FAKULTA
APLIKOVANÝCH VĚD
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI

Západočeská univerzita

Defragmentace pseudoFAT tabulky

Semestrální práce - UPS

Petr Štechmüller

1. ledna 2017

Obsah

Zadání

Vytvořte jednoduchý souborový systém, který bude organizován pomocí pseudoFAT tabulky. Nad souborovým systémem budete provádět následující operace:

- Vytvoření nového souboru
- Smazání souboru
- Vypsat čísla clusterů obsahující data zadaného souboru
- Vypsat obsah souboru
- Vytvoření prázdné složky
- Smazání prázdné sožky
- Vypsat stromovou strukturu celého souborového systému
- Provést úplnou defragmentaci souborového systému

Implementace

Souborový systém se snaží simulovat prostředí v linuxu. Jako znak oddělující jednotlivé soubory a složky od sebe je použito "/". Kořenový adresář se jmenuje "/". Při implementaci byla použity následující struktury pro boot record1 a root directory2:

```
struct boot_record {
    char volume_descriptor[251];
    int fat_type;
    int fat_copies;
    unsigned int cluster_size;
    unsigned long root_directory_max_entries_count;
    unsigned int cluster_count;
    unsigned int reserved_cluster_count;
    char signature[4];
};
```

Listing 1: Struktura pro boot record

```
struct root_directory {
    char file_name[13];
    char file_mod[10];
    short file_type;
    long file_size;
    unsigned int first_cluster;
};
```

Listing 2: Struktura pro root directory

Celý souborový systém je implementován v jazyce C++ a je reprezentován třídou Fat. Třída Fat obsahuje následující veřejné metody3:

```
void loadFat();

std::shared_ptr<root_directory> findFileDescriptor(const std::↩
    string &t_path);

const std::vector<unsigned int> getClusters(std::shared_ptr<↩
    root_directory> t_fileEntry);

```

```
7  void tree();
8
9  void createEmptyFat();
10
11 void createDirectory(const std::string &t_path, const std::string↩
       &t_addr);
12
13 void deleteDirectory(const std::string &t_pseudoPath);
14
15 void insertFile(const std::string &t_filePath, const std::string ↩
     &t_pseudoPath);
16
17 void deleteFile(const std::string &t_pseudoPath);
18
19 const std::string getClusterContent(const unsigned int t_index);
20
21 const std::vector<std::string> readFileContent(const std::↩
     shared_ptr<root_directory> t_rootDirectory);
22
23 const std::vector<std::string> readFileContent(const unsigned int↩
       t_index, const long t_fileSize);
24
25 void writeClusterContent(const unsigned int t_index, const std::↩
     string &t_data);
26
27 void save();
28
29 void printBootRecord();
30
31 void printRootDirectories();
32
33 void printRootDirectory(const std::shared_ptr<root_directory> ↩
     t_rootDirectory);
34
35 void printClustersContent();
36
37 void printFileContent(const std::vector<std::string> &↩
     t_fileContent);
38
39 void printSubTree(const std::shared_ptr<root_directory> ↩
     t_rootDirectory, const unsigned int t_depth);
```

Listing 3: Veřejné metody třídy Fat

Vytvoření nového souboru

Při vytváření nového souboru se nejdříve zkontroluje, jestli existuje kopírovaný soubor. Dále se najde rodičovský adresář4 nového souboru.

```
1 auto separatorIndex = t_pseudoPath.find_last_of(PATH_SEPARATOR);
2 auto realPseudoPath = t_pseudoPath.substr(0, separatorIndex);
3 auto parentDirectory = findFileDescriptor(realPseudoPath);
4 auto parentDirectoryContent = loadDirectory(parentDirectory->↩
    first_cluster);
```

```
5  auto fileName = t_pseudoPath.substr(separatorIndex + 1);
6
7  std::fseek(workingFile, 0L, SEEK_END);
8
9  auto fileSize = std::ftell(workingFile);
10 auto file = makeFile(fileName, "rwxrwxrwx", fileSize, ↩
       FILE_TYPE_FILE, getFreeCluster());
```

Listing 4: Inicializace proměnných pro vložení nového souboru

Smazání souboru

Výpis clusterů obsahující data souboru

Vytvoření prázdné složky

Smazání prázdné složky

Výpis stromové struktury

Defragmentace

Ovldání programu

Závěr

Literatura