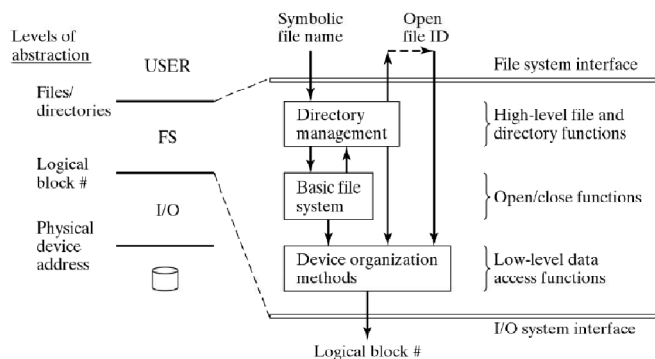


# 8. Filsystemer

## Filsystemet

1. **Directory mgmt:** Informationer om filer og mapper
2. **Basic file system:** Åbne/lukke filer, check rettigheder, open file table (kan tilgås udenfor basic file system)
3. **Device organization methods:** Står for at kommunikere med I/O systemet (som står for de reelle ændringer på disken)



## Open/close

1. **File descriptors:** owner, file type, time of creation, last use, last modification... (inode i Unix)
2. Forklar flowet når en fil åbnes/lukkes
  - a. **Open file:** Verify access rights → find og alloker fri plads i OFT → alloker read/write buffers i main memory → relevant info kopieres fra file descriptor til OFT (hvor på disk, fillængde, osv) → return index eller pointer til OFT indgang.
  - b. **Close file:** Flush modificerede main memory buffers ved at skrive deres indhold på de korresponderende diskblocks → frigør alle buffere → opdater file descriptor med info fra OFT indgang → frigør OFT indgangen.

## Organisering

1. **Filstruktur:** består af logiske records (det mindste stykke man kan henvise til af en fil) som kan organiseres på forskellige måder. **Blocking factor** (records pr blok).
2. Filsystemet ser disken som en lang sekvens af blokke. Logiske file records kan organiseres på forskellige måder (device organization methods):
  - a. **Sammenhængende blokke:** Hurtigt at tilgå, ufleksibelt (ikke særligt dynamisk).
  - b. **Kædede lister:** God udnyttelse af plads.
    - i. Hver blok peger på næste blok. Ineffektivt (mange blokke skal læses – kan være spredt over hele harddisken). Ødelagte blokke kan give manglende pointere.
    - ii. Pointere gemmes i FAT-tabel, som indlæses i RAM.
  - c. **Index tabeller:** 1 index tabel til en fil (i file descriptor)
    - i. multilevel hierarki
    - ii. incremental hierarki
    - iii. I-node indeholder 13 indgange, som kan pege på blokke. De første 10 peger på blokke. 11'te peger på indirekte blok, som peger på 128 blokke, osv.

