FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI

Semestrální práce z KIV/ZOS

PseudoFAT systém

Lukáš Runt (A20B0226P)

lrunt@students.zcu.cz

29. prosince 2022

Obsah

1 Zadání

Tématem semestrální práce je práce se zjednodušeným souborovým systémem založeným na pseudoFAT s podporou následujících příkazů:

- **cp <s1><s2>** - Zkopíruje soubor s1 do umístění s2
- **mv <s1><s2>** - Přesune soubor s1 do umístění s2, nebo přejmenuje s1 na s2
- **rm <s1>** - Smaže soubor s1
- **mkdir <a1>** - Vytvoří adresář a1
- **rmdir <a1>** - Smaže prázdný adresář a1
- **ls <a1>nebo ls** - Vypíše obsah adresáře a1, bez parametru vypíše obsah aktuálního adresáře
- **cat <s1>** - Vypíše obsah souboru s1
- **cd <a1>** - Změní aktuální cestu do adresáře a1
- **pwd** - Vypíše aktuální cestu
- **info <a1/s1>** - Vypíše informace o souboru/adresáři s1/a1 (v jakých clusterech se nachází)
- **incp <s1><s2>** - Nahraje soubor s1 z pevného disku do umístění s2 ve vašem FS
- **outcp <s1><s2>** - Nahraje soubor s1 z vašeho FS do umístění s2 na pevném disku
- **load <s1>** - Načte soubor z pevného disku, ve kterém budou jednotlivé příkazy, a začne je sekvenčně vykonávat. Formát je 1 příkaz/1řádek
- **format <velikost>** - Příkaz provede formát souboru, který byl zadán jako parametr při spuštění programu na souborový systém dané velikosti. Pokud už soubor nějaká data obsahoval, budou přemazána. Pokud soubor neexistoval, bude vytvořen.
- **check** - Zkontroluje, zda je souborový systém nepoškozen
- **bug <s1>** - Poškodí souborový systém, dle vašeho uvážení, tak aby bylo možné ověřit funkcionalitu příkazu check

Více v přiloženém zadání.

2 Teorie

FAT (File Allocation Table) tabulky jsou struktury, které se používají v počítačových systémech k ukládání informací o tom, kde se nacházejí soubory na disku nebo jiném datovém úložišti. FAT tabulky jsou součástí souborového systému, který se používá k ukládání a správě souborů na pevném disku nebo jiném datovém úložišti.

Existují tři hlavní typy FAT tabulek: FAT12, FAT16 a FAT32. Každý z těchto typů má své vlastní specifikace pro ukládání informací o souborech a může být použit v různých typech počítačů nebo operačních systémů.

FAT12 je nejstarší typ FAT tabulky a byl původně vytvořen pro použití s disketami. Jeho kapacita je omezená a není schopný efektivně spravovat velké množství souborů.

FAT16 byl vytvořen pro použití s pevnými disky a má větší kapacitu než FAT12. Je schopný spravovat větší množství souborů a ještě stále se používá v některých starších počítačích.

FAT32 je nejnovější a nejvyvinutější typ FAT tabulky. Má největší kapacitu ze všech tří typů a je schopný spravovat velké množství souborů efektivně. Je široce používán v moderních počítačích a operačních systémech.

3 Implementace

Program je napsán v programovacím jazyce C++. Program se spouští příkazem:

`pseudoFAT.exe <jméno file systému>`

3.1 Struktura file systému

FAT tabulka se skládá z clusterů. Velikost clusteru je pevně daná a to 1024B. Dále se tabulka dělí do následujících 3 částí:

- Bootovací část
- FAT tabulka
- Datová část

3.1.1 Boot

Bootovací část zabírá první cluster a obsahuje následující hodnoty oddělené oddělovacím znakem `0x00`:

- Velikost file systému
- velikost clusteru
- počáteční cluster datové části (cluster, ve kterém se nachází root)

3.1.2 FAT

FAT tabulka obsahuje hodnoty o jednotlivých clusterech. Jednotlivé clustery mohou nabývat následující hodnoty:

- Volný cluster = -2
- Poslední cluster = -1
- Pokračování souboru = Kladné číslo

V reálném FAT systému se používají 2 FAT tabulky, kde druhá slouží jako záloha a kontrola jestli jsou hodnoty v první tabulce správné. Já ve svém programu používám pouze jednu FAT tabulku. Jednotlivé buňky tabulky jsou veliké podle počtu clusterů, aby se nemusela přepisovat celá tabulka, při změně na cluster, který má číslo s více místy.

3.1.3 Data

Clustery můžou být dvou druhů. První druh je typ souboru, který obsahuje pouze bity souboru. Druhý typ je adresář, který má pevně danou následující strukturu:

- Jméno souboru (11 znaků)
- Typ souboru soubor/adresář (1 znak)
- Velikost souboru (počet znaků velikosti file systému)
- Cluster, kde se soubor nachází (počet znaků podle počtu clusterů)

Všechny údaje mají pevně danou velikost, aby nedocházelo k přepisování celé adresářové struktury. Adresář může zabírat pouze jeden cluster, pokud se další soubor nebo adresář nevejde do clusteru, vypíše se chybová hláška o plnosti adresáře. Každý adresář začíná údaji o konkrétním adresáři, kde cluster ukazuje na rodičovský adresář (v případě rootu -1).

3.2 Třídy v programu

3.2.1 Main

Třída se stará o vstup od uživatele. `Main` se ve smyčce ptá uživatele na příkazy, dokud uživatel nezadá příkaz `quit`.

3.2.2 Parser

Třída `Parser` se stará o parsování příkazů. Nejdříve se příkaz rozdělí podle bílých znaků a následně se zjišťuje, který jestli daný příkaz existuje, popřípadě jaký příkaz se má spustit. O jednotlivé příkazy se pak stará třída `Commands`. Příkaz `load` je jediná výjimka, která se provádí ve třídě `Parser`, kvůli tomu, že se načítané řádky rovnou parsují.

3.2.3 Commands

Třída `Commands` se stará o vykonávání jednotlivých příkazů. Třída obsahuje funkce všech příkazů kromě příkazu load. Každý příkaz se pak vykonává pomocí jednotlivých pomocných funkcí, které zařizují unární operace jako změň hodnotu ve FAT tabulce, zjisti velikost souboru, změň cluster souboru a další.

4 Spuštění aplikace

Soubor s aplikací obsahuje následující soubory:

- `main.cpp`
- `Parser.cpp` a `Parser.h`
- `Commands.cpp` a `Commands.h`
- `CMakeLists.txt`
- `PseudoFAT.exe`

Soubor se pak spouští příkazem:

```
PseudoFAT.exe <jméno file systému>
```

Pro překlad byly použit následující příkazy:

Na Windows: `g++ main.cpp Parser.cpp Commands.cpp -o PseudoFAT.exe`

Na Linux: `gcc main.cpp Parser.cpp Commands.cpp -o PseudoFAT.exe`

Při překladu je potřeba kompilátoru s verzí C++ alespoň 17, neboť je v programu použita knihovna `filesystem`, která není obsažena ve starších verzích.

5 Závěr

V rámci semestrální práce, jsem vytvořil program pseudoFAT systému. Tato práce mi pomohla se zdokonalit v programovacím jazyce C++ a také pochopit jak funguje ukládání dat ve File Allocation Table. Kód se dá určitě napsat mnohem lépe, jde o můj první pokus o větší práci v C++. V průběhu práce, jsem si začal uvědomovat pár nevhodně navržených koncepcí, které by šly určitě vymyslet a naimplementovat lépe. I když jsem se při dělání semestrální práce mnoho naučil, nesmím opomenout, že mi tato práce relativně zkazila Vánoce a mému psychickému stavu také vůbec nepomohla. Na závěr tedy doufám, že se můj život alespoň trošku zlepší po odevzdání této práce, aby jsem jí mohl hodnotit pozitivně.