DOCUMENT RESUME

ED 108 626 IR 002 142

AUTHOR Porch, Ann

TITLE Module Design Document: Scan Module; L.A.P. Version I.

INSTITUTION Southwest Regional Laboratory for Educational Research and Development, Los Alamitos, Calif.

REPORT NO SWRL-TN-5-72-27

PUB DATE 24 Apr 72

NOTE 7p.

EDRS PRICE MF-$0.76 HC-$1.58 PLUS POSTAGE

DESCRIPTORS *Computer Programs; Content Analysis; Design; Electronic Data Processing; Language Patterns; *Language Research; Languages; Language Styles; Language Usage; *Morphology (Languages); *Specifications; *Systems Development

IDENTIFIERS Analytical Computer Packages; Computer Software Specifications; *Language Analysis Package; LAP; Scan Module; Stem Analysis

ABSTRACT

A series of computer programs and routines intended to assist researchers in the analysis of language usage--with a power comparable to statistical computer routines--was developed by the Southwest Regional Laboratory (SWRL). This document is one of a series that describes the design specifications for the individual modules which comprise the Language Analysis Package (LAP). The Scan Module presented here will read input text, divide it into words, and check for special characters that the user indicates as identifiers to initiate further processing procedures. (DGC)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. nevertheless, items of marginal *
* reproducibility are ofter encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

I R

ED108626

SWRL

SOUTHWEST REGIONAL LABORATORY TECHNICAL NOTE

DATE: 4/24/72

NO: TN 5-72-27

TITLE: MODULE DESIGN DOCUMENT: SCAN MODULE - L.A.P. VERSION I

AUTHOR: Ann Porch

ABSTRACT

This is one of a series of technical design specifications for individual modules in the Language Analysis Package (L.A.P.).

The Scan Module will read input text, divide it into words, and check for special characters indicating further action needed.

U S DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY .

IR 002 142

This document is intended for internal staff distribution and use. Permission to reprint or quote from this working document, wholly or in part, should be obtained from SWRL, 4665 Lampson Avenue, Los Alamitos, California 90720

MODULE DESIGN DOCUMENT: SCAN MODULE - L.A.P. VERSION I

The following document will serve as design specifications for programming. It is one of a series of technical design specifications for individual modules of the Language Analysis Package (L.A.P.).[1] The section of the system design to which it is related is 6.1.0.

Program Objective

The Scan Module will function as a "front end" for several of the processing modules (KWIC, FREQ, etc.). It will be heavily used, and since it will perform its decision making function by scanning character by character through the input text, special attention will be paid to problems of optimization. Its objectives are three-fold: it will read in data as necessary from the input stream; it will find and return to the calling program the beginning and end points of a word; and it will determine if the delimiter(s) following a word is a "special" character which signals the necessity of some kind of special handling involving an additional subroutine call.

Constraints and Limitations

There will be no constraints or limitations for input data for the SCAN Module.

Options and Defaults

Options and defaults for the SCAN Module will be as follows:

- User specified input record length. Default = 72

[1] See Porch, Ann. TM 5-72-06, "Language Analysis Package (L.A.P.) Version I System Design" for an overview of the entire package.

- User specified "special" characters. Default = "@"
- User specified word definition. Default = alphabet, apostrophe, hyphen and numbers considered as valid word parts.
- User specified EOF indicator. Default = "$$"
- User specified input ID (up to 3 characters. To be flagged with preceding special characters). Default = date
- User specified input data conventions. Default = stream
- Calling program parameter requirements, (i.e., no special character check). Default = position of word beginning and end; special character check

Data File Specifications

Input. Input data will be considered as a stream of characters. No notice will be taken of record boundaries. Data which has been prepared utilizing other conventions can be handled by use of the Translate Module.

Output. Output data will be in the form of parameters passed to the calling program. The main parameters passed will be the following:

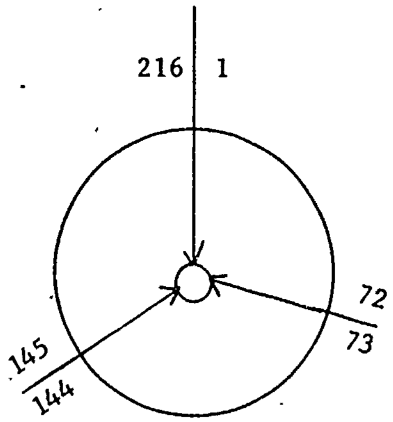
- The array containing three records of input data.
- The array position of the beginning and ending point of the word (or if the calling program requires it, the full word itself).
- A bit flag indicating whether or not the delimiter(s) following the word is a "special" character requiring an additional subroutine call by the program calling SCAN.

ERIC
Full Text Provided by ERIC

Significant Algorithms

There are two significant algorithms connected with the SCAN Module. They are: (1) The Three Record, Circular Read Algorithm, and (2) The Word Finding Algorithm.

Three Record, Circular Read Algorithm. An array will be set up with a length equivalent to three times the length of the input record. Using Record I/O, records will be read into position 1, 2, or 3 (see diagram below) whenever the array pointer reaches a position in the array that is equivalent to a record boundary. The diagram shows a three record array for 72 character records. The array will be initialized to blanks. The first read-in will fill positions 73-144; the second read-in will fill in positions 145-216. At this point, the array pointer will be set with a value of 73 and scanning will begin to determine word boundaries. Each character will be tested in turn and the pointer value incremented after the test. When the pointer value equals 145 (MOD (Pointer value, record length) = 1) read-in will fill positions 1-72. Likewise, when



the pointer value equals 217 (MOD (pointer value, record length) = 1), read-in will fill positions 73-144 (note: the pointer value will be set to 1 any time it equals three times the record length plus one). Read-in will fill positions 145-216 when the pointer equals 73 (MOD (pointer value, record length) = 1). In the diagram, the arrows are used to show which section of the array will be filled at each record boundary.

The Three Record, Circular Read Algorithm will provide highly efficient data handling in the following ways:

- Record I/O rather than Stream I/O can be used.
- Input data need never be relocated into another memory storage area even for output. Consequently less storage and execution time will be needed since no data copying is required. All manipulations, such as finding contexts for KWIC's, can be done simply by finding relative positions within the original array, and printing the appropriate section of the array.

Word Finding Algorithm. Using the three record array described above, the following actions will be taken in order:

- Loop through delimiters (testing for special character, apostrophe and hyphen) to find the beginning of the word by testing character for< 'A'.
- Set the value for the variable associated with word beginning
- Continue incrementing pointer as long as the character is not< 'A'.
- When the next delimiter is reached, set value for the variable associated with word ending (pointer - 1).

ERIC

The Word Finding Algorithm will provide efficient data handling in the following ways:

- No utility function calls are necessary (SUBSTR, LENGTH, INDEX, etc.).
- In most cases a single conditional test will clearly identify the character as either a word builder or a delimiter (worst case is 4 tests).
- Tests that will have to be made on every character in the input stream can be optimally ordered to reflect frequency of the condition being fulfilled.

Significant Variables

There are two significant variables in the SCAN Module. They are: (1) Three Record Array, and (2) Array with Position of Beginning of Word and Position of End of Word.

Error and Other Messages

There are no significant error or other messages associated with the SCAN Module.

Called by and/or Calls

The SCAN Module can be called by the following other modules:

- CONTROL (3.2.0)

The SCAN Module will call no other Modules. It will call the following subroutines:

- TEXT_BRK
- SPECIAL

ERIC Full Text Provided by ERIC