

# Semestrální práce z předmětu **KIV/ZOS**

Kevin Varchola

28. listopadu 2024

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>FAT souborový systém</b>	<b>3</b>
2.1	Boot sekce . . . . .	3
2.2	FAT tabulky . . . . .	4
2.2.1	Datová sekce . . . . .	5
<b>3</b>	<b>Struktura projektu</b>	<b>6</b>
3.1	Popis příkazů . . . . .	6
<b>4</b>	<b>Použití a spuštění programu</b>	<b>8</b>
<b>5</b>	<b>Závěr</b>	<b>8</b>

# 1 Úvod

Tématem této semestrální práce je práce se zjednodušeným souborovým systémem založeným na pseudoFAT (File Allocation Table). Cílem projektu je implementovat základní operace souborového systému.

Po spuštění programu, který přijímá jeden parametr – název souborového systému – bude aplikace čekat na zadání příkazů uživatelem. Všechny příkazy budou umožňovat použití cest jak v absolutní, tak v relativní formě.

Implementace bude zahrnovat základní operace, jako je práce se soubory (vytváření, mazání, přesun, kopírování) a práce s adresáři (vytváření, mazání).

Cílem je vytvořit funkční a efektivní aplikaci, která demonstruje principy práce se souborovými systémy a umožňuje jednoduchou manipulaci se strukturou souborů a adresářů.

## 2 FAT souborový systém

FAT (File Allocation Table) je souborový systém navržený pro jednoduchou správu souborů na úložných zařízeních. Hlavní princip systému spočívá v rozdělení disku na jednotky zvané **cluster**, které jsou spravovány pomocí tabulky FAT. Tento systém propojuje části souborů uložené na různých clusterech a umožňuje efektivní přístup k datům.

FAT souborový systém se skládá ze tří hlavních sekcí:

### 2.1 Boot sekce

Boot sekce se nachází na začátku disku a obsahuje metadata o souborovém systému. Tato data jsou klíčová pro správu disku a obsahují následující informace:

- **Velikost souborového systému(file\_size):**  
Celková velikost souborového systému v bajtech.
- **Velikost tabulky FAT(fat\_size):**  
Velikost jedné tabulky FAT v bajtech.
- **Počet clusterů v tabulce FAT(fat\_cluster\_count):**  
Celkový počet clusterů spravovaných v jedné tabulce FAT.
- **Celkový počet clusterů(cluster\_count):**  
Počet clusterů dostupných v souborovém systému.

- **Začátek první tabulky FAT(fat1\_start):**  
Adresa, kde na disku začíná první tabulka FAT.
- **Začátek druhé tabulky FAT(fat2\_start):**  
Adresa, kde na disku začíná záložní (druhá) tabulka FAT.
- **Začátek datové sekce(data\_start):**  
Adresa, kde na disku začíná datová část, ve které jsou uloženy soubory a adresáře.

Boot sekce slouží jako výchozí bod pro správu souborového systému a zajišťuje, že systém rozpozná strukturu disku.

## 2.2 FAT tabulky

FAT tabulky jsou klíčovou součástí pseudoFAT systému. Slouží ke správě clusterů a k evidenci jejich stavu. Souborový systém obsahuje dvě FAT tabulky – primární a záložní. Záložní tabulka slouží k obnovení dat v případě poškození primární tabulky.

Každý záznam v tabulce FAT odpovídá jednomu clusteru a obsahuje jednu z následujících hodnot:

- **FAT\_FREE = -1:**  
Označuje volný (nepoužitý) cluster.
- **FAT\_EOF = -2:**  
Označuje konec souboru, tedy poslední cluster, který obsahuje data konkrétního souboru.
- **FAT\_BAD = -3:**  
Označuje poškozený cluster, který nelze použít pro ukládání dat.
- **Hodnota n:**  
Pokud cluster obsahuje číslo, jedná se o ukazatel na další cluster, který tvoří pokračování souboru.

Obě FAT tabulky jsou umístěny sekvenčně na disku, kde první je primární tabulka a druhá slouží jako záložní. Každá změna v primární FAT tabulce by měla být zrcadlena i do záložní tabulky, aby byla zajištěna konzistence a možnost obnovy.

### 2.2.1 Datová sekce

Datová sekce je oblast pseudoFAT systému, kde jsou ukládána skutečná data souborů a adresářů. V datové sekci se nacházejí dvě klíčové struktury:

- **Souborová data:**  
Data uložená v souborech zabírají jeden nebo více clusterů. Jejich propojení je definováno v FAT tabulce, která obsahuje ukazatele na další cluster nebo označení konce souboru (`FAT_EOF`).
- **Adresářové záznamy:**  
Adresáře obsahují metadata souborů a podadresářů ve formě záznamů (`DirectoryEntry`).

`DirectoryEntry` je struktura, která ukládá metadata souborů a adresářů, a obsahuje následující informace:

- **Name:**  
Pole bajtů s maximální délkou `MAX_FILE_NAME`, které obsahuje název souboru nebo adresáře.
- **Size:**  
Velikost souboru v bajtech. U adresářů tato hodnota není relevantní.
- **First\_cluster:**  
Číslo prvního clusteru, kde se nachází obsah souboru nebo seznam záznamů podadresáře.
- **Is\_directory:**  
Hodnota indikující, zda se jedná o adresář (1 = true, 0 = false).

Adresářová struktura umožňuje hierarchické uspořádání souborů a adresářů. Každý adresář může obsahovat další `DirectoryEntry` záznamy pro soubory a podadresáře, čímž se vytváří stromová struktura. Práce s těmito záznamy probíhá výhradně v datové sekci, zatímco jejich fyzické umístění je spravováno prostřednictvím FAT tabulek.

## 3 Struktura projektu

Projekt je implementován v jazyce Go a obsahuje několik klíčových souborů a složek, které spolu tvoří celek pro správu souborového systému. Hlavní soubory a složky jsou uspořádány následovně:

<code>data/</code>	<code>% Složka s testovacími daty a skripty</code>
<code>commands.go</code>	<code>% Soubor obsahující implementaci příkazů</code>
<code>file_system.go</code>	<code>% Funkce pro práci se souborovým systémem</code>
<code>help_funcs.go</code>	<code>% Pomocné funkce pro zápis a výpis dat</code>
<code>main.go</code>	<code>% Hlavní vstupní bod aplikace</code>
<code>structures.go</code>	<code>% Definice struktur používaných v aplikaci</code>
<code>go.mod</code>	<code>% Modulová konfigurace pro Go</code>

### 3.1 Popis příkazů

Projekt zahrnuje několik příkazů pro manipulaci se souborovým systémem, které umožňují provádět základní operace se soubory a adresáři:

- `cp` (kopírování souboru):  
Tento příkaz umožňuje zkopírovat soubor z jednoho místa na druhé.
- `mv` (přesun souboru nebo přejmenování):  
Umožňuje přesunout soubor do jiné složky nebo přejmenovat soubor v rámci stejné složky.
- `rm` (odstranění souboru):  
Tento příkaz odstraní soubor a označí jeho clustery jako volné.
- `mkdir` (vytvoření adresáře):  
Tento příkaz slouží k vytvoření nového adresáře.
- `rmdir` (odstranění adresáře):  
Umožňuje smazat adresář, pokud je prázdný.
- `ls` (výpis obsahu adresáře):  
Zobrazuje seznam souborů a adresářů v aktuálním adresáři.
- `cat` (zobrazení obsahu souboru):  
Zobrazuje obsah souboru přímo v konzole.
- `cd` (změna adresáře):  
Umožňuje uživateli přejít do jiného adresáře.

- **pwd** (výpis aktuálního adresáře):  
Tento příkaz vypisuje absolutní cestu k aktuálně aktivnímu adresáři.
- **info** (informace o souboru nebo adresáři):  
Zobrazuje všechny clustery, které jsou přiřazeny souboru nebo adresáři.
- **incp** (import souboru do PseudoFAT):  
Nahrání souboru z reálného souborového systému do systému PseudoFAT.
- **outcp** (export souboru z PseudoFAT):  
Export souboru ze systému PseudoFAT zpět do reálného souborového systému.
- **load** (načtení skriptu):  
Načítá skript obsahující příkazy, které jsou následně vykonány v rámci systému.
- **format** (formátování souborového systému):  
Tento příkaz vymaže všechny data a struktury na disku a znovu naformátuje souborový systém.
- **bug** (simulace poškozeného clusteru):  
Simuluje poškozený cluster tím, že označí cluster souboru nebo adresáře jako `FAT_BAD_CLUSTER`.
- **check** (kontrola integrity systému):  
Prochází celou tabulku FAT a kontroluje případné poškozené clustery.
- **print** (výpis tabulek FAT):  
Tento příkaz vypíše obsah tabulek FAT do souboru pro analýzu.
- **help** (zobrazení nápovědy):  
Vypíše seznam dostupných příkazů s jejich stručným popisem.
- **exit** (ukončení programu):  
Tento příkaz ukončí běh aplikace.

## 4 Použití a spuštění programu

Program se spouští následujícím způsobem:

1. Otevřete terminál a přejděte do složky s projektem.
2. Spusťte příkaz:

```
go run . test.dat
```

3. Argument `test.dat` je soubor, který slouží jako úložiště pro simulaci souborového systému. Pokud soubor neexistuje, program jej vytvoří; pokud již existuje, použije se jeho aktuální obsah.
4. Po spuštění programu můžete zadávat příkazy jako `mkdir`, `ls`, `cp` a další, které jsou popsány výše.

## 5 Závěr

Tento projekt implementuje základní funkce souborového systému inspirovaného strukturou FAT, který umožňuje správu souborů a adresářů a zahrnuje operace jako kopírování, přesun, odstraňování souborů a kontrolu integrity systému.

Projekt byl realizován v jazyce Go, což pro mě představovalo novou výzvu, protože jsem s tímto jazykem neměl žádnou předchozí zkušenost. Ze začátku mi trvalo pochopit, jak souborový systém FAT funguje a jak implementovat jeho základní komponenty. Postupně jsem však získal lepší přehled o jeho struktuře a principu fungování.

Díky tomuto projektu jsem se nejen seznámil s jazykem Go, ale také prohloubil své znalosti o souborových systémech a jejich implementaci v praxi.