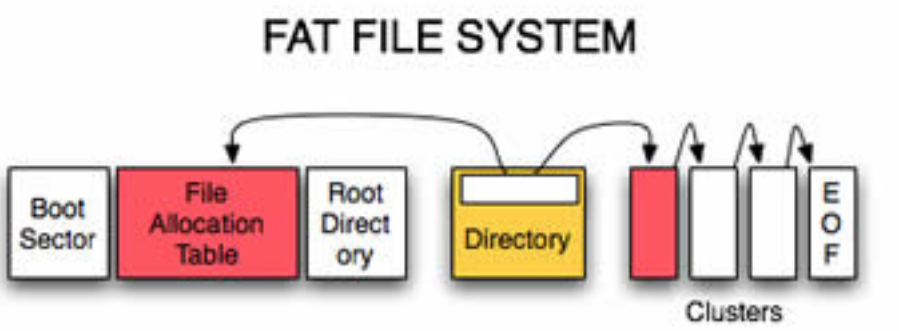
FAT iznutra

FAT je prilično jednostavan datotečni sustav. U prošlom [1] nastavku upoznali smo njegova ograničenja, koja nisu predstavljala problem dok su se koristili diskovi malog kapaciteta. Pratili smo kako se razvijao da bi mogao adresirati sve veće diskove. Na Windowsima ga je s vremenom zamijenio moćniji NTFS. Ali nakon toga FAT nije potiho nestao, već se preselio na mobilne i embedded uređaje. Vrijeme je da upoznamo unutarnju strukturu FAT *filesystema*.

Najbrže ćemo je prikazati ovom slikom:



Na početku diska je boot sektor, čija je zadaća pokretanja OS-a. Boot sektor je tu bez obzira na to koji se operacijski sustav koristi. Ovog se časa nećemo baviti njime. Slijedi FAT tablica (*File Alocation Table*), koja sadrži popis klastera i njihov status, koji može imati jednu od četiri vrijednosti: zauzet (allocated), slobodan, kraj datoteke, loš klaster. Nakon FAT tablice smješten je root direktorij, koji sadrži popis datoteka i poddirektorija, njihov naziv plus početni klaster. Time završava sistemski dio diska i počinje podatkovni dio, na kojem su pohranjene datoteke.

Podsjetimo se: na disku se koristi sektor kao najmanja fizička jedinica za zapis podataka. Klaster je Microsoftov naziv za blok od više sektora, koji datotečni sustav koristi kao najmanju logičku jedinicu za zapis podataka. Klaster može imati veličinu od jednog (512 bajtova) do 128 sektora (65536 bajtova). Veličina klastera upisana je u *boot* sektoru. Ne smijemo smetnuti s uma da sektori unutar klastera uvijek slijede jedan za drugim, dakle klaster je neprekinut slijed zadanog broja sektora. I još jedna činjenica koju uvijek treba imati na umu: klasteri se koriste samo na podatkovnom dijelu diska, dok se u sistemskom dijelu računaju samo sektori!

Pogledajmo oznake za status klastera na FAT12:

0x000 (Free Cluster)
0x001 (Reserved Cluster)
0x002 - 0xFEF (Used cluster; value points to next cluster)
0xFF0 - 0xFF6 (Reserved values)
0xFF7 (Bad cluster)
0xFF8 - 0xFFF (Last cluster in file)

Na FAT16 verziji sve je na prvi pogled isto, samo se koristi više bitova, pa je iza 0x dodana još jedna nula:

0x0000 (Free Cluster)
0x0001 (Reserved Cluster)
0x0002 - 0xFFEF (Used cluster; value points to next cluster)
0xFFF0 - 0xFFF6 (Reserved values)
0xFFF7 (Bad cluster)
0xFFF8 - 0xFFFF (Last cluster in file)

Kad je u pitanju FAT32, stvari se, bar na prvi pogled, malo kompliciraju.

0x?0000000 (Free Cluster)
0x?0000001 (Reserved Cluster)
0x?0000002 - 0x?FFFFFEF (Used cluster; value points to next cluster)
0x?FFFFFF0 - 0x?FFFFFF6 (Reserved values)
0x?FFFFFF7 (Bad cluster)
0x?FFFFFF8 - 0x?FFFFFFF (Last cluster in file)

Čemu upitnici? Radi se o tome da se umjesto 32 bita koji su na raspolaganju koristi samo 28. Ne pitajte zašto, možda FAT28 ne zvuči binarno zaokruženo kao FAT16 i FAT32!? Upitnik iza 0x naprosto označava da tih 4 bita ne treba uzimati u obzir. U praksi, na tom mjestu možemo očekivati nule.

Ahilova peta takve organizacije podataka je u fiksnoj poziciji metapodataka: ako se područje diska koje zauzima FAT tablica na neki način ošteti, datoteke su još na disku, u podatkovnom dijelu, ali bez metapodataka korisnik ne može do njih. Rizik se umanjuje uvođenjem još jedne kopija FAT tablice, smještene odmah iza originalne. Tako na datotečnom sustavu FAT32 koristimo tablicu FAT1 i njenu kopiju FAT2.

Da ne bi sve bilo jednostavno, Microsoft dozvoljava da se isključi "FAT mirroring" i koristi samo jedna od dviju FAT tablica, ne nužno FAT1! Čemu takva komplikacija? Možda je to bio zahtjev korisnika FAT-a? Ili se radi tome da se koristi tablica koja nije oštećena? U svakom slučaju, pri hakiranju FAT32 treba biti spreman na to da će FAT1 u nekim slučajevima biti neaktivna, a FAT2 primarna tablica.

Kad bismo zadali naredbu za brisanje datoteke, namjerno ili slučajno, zapravo bismo samo promijenili prvi bajt naziva datoteke: preko njega se prepiše specijalni znak 0xE5 koji OS-u kaže da zanemari tu datoteku, a zatim se klasteri koje datoteka zauzima proglase slobodnim upisivanjem vrijednosti 0x0000 u FAT tablice. Podaci nisu nestali, samo im više ne možemo pristupiti s razine OS-a. Treba nam neki napredan alat s kojim možemo izravno pristupiti disku, kako bismo poništili brisanje. DOS je imao naredbu debug, elementarni editor diska, ali taj baš i nije bio jednostavan za korištenje, pa su se na tržištu prodavali programi drugih tvrki koji su mogli napraviti "undelete". Osamdesetih godina prošlog stoljeća najpoznatiji su bili Norton Utilites.

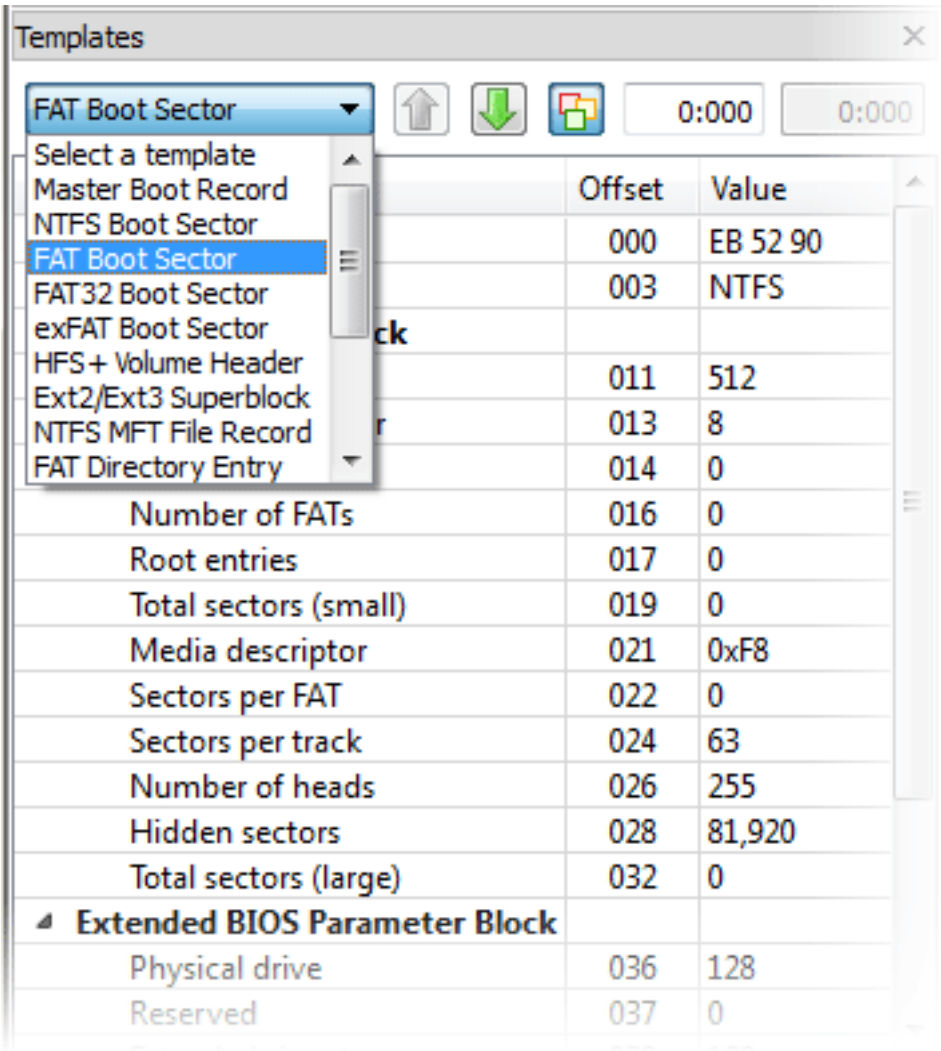
Peter Norton objavio je 1982. svoje alate za DOS, Norton Utilites, skup programa koji su DOS činili moćnijim. Bilo je tu svega i svačega: Beep bi izazvao pištanje zvučnika, što se dalo zgodno iskoristiti u batch programima, Clear bi izbrisao ekran, Reverse bi na tada crno bijelim ekranima zamijenio boje pozadine i teksa, Print bi ispisao datoteku (rane verzije DOS-a nisu imale tu naredbu). Ali već u toj prvoj verziji Nortonovih dodataka bili su alati za spašavanje datoteka s disketa, UnErase, FileFix. Kako su se s vremenom pojavljivale nove verzije DOS-a, Norton je nastavio razvijati svoje Utilities, pa je tako verzija 4,5 iz 1988. donijela Norton Disk Doctora i Disk editor. Mogli smo koristi i Nortonov alat UnErase, ali smo se radije pravili važnima editirajući izravno sadržaj direktorija i FAT tablice pomoću Disk editora.

Tvrtku Norton u međuvremenu je kupio Symantec, ali su zadržali ime Norton Utilites. Dodali su vlastite programe i nastavili razvijati alate za nove verzije Windowsa (do verzije Norton Utilites 16).

Izdali su i verziju za Apple Macove.

S obzirom na to je FAT još uvijek u upotrebi, navedene tehnike spašavanja podataka još uvijek bi mogle zatrebati i mlađim kolegama.

Norton Utilites se još uvijek mogu kupiti, ali nas prvenstveno zanimaju programi otvorenog koda, pa smo se bacili u potragu za slobodnom verzijom Disk Editora. Nije nam dugo trebalo da pronađemo program zanimljivog naziva: Active@Disk Editor. Može se skinuti s adrese http://disk-editor.org [2] i to kao .exe datoteka za Windowse ili .run za Linux. Ovaj je disk editor zaista napredan, ima ugrađene predloške za različite datotečne sustave i olakšava nam snalaženje na njima tako što iz izbornika možemo birati čemu želimo pristupti: u slučaju FAT-a boot sektoru, FAT1/FAT2 tablici, root direktoriju itd. Uz to nam pomaže u tumačenju značenja inače kriptičnih zapisa u obliku niza heksadecimalnih brojeva koji nisu "human friendly".



Active@Disk Editor poznaje i druge datotečne sustave: NTFS, exFAT, ext, XFS, ReFS itd. Dakle bit će nam koristan alat kojeg svakako treba dodati sistemčevu kompletu za preživljavanje. O tome kako se koristi i što se sve s njim može raditi nekom drugom prilikom.

ned, 2018-09-30 17:16 - Aco Dmitrović**Kategorije:** Operacijski sustavi [3]
Vote: 0

No votes yet

story_tag: FAT [4]
datotečni sustav [5]

Source URL: https://sysportal.carnet.hr./node/1823

Links
[1] https://sysportal.carnet.hr/node/1817
[2] http://disk-editor.org
[3] https://sysportal.carnet.hr./taxonomy/term/26
[4] https://sysportal.carnet.hr./taxonomy/term/259
[5] https://sysportal.carnet.hr./taxonomy/term/263