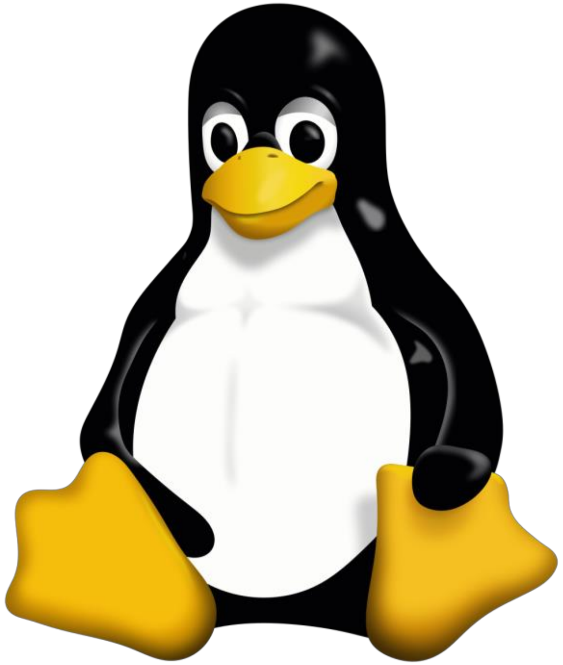


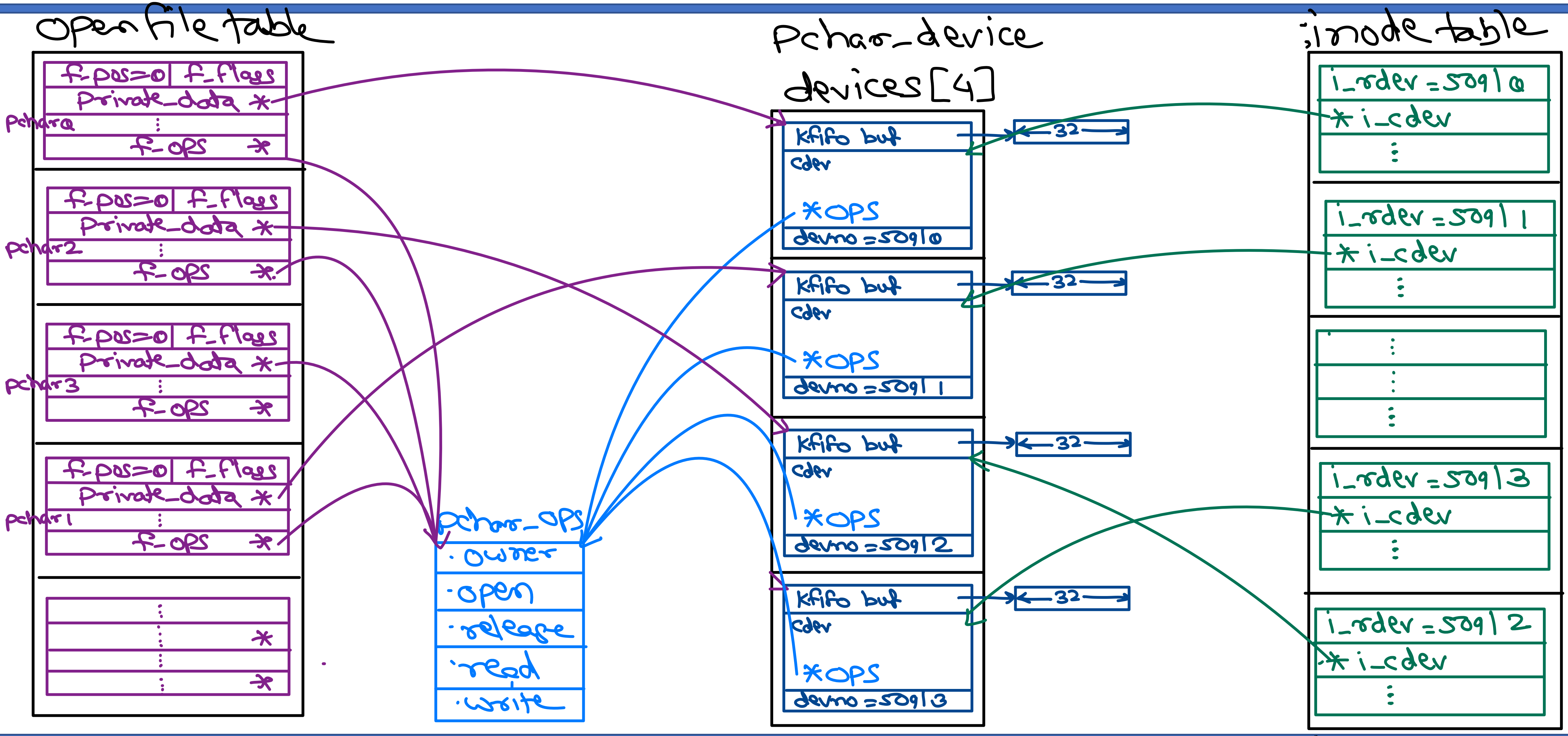
Linux Device Driver

Sunbeam Infotech

Multiple device instances

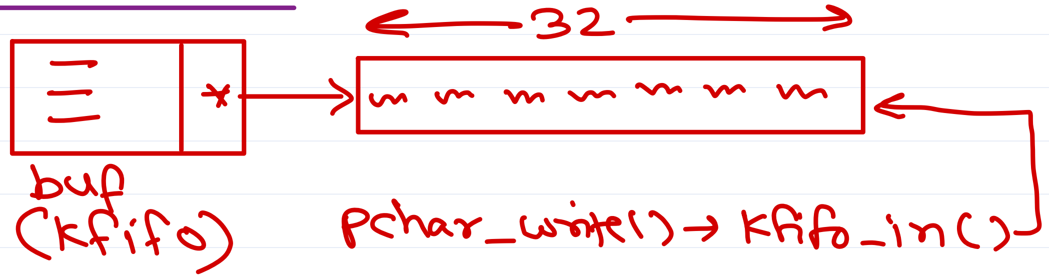
- Pseudo char device driver
  - Operations: open(), close(), read(), write().
  - Device file: /dev/pchar
  - Kernel presence: struct cdev
  - Data hold in: kernel fifo
- For multiple devices of same type (i.e. multiple device instances) single driver is used. Driver should be capable of handling multiple device data independently with the same code base (i.e. driver operations).
  - Operations: open(), close(), read(), write().
  - Device file: /dev/pchar0, /dev/pchar1, /dev/pchar2, …
  - Kernel presence: struct cdev s
  - Data hold in: kernel fifo s
- Each device operation, should be able to access respective kernel fifo. Device file should be associated with it's own data (cdev & kfifo) i.e. private_data.

Execution Flow of Pseudo Char Device Driver (Multiple device instances)



Waiting Queue

pchar device



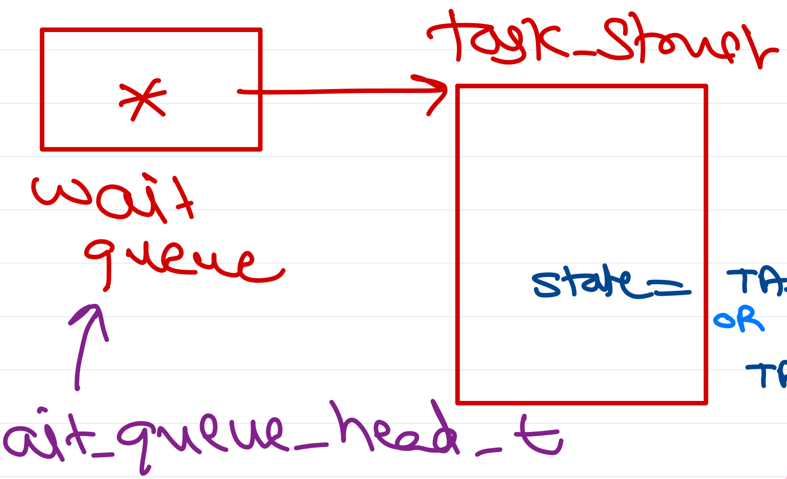
if kfifo is full, then block the writing process.

↳ wait_event_interruptible()

OR

wait_event()

kfifo_is_full()



task_struct ← of writer process (if kfifo is full)

state= TASK_INTERRUPTIBLE → blocked until data is popped from fifo.
OR
signal is received.

OR

TASK_UNINTERRUPTIBLE → blocked until data is popped from fifo.

wake up blocked process, when some data is removed from fifo → kfifo_out() - read op.

wake_up()

OR

wake_up_interruptible()

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>