SIMH Magtape Representation and Handling

Bob Supnik, 03-Mar-03

<u>Magtape Representation</u>

SIMH represents magnetic tapes as disk files. Each disk file contains a series of objects. Objects are either metadata markers, like tape mark or end of medium, or they are data records. Location 0 of the file is interpreted as beginning of tape; end of file is interpreted as end of medium. Pictorially:

```
Location 0:    +--------+
               |  data  |
               | record |
               +--------+
               |  data  |
               | record |
               +--------+
                    :
               +--------+
               |  tape  |
               |  mark  |
               +--------+
               |  data  |
               | record |
               +--------+
                    :
end of file:
```

Metadata markers are 4 bytes stored in little-endian order. The currently defined metadata markers are:

```
     0xFFFFFFFF                    end of medium
     0xFF000000:0xFFFFFFFE         reserved
     0x00000000                    tape mark
```

Data records are consist of an initial 4 byte record length n, (n + 1) & ~1 bytes of data, and a trailing 4 byte record length n that must be the same as the initial record length:

```
bytes 0:3      +--------+
               | record |
               | length |
               +--------+
bytes 4:n+3    |  data  |
               |    :   |
               |    :   |
               +--------+
```

```
bytes n+4:n+7  | record |
               | length |
               +--------+
```

Note that the data is rounded to an even number of bytes. If the record length is odd, the extra byte is undefined but should be 0.

Record lengths are 4 bytes stored in little-endian order. The high order bit is flag, indicating that the record contains an error; the next 7b must be zero; the low 24 bits are the record length:

```
bit<31>        1 = record contains error
               0 = record is error-free
bits<30:24>    must be zero
bits<23:0>     record length, must be non-zero
```

The leading and trailing record lengths allow a record to be accessed either forward or backward.

<u>Magtape Operations</u>

Magnetic tape drives can perform the following operations:

- Read forward
- Read backward
- Write forward
- Space forward record(s)
- Space backward record(s)
- Space forward file(s)
- Space backward file(s)
- Write tape mark
- Security erase
- Write extended gap

On a real magtape, all operations are implicitly sequential, that is, they start from the current position of the tape medium. SIMH implements this with the concept of the *current tape position*, kept in the pos field of the tape drive's UNIT structure. SIMH starts all magtape operations at the current position and updates the current position to reflect the results of the operation:

- Read forward. Starting at the current position, read the next 4 bytes from the file. If those bytes are a valid record length, read the data record and position the tape past the trailing record length. If they are a tape mark, signal tape mark and position the tape past the tape mark. If they are end of medium, or an end of file occurs, signal no more data ('long gap' or 'bad tape') and do not change the tape position.

- Read reverse. If the current position is beginning of tape, signal BOT. Otherwise, starting at the current position, read the preceding 4 bytes from the file. If those bytes are a valid record length, read the data record and position the tape before the initial record length. If they are a tape mark, signal tape mark and position the tape before the tape mark. If they are end of medium, or an end of file occurs, signal no more data (‘long gap’ or ‘bad tape’) and position the tape before the end of medium marker.
- Write. Starting at the current position, write the initial record length, followed by the data record, followed by the trailing record length. Position the tape after the trailing record length.
- Space forward record(s). Starting at the current position, read the next 4 bytes from the file. If those bytes are a valid record length, position the tape past the trailing record length and continue until operation count exhausted or metadata encountered. If those bytes are a tape mark, signal tape mark and position the tape after the tape mark. If they are end of medium, or an end of file occurs, signal no more data (‘long gap’ or ‘bad tape’) and do not change the tape position.
- Space reverse record(s). If the current position is beginning of tape, signal BOT. Otherwise, starting at the current position, read the preceding 4 bytes from the file. If those bytes are a valid record length, position the tape before the initial record length and continue until operation count exhausted, BOT, or metadata encountered. If they are a tape mark, signal tape mark and position the tape before the tape mark. If they are end of medium, or an end of file occurs, signal no more data (‘long gap’ or ‘bad tape’) and position the tape before the end of medium marker.
- Space forward file(s). Starting at the current position, read the next 4 bytes from the file. If those bytes are a valid record length, position the tape past the trailing record length and continue. If those bytes are a tape mark, signal tape mark, position the tape after the tape mark, and continue until operation count exhausted. If they are end of medium, or an end of file occurs, signal no more data (‘long gap’ or ‘bad tape’) and do not change the tape position.
- Space reverse file(s). If the current position is beginning of tape, signal BOT. Otherwise, starting at the current position, read the preceding 4 bytes from the file. If those bytes are a valid record length, position the tape before the initial record length and continue. If they are a tape mark, position the tape before the tape mark and continue until operation count exhausted or BOT. If they are end of medium, or an end of file occurs, signal no more data (‘long gap’ or ‘bad tape’) and position the tape before the end of medium marker.
- Write tape mark. Starting at the current position, write a tape mark marker. Position the tape beyond the new tape mark.
- Security erase. Starting at the current position, write an end of medium marker. Do not update the tape position.
- Write extended gap. All implementations to date treat this as a NOP, because it does not create readable data. This should erase the next object on the tape (as a minimum), but because tape records are only 16b aligned instead of 32b aligned, there is no reliable way to do this.

Magtape Error Handling

The following matrix defines error responses versus events for simulated magtapes. PNU signifies position not updated; PU signifies position updated.

	Unit not attached	Tape mark	End of medium mark	Write locked	End of attached file	Data read or write error
Read forward	Error: unit not ready, PNU	Error: tape mark, PU	Error: bad tape or runaway tape, PNU	ok	Error: bad tape or runaway tape, PNU	Error: parity or data, PNU
Read reverse	Error: unit not ready, PNU	Error: tape mark, PU	Error: bad or runaway tape, PU	ok	Error: bad or runaway tape, PU	Error: parity or data, PNU
Write forward	Error: unit not ready, PNU	na	na	Error: unit write locked, PNU	na	Error: parity or data, PNU
Space records forward	Error: unit not ready, PNU	Error: tape mark, PU	Error: bad or runaway tape, PNU	ok	Error: bad or runaway tape, PNU	Error: parity or data, PNU
Space records reverse	Error: unit not ready, PNU	ok	Error: bad or runaway tape, PU	ok	Error: bad or runaway tape, PU	Error: parity or data, PNU if error on record length, otherwise PU
Space files forward	Error: unit not ready, PNU	Error: tape mark, PU	Error: bad or runaway tape, PNU	ok	Error: bad or runaway tape, PNU	Error: parity or data, PNU
Space files reverse	Error: unit not ready, PNU	ok	Error: bad or runaway tape, PU	ok	Error: bad or runaway tape, PU	Error: parity or data, PNU if error on record length, otherwise PU
Write tape mark	Error: unit not ready, PNU	na	na	Error: unit write locked, PNU	na	Error: parity or data, PNU
Erase	Error: unit not ready, PNU	na	na	Error: unit write locked, PNU	na	Error: parity or data, PNU

The behavior of simulated tapes mirrors that of real tapes, except for errors that make determination of the record length impossible. On a real tape, a read or write error would update the position of the tape. On a simulated tape, this isn't possible; the length of the record is unknown. Real tape drivers would try to recover from the error by backspacing over the erroneous record and trying again. This won't work on a simulated tape.

For intelligent tapes, like the TK50 and the TS11, this problem is handled by reporting "position lost". This status tells the tape driver that tape position is no longer known, and normal error recovery isn't possible. Older tapes do not have this status. Accordingly, these tapes implement a limited form of state "memory" for error recovery. If an error occurs on a forward operation, and the position is not updated, the simulated tape unit "remembers" this fact. If the next operation is a backspace record, the first backspace is skipped, because the simulated tape is still positioned at the start of the erroneous record. If a read is then attempted, the tape will read the record that caused the original error.

Magtape Emulation Library

SIMH provides a support library, sim_tape.c (and its header file sim_tape.h), that implements the standard tape format and functions. The library is described in detail in the associated document, “Writing A Simulator For The SIMH System”.