

2. 4 MTWRITE

MTWRITE (NH, IA, NP)

This procedure is used to write a block of data (or a label block) onto magnetic tape. When used it must be followed by a call of MTCHECK.

NH is the handler number required (in the range 0 to 3).

IA is an integer array which must contain the data block to be written. The first 2 words of the array must not contain significant data, as they will be overwritten when the block is written onto tape.

NP must be set to the number of words to be written in the block. This number includes the 2 locations at the lower end of the array, e.g. if JAA(1) to JAA(200) are to be output on handler 2, the call might be:

```
CALL MTWRITE (2,JAA,200);
```

JAA(1) and JAA(2) will be overwritten. This example assumes that JAA is declared as DIMENSION JAA (300) say.

After the call of MTWRITE array IA must not be referred to or altered until MTCHECK has been called.

The block will be written onto the tape, starting at the current position of the tape under the Read/Write heads. Writing should normally be in sequence along the tape from the header block written by MTOPEN. Each call of MTWRITE leaves the tape positioned ready for the next call. When all blocks of a file have been written MTCLOSE should be used to close for writing. The call of MTCHECK following each call of MTWRITE will give the block number of the block just written, and if necessary the program may keep a record of certain block numbers, so that they can be conveniently found by MTREAD.

If blocks are to be written from a different position, MTREAD should be used to position the tape. As soon as a block is written all other blocks further along the tape should be considered unreadable, (though in exceptional circumstances they may be found using MTREAD to read a specific block number). It is not possible to overwrite odd blocks in the middle of an existing file and preserve the rest of the file. If random blocks are to be updated, the whole file must be copied onto a new tape reel, adding or deleting the necessary blocks as they appear in the copying sequence.

Up to 20 attempts are made to write the block specified, and small patches of bad tape are erased over before the block is written.

If a label block is to be written, parameter NP should be given value (number of words - 131071 - 1).

2.5 MTREAD.

MTREAD (NH, IA, N)

This procedure is used to find and read a specified block on a magnetic tape reel. Since all blocks are numbered, a block with a given number may be found, searching forwards or backwards along the tape.

NH is the handler number required (in the range 0 to 3).

IA is an integer array which is used as a buffer for input of blocks. The whole array is used. After reading and successfully checking with MTCHECK the block specified by N will be held in IA, with the first two words holding special information inserted by MTWRITE when the block was written.

N parameter specifies the block to be found and read. If N = 0 the next block in sequence is read. If N is positive, non-zero then the block with number N is found.

The array IA should not be referred to until after the call of MTCHECK following MTREAD. If the actual block specified to be read is a label block then MTCHECK returns value +0. However, the block will still be read correctly into array IA, with the block number in the first location of IA.

The search for a numbered block will start with a rewind or a number of backspace movements if necessary, then will continue forward till the block is found. The search forward is not fundamentally quicker than reading repetitively with MTREAD (NH, IA,), unless there are a number of bad patches on the tape.

Up to 20 attempts will be made to read the actual block specified, other blocks are ignored. The actual block length (number of words) will be found in the second location of IA, unless a label is read, when this location holds the length - 131072.

2.6 MTCLOSE

MTCLOSE (NH, NT)

This procedure is used to 'close' a magnetic tape file after reading or writing is complete. Files that have been written or updated must be closed for writing once the last block has been added. In general, at the end of a program run, and always before a reel is removed, any reels used by the program should be closed.

MTCLOSE may also be used to rewind a tape without closing the file.

NH is the handler number required, in the range 0 to 3.

NT is a parameter giving the type of operation required as follows:

- NT = 1 Close file for writing. Write an End of File label and rewind tape.
- NT = 2 Close file for writing and reading. Write an End of File label and rewind.
- NT = 3 Close file for reading only. Rewind tape (Only use if file not opened for writing).
- NT = 0 Rewind tape without closing file.

After closing for writing ($T = 1$ or $T = 2$), MTCHECK must be used before any other reference to any of the tapes. After this (and after any other call of MTCLOSE) MTCHECK may be used at any time to check whether the rewind operation is complete.

Chapter 3: STORE USED

Approximate core store used:

MTCHECK, MTOPEN, MTREAD, together = 600 words

MTCLOSE 65 words

MTWRITE 110 words

Total: 775 words

Chapter 4: TIME TAKEN

The time taken for these procedures is governed almost entirely by the tape movement times, and it is virtually independent of the processor speed.

Approximate times for 9kHz tape, assuming no re-reads or re-writes due to tape errors, may be calculated as follows, if W is the number of words in a block:

MTWRITE	$18 + 0.33W$ millisec.
MTOPEN	28 millisec. plus rewind time, to open for reading or updating
	80 millisec. plus rewind time, to open for writing.
MTCLOSE	20 millisec. to close for writing. 1 millisec. to close for writing.
MTCHECK	0.8 millisec (100 microsec. on 905, 1μS store)
MTREAD	$18 + 0.33 W$ millisec. to read next block.

If MTREAD is used to search for a given block number N and L is the number of the last block read or written, then the time taken in seconds is very approximately given by:

```
if N > L then (N-L) * (18 + 0.33 W)/1000  
else if N ≤ L/2 then (L/150) * (W/66 + 0.75) + (N/1000) * (18 + 0.33 W)  
else (L - N + 2) * (18 + 0.33 W)/1000;
```

In this case W is the average number of words in a block.