I/O MANAGEMENT

BY: MAANASA S (20PW19)

I/O DEVICES

- HUMAN READABLE
  - Suitable for communicating with the computer user.
- MACHINE READABLE
  - Suitable for communicating with electronic equipment.
- COMMUNICATION
  - Suitable for communicating with remote devices.

ORGANISATION OF I/O

- PROGRAMMED I/O
  - The processor issues an I/O command, on behalf of a process, to an I/O module; that process then busy waits for the operation to be completed before proceeding.

- INTERRUPT- DRIVEN I/O
  - If the I/O instruction from the process is non blocking, then the processor continues to execute instructions from the process that issued the I/O command. If the I/O instruction is blocking, then the next instruction that the processor executes is from the OS, which will put the current process in a blocked state and schedule another process.

- DIRECT MEMORY ACCESS
  - The processor sends a request for the transfer of a block of data to the DMA module and is interrupted only after the entire block has been transferred.

EVOLUTION OF I/O

- Processor directly controls a peripheral device.
- A controller or I/O module is added.
- Controller or I/O module with interrupts.
- The I/O module is given direct control of memory via DMA.
- The I/O module is enhanced to become a separate processor, with a specialized instruction set tailored for I/O. (I/O Channel)
- The I/O module has a local memory of its own (I/O Processor)

I/O DEVICES
(Key Differences)

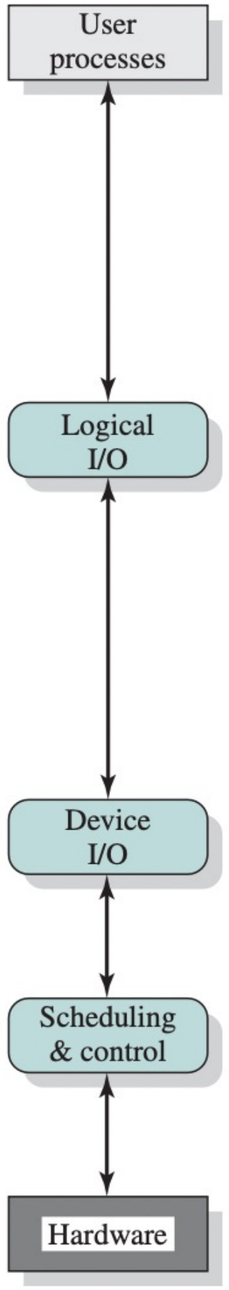
- Data Transfer
- Application
- Complexity of Control
- Unit of Transfer
- Data representation
- Error condition

How is OS related to I/O Devices?

An operating system’s main function is used to control access to the input and output devices. Operating systems and specific hardware devices communicate with each other via this device.

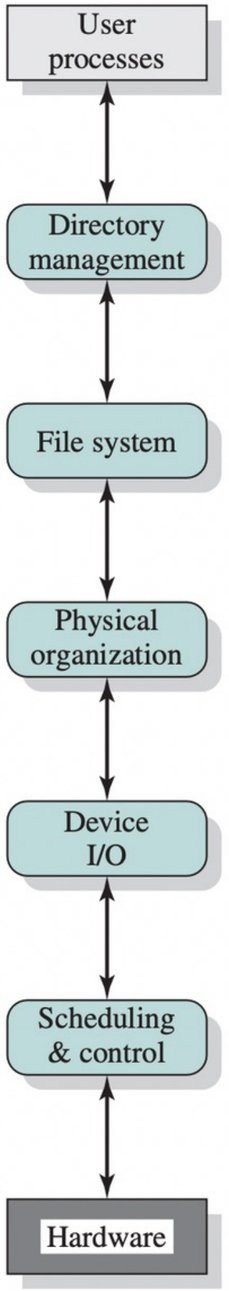
LOGICAL STRUCTURE OF I/O

- LOGICAL I/O:
  - Concerned with managing general I/O functions on behalf of user processes, allowing them to deal with the device in terms of a device identifier and simple commands such as open, close, read, & write.
- DEVICE I/O:
  - The requested operations and data are converted into appropriate sequences of I/O instructions, channel commands, and controller orders. Buffering techniques may be used to improve utilization.
- SCHEDULING & CONTROLLING:
  - The actual queuing and scheduling of I/O operations occurs at this level. Interrupts are handled and I/O status is reported.



LOGICAL STRUCTURE OF I/O

- DIRECTORY MANAGEMENT:
  - Symbolic file names are converted to identifiers. This level is also concerned with user operations that affect the directory of files, such as Add, Delete, and Reorganize.
- FILE SYSTEM:
  - Deals with logical structure of files. Open, Close, Read, Write. Access rights are handled in this level.
- PHYSICAL ORGANISATION:
  - References to files are converted to physical secondary storage addresses, taking into account the physical track and sector structure of the secondary storage device. Allocation of secondary storage space and main storage buffers is handled in this level.



I/O BUFFERING

Why do we need buffering?

Why do we need buffering?

A process waits either by, Busy waiting or Process suspension on an interrupt

PROBLEMS CAUSED:

- Program waits for the I/O to finish execution.
- Risk of single-process deadlock.

I/O BUFFERING

- BLOCK ORIENTED DEVICES:
  - Stores information in blocks that are usually of fixed size, and transfers are made one block at a time.
  - Ex: Disks, USBs

- STREAM ORIENTED DEVICES:
  - Transfers data in and out as a stream of bytes, with no block structure
  - Ex: Printers, Mouse

TYPES OF BUFFER: Single Buffer, Double Buffer, Circular Buffer

SINGLE BUFFER

When a user process issues an I/O request, the OS assigns a buffer in the system portion of main memory to the operation.

- BLOCK ORIENTED DEVICES:
  - Input transfers are made to the system buffer.
  - Upon completion of transfer, the process moves the block into user space.
  - Immediately requests another block. [Reading ahead]

- STREAM ORIENTED DEVICES:
  - Line at a time fashion. Ex: Line printer
  - Byte-at-a-time operation. Ex: Sensors

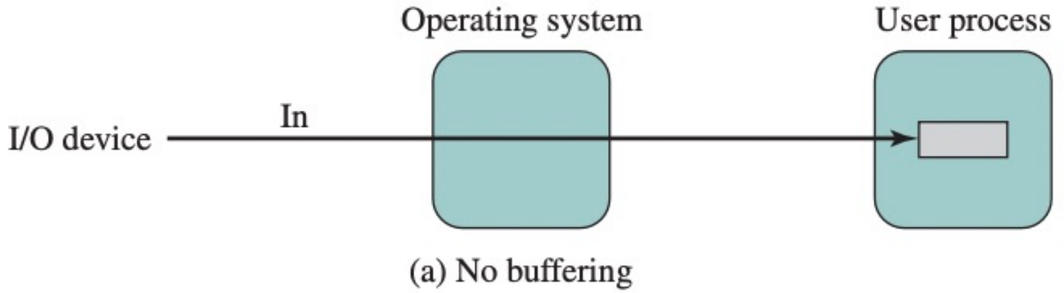
DOUBLE BUFFER

A process transfers data to (or from) one buffer while the operating system empties (or fills) the other. This technique is known as double buffering or buffer swapping.

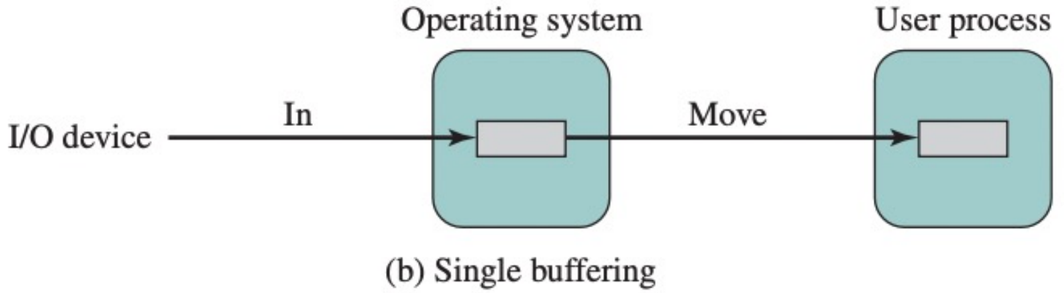
For line-at-a-time I/O, the user process need not be suspended for input or output, unless the process runs ahead of the double buffers.

CIRCULAR BUFFER

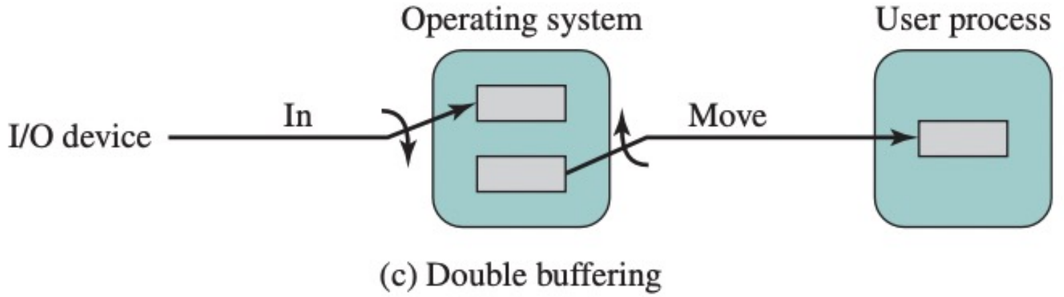
When more than two buffers are used then collection of buffers is known as circular buffer with each individual buffer being one unit of the circular buffer.



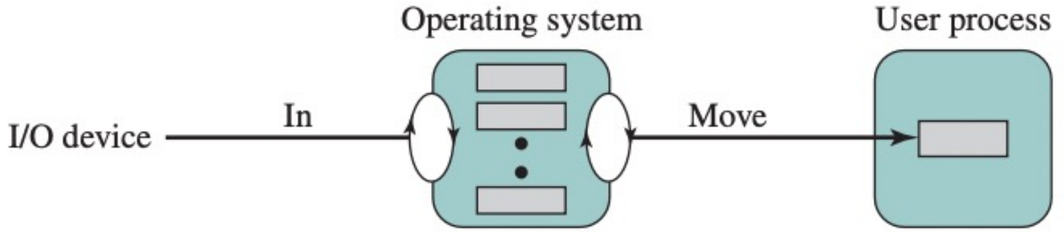
(a) No buffering



(b) Single buffering



(c) Double buffering



(d) Circular buffering

UTILITY OF BUFFER

- Buffering is a technique that smoothes out peaks in I/O demand.
- Even with multiple buffers, all of the buffers will eventually fill up and the process will have to wait after processing each chunk of data.