**File Sistemet**

Kërkesat e përdoruesit sa i përketë ruajties së të dhënave janë : *Qëndrushmëria* – Të dhënat të ruhen edhe nëse pasija fiket. *Shpejtësi* – Qasje e shpejt në të dhënat e ruajtura. *Madhësi* – Hapsirë të mjaftueshme për të ruajtur të dhënat. *Shpërndarje dhe Mbrojtie*. *Lehtësim* – Të dhënat e ruajtura të gjinden në mënyr të lehtë.

Hardueri i plotëson tri kërkesat e para : *Qëndrushmërinë* – me anë të memorieve non-valotine që nuk kanë nevoj për energji elektrike që ti ruajn të dhënat. *Shpejtësinë* – Me SSD. *Madhësinë* – Kapaciteti i disqeve rritet shum me kalimin e kohës dhe ndryshimin e teknologjisë.

OS i plotëson këto kërkesa të userave për ruajtien e të dhënave : *Qëndrushmërinë* – Me anë të redudancën mundëson rikuperimin e të dhënave në raste të dështimit (RAID). *Shpërndarjen dhe Mbrojtien* : duke ofruar privilegje për shkrim, lexim ose modifikim. *Lehtësimin* – I asocon fajllave emër i organizon fajllat në direktoriume dhe na ofron kërkim.

*File* është njësi logjike e hapsirës së ruajtur në disk, mund të përmbajn programe(kod) ose të dhëna.

*File* mund të jenë të *strukturuar(IBM)* ose të *pa strukturuar(UNIX)*.

*File* ka këto *atribute* kryesore : *Emri, mbrojtia, lokacioni, tipi, madhësia, koha e krijimit*.

*File Descriptor* tregon lokacionin e File në disk si dhe mban *atributet e* File. Ruhen në disk si File.

*Operacionet që File Sistemi ofron në të dhënat e file* : Create(), Delete(), Open(), Close(), Read(), Write(), Seek().

*Operacionet që File Sistemi ofron në emërtime : Rename(), HardLink(), SoftLink(), SetAttribute(), GetAttribute()*.

OS e ruan dhe e menaxhon një tabelë të quajtur *Open File Table* në të cilën gjinden të gjith File e hapur së bashku me atributet e tyre, numrin e hapjeve, lokacionin në memorie, lokacionin në disk. Kjo tabelë përdoret nga të gjitha processet e sistemit.

Gjithashtu OS man edhe *Per Process Table* kjo tabel është e veqant për secilin process dhe mbushet me rekorde saher që procesi hap një fajll, në rekordin e shtuar do të gjejmë një referenc për tek *OFT*, pozita e ati procesi në atë file si dhe mënyra e qasjes (r, w rw).

Kur krijojmë një File OS së pari e alokon hapsirën për atë File, pastaj e krijon një *File Descriptor* në të cilin vendos emrin e File, lokacionin dhe atributet e tjera.Gjithashtu edhe atributet zgjedhore. *Atribute zgjedhore*  i quajm ekstenzionet të cilat e reprezentojnë llojin e File, ekstenzionet e shpejtësojnë procesin e hapjes dhe detektojnë gabimet mirpo nuk janë shumë fleksibile për përdoruesit.

Kur ne e fshijmë një File OS së pari e gjen direktoriumin ku ai File gjindet, i liron blloqet që ai File i ka zënë dhe e largon *File Desc* që i përket ati File nga direktorium.

Kur hapin një File OS së pari e pranon ID dhe emrin e File të cilin duam ta hapim, në bazë të ktyre parametrave kontrollon në OFT nëse ai File është i hapur nëse nuk është e mer *File Desc* e tij dhe e vendos në OFT, pastaj e kontrollon mbrojtien e File në File Desc dhe e krahason me mënyrën që është kërkuar të hapet nëse nuk përputhen e ndërpren procesin e hapjes. Nëse përshtaten e rrit numrin e hapjeve në OFT dhe e shton një rekord në *Per Process Table* i cili referencon rekordin e sapo krijuar në OFT.

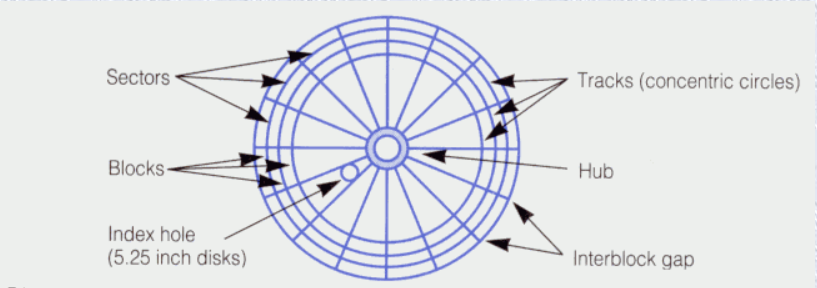
Kur duam të mbyllim një File OS e mer ID e ati File në bazë të cilës e largon nga *Per Proces Table* rekodrin që e përmban referenc për atë File në OFT, dhe ulet numri i hapjeve në OFT për rekordin e ati File. Nëse numri i hapjeve është 0 OS e largon File nga OFT.

OS ofron dy forma të leximit nga File : 1.*Qasja Sekuenciale* – Të dhënat e File lexohet bit për bit, kjo arrihet duke mbajtur referenc për bllokun e rradhës që do të lexohet në secilin bllok. Kjo qasje përdoret nga shumica e programeve si psh kompajllerët. *2.Qasja Direkte* – Të dhënat qasen me anë të vlerave kryesore për të cilave janë të asocuar blloqet ku gjenden pjesët e File. Kjo arrihet me anë të Offset që gjendet brenda File. Këtë qasje zakonisht e përdorin Databazat.

*Direktoriumet* janë struktura të të dhënave që OS i përdorë për të ruajtur *File Descriptor*.

Ekzistojnë tri strategji të emërimit të *Direktoriumeve* : 1.*Një shtresorë* – Në të gjith sistemin ekziston vetëm një Direktorium i cili ruhet në një pjesë të veqantë të diskut, ai përmbanë qifte ID – Emër për gjdo File, emri i njëjtë nuk mund të përdoret dy herë për File, Kjo formë është përdorur nga kompjuterët e mëhershëm shkaku i hapsirës. *2.Direktoriumet Dy Shtresore* – Për secilin shfrytëzues krijohet nga një direktorium i veqantë në të cilin i vendos të gjith File e vetë, edhe pse janë dy direktoriume të ndara emërtimin i File me emër të njëtë nuk lejohet. *3.Direktoriumet Shumë Shtresore* – Direktoriumet organizohen si pemë, secili direktorium ruhet si File në disk dhe ka *File Desc* i cili përmban edhe një bit etiketues për të treguar se i përket një direktoriumi.

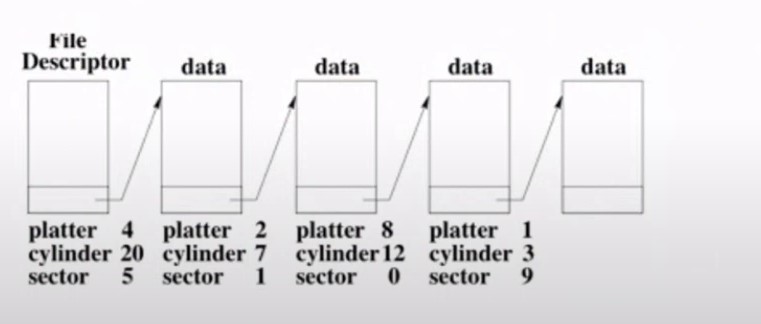
Për të ju referencuar File në mënyrë indirekte kemi dy mënyra : *1.Hard-Link* : Ka madhësinë e njëjtë me madhësinë e File org, gjdo ndryshim në File org reflekon edhe në Hard-Link. Nëse fshihet File org ne ende mund ta përdorim Hard-Link. *2.Soft-Link* : Ose short-cut ka madhësi shumë më të vogël se File org nëse File org fshihet, short cut bëhet i pa përdorshëm.



Format e menagjimit të diskut :

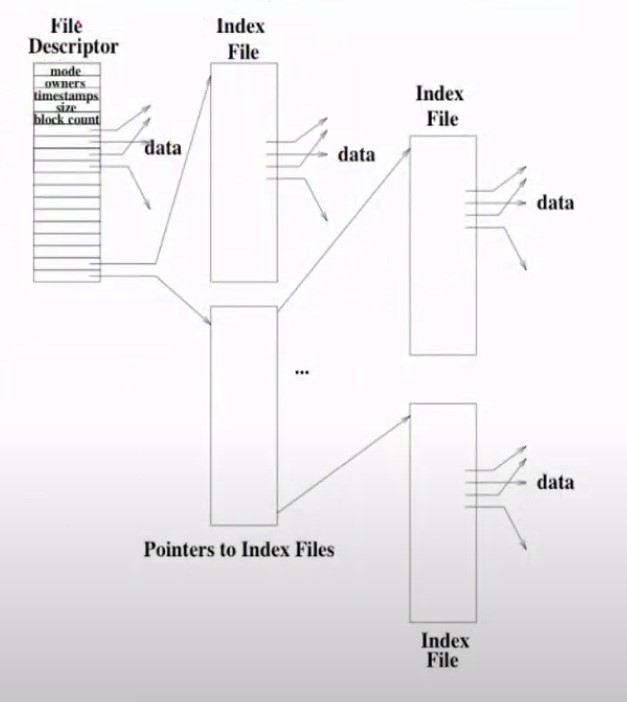
*Alokimi i vazhdueshëm i diskut* : OS e bën alokimin e vazhdueshëm me anë të një liste në të cilën gjenden të renditura blloqet e lira, kur një File krijohet dhe ruhet në disk OS i alokon hapsirë të vazdhueshme nga kjo listë dhe në File Desc të ati fajlli vendosë pozitën startuese dhe madhësinë e File. Përparësi e ksaj forme të menagjimit të diskut është shpejtësia dhe thjeshtësia mirpo ka mangësi në rastin se një File i para alokuar dëshiron të rritet në madhësi nëse nuk ka hapsir të vazhdueshme duhet bëhet lëvizja e komplet file, gjithashtu nëse ndonjë File fshihet do kemi problem me fragmentim.

*File e ndërlidhur* : OS përsëri mban listën me blloqet e lira, kur kriohet një File dhe ruhet OS e vendos File në disk nëpër blloqe, në File Desc vendos referenc drejt bllokut të parë të ati File dhe secili bllok pastaj përmban referenc për në bllokun e rradhës. Përparësi e ksaj forme të menagjimit është menjanimi i fragmentit dhe mundësia për ndryshim të madhësis pa pasur nevoj lëvizjen e File nëpër disk. Mangësi e ksaj forme të menagjimit të diskut është numri i madh i kërkimeve dhe nuk lejon qasjen e randomtë. Është përdorur nga *FAT dhe MS-DOS*.



*File e indeksuar* : OS mbanë një listë me referenca të blloqeve për secilin file, ose më saktë mbanë File Desc të File dhe secili File Desc në vete mbanë referenca drejt bllokut ku gjendet File. Madhësia maksimale e File duhet paracaktuar. Me këtë formë nuk humbasim hapsirë mirpo madhësia është e para definuar dhe kryhen shumë kërkime shaku që nuk është memoria e vazhduar.

*File e indeksuar shumë shtresorë* : Secili *File Desc* le të themi se mbanë 14 tregues, 12 të parët do jenë referenca për në blloqe që kanë të dhëna ndërsa i 13 do mbajë referenc drejt një blloku i cli mbanë referenca për në blloqe me të dhëna, kurse i 14 do të mbajë referenca drejt blloqeve me referenca drejt blloqeve që kanë referenca në të dhëna. Përparësi e ka se e lejon ndryshimin e madhësisë manësi i shfaqet kur duam të nëjm qasje direkte në fajlla të mëdhenjë dhe kryhen kërkime të mëdha,



Për të gjetur blloqet e lira në disk mund të përdoret *bitmap* secilit bllok i asocohet një bit 1 ose 0, që tregon nëse 1-është i lirë ose 0-është i zënë. Përdorimi i *bitmapit* nuk është i përshatshëm në disqet e mëdha sepse kemi të bëjmë me shumë blloqe dhe duhet të ruhet shumë *bitmap* të cilët marin hapsirë. Ose në vend të *bitmap* mund të përdorim *ndërlidhjen e blloqeve të lira,* në cache memorien e kernellit vendoset referenc drejt kreut të ksaj liste.