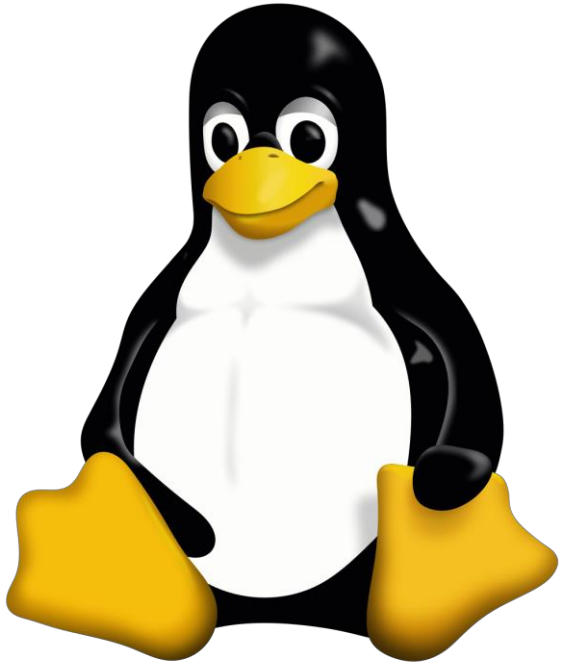
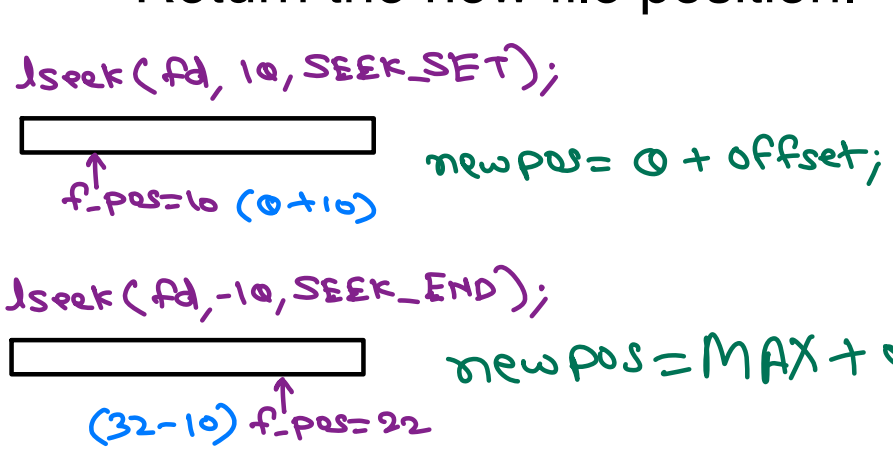


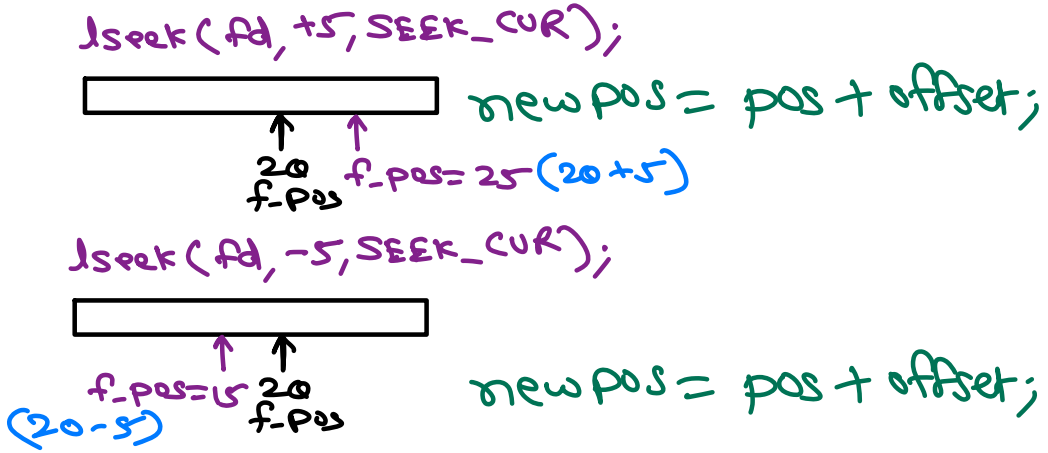
Linux Device Driver

Sunbeam Infotech

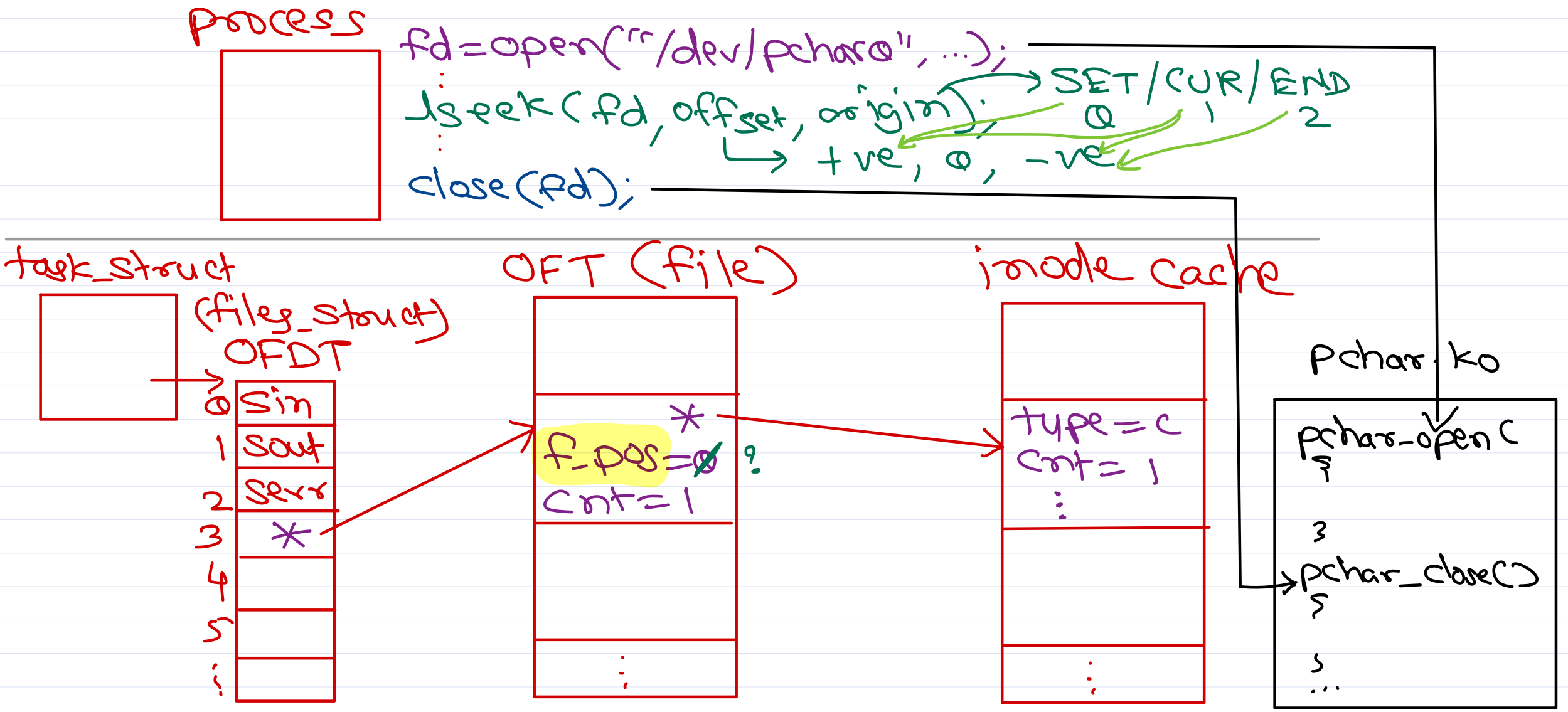
llseek() operation

- Modify current file position of the device.
- Pseudo char device driver llseek() implementation:
  - Calculate new position depending on offset and origin.
    - If origin is SEEK_SET (0), new pos = 0 + offset. (offset is +ve)
    - If origin is SEEK_CUR (1), new pos = cur pos + offset. (offset is +ve/-ve)
    - If origin is SEEK_END (2), new pos = ~~cur pos~~ MAX + offset. (offset is -ve)
  - Ensure that new file position is valid. Otherwise do the necessary adjustment.
  - Return the new file position.

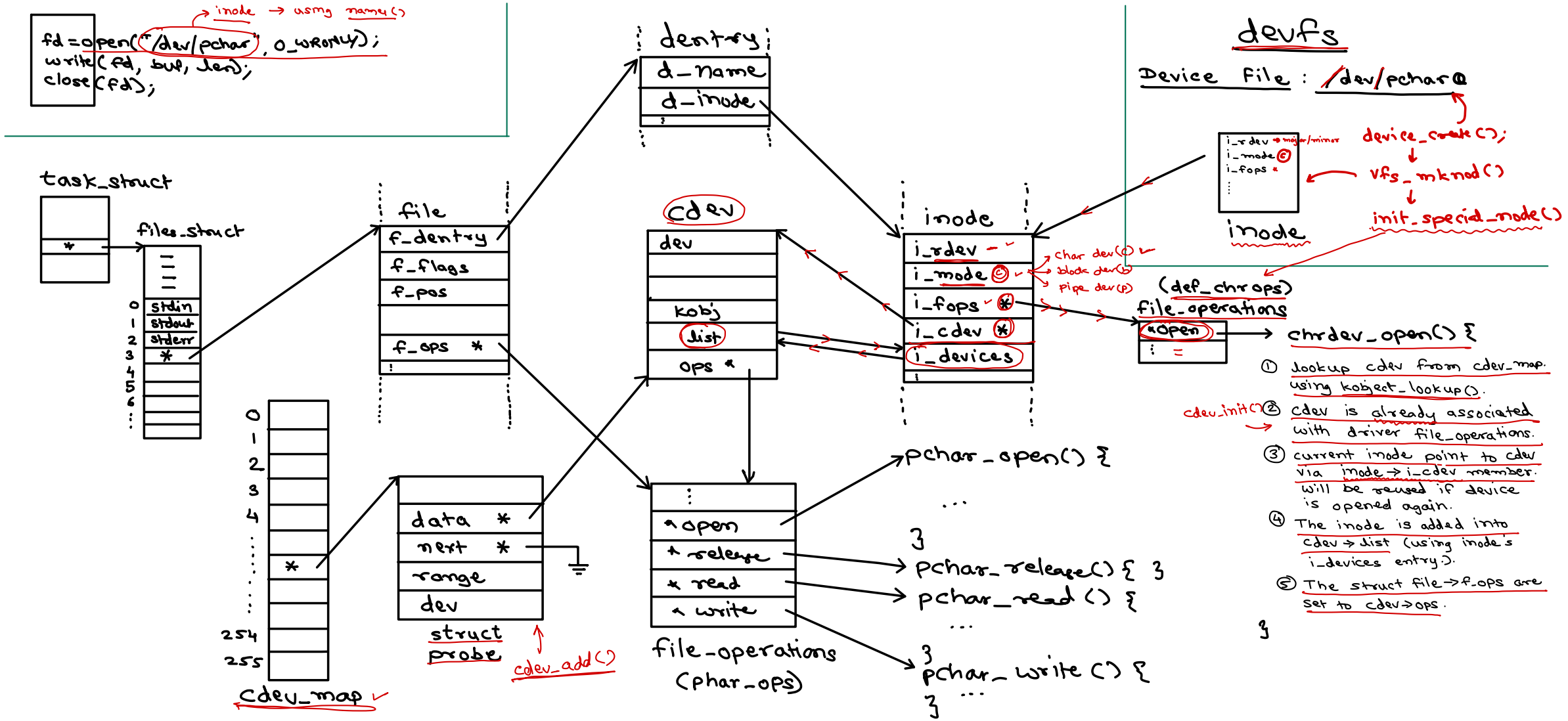




llseek() operation



Execution Flow of Pseudo Char Device Driver



ioctl() operation

- read(), write() are typical IO operations on the device.
- ioctl() system call: #include <sys/ioctl.h>
  - int ioctl(int fd, unsigned long cmd, ...);
- ioctl() is special ad-hoc operation that can be used for arbitrary purposes.
  - Manipulating device state directly.
  - Monitoring device state (debugging).
  - Direct hardware control operations.
- Example: handling CD-ROM using ioctl().
  - https://www.kernel.org/doc/Documentation/ioctl/cdrom.txt
  - e.g. ioctl(fd, CDROMEJECT, 0);
- Newer kernel version replace ioctl() with unlocked_ioctl() implementation.
  - long (*unlocked_ioctl)(struct file *pfile, unsigned int cmd, unsigned long param);

user space

```
int main() {
  int fd;
  fd = open("/dev/sr0", O_RDWR | O_NONBLOCK);
  ioctl(fd, CDROMEJECT, 0);
  close(fd);
}
```

ioctl() operation

- ioctl() operation – 2nd argument *cmd*:
  - Should be unique value throughout the kernel.
  - Old kernel version: 16 bit cmd = 8 bit device magic no (type) + 8 bit sequential value.
  - Refer Documentation/ioctl-number.txt for list of magic numbers used in kernel.
  - New kernel version: 32 bit cmd = 8 bit type (magic) + 8 bit ordinal number + 2 bit direction (NONE, READ, WRITE, READ|WRITE) + 13-14 bit width of data transfer (arch depend)
  - *cmd* argument is created using _IO(), _IOR(), _IOW(), _IOWR()
- pseudo char device driver ioctl() implementation:
  - pchar_ioctl.h: define struct for data transfer and define ioctl commands.
    - #define FIFO_CLEAR _IO('x', 1)
    - #define FIFO_GET_INFO _IOR('x', 2, info_t*)
    - #define FIFO_RESIZE _IOW('x', 3, long)
  - Implement each cmd in device ioctl operation.
  - Use copy_to_user() / copy_from_user() to transfer data from 3rd argument (if pointer).