

Virtual Filesystem

Tomáš Klepač

KIV/ZOS – Základy operačních systémů

Fakulta aplikovaných věd, Západočeská univerzita v Plzni

2025

1 Cíl práce

Cílem práce bylo vytvořit jednoduchý virtuální souborový systém v jazyce C++, který je uložen do jednoho binárního souboru. Program simuluje základní principy reálných souborových systémů (např. ext2, MINIX) – práci s i-uzly, alokací bloků dat, hierarchickou adresářovou strukturou a operacemi na souborech a adresářích.

2 Architektura

2.1 Fyzická organizace disku

Virtuální disk je rozdělen do následujících sekcí:

- **Superblock** - metadata o systému,
- **I-node bitmap** - mapa volných i-uzlů,
- **Data bitmap** - mapa volných datových bloků,
- **I-node tabulka** - samotné i-uzly,
- **Datová oblast** - datové bloky pro uložení dat souborů.

2.2 Datové struktury

2.2.1 Superblock

Obsahuje globální metadata o systému:

- Signaturu a popis svazku,
- Celkovou velikost disku,
- Velikost datových bloků,
- Adresy všech tabulek a bitmap v binárním souboru.

2.2.2 I-uzel (Inode)

Popisuje soubor nebo adresář a obsahuje:

- Identifikátor i-uzlu,
- Typ (soubor nebo adresář),
- Počet odkazů,

- Velikost v bytech,
- **Přímé odkazy** na datové bloky,
- **Nepřímé odkazy** pro větší soubory.

2.2.3 Přímé a nepřímé bloky

Systém pracuje s dvěma typy odkazů na data:

- **Přímé bloky:** Prvních 5 odkazů v i-uzlu přímo ukazují na datové bloky,
- **Indirect1:** Jeden blok s ukazateli na další datové bloky,
- **Indirect2:** Jeden blok s ukazateli na bloky, které obsahují další ukazatele.

Tímto přístupem lze dosáhnout velkých maximálních velikostí souborů.

2.2.4 Directory item

Položka v adresáři mapuje:

- Název souboru nebo adresáře,
- Identifikátor odpovídajícího i-uzlu.

3 Implementace

Projekt je rozdělen do následujících modulů:

- **filesystem.h** - hlavička se třídou FileSystem,
- **structures.h** - definice datových struktur,
- **filesystem_core.cpp** - jádro (formatting, alokace, čtení/zápis i-uzlů),
- **filesystem_dir.cpp** - operace s adresáři,
- **filesystem_file.cpp** - operace se soubory,
- **main.cpp** - interaktivní shell pro uživatele.

4 Podporované příkazy

4.1 Základní příkazy

<code>format [MB]</code>	- vytvoří nový systém
<code>mkdir [jméno]</code>	- vytvoří adresář
<code>rmdir [jméno]</code>	- smaže prázdný adresář
<code>ls [jméno]</code>	- vypíše obsah adresáře
<code>cd [jméno]</code>	- změni aktuální adresář
<code>pwd</code>	- vypíše cestu k aktuálnímu adresáři
<code>touch [soubor]</code>	- vytvoří prázdný soubor
<code>cat [soubor]</code>	- vypíše obsah souboru
<code>write [soubor] [text]</code>	- přepíše obsah souboru
<code>rm [soubor]</code>	- smaže soubor
<code>info [položka]</code>	- zobrazí metadata souboru/adresáře
<code>statfs</code>	- zobrazí statistiku systému

4.2 Operace se soubory

<code>cp [zdroj] [cíl]</code>	- zkopíruje soubor
<code>mv [zdroj] [cíl]</code>	- přesune nebo přejmenuje soubor
<code>xcp [s1] [s2] [s3]</code>	- spojí soubory s1 a s2 do s3 (ROZŠÍŘENÍ)
<code>add [s1] [s2]</code>	- připojí obsah s2 na konec s1 (ROZŠÍŘENÍ)

4.3 Host integrace

<code>incp [host] [vfs]</code>	- importuje soubor z hostitele
<code>outcp [vfs] [host]</code>	- exportuje soubor na hostitele
<code>load [skript]</code>	- spustí dávkové příkazy

5 Klíčové vlastnosti

- Hierarchická struktura adresářů s podporou cest,
- Dynamická alokace datových bloků a i-uzlů,
- Podpora velkých souborů pomocí nepřímých odkazů,
- Dávkové spouštění příkazů ze skriptů,
- Import a export souborů z hostitele,
- Zobrazení metadat souborů a statistiky systému.

6 Závěr

Projekt implementuje zjednodušený virtuální souborový systém s hierarchickou strukturou adresářů a souborů, který podporuje přímé i nepřímé datové bloky. Jsou implementovány všechny povinné příkazy dle zadání včetně rozšiřujících příkazů `xcp` a `add`. Práce slouží jako demonstrace základních principů správy souborových systémů a porozumění vnitřnímu fungování operačních systémů.