Lecture 3: Secondary Storage Devices

Last Day: Magnetic Disks
- disk organization
- access time

Today: Magnetic Tape
- tape organization
- example

Buffering

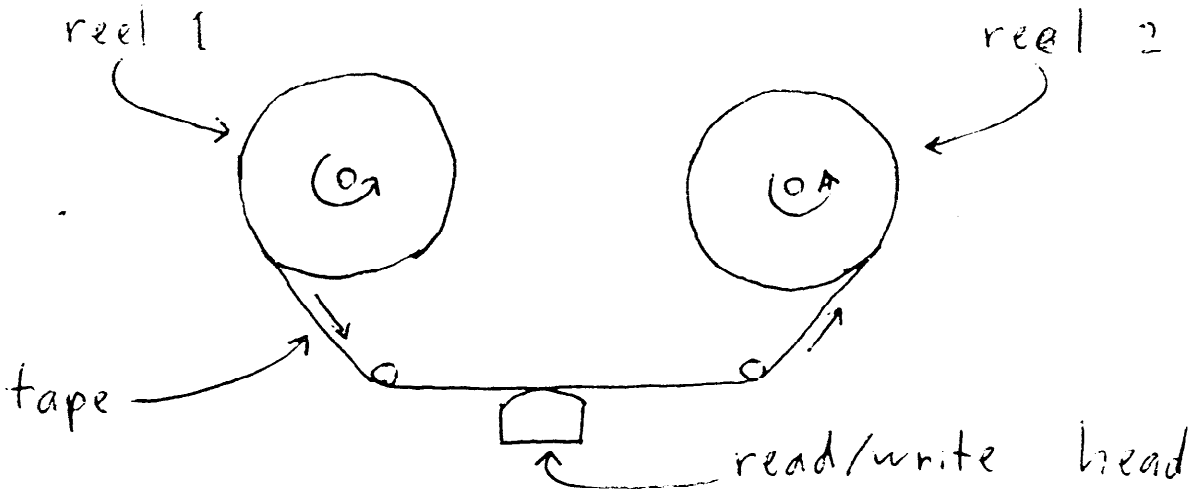
Zoellick & Folk §3.2, 3.3, 3.4, 3.6

Magnetic Tape

- A sequence of bits (1's and 0's) are stored on magnetic tape.

- For storage, the tape is wound on a reel.

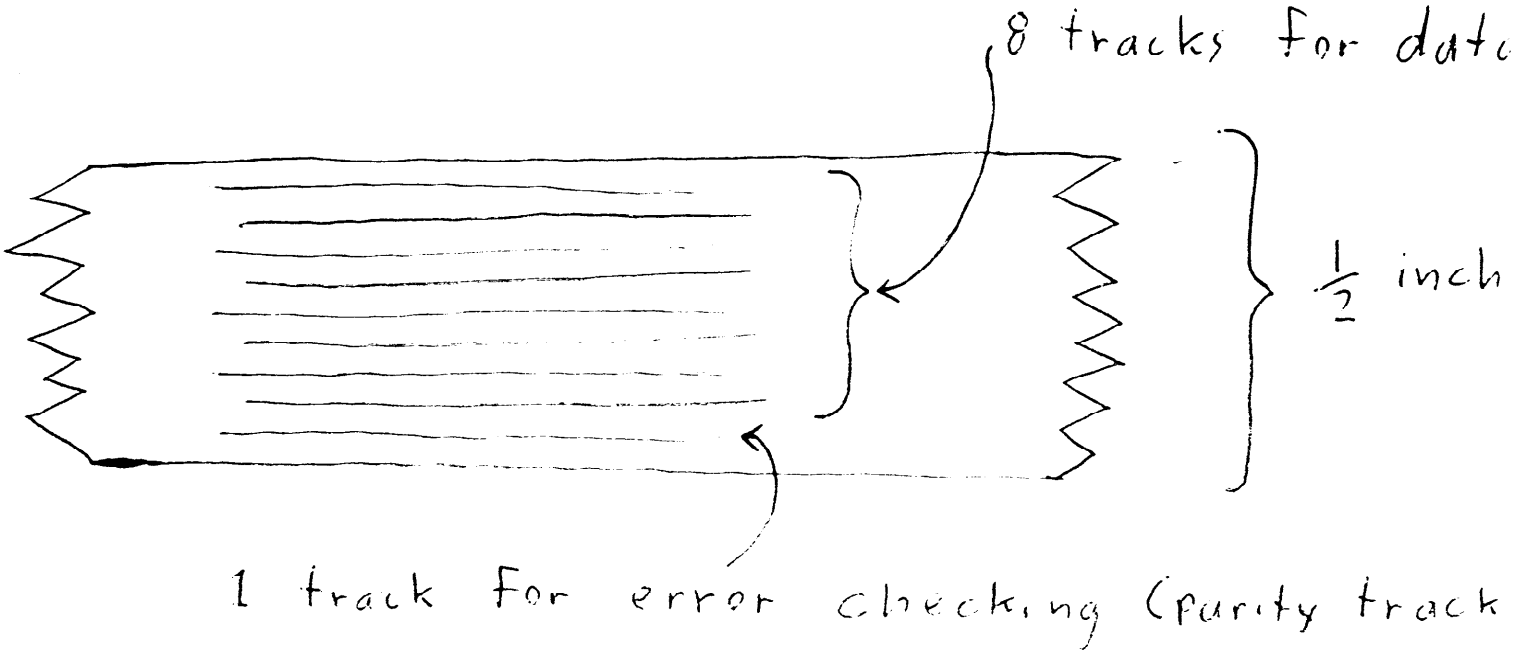
- To access the data, the tape is unwound from one reel to another.



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

Tracks

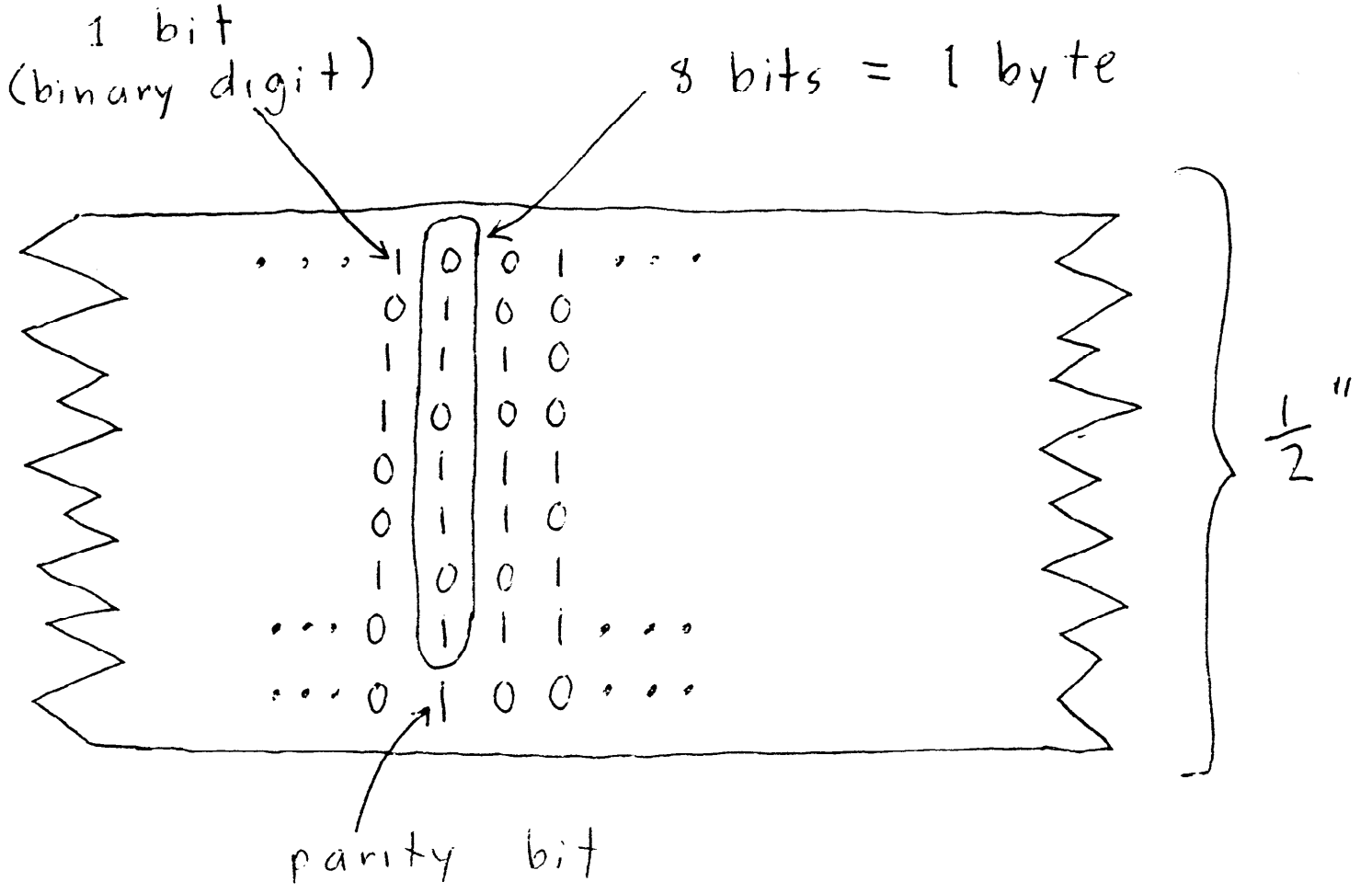
- Typically, data on tape is stored in 9 separate bit streams, or tracks.



- 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.

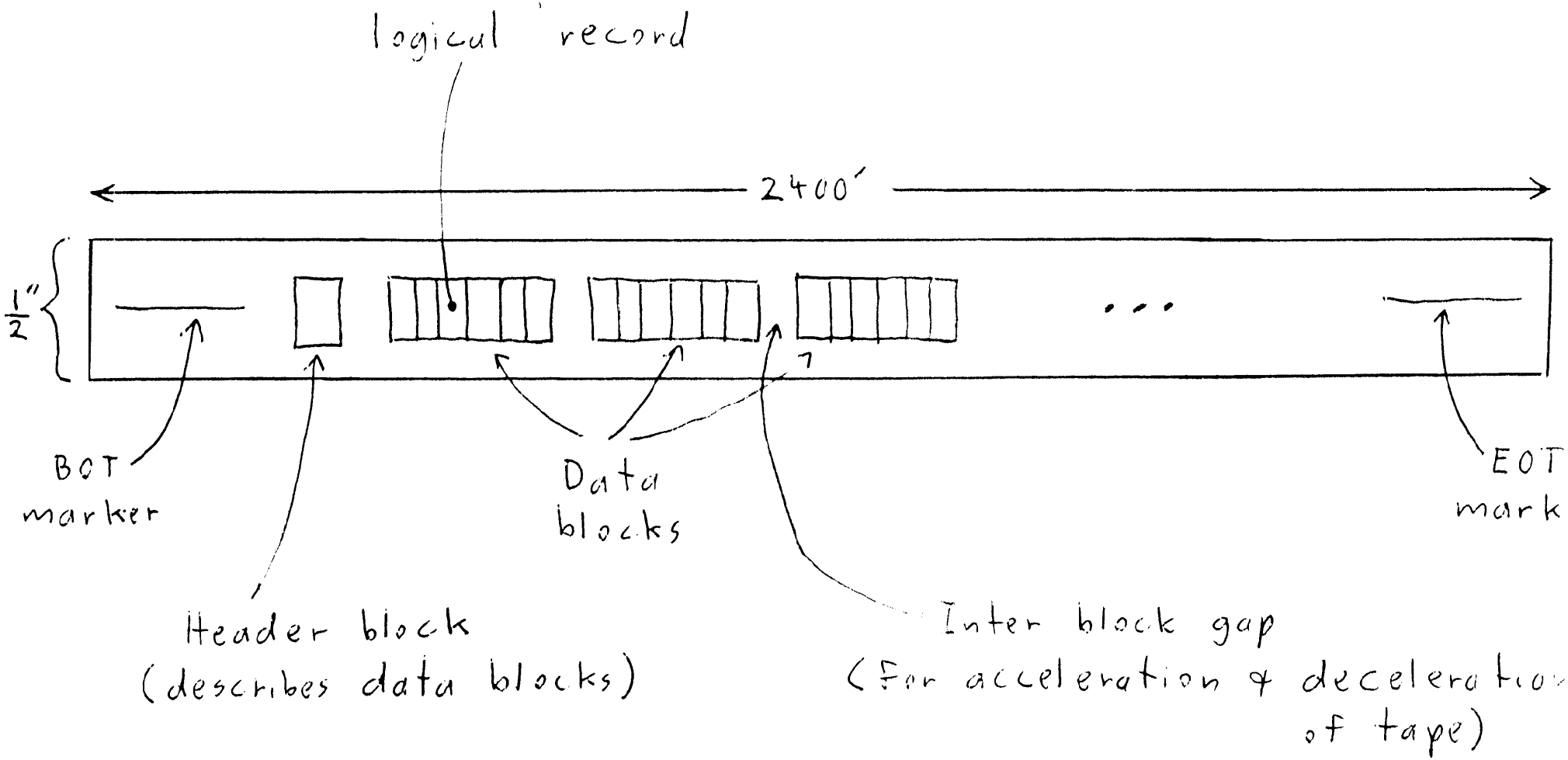
In Detail



* parity bit = { 0 if # of 1's in byte is even.
1 if # of 1's in byte is odd.

Error checking: If * does not hold when byte is read, then the tape has been corrupted

Tape Organization



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 & 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

Example: Tape Capacity

Given the following tape:

recording density = 1600 bpi

tape length = 2400'

inter block gap (ibg) = $\frac{1}{2}$"

512 bytes per record

blocking factor = 25

How many records can we write on the tape? (ignoring BOT & EOT markers and the header block, for simplicity).

$$\#\text{bytes/blk} = \overbrace{512 \text{ bytes/rec}}^{1 \text{ record}} \times \overbrace{25 \text{ rec/blk}}^{\text{blocking factor}}$$

$$= 12{,}800 \text{ bytes/blk}$$

$$\text{block length} = \frac{\#\text{ bytes/blk}}{\#\text{ bytes/inch}}$$

$$= \frac{12{,}800}{1600} \text{ inches} = 8 \text{ inches}$$

$$\text{block + gap} = 8'' + \tfrac{1}{2}'' = 8.5''$$

$$\text{tape length} = 2400 \text{ Ft.} \times 12 \text{ in/ft} = 28{,}800 \text{ in}$$

$$\#\text{blocks} = (\text{tape length}) / (\text{block + gap})$$

$$= 28{,}800 / 8.5 = 3388 \text{ blks}$$

$$\#\text{records} = \#\text{blks} \times \#\text{recs/blk}$$

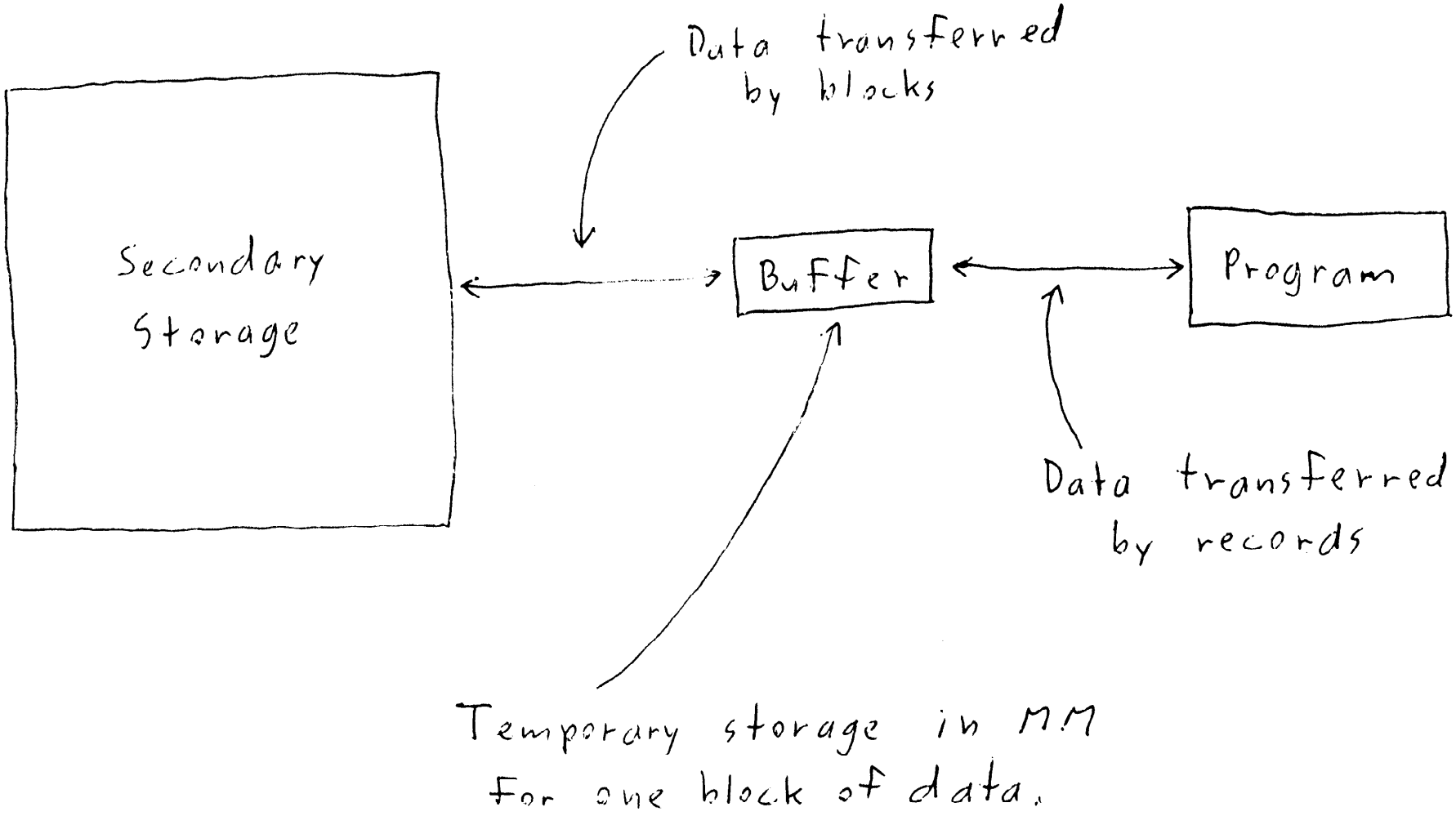
$$= 3388 \times 25 = \underline{84{,}700 \text{ records}}$$

Buffering

Zoellick & Folk, § 3.6

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

Buffers


Data transferred by blocks
Secondary Storage
Buffer
Program
Data transferred by records
Temporary storage in MM For one block of data.

Buffering

- Buffers are managed by a computer's operating system (O.S)
- The O.S. uses buffers to mediate between secondary storage and user programs.
- ie, to translate between blocks & records, & 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 M.M.)

- 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, as before.

- In this way, the O.S. hides the details of reading & writing from the user.

- ii, The O.S. manages the data blocks for the user.