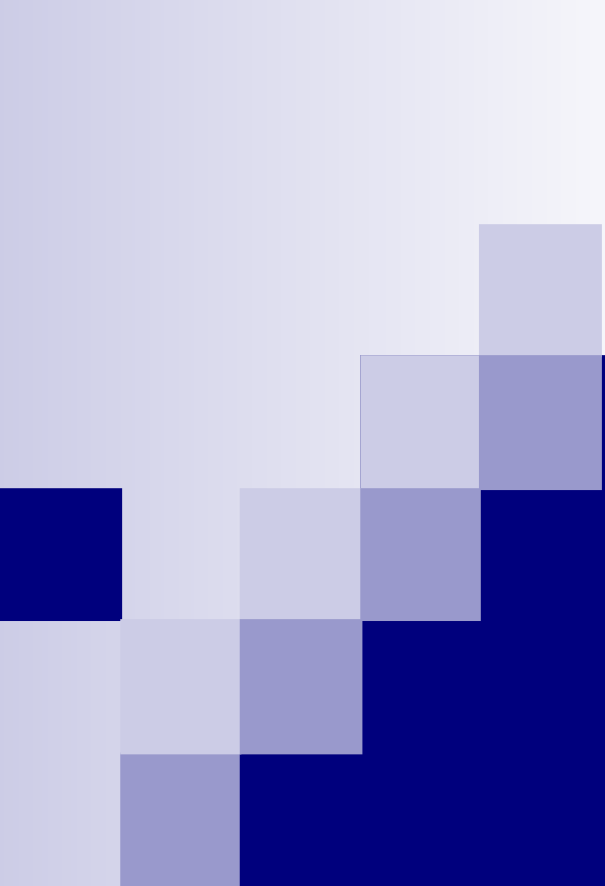
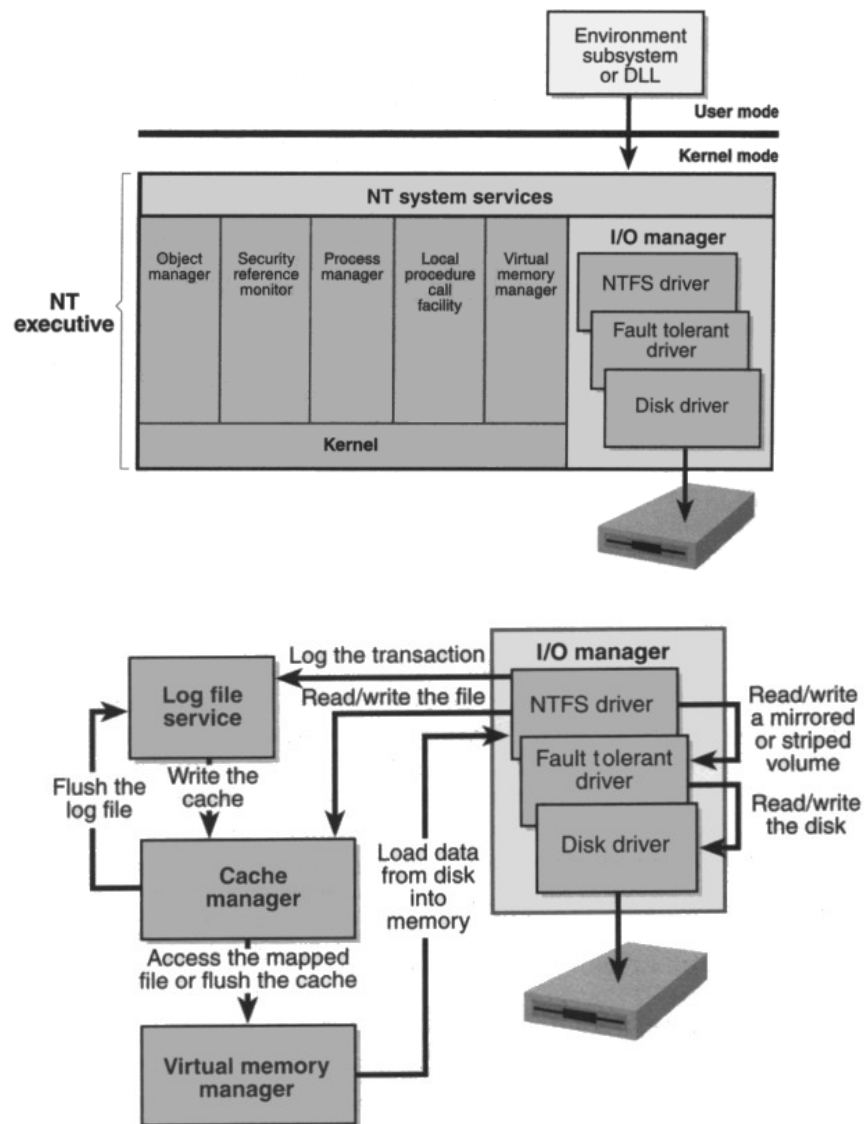




Interna struktura NTFS

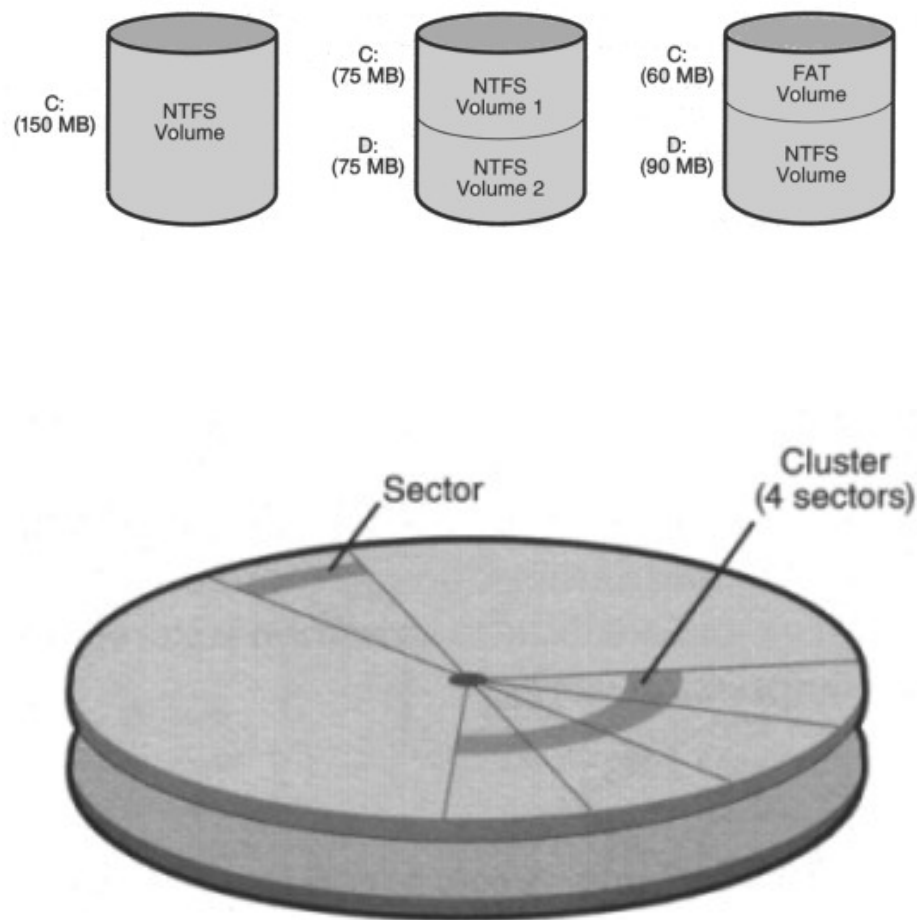
Osnovna struktura

- Predstavljen kot naložljiv gonilnik
- Okoljski podsystemi NT kličejo sistemske servise NT, ti pa nato ustrezne gonilnike.
- Gonilniki se povezujejo z:
 - Servisom LFS (Log Service File) (beleženje dogodkov)
 - Upraviteljem medpomnilnika
 - Upraviteljem navideznega pomnilnika
- Aplikativni programi dostopajo preko ročajev na objekte.



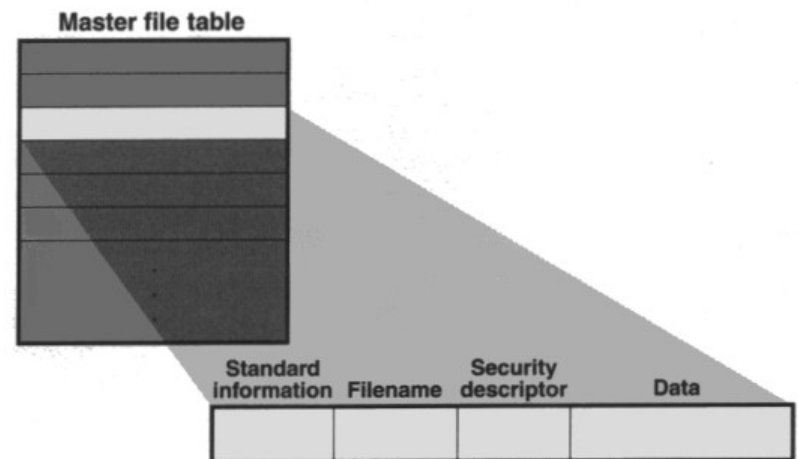
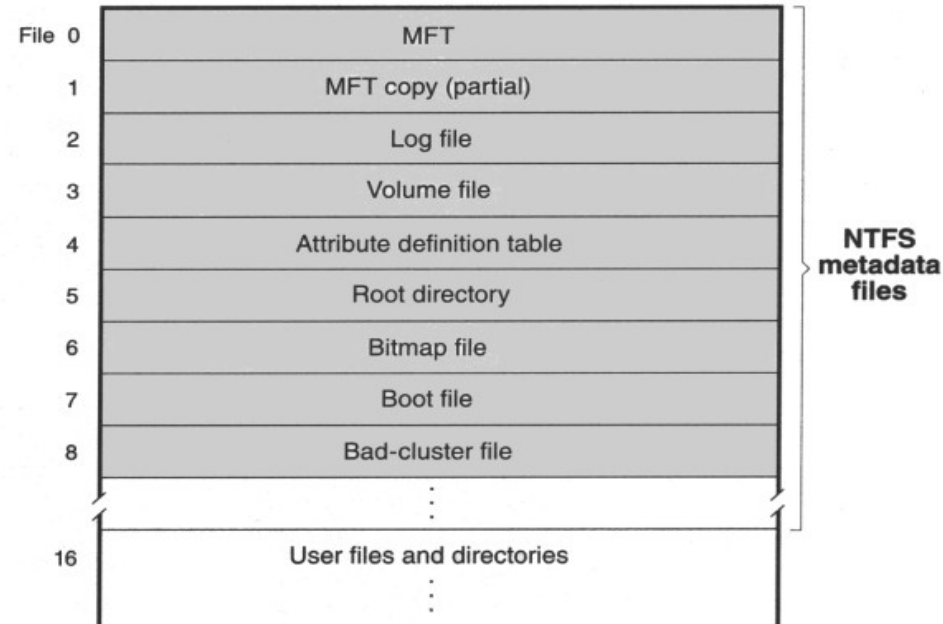
Osnovna struktura

- Osnova je zvezek (volume).
- Osnovna alokacijska enota na disku je grozd (cluster). NTFS je neodvisen od fizične velikosti diska.
- Naslavljanje lokacij preko LCN (Logical Cluster Numbers).
- Do podatkov v posameznih datotekah pridemo preko VCN (Virtual Cluster Numbers), ki se za vsako posamezno datoteko štejejo od 0 naprej.



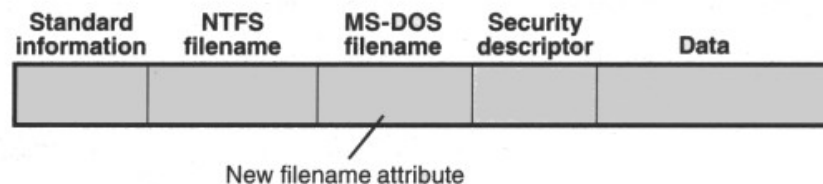
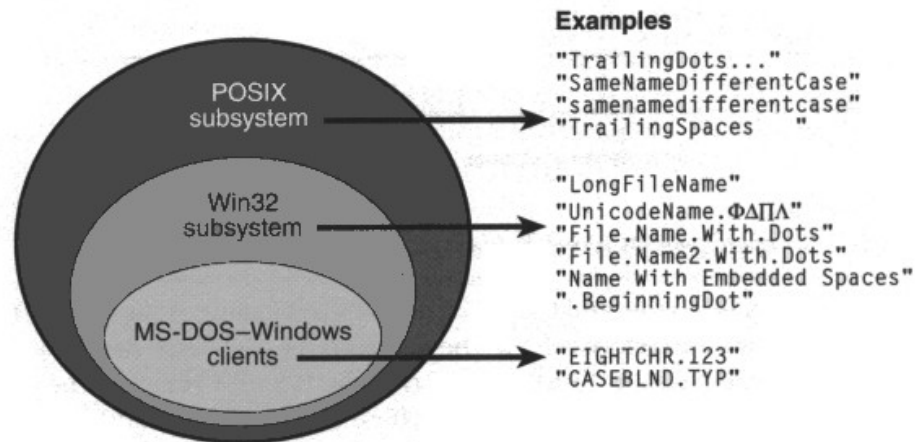
Osnovna struktura

- Srce strukture je glavna tabela datotek (MFT, Master File table).
- Vsak zapis je dolg 1k. Za vsako datoteko na disku ima MFT eno vrstico, vključno z vrstico za samo MFT.
- Množica metadata datotek.
- Vsak datotečni atribut pomnimo kot ločen tok bajtov v datotekah.



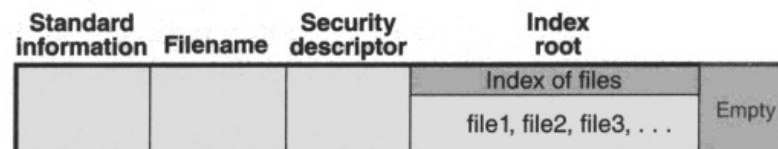
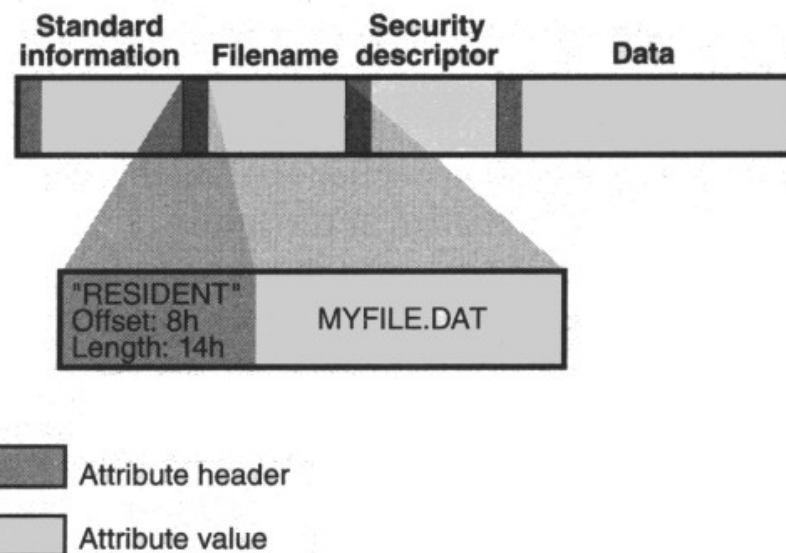
Osnovna struktura

- 255 mest dolga imena datotek.
- Imenski prostor POSIX je enak imenskemu prostoru NTFS.
- MS DOS imena datotek pomnimo v istem datotečnem zapisu kot dolga imena datotek



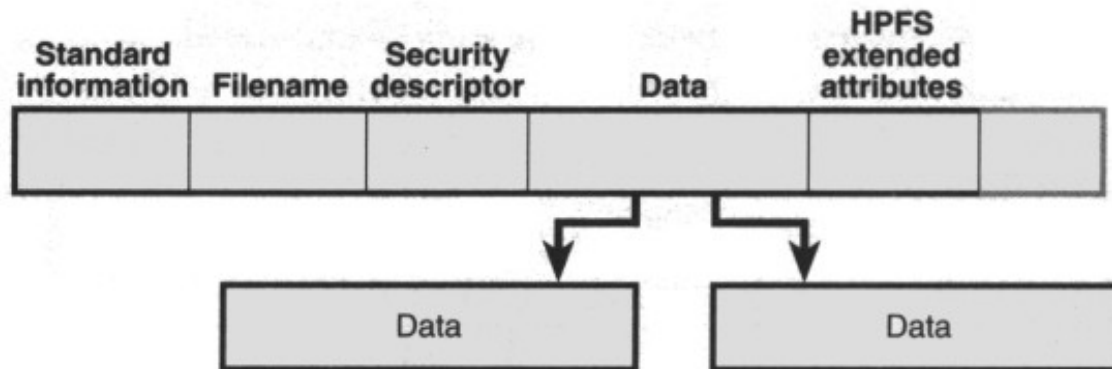
Rezidenčni in nerezidenčni atributi

- Vsak atribut začneja zaglavje (header), v katerem so podatki o atributu.
- Če je vrednost nekega atributa pomnjena kar v datotečnem zapisu, je to rezidenčni atribut.
- Če je datoteka majhna, pomnimo vse attribute in tudi njihove vrednosti (podatke) v datotečnem zapisu. Hiter dostop.
- Majhen direktorij je primer take, kratke datoteke.



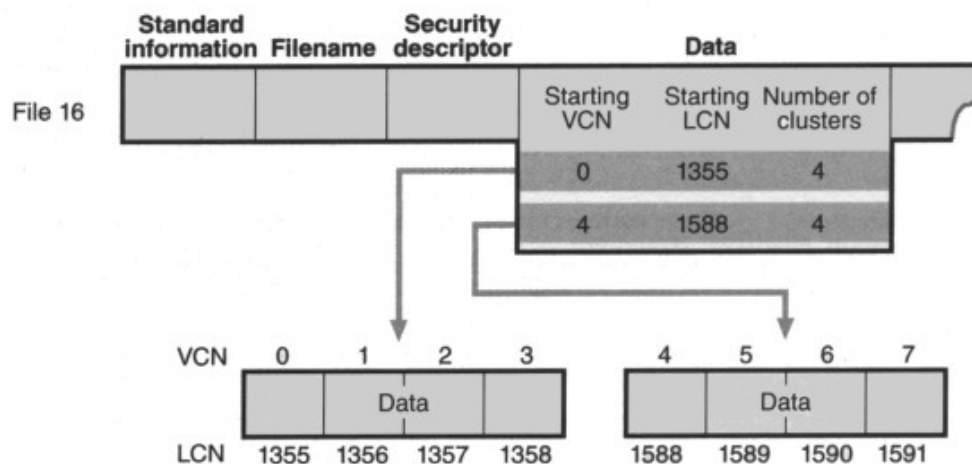
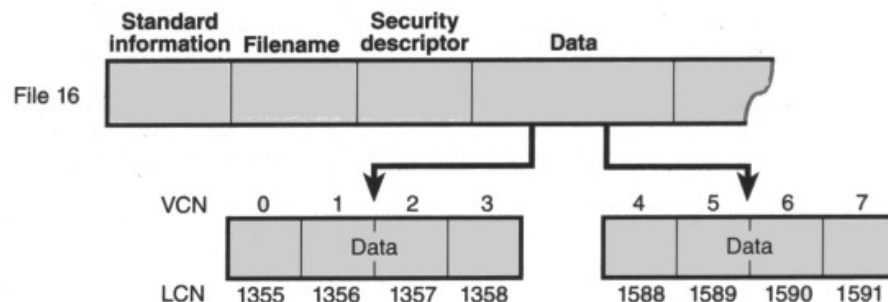
Rezidenčni in nerezidenčni atributi

- Za attribute z dolgimi vrednostmi uvede NTFS izven tabele MFT nekaj kB velik podaljšek, v katerem pomni vrednost atributa
- Atributi, katerih vrednosti so pomnjene izven MFT, so nerezidenčni atributi.



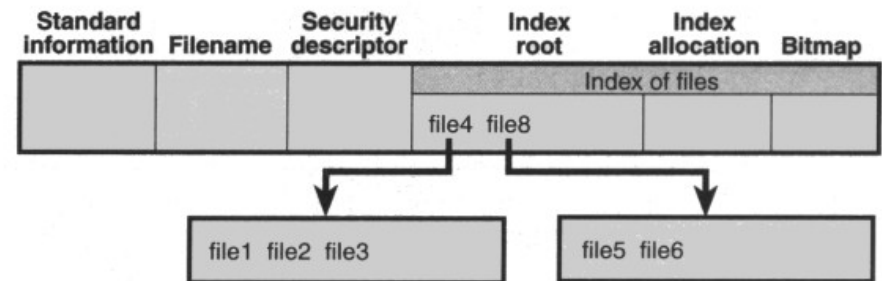
Večje datoteke

- Če atributov datoteke (ali direktorija) ne moremo shrani v datotečni zapis MFT, potrebujemo dodatne, ločene
- Slika kaže , kako zaglavje podatkovnega atributa pomn preslikavo med virtualnimi in logičnimi številkami grozdov.



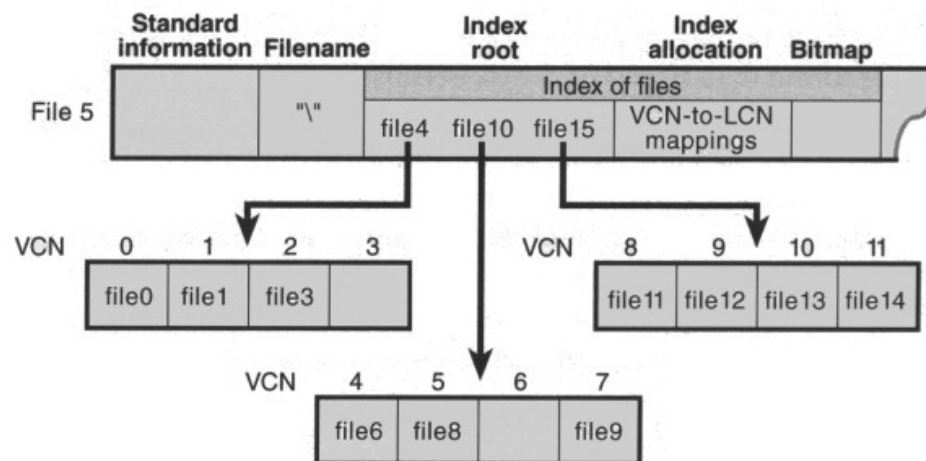
Večje datoteke

- Samo tisti atributi, ki lahko rastejo, so lahko nerezidenčni. Pri datotekah so lahko taki atributi opisniki varnosti (security descriptors), podatki, pa tudi seznam atributov.
- Standardna informacija in atribut z imenom datoteke sta vedno le rezidenčna.
- Slika kaže primer večjega direktorija. Le del indeksov datotek je pomnjen v korenskem atributu, preostali pa so pomnjeni v nerezidenčnih podaljških (ki jim pravimo "index buffers").



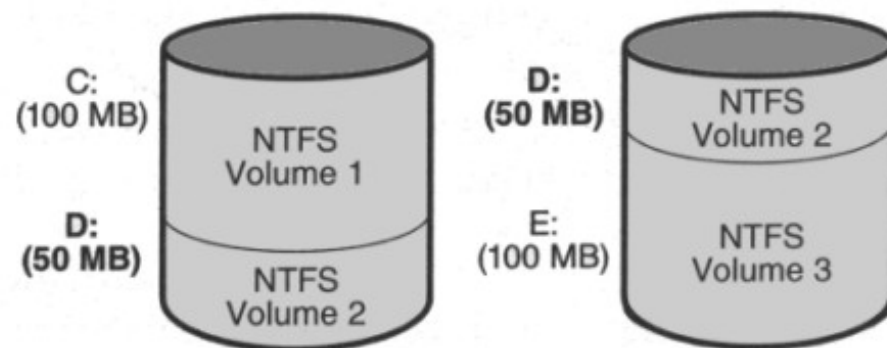
Večje datoteke

- Če ima neka datoteka preveč atributov, da bi jih lahko pomnili v MFT v enem datotečnem zapisu, uporabimo dodatni datotečni zapis in uvedemo dodatni atribut "attribute list". Pri NTFS so datotečni direktoriji kar indeksirana zbirka datotečnih imen in njihovih naslovov.
- Slika prikazuje datotečni zapis korenskega direktorija v nekem zvezku.



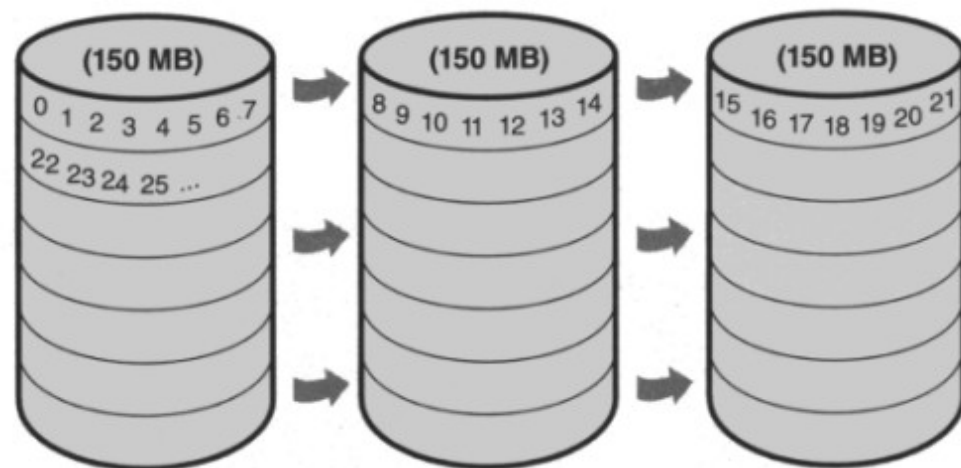
Toleranca napak

- Volume set uporabljamo za konsolidiranje večjega števila neizkoriščenih prostorov na diskih v enoten zvezek.
- Z orodjem "Windows NT Disk Administrator" združimo ta področja v enoten volume set, in ga formatiramo za izbrani datotečni sistem.
- Slika kaže tak volume set, ki je označen kot pogon D:



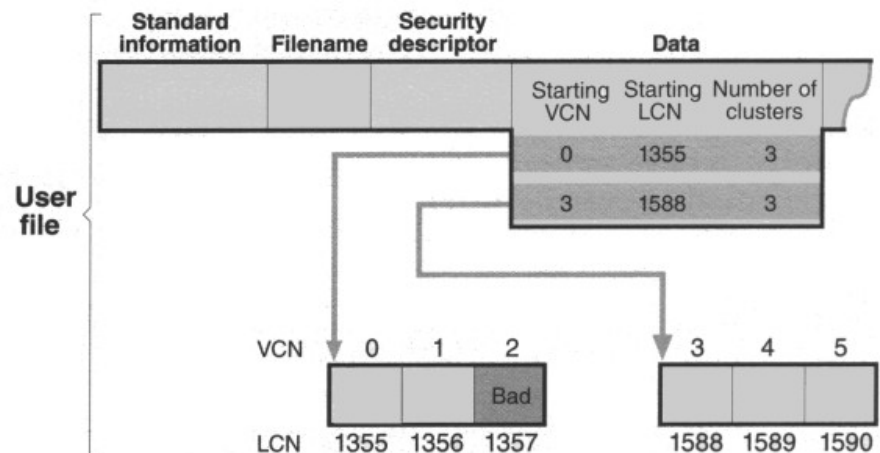
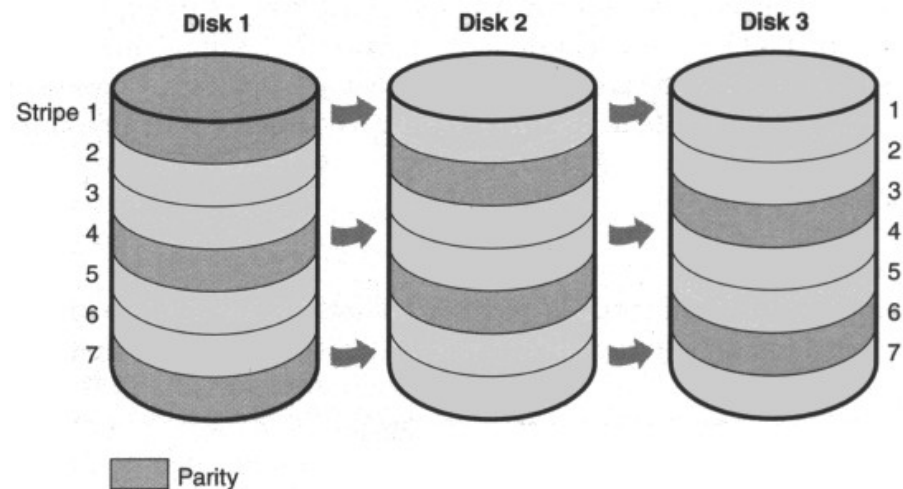
Toleranca napak

- Stripe set (stripe = proga) je zaporedje več particij, ki skupaj predstavljajo en logični zvezek.
- Sistem tak zvezek optimizira glede na čase dostopa tako, da podatke primerno porazdeli po več fizičnih diskih.
- Taka organizacija je primerna predvsem pri sistemih z veliko gostoto dostopov do diskov.



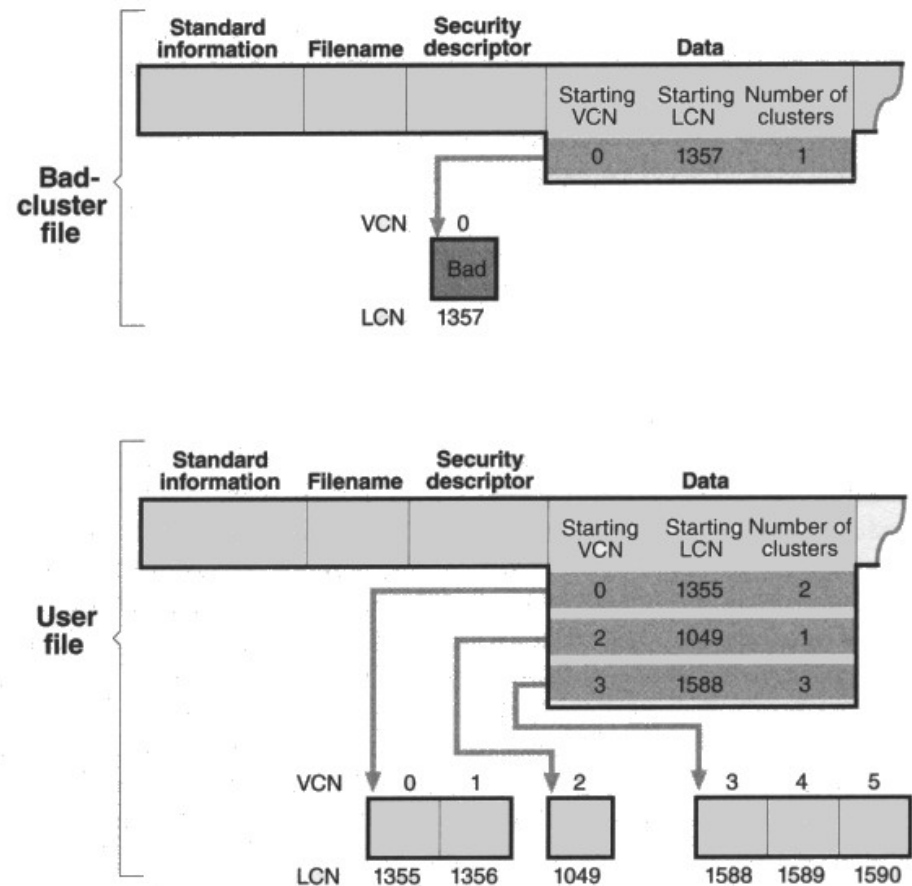
Toleranca napak

- Stripe set s parnostjo zagotavlja tolerantnost do napak tako, da rezervira ekvivalent enega diska za vsak stripe. Vidimo, da parnost kroži med diski, kar povečuje hitrost prenosa podatkov.
- Če sistem odkrije slab sektor, restavrira podatke in slab sektor nadomesti, če se da.



Toleranca napak

- Ko NTFS odkrije slab sektor, preveže celoten grozd, v katerem se tak sektor nahaja.
- V primeru, da se je to zgodilo med branjem na redundantnem zvezku, lahko restavrira podatek, sicer pa le vrne ugotovitev, da je podatek napačen, v bodoče pa tak slab sektor ne bo več klican.





Viri:

- <http://colos1.fri.uni-lj.si/~sis/>
- <http://www.digit-life.com/articles/ntfs/>
- <http://en.wikipedia.org/wiki/Ntfs>