain amount of duplication in the various word lists. First, the 100 most common words in the English language are also in the general vocabulary word list. Second, the technical word lists contain words that are also in the general vocabulary but are used with a different or more precise meaning within the technical field. An example is the word "compact" which is in the mathematical word list in addition to the general vocabulary because of its specialized meaning in Topology. There is also a certain amount of duplication between the technical word lists.

SECTION 4.1: THE 100 MOST COMMON WORDS IN ENGLISH

This list was obtained from the works of Kucera and Francis [Kuc67]. Kucera and Francis analyzed 500 fragments of text, each fragment containing approximately 2000 words. The fragments of text were all originally published in 1961 and were distributed among 15 categories including newspaper articles, technical writing, and fiction. In all, the text contained 1,014,232 words.

One of the simpler things that Kucera and Francis did with this body of text was to generate a rank list of all 50,406 distinct words in the text. The list in this section is the first part of their rank list. In the entire body of text, these 100 words constituted 47.419 percent of the total words. This percentage appears to be somewhat lower in technical writings.

The 100 most common words in the English language, arranged in order with the most common first, are:

1. the	21. this	41. we	61. can	81. man
2. of	22. had	42. him	62. only	82. me
3. and	23. not	43. been	63. other	83. even
4. to	24. are	44. has	64. new	84. most
5. a	25. but	45. when	65. some	85. made
6. in	26. from	46. who	66. could	86. after
7. that	27. or	47. will	67. time	87. also
8. is	28. have	48. more	68. these	88. did
9. was	29. an	49. no	69. two	89. many
10. he	30. they	50. if	70. may	90. before
11. for	31. which	51. out	71. then	91. must
12. it	32. one	52. so	72. do	92. through
13. with	33. you	53. said	73. first	93. back
14. as	34. were	54. what	74. any	94. years
15. his	35. her	55. up	75. my	95. where
16. on	36. all	56. its	76. now	96. much
17. be	37. she	57. about	77. such	97. your
18. at	38. there	58. into	78. like	98. way
19. by	39. would	59. than	79. our	99. well
20. I	40. their	60. them	80. over	100. down

The purpose of incorporating this list into the Word Scanning Program is to enable the program to quickly eliminate as many words as possible from further processing. Since these words are eliminated before the remaining words are sorted, large savings in execution time can accrue.

SECTION 4.2: THE GENERAL VOCABULARY

The original source of the words in this word list was a tape that was a byproduct of some work done by Hanna et al. [Han66]. This tape contained approximately 17,000 words. Normally the tape contained only a single root word and not all of the derived words so these derived words had to be added to the word list. Thus the original tape contained the word "activate" while the final word list also contains "activates", "activated", "activating", and "activation". Later, a second source of words was merged into this file. The result is a word list with more than 78,800 words [8] with an average length of 8.90 characters.

Merriam-Webster's dictionary [MER84] contains multiple spellings for many words. For example, it includes both "canceled" and "cancelled". The first form is given first in that dictionary and only that form is in this word list. The second form of these words is in the extended vocabulary.

SECTION 4.3: THE WORD LIST OF MATHEMATICAL TERMS

At present, this word list contains over 2800 words with an average length of 9.37 characters. The principal source of words for this list was the James's Dictionary [Jam64].

SECTION 4.4: THE WORD LIST OF COMPUTER SCIENCE TERMS

At present, this word list contains over 1900 words with an average length of 8.28 characters. The principal source of words for this list were Meek's Glossary [Mee72] and Weik's Dictionary [Wei69].

SECTION 4.5: THE WORD LIST OF PHYSICS TERMS

At present, this word list contains over 2500 words with an average length of 9.40 characters. The principal source of words for this list was Elsevier's Dictionary [Cla62].

SECTION 4.6: THE WORD LIST OF GEOGRAPHIC TERMS

At present, this word list contains over 1600 words with an average length of 7.08 characters. The principal source of words for this list was the World Almanac [Del79]. The words selected were the names of the states of the United States and the provinces of Canada; the names of their capital cities and all cities with a population of more than 100,000; the names of all of the countries in the world; the names of their capital cities and all cities with a population of more than 1,000,000; and the names of the continents, major rivers, lakes, and oceans. The list also contains words of the form "american", "americans", and "scottish". In addition the list contains most of the geographic names in the SLAC Preprints List [SLA77].

SECTION 4.7: THE EXTENDED VOCABULARY

This word list contains unusual or seldom used words. It should not normally be used when running the Word Scanning Program because a misspelling of a simple word could find a match in this list. This word list also contains the secondary spellings of words as defined by Merriam-Webster's dictionary [MER84]. At present, this word list contains over 7300 words with an average length of 8.54 characters.

SECTION 5: IMPLEMENTATION NOTES

The Word Scanning Program and the Word List Utility Program are PL/1 programs that were originally written to run on an IBM/370 Model 168 running OS/VS2 Release 1.6. This document describes the current version of the program which runs on an IBM 3081 running VM/SP Release 1.

The conversion of these programs from SVS to VM posed many problems. PL/1 under VM is a butchered-up emasculated subset of PL/1 under SVS. In fact, these programs will not run under an unmodified VM environment because both programs invoke the sort package. In an effort to reduce this and any other future conversion problems, all possible Assembler Language modules were converted to PL/1. The only Assembler Language module left is the one used to determine the execution time of the program.

The earlier versions had Assembler Language modules to do the translation of the input records and to do the comparison of character strings. These modules were not written by accident; they greatly speed up the execution of the programs. In the case of the translation functions, the PL/1 compiler does more work than required and unnecessarily allocates temporary storage.

The problem with the character string comparison is more fundamental, and, in my opinion, shows that the PL/1 Compiler in use does not deserve to be called an "Optimizing" Compiler. In the Word Scanning Program, the basic operation in matching the sorted words from the document with the word lists is to determine the relation between the current document word and the current word list word. This is done in PL/1 with two of the following statements:

```
IF CUR_DOC_WORD < CUR_WLW_WORD THEN ...
IF CUR_DOC_WORD = CUR_WLW_WORD THEN ...
IF CUR_DOC_WORD > CUR_WLW_WORD THEN ...
```

The PL/1 code for each comparison consists of the generation of an argument list and the calling of a subroutine followed by a BL, BE, or BH instruction. When more than one of the above comparisons must be made, it is clear that the subroutine need only be called once with more than one of the comparison instructions following the call. Unfortunately, however the above statements are ordered or nested, PL/1 never consolidates the subroutine calls. The result is that the most time consuming part of the Word Scanning Program is massively inefficient.

Another problem with VM is the terrible support it gives to I/O by PL/1 and FORTRAN programs. Under SVS, the OS macros used by these programs were executed directly. Under VM, these I/O macros are interpreted and converted to VM macros at execution time. The execution of these VM I/O macros then involves another level of interpretation. Simple tests show that I/O bound programs use, according to system supplied statistics, between 10 and 100 times more CPU cycles under VM than under SVS.

As a result of the abysmally deficient support of PL/1 under VM and of the sloppiness of the PL/1 "Optimizing" Compiler itself, the VM version runs only slightly faster than the SVS version even though the IBM 3081 is allegedly four times faster than the IBM/370 Model 168.

FOOTNOTES AND OTHER OBSCURE THOUGHTS

This section contains all of the footnotes that have been referenced in this document.

[1] If you think stutters are easy to spot, read the footnoted sentence again very carefully.

[2] To see how punctuation can alter the meaning of a sentence, consider: "Woman without her man would be a savage" and "Woman: without her, man would be a savage".

[3] The book Thy Neighbor's Wife [Tal80] has achieved notoriety as one of the most poorly edited books to have been published in recent years. For example, it contains the sentence: "After completing high school in 1949, his sister wrote that she had arranged for him an appointment to Annapolis". It is of course the brother, not the sister, who completed high school in 1949.

[4] In late 1977, The Standard, an English-language newspaper published in Nairobi Kenya referred to John Vorster, the Prime Minister of South Africa, as a "white friend". The next day the paper ran a correction saying that the word they had intended to use was "fiend"; the correction did not fully stem the tide of letters, telegrams, and phone calls.

[5] In 1631, a Bible was published in England with the word "not" omitted from the seventh commandment. The good people of England were thus told "Thou shalt commit adultery". This Bible became known as the "Wicked Bible" and England became known as "Merrie Olde England".

[6] Consider a situation in which a teacher is returning a test on English grammar to two students, John and Jim. The following situation could then apply:

> John, where Jim had had "had", had had "had had".
> "Had had" had had the instructors approval.

Eleven identical words in a row!

[7] This restriction should not present any serious problem. The longest word in The Oxford English Dictionary [Mur82] is the 29 letter "floccinaucinihilipilification" [9]. This restriction does, however, means that the 34 letter word "supercalifragilisticexpialidocious" can not be added to the word lists.

[8] If you think that this is a large number, remember that The Oxford English Dictionary [Mur82] contains 414,825 words, and this does not count all derivatives of the root words.

[9] Meaning, of course, the action or habit of estimating something as worthless.

REFERENCES

This section contains a list of all of the publications that have been referenced in this document.

[ANS78] American National Standard, Programming Language, FORTRAN, American National Standards Institute, Inc., Document Number ANS X3.9-1978 (1978).

[Ber69] Gerald M. Berns, Description of FORMAT, a Text-Processing Program, Communications of the Association for Computing Machinery, Volume 12, Number 3 (March 1969), pages 141-146.

[Cla62] W. E. Clason, Elsevier's Dictionary of General Physics, Elsevier Publishing Company, Amsterdam Netherlands and New York New York, (1962).

[Coo75] A. James Cook and L. J. Shustek, A User's Guide to MORTRAN2, Stanford Linear Accelerator Center, Stanford California 94305, CGTM Number 165 (February 1975, Revised June 1975).

[Del79] George E. Delury (editor), The World Almanac and Book of Facts, 1979 Newspaper Enterprise Association, New York New York 10017, (1979).

[Ehr71] John R. Ehrman and Gerald M. Berns, FORMAT, A Text Processing Program, Stanford Linear Accelerator Center, Stanford California 94305, SLAC Report Number 135 (July 1971).

[Han66] Paul R. Hanna, Jean S. Hanna, Richard E. Hodges, and Edwin H. Rudorf Jr., Phoneme-Grapheme Correspondences as Cues to Spelling Improvement, U. S. Government Printing Office, Report Number OE-32008, (1966).

[Hen79] Tom Henkel, Software Detects, Fixes Spelling Errors in Copy, ComputerWorld, (27 August 1979), page 18. This article describes a project done by the Chemical Abstracts Service under a $153,000 National Science Foundation grant.

[IBM64] IBM System/360 Operating System, Assembler Language, International Business Machines Corporation, Form Number C28-6514 (1964).

[IBM65a] IBM System/360 Fortran IV Language, International Business Machines Corporation, Form Number C28-6515 (1965).

[IBM65b] IBM System/360 Operating System, COBOL Language, International Business Machines Corporation, Form Number GC28-6516 (1965).

[IBM66] IBM System/360 Operating System, PL/1 (F), Language Reference Manual, International Business Machines Corporation, Form Number GC28-8201 (1966).

[ISC79] Advertisement in ComputerWorld, (17 September 1979), page 21. The SPROOF program is available on a 10 day trial basis for $800 from Intelligent Software Company, 276 Harris Avenue, Needham Massachusetts 02192.

[Jam64] Glenn James and Robert C. James, Mathematics Dictionary, D. Van Nostrand Company Inc., Princeton New Jersey, (1964).

[Knu79] Donald E. Knuth, TEX and METAFONT, New Directions in Typesetting, American Mathematical Society, (1979).

[Kuc67] Henry Kucera and W. Nelson Francis, Computational Analysis of Present-Day American English, Brown University Press, Providence Rhode Island, (1967).

[Mee72] C. L. Meek, Glossary of Computing Terminology, CCM Information Corporation, New York New York 10022, (1972).

[MER84] Webster's Ninth New Collegiate Dictionary, Merriam-Webster Inc., Springfield Massachusetts, (1984).

[Mur82] James A. H. Murray, Henry Bradley, W. A. Craigie, and C. T. Onions (editors), The Oxford English Dictionary, Oxford University Press, Oxford England, (originally published in 1882 through 1928 in numerous small sections, republished in 1933 in 10 Volumes, and reprinted in 1961 in 12 Volumes plus a Supplement).

[Sco77] R. S. Scowen, Some Aids for Program Documentation, Software Practice and Experience, Volume 7, (1977), pages 779-792.

[SLA77] Preprints in Particles and Fields, 1229 Addresses, Stanford Linear Accelerator Center, Stanford California 94305, (December 1977).

[Tal80] Gay Talese, Thy Neighbor's Wife, Doubleday, Garden City New York, (1980).

[Wat78] Waterloo SCRIPT Reference Manual, University of Waterloo, Waterloo Ontario Canada, (January 1978).

[Wei69] Martin H. Weik, Standard Dictionary of Computers and Information Processing, Hayden Book Company Inc., New York New York, (1969).