



ZÁPADOČESKÁ UNIVERZITA V PLZNI

ZÁKLADY OPERAČNÍCH SYSTÉMŮ

KIV/ZOS

Dokumentace semestrální práce

Vojtěch DANIŠÍK
A16B0019P
danisik@students.zcu.cz

24. ledna 2019

Obsah

1	Zadání	2
2	Programátorská dokumentace	2
2.1	Datové struktury	2
2.2	Implementace	4
2.2.1	Seznam validních příkazů	4
2.2.2	Použité parametry v pseudoNTFS	5
2.2.3	Start aplikace	5
2.2.4	Vytváření složky	5
2.2.5	Import souboru	5
2.2.6	Export souboru	5
2.2.7	Defragmentace	6
3	Uživatelská dokumentace	6
3.1	Překlad aplikace	6
3.2	Spuštění aplikace	6
4	Závěr	7

1 Zadání

Letošním tématem semestrální práce bude práce s pseudosystémem NTFS (značně zjednodušeným). Zadání práce je společné pro všechny studenty vyjma posledního úkolu, který je určen dle prvního písmene vašeho loginu. Všechny potřebné informace najdete v pdf zadání semestrální práce na této stránce. K dispozici je i zip soubor s popisem základních datových struktur a vysvětlujícím obrázkem. Tento popis je doporučený, datové struktury si můžete nadefinovat i jinak (ale v souladu s principem daného pseudosystému).

Celé zadání semestrální práce lze najít ZDE

2 Programátorská dokumentace

Tuto aplikaci jsem vyvíjel pomocí programovacího jazyka C.

2.1 Datové struktury

- VFS - Virtuální file system, ve kterém se aplikace pohybuje
 - (BOOT_RECORD *) boot_record - Bootovací záznamy
 - (MFT *) mft - Master Field Table, uchovávání záznamů (soubory a složky)
 - (BITMAP *) bitmap - Bitmapa reflektující disk
 - (PATH *) actual_path - Uchovává aktuální cestu, ve které se uživatel nachází
 - (char *) filename - Název datového souboru
 - (FILE *) FILE - Odkaz na datový soubor
- BOOT_RECORD - Bootovací záznamy
 - (char [9]) signature - Název systému
 - (char [251]) volume_descriptor - Popis systému
 - (int32_t) disk_size - Velikost disku
 - (int32_t) cluster_size - Velikost jednoho clusteru
 - (int32_t) cluster_count - Počet clusterů
 - (int32_t) mft_start_address - Startovací adresa pro ukládání dat u MFT

- (int32_t) bitmap_start_address - Startovací adresa pro ukládání dat u Bitmapy
- (int32_t) data_start_address - Startovací adresa pro ukládání dat u datového bloku
- (int32_t) mft_max_fragment_count - Maximální počet fragmentů pro jeden záznam v MFT
- MFT - Master Field Table
 - (int32_t) size - Počet záznamů v MFT tabulce
 - (MFT_ITEM **) items - Tabulka MFT záznamů
- MFT_ITEM - Reprezentace záznamu (složky, soubory)
 - (int32_t) uid - ID záznamu
 - (int32_t) parentID - ID nadřazené složky
 - (int) isDirectory - Indikátor, zda záznam je složka (isDirectory = 1) nebo soubor (isDirectory = 0)
 - (int8_t) item_order - Pořadí v MFT při více souborech
 - (int8_t) item_order_total - Celkový počet položek v MFT
 - (char [12]) item_name - Název záznamu
 - (long) item_size - Velikost záznamu v Bytech
 - (int32_t[]) fragment_start_address - Startovací adresa fragmentu v datovém bloku
 - (int32_t[]) fragment_count - Počet clusterů rezervovaných pro záznam
 - (int32_t[]) start_cluster_ID - ID prvního clusteru přiřazeného záznamu
 - (int) fragments_created - Počet fragmentů vytvořených pro záznam
- FRAGMENT_TEMP - Dočasný záznam o fragmentu
 - (int) successful - Indikace zda se podařilo vytvořit dočasný záznam
 - (int32_t) start_cluster_ID - ID prvního clusteru přiřazeného záznamu
 - (int) count - Počet přiřazených clusterů

- BITMAP - Bitmapa reflektující disk
 - (int32_t) length - Velikost bitmapy
 - (unsigned char *) data - Pole hodnot (0 - cluster je volný, 1 - cluster je zabraný)
- PATH - Aktuální cesta, ve které se uživatel nachází
 - (path *) path - Aktuální cesta v systému

2.2 Implementace

2.2.1 Seznam validních příkazů

Příkaz	Použití	Popis
cp	cp s1 s2	Kopírování souboru
mv	mv s1 s2	Přesun souboru
rm	rm s1	Smazání souboru
mkdir	mkdir a1	Vytvoření adresáře
rmdir	rmdir a1	Smazání prázdného adresáře
ls	ls a1	Výpis obsahu adresáře
cat	cat s1	Výpis obsahu souboru
cd	cd a1	Změna aktuálního adresáře
pwd	pwd	Výpis aktuální cesty
info	info a1, info s1	Výpis informací o adresáři/souboru
incp	incp s1 s2	Nahrání souboru z pevného disku
outcp	outcp s1 s2	Nahrání souboru na pevný disk
load	load s1	Vykonání sekvence příkazů ze souboru
format	format 600MB	Zformátování datového souboru a nastavení velikosti disku
defrag	defrag	Defragmentace disku
finfo	finfo	Výpis veškerých informací o pseudoNTFS
quit	quit	Ukončení programu
help	help	Výpis dostupných příkazů

2.2.2 Použité parametry v pseudoNTFS

Parametr	Hodnota
Velikost disku	10 KB
Velikost clusteru	4 KB
Maximální počet fragmentů pro mft item	4
Velikost složky	1 B

2.2.3 Start aplikace

Při startu si aplikace zjistí, zda v aktuálním adresáři kde je spouštěna, existuje datový soubor s názvem ntfs.dat. Pokud existuje, naplní své struktury z dat uložených v datovém souboru. Pokud neexistuje, nainicializuje své struktury pomocí předem stanovených parametrů v kapitole 2.2.1

2.2.4 Vytváření složky

Pro vytvoření složky si aplikace zjistí, zda zadané jméno složky existuje, případně zda zadaná cesta je validní. Poté se kontroluje, zda ve složce, ve které chceme vytvořit novou složku, existuje soubor nebo složka se stejným názvem. Pokud neexistuje, aplikace se snaží vytvořit ve své struktuře MFT nový záznam (pokud máme stále volné místo) a přidělí tomuto záznamu místo (v bitmapě) pro ukládání svých dat.

2.2.5 Import souboru

Pro import souboru do našeho systému je potřeba zadat validní cestu jak v původním systému, tak i v tom našem. Provádí se kontrola, zda tyto cesty jsou validní a pokud ne, aplikace se dále o nic nesnaží. Následně se aplikace snaží vytvořit záznam v MFT pro importující soubor a přidělí mu místo pro zapisování svých dat na základě zjištěné velikosti importovaného souboru. Následně jsou data ukládána do datového souboru na předem určené místo, které mu bylo přiděleno naším systémem.

2.2.6 Export souboru

Export souboru probíhá tak, že se nejdříve zkontroluje, zda obě dvě zadané cesty (jak source, tak i destination) jsou validní. Následně systém vytvoří soubor na zadané destination a začne do něj ukládat data, která jsou uložena v datovém souboru systému na určených adresách, které byly přiděleny souboru, který chceme vyexportovat, při inicializaci.

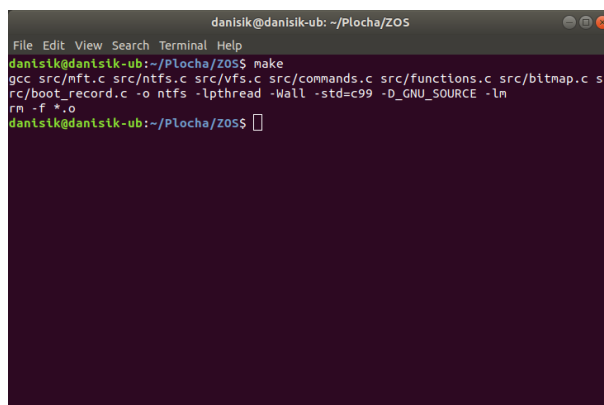
2.2.7 Defragmentace

Při defragmentaci postupujeme tak, že si nejdříve zkopírujeme všechna data z datového bloku do nově vytvořeného dočasného datového souboru. Následně nastavíme všechny hodnoty v bitmapě na 0 (indikuje, že v datovém souboru nic není uloženo). Po nastavení celé bitmapy probíhá nová inicializace všech souborů a následné překopírování dat patřících k těmto souborům. Nakonec smažeme pomocný datový soubor.

3 Uživatelská dokumentace

3.1 Překlad aplikace

Přeložení zdrojových souborů se provádí zadáním příkazu `make` v terminálu v kořenovém adresáři naší aplikace. Zdrojové soubory najdeme ve složce `src`.

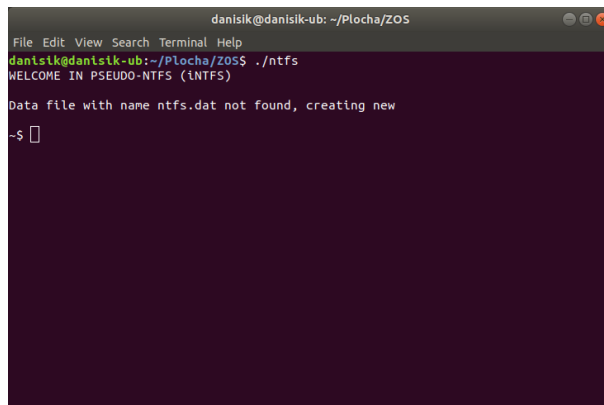


```
danisik@danisik-ub: ~/Plocha/ZOS
File Edit View Search Terminal Help
danisik@danisik-ub:~/Plocha/ZOS$ make
gcc src/nft.c src/ntfs.c src/vfs.c src/commands.c src/functions.c src/bitmap.c s
rc/boot_record.c -o ntfs -lpthread -Wall -std=c99 -D_GNU_SOURCE -lm
rm -f *.o
danisik@danisik-ub:~/Plocha/ZOS$
```

Obrázek 1: Překlad aplikace

3.2 Spuštění aplikace

Aplikace se spouští pomocí bez parametrů příkazu `./ntfs`.

A screenshot of a terminal window titled 'danisik@danisik-ub: ~/Plocha/ZOS'. The terminal shows the command './ntfs' being executed. The output is 'WELCOME IN PSEUDO-NTFS (iNTFS)' followed by a message 'Data file with name ntfs.dat not found, creating new' and a prompt '-\$ '.

Obrázek 2: Spuštění aplikace

4 Závěr

Podařilo se mi implementovat poměrně komplexní celek, který reprezentuje virtuální souborový systém NTFS (dost ořezaný, nelze například vytvářet/upravovat soubory, pouze je můžeme importovat). Asi nejvíce času mi zabralo implementování fragmentů, přiřazování adres v datovém bloku a samotné ukládání do datového souboru.