APLICAȚIE

SIMPLE FILESYSTEM



Îndrumător: Realizat de:

Slt. Adina VAMAN Sd. Sg. Ștefan Gheorghe Sd. Sg. Sebastian Rădulescu Grupa C113E

**Cuprins**

Capitol 1 - Introducere 3

* 1. [Scopul proiectului 3](#_TOC_250004)
  2. [Lista definițiilor 3](#_TOC_250003)
  3. [Structura documentului 4](#_TOC_250002)

[Capitolul 2 – Arhitectură și componente SW 5](#_bookmark0)

* 1. [Descrierea produsului SW 5](#_bookmark1)
  2. [Detalierea platformei SW/HW 5](#_TOC_250001)
  3. [Actori 5](#_bookmark1)
  4. [Arhitectura internă 6](#_TOC_250000)

**Capitolul 3** – **Interfața grafică** 7

* 1. **Arbore funcționalități** 7
  2. **Descriere funcționalități** 8

## Capitolul 4 – Testare funcționalități 9

**Capitolul 1 – Introducere**

## Scopul proiectului

Scopul acestui proiect este de a dezvolta o aplicație care să permită utilizatorilor să creeze și să gestioneze o partiție de tip „raw“ ca un sistem de fișiere simplu. Această aplicație va oferi funcționalități de citire și scriere în partiția „raw“ și va furniza o interfață de utilizator pentru a efectua operații pe sistemul de fișiere simulat.

## Lista definițiilor

File System – sistem de fișiere.

Partiție “raw” – o zonă de stocare pe disc care nu este formatată cu un sistem de fișiere specific, permițând gestionarea directă a datelor brute.

Citire – operația de extragere a datelor din partiție sau sistemul de fișiere.

Scriere – operația de adăugare a datelor în partiție sau sitemul de fișiere.

Interfață de utilizator – o componentă a aplicației care permite utilizatorilo să interacționeze cu sistemul de fișiere.

Simulare – procesul de imitare a comportamentului unui sistem de fișiere real în cadrul aplicației.

## Structura documentului

Documentul este împărțit în trei capitole. Capitolul 1 reprezintă introducerea. Capitolul 2 prezintă actorii (tipurile de utilizatori ai aplicației) și arborele de fișiere al aplicației. Capitolul 3 prezintă interfața grafică și

funcționalitățile pe care aceasta le-o oferă utilizatorului. Capitolul 4 cuprinde exemple de testare a funcționalităților aplicației.

# Capitolul 2 – Arhitectură și componente SW

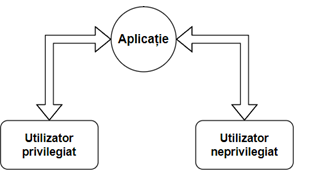
* 1. **Descrierea produsului software**

Aplicația va fi dezvoltată în limbajul de programare C/C++, iar interfața grafică va fi realizată cu ajutorul framework-ului Qt sau cu ajutorul bibliotecii GTK.

## Detalierea platformei SW/HW

Produsul software este dezvoltat pentru dispozitivele pe care rulează sistemul de operare Linux. Vom utiliza platforma Qt Creator pentru crearea interfeței prietenoase cu utilizatorul și mediul de dezvoltare Microsoft Visual Studio Code.

## Actori



Utilizator obișnuit (neprivilegiat)- poate efectua operații de scriere,citire, creare, ștergere și navigare.

Utilizator administrator (privilegiat)- poate accesa toate opțiunile oferite de aplicație.

## Arhitectura internă

Arhitectura internă a aplicației noastre de sistem de fișiere este structurată în jurul a două componente principale:

1. *Interfața grafică (GUI):*

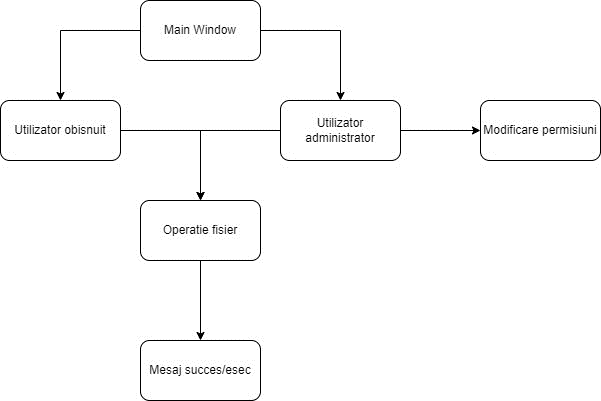
* Interfața grafică oferă utilizatorilor obișnuiți o modalitate intuitivă de a interacționa cu sistemul de fișiere. Acesta include elemente precum ferestre, butoane, meniuri și casete de dialog pentru a efectua operații precum citire, scriere, creare, ștergere și navigare.
* Interfața grafică poate fi configurată pentru utilizatori cu privilegii de administrator pentru a efectua operații de administrare.

1. *Logică de sistem de fișiere:*

* Această componentă este responsabilă de gestionarea partiției „raw“ și a operațiilor de citire și scriere în cadrul acesteia.
* Administrează structura și organizarea datelor pe partiție, precum și asigură gestionarea corectă a permisiunilor și a accesului utilizatorilor.
* Interacționează cu interfața grafică pentru a afișa informații, a primi comenzi și a returna rezultatele operațiilor.
* Pentru utilizatorii obișnuiți, interfața grafică reprezintă principalul punct de interacțiune, în timp ce pentru utilizatorii cu privilegii de administrator, aceștia pot accesa funcționalități suplimentare pentru gestionarea și configurarea partiției „raw“.
* Arhitectura noastră internă asigură o separare clară între interfața grafică și logică de sistem de fișiere, facilitând dezvoltarea, întreținerea și extinderea aplicației noastre de sistem de fișiere.

# Capitolul 3 – Interfața grafică

* 1. **Arbore funcționalități aplicație**



* 1. **Descriere funcționalități**

După rularea programului, se va deschide o interfață grafică, prietenoasă cu utilizatorul. Prima fereastră deschisă este ”Main Window”, care va fi aferentă pentru tipul de utilizator – obișnuit/administrator.

În funcție de tipul de utilizator, vor apărea butoane pentru a executa operațiile de scriere, citire, ștergere, creare asupra unui fișier – pentru ambii utilizatori.

În plus, pentru administrator va exista opțiune de a modifica drepturile pentru anumiți utilizatori sau pentru anumite fișiere.

Fereastra ”Main Window”:

Implicit, aplicația va verifica existenței partiției “raw”.

În funcție de opțiunea aleasă va fi necesară completarea anumitor câmpuri care ulterior vor forma utilitarul care va executa procesele aferente pe partiție.(ex. fișier, conținut etc.)

La final, aplicația va afișa un mesaj de succes sau eșec pentru ceea ce s-a dorit.

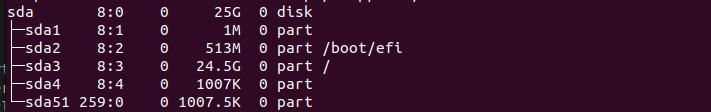
De asemenea, aplicația va deschide fișierul dacă opțiunea de “deschidere dupa rulare” este bifată – această opțiune este disponibila doar pentru anumite operații asupra fișierelor.

**Capitolul 4 – Testare funcționalități**

**Test 1- Verificare existență partiție “raw”**

Input: lsblk

Output:



**Test 2- Verificare funcționalitate scriere pe partiție**



**Test 3- Verificare funcționalitate creare fișier dat ca argument**



Aici trebuie sa facem noi sistem de open() si alte functii pentru fisiere

Write a file system from scratch – de cautat -> respository uri

Operatii basic – creare fisier, creare director

Exista pe github si peste tot implementare de sistem de fisiere....