ELLIOTT 900

Volume 2: PROGRAMMING INFORMATION

Part 7: MAGNETIC TAPE ROUTINES

Section 1: INTRODUCTION TO TAPE ROUTINES

CONTENTS

July, 1968.

Printed in England by
Engineering Unit, English Electric Computers Ltd.

Chapter 1: INTRODUCTION

1.1 General

The 903 Program Library will contain a set of routines for using Magnetic Tape. It is strongly recommended that any other 903 program making any use of magnetic tapes should use these routines, and not address any instruction directly to the tape hardware. Where it is impossible to use the standard routines intact due to special requirements of on-line systems the use of the tapes should follow the same principles. The routines handle error recovery, bad patches on tape, etc., and the hardware and standard software depends on each others characteristics for consistent and reliable operation.

A set of standard Data Formats are used by the software. The data formats on the tape are described in Chapter 2. SIR programs which access the tapes pass information by means of blocks held in core store, known as file tables, described in Chapter 3.

1.2 Routines Provided

(a) Basic programs for handling data at block and record level:

Open file, close file, Find and read block, Write block, Get Record, Put Record. These are described in Sections 2.7.2 to 2.7.8.

(b) Utility Routines

Initialise file, Dump program, retrieve program, list contents of file. These are described in Section 2.7.9.

(c) High level Language Facilities

ALGOL and FORTRAN routines for handling data at block level. These are described in Sections 2.1.4 and 2.1.5 respectively.
Elementary operating system.

Chapter 2: DATA FORMATS ON TAPE

All the standard software routines will read and write tapes according to the formats described below.

2.1 Blocks

The natural unit of information on magnetic tape is the block. written onto or read from the tape by hardware in one continuous operation. Data is recorded on the tape in 6-bit characters (with parity in the seventh track) at 200 characters per inch for the length of a block. Between each block there is an inter-block gap of 0.75 inches nominal. Odd parity is used by all the standard software routines.

From the programmer's viewpoint blocks consist of an integral number of 18-bit words, of which the first two words are reserved for use by the software. Blocks may be of any length from 5 to 2047 words, the lengths of the block is under the programmer's control, and block lengths may vary at random along a tape. A block must contain some non-zero words, blocks consisting entirely of zeros may be ignored on reading.

2.2 Summary of Tape Operation

Data is always written or read while the tape is moving in the forward direction, i.e. from the tape reel on the right to the take-up reel on the left. Reverse movement can be obtained by a backspace, over one block, or a fast rewind to the beginning of the tape. The beginning of the tape, from the viewpoint of the program, is the load point, indicated by a reflective marker on the tape. When a reel is first loaded it is always positioned on the load point before program use. The end of tape is indicated by another reflective marker, which causes a hardware warning when the physical end of tape is within 7 metres of the read and write heads. The standard software only allows one block to be written completely beyond this point, to guard against a program unwinding the tape completely from a reel.

Blocks must be written in sequence along the tape, starting from the load point. A block should only be written if it follows the load point, or another block just written or read, or if it replaces a block just backspaced over.

Any information beyond the block last written should be considered to be lost, and an attempt to read past this block may lead to errors.

2.3 Block Numbering and Indicators

The first two words of each block are reserved for software use. When a block is written a block number is inserted in the first word and a block length, plus check and indicator bits, in the second word.

The block numbers of each block are 18-bit integers which increase serially along the tape, starting with +1 for the header block, which is immediately after the load point. Thus each block on the tape has a unique number. This is essential for proper error recovery, and is made use of by the QMFIND routine.

The second word of a block contains:

Bit 18 an indicator which is one if the block is a header or label.

Bits 17 to 13 contain a check character, formed from the block number. The read block routine checks this character, to ensure that the block is a valid one, written by standard software. On exit from the read routine the block in core store has bits 17 to 13 set to zero.

Bits 12 to 1 contain the block length as a positive integer in the range 0 to 2047 (Lengths less than 5 are illegal). The block length is the total number of 18-bit words in a block including the first two words.

2.4 Layout of Blocks on a Tape

Block 1, following the load point, must be a header block (see 2.7) written by the Open File or Initialise routines. At the end of any program run, and particularly when a tape is unloaded, the last significant block on the tape must be an EOF block (see 2.8.1). Between the header block and the EOF or EOV block there may be any number of blocks (or no blocks) written under the control of the user.

These intervening blocks may be in any format specified by the user (with the block number, etc. written by standard software). However, the user(particularly where there is more than one type of data on a tape), may include label blocks (see 2.8) at the start of each section of data.

2.5 Files

A file as the general name given to a quantity of data contained in several blocks on the tape. It is convenient to have one file per tape reel for a majority of applications. The header block on a tape contains the file name. 900 Software does not allow more than one file on a single tape reel, but this restriction can easily be overcome by splitting the information on a reel into a series of sub-files, each preceded by a label giving the sub-file name. The user may specify a file which uses more than one reel; the data formats allow for files extending over any integral number of reels, but each file must start of the beginning of a reel. If reel numbering is used it is the user's responsibility.

2.6 Record Structure

The user writing in SIR may handle data in units of one block at a time, or one record at a time. For 900 Software a record is regarded as a sub-unit of a block. A record consists of a integral number of words ≥ 2 Typical record lengths would range from 2 to 40 words, while typical block lengths range from 100 to 1000 words.

Records may be 'fixed-length' or 'variable length'. If 'fixed-length' then all the records on a file, or a sub-section of a file between label blocks, must have the same length. If'variable length', the individual records may be of any length from two words upwards, but a word is added to the beginning of the record, containing an integer giving the total length of the record.

The first word of a fixed length record must not be zero.

For efficient use of the tape with fixed length records, the number of words in a block should be an exact multiple of the number of words in a record (plus two words for block number and length).

A file may consist of individual sections with data in block units of various kinds, fixed length records and variable length records, but in this case there must be label blocks in between each section.

2.7 Header Blocks

A header block is the first block on a magnetic tape reel. It identifies the file on that reel and contains space for additional information on the file. Not all this information is used by the standard software at present, but it specified for possible future use.

All the standard software requires that a header block of the correct format is written on a tape before it can be used. The only exception to this rule is MTINIT, the tape initialising routine. A program can only obtain access to a file if the name of that file (held in the header block) is supplied by the user.

Header block layout,

(Note that SIR conventions are used; e.g. word 5; is the 6th word of the block).

Word number (relative position in block)	Contents	Example
0;	Block Number	+1
1;	/0 n (n=number of words in block)	/0 28
2; to 5;	Name of file, 12 characters in SIR internal code, ending with spaces (zeros) if not 12 characters long.	£NAM £E12 £345 £6
6;	Serial Number	+9876
7;	Generation Number	+0
8;	Reel Number for multi-reel file (zero if file on one reel only)	+0
9;	Creation date (zero if none)	+0
10;	Expiry data (zero if none)	+0
11; to 14;	Reserved	>4
15;	Zero if no record structure Negative if variable length records, Positive integer if number of words in fixed length record.	+0
16;	Reserved	+0
17;	If negative file can only be opened for reading or updating, it cannot be opened for writing.	+0
18; to 27;	Owners optional information If word 18 is zero the other words are undefined.	£JOH £NSM £ITH >7

The dates in words 9; and 10; are stored in the form:

Bits 16 to 10 :	Year number - 1900
Bits 9 to 6 :	Month number
Bits 5 to 1 :	Day number in month.

e.g. 1st January 1968 (1/1/68) would be +34849.

2.8 Label Blocks

Label Blocks are used to divide a file on Magnetic Tape into sections. In general the contents of a label block gives information about the data blocks on the section of tape following. A label block is indicated by a second word of the block being negative (i.e. bit 18 = 1). Special cases of label blocks are header blocks (2.7) and end of tape blocks (2.8.1).

2.8.1 End of Tape Blocks (EOF)

Whenever a file is written, the last significant data block on the tape should be followed by an end of tape label block. This will be written by using the Close File subroutine.

The only form of end of tape label allowed by the standard software is an EOF block with the following format:

Word 0;	Block number
Word 1;	/0 n (n = number of words in block)
Word 2;	EOF (in SIR internal code £EOF)
Words 3; to 5;	Content unspecified, reserved for future use.

2.8.2 Users Labels

The user may freely introduce label blocks into his files, keeping to the following format:

Word 0;	Block number
Word 1;	/0 n (number of words in block)
Words 2; and 3;	Any alphanumeric name, in SIR internal code, with the restrictions detailed below.
Words 4; to 27;	Any information inserted by the user, in any format. Label blocks should not normally be more than 28 words long.

The restrictions on the name are

(a) The first character must be a letter, thus making word 2; negative.

(b) The name must not start with the first two letters EO, as this indicates an end of tape label.

2.8.3 Data Labels

Data labels are a special type of label block, which may be used by standard software routines. They are distinguished from user's labels by the contents of word 2; being positive. Their format is as follows:

Word 0;	Block number
Word 1;	/0 n (number of words on block)
Word 2;	Data type number (positive integer)
Words 3; and 4;	Data name, 6 alphanumeric characters.
Words 5; onwards	Format depends on data type number.

The data type numbers indicate the type of data following. The numbers allocated so far are:

+1	Program dump
+2 +3 +4	Reserved for future use
+5	Mnemonic program
+6	Fixed length records
+7	Variable length records

If data type = 6 (fixed length records) then word 5; holds the number of words in a record, as a positive integer.

The user need only concern himself with data labels if he specifically arranges for his file to contain a mixture of different data types, e.g. he uses MSDUMP to dump a program on his own file.

If a file consists of different sections with different types of record structure data labels type 6 or 7 may be inserted, but their use is optional.

2.8.4 Action on Detecting a Label

In general an individual program which reads a data file should detect the presence of a label by checking the second word (word 1;) of the block read into core store. If a label is detected, the program should check whether its type is of interest at that stage. If it is not, blocks should be read and ignored until the next label is detected. If the label is a EOF indicator no more blocks should be read.

Chapter 3: FILE TABLE FORMAT

All SIR programs using the magnetic tape must reserve one or more areas of store, known as file tables. In general there will be one table for each tape handler used. The methods of using the file table are specified in individual program descriptions; the complete format is specified below.

Word 0; Handler number (a positive integer in the range 0 to 3, indicating the handler to be used).

Word 1; Address of the area in core store (the buffer) to or from which complete data blocks are read or written. The user must reserve this block within his own data areas. The first two words of this block are reserved for the block number and length, set by software.

Word 2; The length of the buffer area in core store. A positive integer $5\leq i\leq 2047$ giving the actual length of block to be written or the maximum length of block that can be read.

If a label block is to be written bit 18 must also be set to one.

Words 3; and 4; Reserved for use by the magnetic tape software.

Word 5; Address/Marker for record handling.

Words 6; to 9; Reserved. Words 8; and 9; are affected when the file is opened.

Words 10; to 35; Correspond exactly to words 2; to 27; of the header block format (see 2.7). When opening for writing the header block is copied from this position.

Words 10; to 13; Name of file, 12 characters.

Word 14; Serial number.

Word 15; Generation number.

Word 16; Reel number.

Word 17; Creation date.

Word 18; Expiry date.

Words 19; to 22; Reserved.

Word 23; Record structure indicator

Word 24; Reserved

Word 25; Protection marker.

Words 26; to 35; Owners optional information. If word 26; is zero the file table is assumed to end at that word.

Most of the file table is only referred to on opening a file. Many items of information are not used by the standard software. Some of these may be used by future software routines, but in general the user can ignore them or use them for his own purpose as necessary. If an item is not used it should be set to zero. Reserved words must not be used, and other words must be given the meanings and the format described here or in 2.7.

Chapter 4: FILE PROTECTION

File protection is arranged by only allowing programs to read or write on a file if the file is "opened", an operation which checks the identity of the file.

If a program attempts to read or write a magnetic tape before the appropriate "Open" subroutine is used, an error is generated. The "Open File" subroutine will only work if the name in the file header block is specified by the user program, or if the file is a SCRATCH file.

Only one special utility program is allowed to write onto a magnetic tape reel without the above checks. This is the initialising program MTINIT, which must be used to prepare any new tape for use by the other software.