



# KIV/ZOS

Zjednodušený filesystem založený na i-uzlech

Martin Bruna (A20B0066P)

email: [brunam@students.zcu.cz](mailto:brunam@students.zcu.cz)

8. února 2021

# Obsah

<b>1</b>	<b>Zadání</b>	<b>2</b>
<b>2</b>	<b>Teorie</b>	<b>3</b>
2.1	Super blok . . . . .	3
2.2	I-node a cluster bitmapa . . . . .	3
2.3	Sektor i-nodu . . . . .	3
2.3.1	I-node . . . . .	3
2.4	Sektor clusterů . . . . .	3
<b>3</b>	<b>Uživatelská dokumentace</b>	<b>4</b>
<b>4</b>	<b>Programátorská dokumentace</b>	<b>4</b>
4.1	Console - Console.hpp + Console.cpp . . . . .	4
4.2	System - System.hpp + System.cpp . . . . .	4
4.3	Directory - Directory.hpp + Directory.cpp . . . . .	4
4.4	INode - INode.hpp + INode.cpp . . . . .	4
4.5	MemoryIterator - MemoryIterator.hpp + MemoryIterator.cpp . . . . .	4
4.6	FileSystem - FileSystem.hpp + FileSystem.cpp . . . . .	5

# 1 Zadání

Tématem semestrální práce bude práce se zjednodušeným souborovým systémem založeným na i-uzlech.

Systém bude podporovat příkazy: cp, mv, rm,mkdir, rmdir, ls, cat, cd, pwd, info, incp, outcp, load, format, ln

Předpokládáme korektní syntax zadávaných příkazů, ale ne sémanticky. Délka souboru je omezena na 8+3 znaků plus ukončující 12. byte.

## 2 Teorie

Souborový systém založený na i-nodech se skládá z několika částí

### 2.1 Super blok

Hlavička systému, obsahuje počáteční adresy jednotlivých částí a další konfiguraci celého systému.

### 2.2 I-node a cluster bitmapa

Dvě sekce, která každá představuje jedno bitové pole označující, které části i-node/cluster sektoru jsou zaplněny/volné.

### 2.3 Sektor i-nodu

Sektor, kde se nachází všechna informační data i-nodu.

#### 2.3.1 I-node

I-node je identifikátor a nositel informací pro jednotlivé soubory. Má na sobě flag zda jde o složku nebo běžný soubor. Zároveň obsahuje 5 direct linků (přímý odkaz na paměť v cluster sektoru), adresu na indirect cluster, který v sobě má seznam adres na reálná data a v poslední řadě odkaz na 2x nepřímý odkaz (odkaz na seznam odkazů na odkazy do reálných dat).

Kromě toho má i-node na sobě uloženo, jaká je velikost dat a počet složek ve kterých se na daný i-node odkazuje.

### 2.4 Sektor clusterů

Sektor, kde se nachází data souborů.

### 3 Uživatelská dokumentace

Program byl testován s překladem pomocí cmake ve verzi 3.19.

Jako parametr program přijímá cestu k souboru do kterého je/bude uložen celý filesystem. Pokud parametr není zadán je defaultně zvolen soubor fs.dat

Před používáním nového filesystemu je potřeba provést formátování pomocí příkazu format.

Příkazy: viz. zadání

## 4 Programátorská dokumentace

Hlavním vstupem do programu je `main.cpp`, který spustí instanci `Console`

### 4.1 Console - `Console.hpp` + `Console.cpp`

Tvoří uživatelský interface aplikace a předává uživatelem zadané příkazy dál

### 4.2 System - `System.hpp` + `System.cpp`

Obsahuje veškerou vysokoúrovňovou logiku - poskytuje implementaci jednotlivých příkazů

### 4.3 Directory - `Directory.hpp` + `Directory.cpp`

Poskytuje možnost práce se složkami

### 4.4 INode - `INode.hpp` + `INode.cpp`

Obalka logiky kolem samotného inodu, primárně poskytuje vstupní/výstupní stream.

### 4.5 MemoryIterator - `MemoryIterator.hpp` + `MemoryIterator.cpp`

Vlastní logika průchodu daty inodu, vytváření nových odkazů/mazání nepotřebných

### 4.6 FileSystem - `FileSystem.hpp` + `FileSystem.cpp`

Nejnižší úroveň - přístup k zapisování přímo na filesystem, řeší správu bitmap, formátování zápis a čtení z cluterů.

## 5 Závěr

Výsledkem práce je funkční zjednodučený i-node based filesystem. Systém je zpracovaný v C++. Práce zabrala přibližně 30 hodin.