# Popis souborového systému SZ-FAT

# (semestrálka ZOS FAT, verze: 2015.12.09. 11:54)

Datový blok začíná strukturou **boot record**, která obsahuje informace nutné pro další práci s daty

struct boot\_record {

char volume\_descriptor[251];

//popis

int fat\_type;

//velikost FAT (12,16,32)

int fat\_copies;

//počet FAT tabulek

unsigned int cluster\_size;

//velikost clusteru

long root\_directory\_max\_entries\_count;

//počet zaznamu v rd

unsigned int cluster\_count;

//celkovy počet clusteru na FAT

unsigned int reserved\_cluster\_count;

//počet rezervovaných clusterů (o kolik se zmenší FAT použitelnost)

char signature[4];

//OK/NOK

}

Následuje fat\_copies počet **FAT tabulek**. Položky tabulek clusterů jsou datového typu unsigned int.

FAT může obsahovat vyhrazené hodnoty

…

FAT\_UNUSED = 65535;

FAT\_FILE\_END = 65534;

FAT\_BAD\_CLUSTER = 65533;

Nebo adresu clusteru s další částí souboru

Následuje **root\_directory**, jenž obsahuje položky typu

struct root\_directory {

char file\_name[13];

//jmeno 8+3

char file\_mod[10];

//atributy

short file\_type;

//0 – adresar, 1 - soubor

long file\_size;

//velikost souboru

unsigned int first\_cluster;

//kde soubor v datech zacina – index FAT

}

Adresa za adresářem je offset, od kterého se adresují datové clustery. Tedy první cluster s daty má adresu 1 + offset (boot\_record + n \* fat + root\_directory + cluster\_count – reserved\_cluster\_count)

* Všechny data ve FAT budou ukládána ve formě řetězců ve formátu C (tedy XXX/0) – Délka v definicích struktur již počítá s tímto znakem navíc.
* V datech souboru (clustery) pak je potřeba počítat s tím, že pokud cluster má velikost 32B (32 znaků) bude významových pouze 31, 32 bude vždy /0.
* Pokud bude jakýkoliv řetězec kratší, jak maximální délka, bude ukončen /0 a doplněn nevýznamovými znaky do dané délky (bude vidět v datech)
* Chybný datový cluster bude obsahovat na začátku a na konci dat řetězec „FFFFFF“ a poté významové data – vyskytovat se bude pouze v úloze se špatnými sektory. Tento řetězec se nikde jinde ve významových datech vyskytovat nebude (délka významových dat tedy bude o 2x8 + 1 (/0) kratší, jak velikost clusteru). U neúplného clusteru bude opět po FFFFFFToto je textFFFFFF/0 doplněno nulami do zbytku clusteru.
* Data jsou generována na 64-bit systému, je vhodné kompilovat Vaše programy také na 64-bit systému
* Můžete předpokládat největší FAT soubor s daty o velikosti 4GB (tedy všechna data se vejdou do paměti)
* Bloky root\_directory, boot\_record je třeba načítat jako celé struktury (definice viz. Strana 1.) Ukádány jsou pomocí funkce:

fwrite(&promena\_struktury, sizeof(promena\_struktury), 1, \*FILE);