REPORT FOR ASSIGNMENT 4

TASKA:

Learn what is a named pipe. Learn how to create and use a named pipe using command line in Linux (or any UNIX system).

A FIFO special file (a named pipe) is similar to a pipe, except that it is accessed as part of the file system. It can be opened by multiple processes for reading or writing. When processes are exchanging data via the FIFO, the kernel passes all data internally without writing it to the file system. Thus, the FIFO special file has no contents on the filesystem; the filesystem entry merely serves as a reference point so that processes can access the pipe using a name in the file system.
The kernel maintains exactly one pipe object for each FIFO special file that is opened by at least one process. The FIFO must be opened on both ends (reading and writing) before data can be passed. Normally, opening the FIFO blocks until the other end is also opened.

A process can open a FIFO in nonblocking mode. In this case, opening for read-only succeeds even if no one has opened on the write side yet and opening for write-only fails with ENXIO (no such device or address) unless the other end has already been opened.

Under Linux, opening a FIFO for read and write will succeed both in blocking and nonblocking mode. POSIX leaves this behavior undefined. This can be used to open a FIFO for writing while there are no readers available. A process that uses both ends of the connection in order to communicate with itself should be very careful to avoid deadlocks.

command to create a named pipe is mkfifo pipe2

TASKB:

1. Run one consumer and one producer concurrently
   - Kill the producer with Ctrl-C. Leave consumer running. What happens and why?
     Once the producer started writing and consumer started reading when I killed the producer, consumer showed error reading ret=0 errno=0 perror:success, because there is nothing to read and in FIFO generally both the end should be open.
   - Kill the consumer with Ctrl-C. Leave producer running. What happens and why?
     Once the producer started writing and consumer started reading when I killed the consumer, producer showed error writing ret=-1 errno=32 perror: broken pipe, error number 32 specifies broken pipe error the other end of the pipe is borken so producer will show this error.

2. Run one consumer and multiple producers concurrently.
When I ran one consumer and multiple producers concurrently consumer was reading lines written by both the producers but not in order that is sometimes it was reading one producers and sometimes it was reading lines written by different producers. Any process can access named pipes, subject to security checks, making named pipes an easy form of communication between related or unrelated processes.

3. Multiple consumers and one Producer.
When I ran one producer and multiple consumers few lines written by the producer was read by one consumer and few lines was read by other consumers because there is no synchronization in named pipe. Any process can access named pipes, subject to security checks, making named pipes an easy form of communication between related or unrelated processes.

4. Run multiple consumers and multiple producers concurrently.
When I ran multiple producers and multiple consumers each consumer was reading different lines of different consumers this is because lack of synchronization.

Submit brief plan for how to solve Task C.
I am planing to use mutex and semaphores to have the synchronization.