Secondary Storage Devices: Magnetic Tape

A sequence of bits (1’s and 0’s) is stored on magnetic tape. For storage, the tape is wound on a reel. To access the data, the tape is unwound from one reel to another.

As the tape passes the head, bits of data are read from or written onto the tape.

Tracks:

Typically, data on tape is stored in 9 separate bit streams, or tracks. Each track is a sequence of bits.
Recording density = # bits per inch (bpi), typically 800 or 1600 bpi, 30,000 bpi on some recent devices.

A closer look:

Parity bit:

\[
\begin{align*}
\text{Parity bit} & : 0 & \text{if } \# \text{ of 1's in byte is even} \\
& : 1 & \text{if } \# \text{ of 1's in byte is odd}
\end{align*}
\]

Error Checking: If the parity bit rule does not hold when the byte is read, then the tape has been corrupted.

Tape organization:
BOT= beginning of tape marker
Header block: describes data blocks
Inter block gap: For acceleration and deceleration of tape
Blocking factor: # records per block

Data blocks and records

- Each data block is a sequence of contiguous records
- A record is the unit of data that a user’s program deals with.
- For efficiency, the tape drive reads an entire block of records at once.
- Unlike a disk, a tape starts and stops
- When stopped, the read/write head is over an interblock gap (i.g)
- To read a block, the tape uses the i.g to accelerate. Then it reads all records in the data block. Then it uses the next i.g to decelerate.

Tape capacity:

Given the following tape:
Recording density = 1600 bpi
Tape length = 2400’
Inter block gap = ½”
512 bytes per record
Blocking factor =25

How many records can we write on the tape (ignoring BOT and EOT markers and the header block0
bytes/block = 512 bytes/rec * 25 rec/block = 12,800 bytes/block
Block length = #bytes per block / # bytes per inch
Block + gap = 8" + 1/2" = 8.5"

Tape length = 2400 ft * 12 in/ft = 28,800 in
#blocks = (tape length) / (block + gap)
#blocks = 28,800 / 8.5 = 3388 blocks
#records = #blocks * # rec per block
# records = 3388 * 25 = 84,700 records

Buffering:
- A user wants to read or write one record at a time
- but, a secondary storage device (disk or tape) reads and writes an entire block all at once.
- Question: how do we resolve this mismatch
- Solution: Buffers

Buffers:
- Buffers are managed by the computer’s operating system (O.S)
- The O.S uses buffers to mediate between secondary storage and user
  programs, i.e. to translate between blocks and records et vice versa.
- The whole process is invisible to the user, who thinks he is reading or
  writing one record at a time to secondary storage (S.S)
- When a user’s program asks for a record, the O.S automatically reads
  an entire block of data from S.S to a buffer in M.M
- The O.S then passes the first record to the user’s program
- When the user’s program asks for the next record, the O.S gives it the
  2nd record in the buffer, without accessing S.S

Note: The buffer is 10,000 times faster than S.S because it is in the MM
This process continues with the 3rd, 4th, 5th … records in the buffer being
passed to the user’s program with each read request.
- Eventually, all the records in the buffer may be passed to the user’s
  program.
- If the program then asks for another record, the O.S will read
  another block of data from SS into the buffer (overwriting the old
  buffer contents)
- The O.S then passes the first record in the buffer to the user’s
  program, etc.
- This way, the O.S hides the details of reading and writing from the
  user, it means that the O.S manages the data blocks for the user.