

# Semestrální práce

KIV/ZOS

2020/2021

# Obsah

Zadání .....	2
Popis implementace.....	3
Struktura souborů .h .c .....	3
Struktura souborového svazku .....	3
Uživatelská příručka.....	4
Popis příkazů .....	4
Parametry programu .....	4

## **Zadání**

Zadáním je vytvořit pseudosystém souborů založený na i-uzlech.

Více na:

<https://courseware.zcu.cz/portal/studium/courseware/kiv/zos/samostatna-prace.html>

<https://courseware.zcu.cz/CoursewarePortlets2/DownloadDokumentu?id=187948>

## Popis implementace

Základem systému je soubor, jehož název je předaný prvním a jediným argumentem. Tento soubor se upravuje pomocí zadaných příkazů. Po naformátování souboru se v něm vytvoří struktura reprezentující souborový systém.

### Struktura souborů .h .c

Program je rozdělen do 5 zdrojových souborů.

Main.c – hlavní soubor aplikace

Commands.h – hlavičkový soubor příkazů

Commands.c – zdrojový soubor příkazů

Ext.h - hlavičkový soubor obsahující struktury zapisující se do svazku

Ext.c – implementace funkcí pracujících se svazkem

### Struktura souborového svazku

Naformátovaný svazek obsahuje několik bloků. Prvním je superbloc. Tato struktura obsahuje základní informace o svazku. Dalším blokem je pole bytů o 10 ti prvních reprezentující zabrané inody. Tato velikost se dá změnit přepsáním directive preprocesoru *BITMAP\_SIZE*. Obdobnou strukturou je pole reprezentující zabrané clustery. Jeho velikost se dá změnit přepsáním hodnoty *BITMAP\_SIZE*. Následuje výpis struktury pseudo\_inode, která reprezentuje i-uzel. Po skončení inodů, následují samotná data. Data lze rozdělit na 3 typy podle toho, zde se jedná o adresář, soubor nebo symbolický link. V prvním případě je v prvním clusteru několik struktur directory\_item. Tato struktura obsahuje inode\_id a jméno souboru. Directory\_item na prvním místě je odkaz na rodiče složky. Pokud je soubor prázdný je jeho inode\_id nastaveno na ID\_ITEM\_FREE, což odpovídá hodnotě -1. Jestliže se jedná o soubor, pak jsou jeho data buď přímo v clusterech, na které směřují jeho přímé odkazy, nebo je odkaz na jeho data v nepřímém linku.

# **Uživatelská příručka**

## **Popis 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 - Vypíše obsah adresáře a1

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 v pseudoNTFS

outcp s1 s2 - Nahraje soubor s1 z pseudoNTFS 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/ládek

format 600MB - 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.

## **Parametry programu**

Program má pouze jediný parametr a tím je název svazku, se kterým se bude pracovat.

## **Závěr**

Myslím, že zadání jsem v rámci mezí splnil. Příkazy mi fungují a systém běží stabilně. Mám hodně nápadů na vylepšení mé práce. Pokud bych měl práci dále rozšiřovat, zaměřil bych se na absolutní odladění chyb a případně bych se pokusil naddimenzovat systém, aby bylo možné ho plnohodnotně používat. Dalším vylepšením by mohlo být šifrování souboru svazku, podobně jako program VeraCrypt.