

Linux Internals & Networking

System programming using Kernel interfaces

Team Emertxe



Contents



Linux Internals & Networking

Contents

- .Introduction
- .Transition to OS programmer
- .System Calls
- .Process
- .IPC
- .Signals
- .Networking
- .Threads
- .Synchronization
- .Process Management
- .Memory Management



Inter Process Communications (IPC)



Communication

In real world

- .Face to face
- .Fixed phone
- .Mobile phone
- .Skype
- .SMS



Inter Process Communications

Introduction

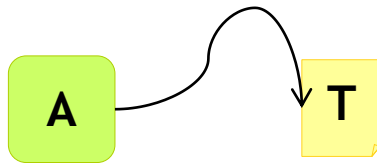


- *Inter process communication (IPC)* is the mechanism whereby one process can communicate, that is exchange data with another processes
- There are two flavors of IPC exist: System V and POSIX
- Former is derivative of UNIX family, later is when standardization across various OS (Linux, BSD etc..) came into picture
- Some are due to “UNIX war” reasons also
- In the implementation levels there are some differences between the two, larger extent remains the same
- Helps in portability as well

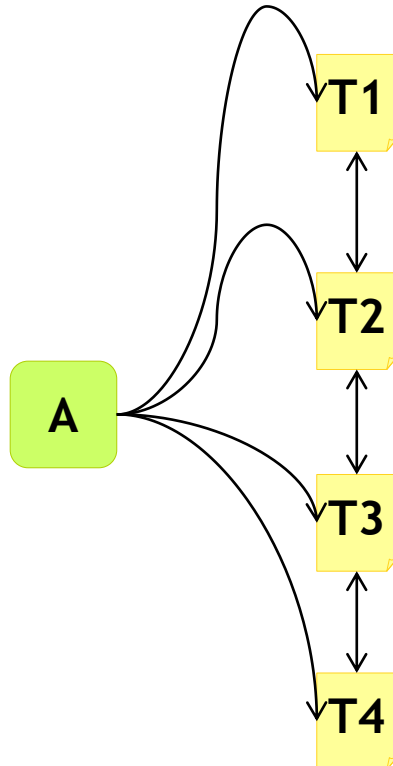
Team Emertxe



Application & Tasks



Example: Read from a file
`$ cat file.txt`



Example: Paper jam handling
in printer

Team Emertxe

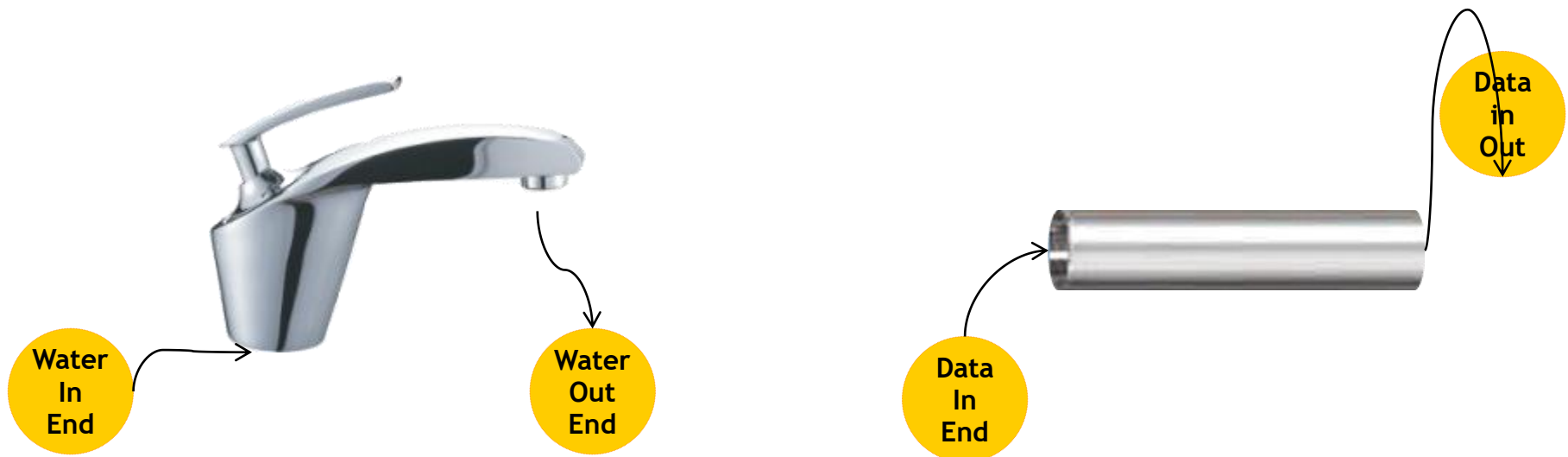


Inter Process Communications

Pipes



- A pipe is a communication device that permits unidirectional communication
- Data written to the “write end” of the pipe is read back from the “read end”
- Pipes are serial devices; the data is always read from the pipe in the same order it was written



Team Emertxe

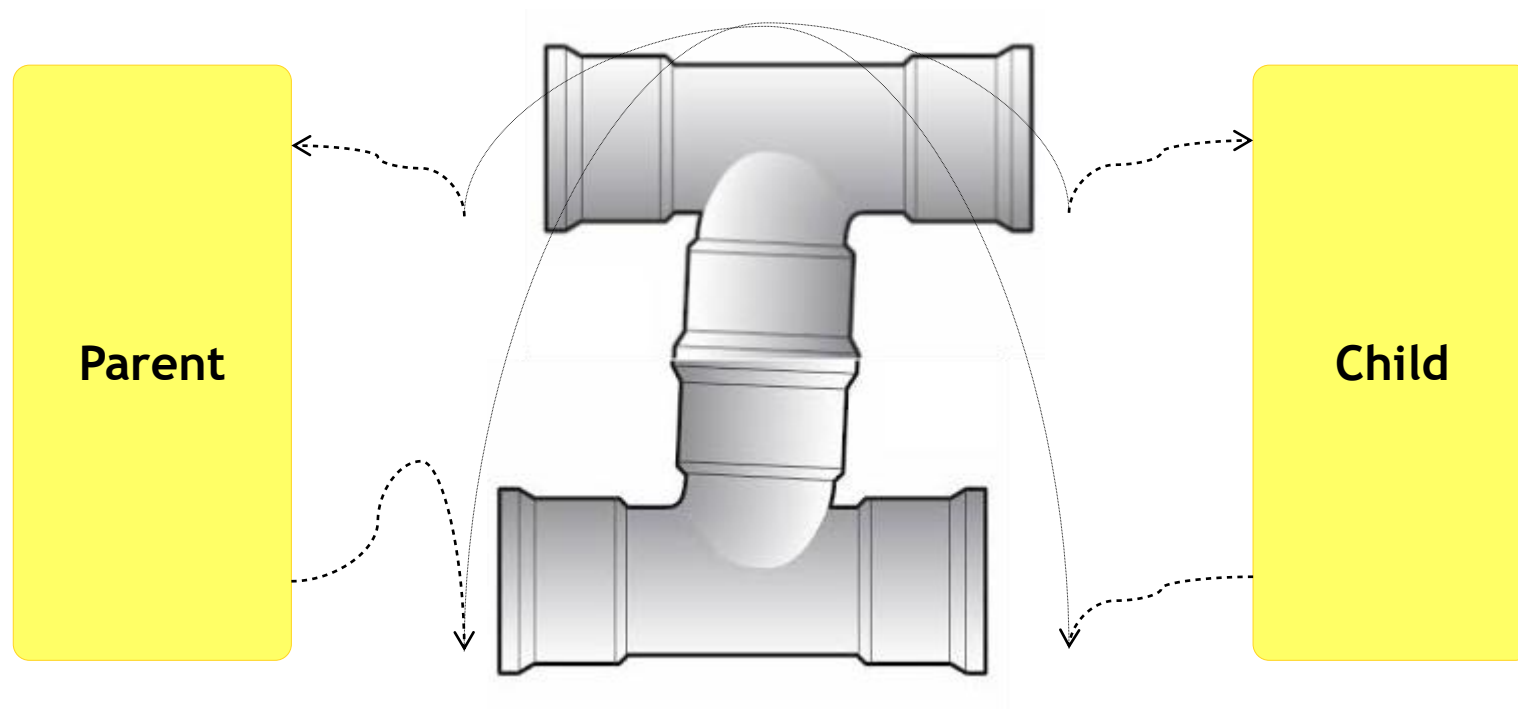


Inter Process Communications

Pipes – Direction of communication



- Let's say a Parent wants to communicate with a Child
- Generally the communication is possible both the way!

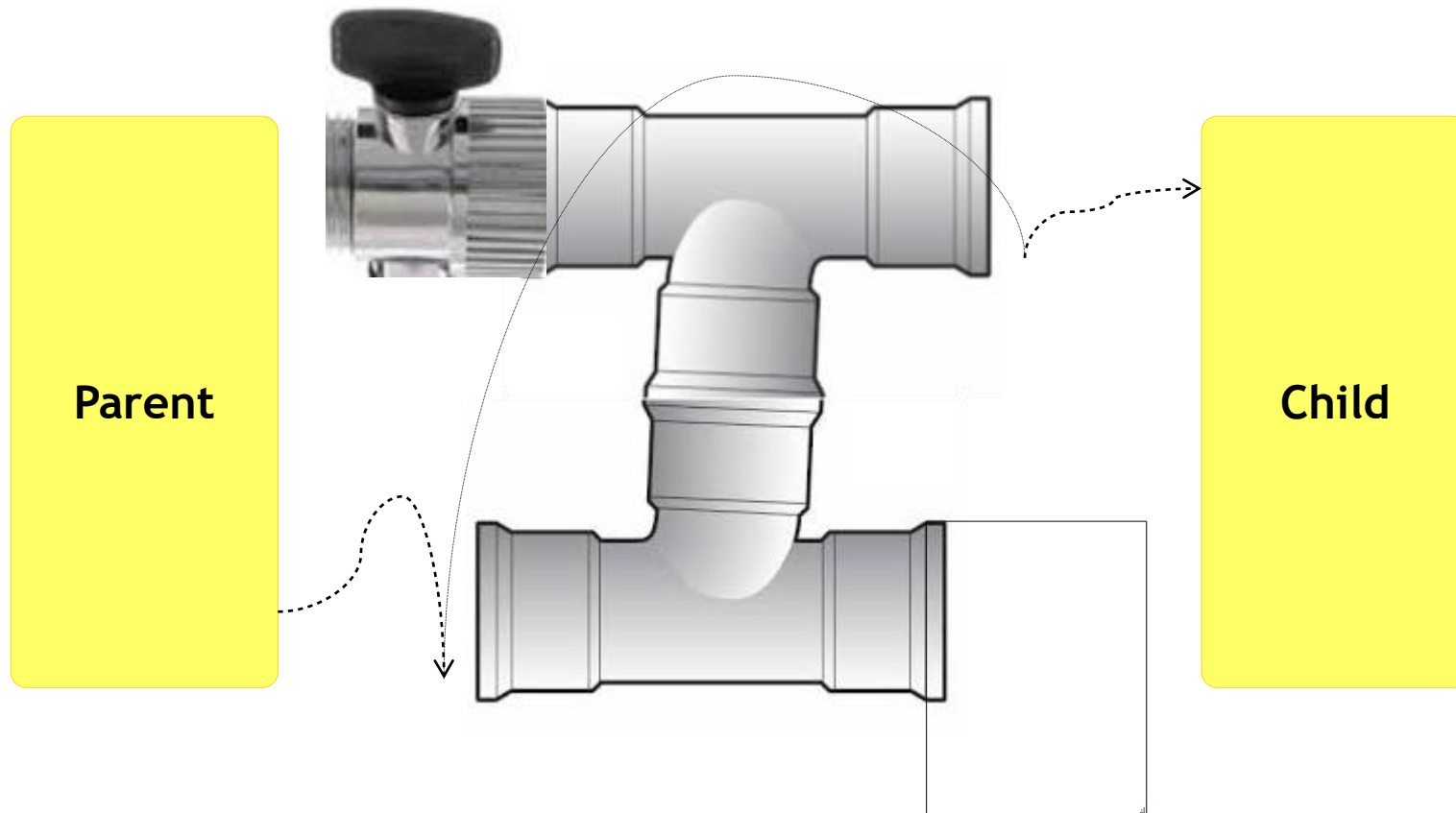


Inter Process Communications

Pipes – Direction of communication

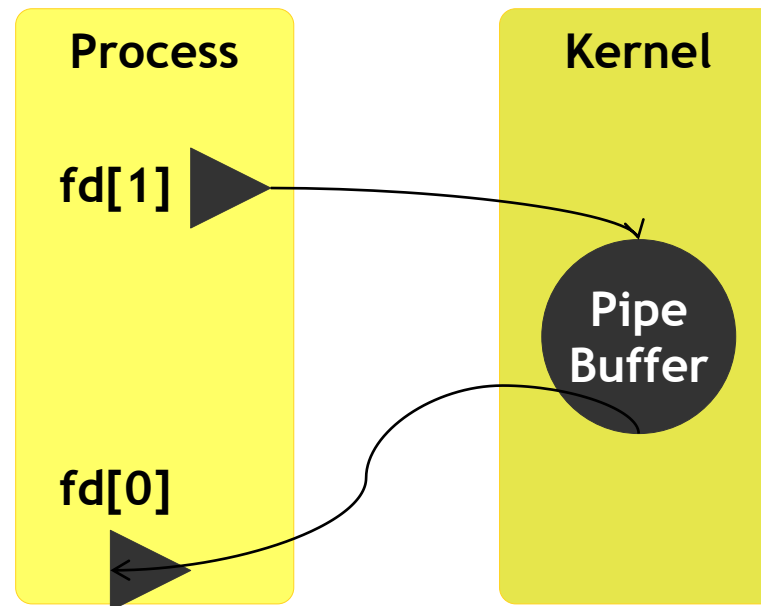


.So it necessary to close one of the end form both sides



Inter Process Communications

Pipes – Working



Inter Process Communications

Pipes - Pros & Cons



PROS

- .Naturally synchronized
- .Simple to use and create
- .No extra system calls required to communicate (read/write)

CONS

- .Less memory size (4K)
- .Only related process can communicate.
- .Only two process can communicate
- .One directional communication
- .Kernel is involved

Inter Process Communications

Summary



.We have covered

Data exchange

Communication

- .Pipes
- .FIFO
- .Shared memory
- .Signals
- .Sockets

Resource usage/access/control

Synchronization

- .Semaphores

Team Emertxe



Thank You