

Task A:

A named pipe is a named, one-way or duplex pipe for communication between the pipe server and one or more pipe clients. All instances of a named pipe share the same pipe name, but each instance has its own buffers and handles, and provides a separate conduit for client/server communication

Creating on named pipe on Linux: **mknod mypipe p**

Reference:

https://documentation.progress.com/output/ua/OpenEdge_latest/index.html#page/dvpin/creating-a-named-pipe.html

Task B:

1. One consumer & one producer concurrently

- a. **Killing Producer:** As soon as we kill producer it will stop producing item and stop filling buffer. In our case it will stop writing to the pipe, so consumer will not be able to read(consume) and start giving error.
- b. **Killing Consumer:** As soon as we kill consumer it will stop consuming item and buffer will be overflowed and producer won't be able to produce more items. In our case consumer will stop reading from the pipe, so producer will not be able to write(produce) and start giving error.

2. One consumer & multiple producer:

As there are multiple producers which will be producing items, the consumer will be consuming everything all the things. In our case producers will write many lines at same times and consumer will read all the lines.

3. Multiple consumer & one producer:

As there is only one producer it will be filling buffer, but we have many buffer so as soon as

something is produced the consumer which gets first wake up call will consume the item.

4. **Multiple consumer & multiple producer:** Here we can not say which consumer will consume which produced item as there is no synchronization.

Task C:

For coming over this problem I'll use semaphores or monitors.