FIFO And Disk Management

1. What is FIFO in embedded C

➔ FIFO stands for "First In, First Out," and it is a type of data structure commonly used in embedded systems programming. It is a type of queue, where data is added to one end (the "tail") and removed from the other end (the "head"). Data added first is removed first, hence the name "First In, First Out." This is in contrast to a LIFO (Last In, First Out) data structure, such as a stack, where the last item added is the first item removed. FIFOs are often used for communication between different parts of a system or between different systems, as they provide a way to store and retrieve data in a predictable order.

2. How FIFO is used to communicate between the processes

➔ FIFO allows for data to be passed between processes in a predictable order, with the oldest data being removed first.

➔ One way FIFOs are used for inter-process communication is through named pipes.

➔ Another way FIFO is used for inter-process communication is through message queues

➔ FIFO enables the processes to communicate with each other in a controlled and predictable manner, ensuring that data is not lost and the order of the data is preserved.

3. What is Named Pipe

➔ A named pipe is a special file that acts as a buffer for data being passed between processes. One process writes data to the named pipe, and another process reads that data from the pipe. Since the data is read in the same order that it was written, the named pipe behaves like a FIFO queue.

4. What is Message Queue

➔ A message queue is a data structure that stores messages and allows multiple processes to send and receive messages in a predictable order. Each message is added to the tail of the queue and removed from the head, maintaining the FIFO order.

5. Descriptive information about named pipe in fifo

➔ A named pipe, also known as a FIFO (First In, First Out) pipe, is a special file that acts as a buffer for data being passed between processes in an operating system. A named pipe can be thought of as a two-way communication channel, where one process writes data to the pipe and another process reads that data.

➔ One process creates the named pipe using the mkfifo() system call and giving it a unique name. Once the named pipe is created, it can be opened by other processes using the open() system call. One process opens the pipe for writing and another process opens it for reading. The writing process writes data to the named pipe, and the reading process reads the data from the pipe

➔ Since the data is read in the same order that it was written, the named pipe behaves like a FIFO queue. This allows for data to be passed between processes in a predictable order, with the oldest data being removed first. Named pipes are often used for inter-process communication when the processes involved need to synchronize their actions, for example when one process is producing data and another process is consuming it.

➔ Additionally, Named pipes support blocking and non-blocking I/O, meaning that the process can choose to block when it tries to read from an empty pipe or write to a full pipe, or it can choose to return an error instead.

➔ Named pipes are a powerful tool for inter-process communication, they are widely used in many operating systems and in different programming languages.

6. Where is FIFO used?

➔ Data Structures:

  ◆ Certain data structures like Queue and other variants of Queue use the FIFO approach for processing data.

➔ Disk scheduling:

  ◆ Disk controllers can use the FIFO as a disk scheduling algorithm to determine the order in which to service disk I/O requests.

➔ Communications and networking:

  ◆ Communication network bridges, switches and routers used in computer networks use FIFOs to hold data packets en route to their next destination.

7. Operating System :

➔ Most computer systems employ secondary storage devices (magnetic disks). It provides low-cost, non-volatile storage for programs and data (tape, optical media, flash drives, etc.).

➔ Programs and the user data they use are kept on separate storage devices called files.

➔ The operating system is responsible for allocating space for files on secondary storage media as needed.

➔ The operating system must be able to locate each file and perform read and write operations on it whenever it needs to.

- ➔ Therefore, the operating system is responsible for configuring the file system, ensuring the safety and reliability of reading and writing operations to secondary storage, and maintaining access times (the time required to write data to or read data from secondary storage).

8. Disk Management:

- ➔ It includes Disk format,Booting from disk,bad block recovery.

- ➔ The Low-Level Format or physical format:
  - ◆ It is conducted in two stages:
    - Divide the disc into multiple cylinder groups. Each is treated as a logical disk.
    - Logical format or “Create File System”. The OS stores the data structure of the first file system on the disk. Contains free space and allocated space.
  - ◆ For efficiency, most file systems group blocks into clusters. Disk I / O runs in blocks.File I / O runs in a cluster.

- ➔ Boot Block:
  - ◆ When the computer is turned on or restarted, the program stored in the initial bootstrap ROM finds the location of the OS kernel from the disk, loads the kernel into memory, and runs the OS. start.
  - ◆ To change the bootstrap code, you need to change the ROM and hardware chip. Only a small bootstrap loader program is stored in ROM instead.
  - ◆ The full bootstrap code is stored in the “boot block” of the disk.
  - ◆ A disk with a boot partition is called a boot disk or system disk.

➔ Bad Blocks:

- ◆ Disks are error-prone because moving parts have small tolerances.
- ◆ Most disks are even stuffed from the factory with bad blocks and are handled in a variety of ways.
- ◆ The controller maintains a list of bad blocks.
- ◆ The controller can instruct each bad sector to be logically replaced with one of the spare sectors. This scheme is known as sector sparing or transfer.
- ◆ A soft error triggers the data recovery process.
- ◆ However, unrecoverable hard errors may result in data loss and require manual intervention.