FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI

Katedra informatiky a výpočetní techniky

Semestrální práce z předmětu
Základy operačních systémů

Souborový systém s i-uzly

Autor:
Stanislav Kafara
A21B0160P
skafara@students.zcu.cz

Obsah

1 Zadání **2**

2 Popis implementace **6**
2.1 Reprezentace dat 6
2.2 Vybrané operace 6
2.2.1 Formátování souborového systému 6
2.2.2 Kopírování souboru 7
2.2.3 Přesouvání/přejmenování souboru 7
2.2.4 Mazání souboru 7
2.2.5 Vyhodnocení cesty 8
2.2.6 Symbolický odkaz 8

3 Uživatelská příručka **9**
3.1 Překlad a sestavení 9
3.2 Spuštění 9

4 Závěr **10**

Kapitola 1

Zadání

Tématem semestrální práce bude práce se zjednodušeným souborovým systémem založeným na i-uzlech. Vaším cílem bude splnit několik vybraných úloh.

Program bude mít jeden parametr a tím bude název Vašeho souborového systému. Po spuštění bude program čekat na zadání jednotlivých příkazů s minimální funkčností viz níže. Všechny soubory mohou být zadány jak absolutní, tak relativní cestou.

1. `cp s1 s2`

   Zkopíruje soubor s1 do umístění s2.

   Možný výsledek:

   ```
   OK
   FILE NOT FOUND
   PATH NOT FOUND
   ```

2. `mv s1 s2`

   Přesune soubor s1 do umístění s2, nebo přejmenuje s1 na s2.

   Možný výsledek:

   ```
   OK
   FILE NOT FOUND
   PATH NOT FOUND
   ```

3. `rm s1`

   Smaže soubor s1.

   Možný výsledek:

```
OK
FILE NOT FOUND
```

4. `mkdir a1`

   Vytvoří adresář a1.

   Možný výsledek:

   ```
   OK
   PATH NOT FOUND
   EXIST
   ```

5. `rmdir a1`

   Smaže prázdný adresář a1.

   Možný výsledek:

   ```
   OK
   FILE NOT FOUND
   NOT EMPTY
   ```

6. `ls a1`

   `ls`

   Vypíše obsah adresáře a1.

   Možný výsledek:

   ```
   -FILE
   +DIRECTORY
   PATH NOT FOUND
   ```

7. `cat s1`

   Vypíše obsah souboru s1.

   Možný výsledek:

   ```
   OBSAH
   FILE NOT FOUND
   ```

8. `cd a1`

   Změní aktuální cestu do adresáře a1.

   Možný výsledek:

   ```
   OK
   PATH NOT FOUND
   ```

9. `pwd`

   Možný výsledek:

   ```
   PATH
   ```

10. `info a1`

    `info s1`

    Vypíše informace o souboru/adresáři s1/a1 (v jakých clusterech se nachází).

    Možný výsledek:

    ```
    NAME - SIZE - i-node NUMBER - přímé a nepřímé odkazy
    FILE NOT FOUND
    ```

11. `incp s1 s2`

    Nahraje soubor s1 z pevného disku do umístění s2 ve vašem FS.

    Možný výsledek:

    ```
    OK
    FILE NOT FOUND
    PATH NOT FOUND
    ```

12. `outcp s1 s2`

    Nahraje soubor s1 z vašeho FS do umístění s2 na pevném disku.

    Možný výsledek:

    ```
    OK
    FILE NOT FOUND
    PATH NOT FOUND
    ```

13. `load s1`

    Načte soubor z pevného disku, ve kterém budou jednotlivé příkazy, a začne je sekvenčně vykonávat. Formát je 1 příkaz/1řádek.

    Možný výsledek:

    ```
    OK
    FILE NOT FOUND
    ```

14. `format 600MB`

    Příkaz provede formát souboru, který byl zadán jako parametr při spuštení programu na souborový systém dané velikosti. Pokud už soubor nějaká data obsahoval, budou přemazána. Pokud soubor neexistoval, bude vytvořen.

    Možný výsledek:

    ```
    OK
    CANNOT CREATE FILE
    ```

15. `slink s1 s2`

    Vytvoří symbolický link na soubor s1 s názvem s2. Dále se s ním pracuje očekávaným způsobem, tedy např. cat s2 vypíše obsah souboru s1.

Budeme předpokládat korektní zadání syntaxe příkazů, nikoliv však sémantiky (tj. např. cp s1 zadáno nebude, ale může být zadáno cat s1, kde s1 neexistuje).

Maximální délka názvu souboru bude 8+3=11 znaků (jméno.přípona) + \0 (ukončovací znak v C/C++), tedy 12 bytů.

Každý název bude zabírat právě 12 bytů (do délky 12 bytů doplníte \0 - při kratších názvech).

Kapitola 2

Popis implementace

2.1 Reprezentace dat

Obsah souborového systému je reprezentován souborem mapovaném v hlavní paměti. Tento soubor je reprezentován třídou `MMappedFile`.

Každá část souborového systému, tj. superblok, bitmapy, i-uzly a datové bloky jsou reprezentovány odpovídající třídou.

Všechny tyto třídy splňují rozhraní `I_ReadableWritable` pro jednotný a jednoduchý zápis a čtení z podlehlé reprezentace souborového systému souborem mapovaným v hlavní paměti.

2.2 Vybrané operace

2.2.1 Formátování souborového systému

Souborový systém vyplní hodnoty v superbloku, inicializuje kořenový adresář a vše zbylé vyplní prázdnými hodnotami. Těmi jsou:

- 0 pro neobsazené bity v bitmapě,
- -1 pro nepoužité přímé odkazy v i-uzlu,
- 0 pro nepoužité nepřímé odkazy v i-uzlu a
- 0 pro nepoužitý prostor v datovém bloku.

Počet i-uzlů a datových bloků a skutečná velikost souborového systému je odvozena z požadavků na souborový systém.

Ty jsou definovány následovně:

1. Velikost souborového systému je maximální možná.
2. Počet i-uzlů a datových bloků je dělitelný 8.
3. Předpokládá se ukládání souborů průměrně zabírající 10 datových bloků.

Výsledkem jsou vztahy pro výpočet parametrů souborového systému.

2.2.2 Kopírování souboru

Před zahájením kopírování je vypočten počet potřebných datových bloků k uložení kopírovaného souboru, a sice datových bloků uchovávající skutečný obsah souboru, ale i nepřímých datových bloků uchovávající reference na datové bloky, které obsahují skutečný obsah souboru.

Tyto datové bloky jsou pak rezervovány a používány při kopírování. Proces procházení přímých i nepřímých datových bloků je realizován promocí iterátoru, který zpřístupňuje obsah v datových blocích, které mají uchovávat skutečný obsah souboru.

Postupně jsou tedy tyto datové bloky procházeny, v bitmapě jsou označeny jako použité a je do nich zapisován obsah vyrovnávací paměti obsahující části kopírovaného souboru.

Při kopírování souboru ze souborového systému je postup opačný. Datové bloky jsou postupně procházeny, jejich obsahem je plněna vyrovnávací paměť, která je následně zapisovaná do souboru.

2.2.3 Přesouvání/přejmenování souboru

K přesunu/přejmenování souboru dochází pouze tím, že je vymazán záznam o daném souboru ve zdrojovém adresáři a je přidán záznam o daném souboru v cílovém adresáři.

Procházení záznamů o souborech v adresáři je realizováno pomocí iterátoru. Iterátor zároveň zpřístupňuje připojení nového záznamu a vymazání záznamu.

2.2.4 Mazání souboru

Mazání souboru probíhá tak, že jsou pomocí iterátoru procházeny datové bloky obsahující jak skutečný obsah souboru, tak i odkazy na ně a jsou uvolněny. Uvolnění datového bloku proběhne tak, že je mu v bitmapě nastaven příznak jako nepoužitý a obsah datového bloku je nastaven na prázdnou hodnotu.

2.2.5 Vyhodnocení cesty

Cesta je vyhodnocována postupným procházením datových bloků adresářů a hledání v nich daných jmen sekvenčním procházením pomocí iterátoru zpřístupňující záznamy o souborech v datových blocích.

Celý proces je opakován do doby, než je celá cesta zpracována, nebo dojde k chybě a cesta není nalezena.

2.2.6 Symbolický odkaz

Symbolický odkaz je samostatný soubor (soubor nebo adresář). Navíc má příznak, že je to symbolický odkaz. Obsahem takového souboru je relativní nebo absolutní cesta, na kterou byl při jeho vzniku navázán.

Při vzniku obsahuje platnou cestu. Nicméně podlehlý soubor nebo adresář může být odstraněn, přejmenován nebo přesunut a symbolický odkaz nepřestane existovat, ale stane se neplatným, protože ukazuje na neexistující soubor/adresář.

Správná interpretace symbolického odkazu je zajištěna expandováním cesty, kterou obsahuje, když je obsažen ve vyhodnocované cestě.

Některé operace vyžadují, aby cesta byla expandovaná celá. Těmi jsou např. `cd`, `cat`, `cp`. Některé naopak vyžadují, aby cesta byla expandovaná celá vyjma případné existence symbolického odkazu na konci cesty. Těmi jsou např. `mv`, `rm`, ... To plyne z požadavků na symbolický odkaz.

Kapitola 3

Uživatelská příručka

Program vyžaduje překladač jazyka C++ podporující standard jazyka `C++20` a utilitu `cmake` verze alespoň 3.0.

3.1 Překlad a sestavení

Program je přeložitelný a spustitelný v prostředí Unix. Překlad a sestavení programu je řízeno skriptem utility `cmake`. K překladu a sestavení programu dojde po zadání příkazu v kořenovém adresáři:

```
$ cmake -Bbuild -Ssrc && cd build && make
```

Výsledný spustitelný soubor se nachází v adresáři `build` a jmenuje se `fsinodes`.

3.2 Spuštění

Program přijímá právě jeden vstupní argument. Vzor příkazu spuštění je:

`fsinodes` ⟨`fs-data`⟩

Symbol ⟨`fs-data`⟩ představuje cestu k souboru obsahující obsah souborového systému.

Příkaz spuštění programu může vypadat následovně:

```
fsinodes fs.dat
```

V případě zadání nesprávného počtu nebo neplatných hodnot argumentů program vypíše chybovou hlášku a příručku k ovládání programu.

Kapitola 4

Závěr

Implementované řešení splňuje veškeré požadavky zadání.

Zdrojový kód programu trochu ztrácí na přehlednosti, což z časových důvodů nebylo přepracováno. Nicméně i tak je dostatečně čitelný a rozšiřitelný.