



Pokročilé operační systémy

Souborové systémy

Petr Krajča



Katedra informatiky
Univerzita Palackého v Olomouci



- nejnižší úroveň abstrakce
- není potřeba přístup k HW
- lze emulovat v uživatelském prostoru
- viz `blk_dev.h`, `file_blk_dev.c`, `file_blk_dev.h`



- 1 superblock – metadata souborového systému
- 2 oblast inode – metadata jednotlivých souborů (pevná velikost)
- 3 bitmapy
 - volných inode
 - volných bloků
- 4 datová oblast
 - praktické implementace mají několik instancí (groups) těchto struktur
 - viz `mufs.h`, `mufs.c` (pomocné funkce pro práci s inode, bitmapami, clustery)

bloky vs. clustery

- blok – nejmenší jednotka dat poskytovaná blokovým zařízením (často sektor)
- cluster – shluk bloků optimalizující přístup k zařízení (jiné označení – block, zone)
- s metadaty vhodnější pracovat po blocích, s daty po clusterech



```
int mufs_format(struct blk_dev *dev, uint32_t blks_per_cluster,  
               uint32_t inodes_cnt);
```

```
int mufs_open(struct mufs *fs, struct blk_dev *dev);
```

```
inode_id mufs_inode_create(struct mufs *fs, uint32_t type,  
                           struct inode *inode);
```

```
int mufs_write(struct mufs *fs, struct inode *inode,  
               unsigned char *data, uint64_t offset, uint64_t size);
```

```
int mufs_read(struct mufs *fs, struct inode *inode, unsigned char *buf,  
              uint64_t offset, uint64_t size);
```

```
int mufs_truncate(struct mufs *fs, struct inode *inode, uint64_t size);
```



- vhodné pracovat po celých clusterech
- tři typy záznamů soubor, adresář, volné místo
- procházení iterátorem

```
struct dir_entry {
    uint16_t rec_len; /* velikost zaznamu vctne hlavicky */
    uint8_t type; /* typ zaznamu */
    uint8_t reserved; /* vyhrazeno, zarovnani */
    inode_id inode; /* inode souboru */
    char fname[255]; /* jmeno souboru vctne \0 */};

int mufs_dir_create(struct mufs *fs, inode_id parent_dir,
                  struct inode *dir_inode);

int mufs_dir_query(struct mufs *fs, struct dir_query *query,
                  struct inode *dir_inode, unsigned char *cluster_buf,
                  uint8_t query_type, void *constraint);

struct dir_entry *mufs_dir_next(struct dir_query *query);

int mufs_dir_link(struct mufs *fs, struct inode *dir_inode, inode_id inode,
                  char *fname, int is_dir);

int mufs_dir_unlink(struct mufs *fs, struct inode *dir_inode, char *fname);
```



- inode jako struktura v paměti (sdílení přístupu)
- práce s cestami ve FS
- ukazatele do souboru
- manipulace s obsahem adresářů (počítání odkazů)

```
struct mufs_fd *mufs_file_open(struct mufs *fs, char *path, int flags);
int mufs_file_close(struct mufs_fd *fd);
uint64_t mufs_file_write(struct mufs_fd *fd, unsigned char *buf,
                        uint64_t size);
uint64_t mufs_file_read(struct mufs_fd *fd, unsigned char *buf,
                        uint64_t size);
uint64_t mufs_file_seek(struct mufs_fd *fd, int64_t offset, int whence);
int mufs_file_eof(struct mufs_fd *fd);
int mufs_file_unlink(struct mufs *fs, char *path);
int mufs_hardlink(struct mufs *fs, char *source_path, char *dest_path);
```



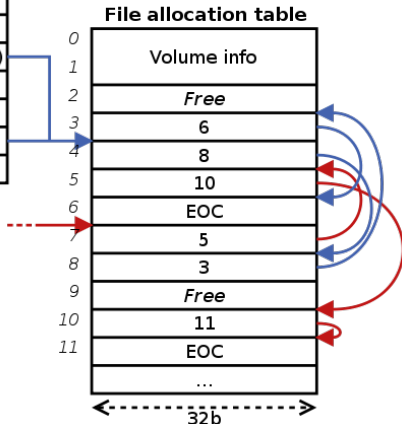
```
int mufs_mkdir(struct mufs *fs, char *path);
struct mufs_dird *mufs_dir_open(struct mufs *fs, char *path, int flags);
struct dir_entry *mufs_readdir(struct mufs_dird *dird);
void mufs_dir_close(struct mufs_dird *dird);
int mufs_rmdir(struct mufs *fs, char *path);
```

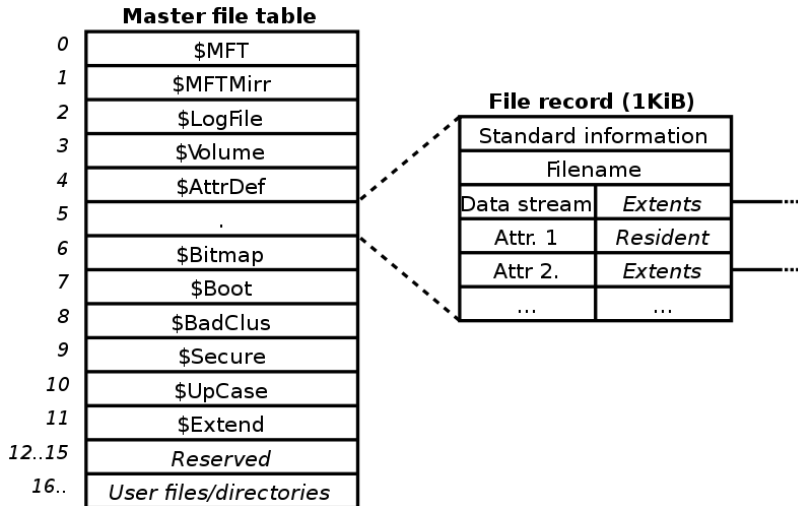


- práce s bitmapami
- odkazy na bloky dat
- struktura adresářů
- přístup k datům
- cf.
 - FAT*
 - NTFS
 - XFS
 - BtrFS

Directory table entry (32B)

Filename (8B)
Extension (3B)
Attributes (1B)
Reserved (1B)
Create time (3B)
Create date (2B)
Last access date (2B)
First cluster # (MSB, 2B)
Last mod. time (2B)
Last mod. date (2B)
First cluster # (LSB, 2B)
File size (4B)





```
struct btrfs_disk_key {
    __le64 objectid;
    u8 type;
    __le64 offset;
}
```





- 1** Vytvořte aplikaci nebo nástroj, která bude manipulovat s obsahem souborového systému přes jeho vysoúrovňové rozhraní. Tj. bude vytvářet, číst, zapisovat soubory, případně manipulovat s adresáři.
- 2** Vytvořte aplikaci nebo nástroj, která zobrazí metadata souborového systému a souborů.
 - Pro souborový systém zobrazí jeho vlastnosti včetně informací o aktuálně využitých inodech a clusterech dat.
 - Pro soubor zobrazí dostupné informace včetně toho, v kterých blocích se jeho data nachází.