

FIFO

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.