

București - 2025



Universitatea Tehnică de Construcții din București
Facultatea de Hidrotehnică
Specializarea: Automatică și Informatică Aplicată

Proiect PCLP II

Coordonatori:

Conf. Giorgian Neculoiu

Ing. Cosmin Fudulu

Ing. Gabriela Olteanu

Student:

Cîrlea Ioan Călin

București - 2025



Universitatea Tehnică de Construcții din București
Facultatea de Hidrotehnică
Specializarea: Automatică și Informatică Aplicată

Editor de text

Cuprins

1. Introducere
 - a. Motivare alegere tema
 - b. Obiective
2. Tehnologii utilizate
3. Studiu De caz - Tema aleasa
 - a. Implementare pas cu pas
4. Concluzie
5. Bibliografie
6. Anexe

1. Introducere

Motivarea alegerii temei

Am ales sa dezvolt acest editor de text simplu deoarece este un instrument util in viata de zi cu zi. Un editor de text permite utilizatorilor sa creeze, sa citeasca si sa modifice fisiere text rapid si eficient. Acest tip de program poate fi folosit pentru o varietate de sarcini, cum ar fi scrierea de note, sau chiar scrierea de cod sursa. Prin dezvoltarea acestui program, am dorit sa creez un instrument care sa fie usor de utilizat si sa raspunda nevoilor utilizatorilor.

Un alt motiv important pentru alegerea acestei teme a fost dorinta de a deveni mai bun la utilizarea fisierelor in limbajul C. Lucrul cu fisiere este o parte esentiala a programarii, iar intelegerea modului in care se deschid, se citesc, se scriu si se inchid fisierele este importanta pentru dezvoltarea de aplicatii eficiente. Prin dezvoltarea acestui editor de text, am avut ocazia sa exersezi si sa imi imbunatatesc abilitatile in manipularea fisierelor, invatand cum sa gestionez erorile si sa asigur ca programul functioneaza corect in diverse situatii.

Obiectivele Proiectului

Unul dintre principalele obiective ale acestui proiect a fost dezvoltarea unui program care sa functioneze eficient si fara erori in mediul de consola. Acest lucru implica:

1. Gestionarea Corecta a Resurselor: Asigurarea ca toate fisierele deschise sunt inchise corect dupa utilizare pentru a preveni pierderi de resurse.

2. Validarea si Gestionarea Erorilor: Implementarea de verificari pentru a asigura ca operatiunile cu fisiere (cum ar fi deschiderea, citirea, scrierea si stergerea) sunt efectuate fara erori.

3. Optimizarea Performantei: Asigurarea ca programul raspunde rapid si eficient la cerintele utilizatorului, chiar si in cazul in care fisierele sunt de dimensiuni mari.

Un alt obiectiv major al proiectului a fost implementarea functionalitatilor de baza pentru manipularea fisierelor text direct din consola:

1. Scrierea in Fisiere: Permitearea utilizatorului sa introduca text si sa il salveze intr-un fisier specificat. Aceasta functionalitate este esentiala pentru crearea si editarea de fisiere text.

2. Citirea din Fisiere: Permitearea utilizatorului sa citeasca continutul unui fisier text si sa il afiseze in consola. Aceasta functionalitate este utila pentru revizuirea si editarea fisierelor existente.

3. Stergerea Fisierelor: Permite utilizatorului sa stearga un fisier specificat. Aceasta functionalitate este importanta pentru gestionarea fisierelor si eliberarea spatiului de stocare.

2. Tehnologii utilizate

Limbajul de programare C

Proiectul a fost dezvoltat folosind limbajul de programare C, care este cunoscut pentru controlul sau de nivel scazut asupra resurselor sistemului. C este un limbaj puternic si versatil, ideal pentru dezvoltarea de aplicatii care necesita manipularea directa a fisierelor si gestionarea eficienta a memoriei.

De asemenea este un limbaj usor de inteles si cu care se poate lucra usor. Principalul motiv pentru alegerea de a lucra in C si nu in C++ este acela de a ma familiariza mai mult cu scrierea in C si pentru a aduna mai multa experienta pentru viitor.

Medii de Dezvoltare Integrata (IDE)

Pentru dezvoltarea si testarea programului, au fost utilizate urmatoarele medii de dezvoltare integrata (IDE):

GDB Online: Un mediu de dezvoltare online care permite scrierea, compilarea si executarea codului C direct in browser. Acest instrument este util pentru testarea rapida a codului fara a fi necesara instalarea unui mediu de dezvoltare local acesta necesitand doar o conexiune la internet si un browser.

Code::Blocks: Un mediu de dezvoltare integrata open-source care suporta multiple limbaje de programare, inclusiv C. Code::Blocks este cunoscut pentru usurinta sa de utilizare si pentru functionalitatile sale avansate de depanare, fiind ideal pentru dezvoltarea si testarea aplicatiilor

locale. De asemenea stiu foarte bine sa il utilizez si sa navighez interfata datorita experientei mele dobandite pe parcursul anilor de liceu.

Platforma GitHub

Pentru gestionarea versiunilor si colaborarea la proiect, a fost utilizat GitHub, o platforma de gestionare a codului sursa bazata pe Git. GitHub este o platforma des folosita in mediul de programare datorita functiilor ei utile.

Uploadarea si Stocarea Codului: Codul sursa al proiectului a fost incarcat pe GitHub pentru a fi stocat intr-un repository centralizat, accesibil de oriunde.

Colaborare si Distribuie: Platforma faciliteaza colaborarea intre dezvoltatori si distribuirea codului catre utilizatori sau alte parti interesate.

3. Studiu de caz - Tema aleasa

Implementare pas cu pas

A. Declararea variabilelor si initializarea programului

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main() {
    int max_size=1000;
    char input[max_size];
    char filename[100];
    FILE *file;
    char choice;
```

La inceputul programului, sunt definite variabilele esentiale pentru functionarea aplicatiei. Acestea includ un buffer generos pentru manipularea textului introdus de utilizator, o variabila pentru retinerea numelui fisierului asupra caruia se va lucra, precum si un pointer de tip FILE necesar pentru deschiderea si manipularea fisierelor prin operatii standard (citire, scriere, stergere).

In plus, se defineste o variabila care va retine optiunea aleasa de utilizator din meniul aplicatiei. Aceasta structura de baza pregateste cadrul general al programului si asigura un control minim asupra dimensiunilor

datelor manipulate, prevenind erori frecvente precum depasirea limitelor de memorie sau lucrul cu fisiere inexistente.

B. Interfata cu utilizatorul si preluarea comenzilor

```
printf("Options:\n");
printf("W (Write)\n");
printf("R (Read)\n");
printf("D: (Delete)\n");
printf("Awaiting input: ");
scanf("%c", &choice);

getchar();

printf("Enter the file name: ");
scanf("%s", filename);
getchar();
```

Aceasta parte a programului reprezinta interfata initiala cu utilizatorul si marcheaza punctul in care aplicatia incepe sa interactioneze activ cu acesta. Se afiseaza un meniu simplu, care descrie cele trei optiuni disponibile: scriere intr-un fisier (W), citirea continutului unui fisier (R) sau stergerea unui fisier (D). Dupa afisarea optiunilor, programul asteapta ca utilizatorul sa introduca o comanda printr-un caracter specific.

Ulterior, se cere introducerea numelui fisierului asupra caruia urmeaza sa se aplice actiunea selectata. Pentru a asigura functionarea corecta a citirii datelor din consola, se foloseste `getchar()` pentru a consuma eventualul caracter newline ramas in buffer dupa `scanf`, prevenind astfel probleme in

citirea ulterioara. Aceasta secventa pregateste contextul operational pentru executarea efectiva a comenzii alese.

C. Procesarea optiunii de scriere in fisier

```
switch (choice) {  
    //scris  
    case 'W':  
    case 'w':  
        file = fopen(filename, "w");  
        if (file == NULL){  
            perror("Error");  
            return 1;  
        }  
  
        printf("Insert text: \n");  
        while (fgets(input, max_size, stdin) != NULL){  
            if(strcmp(input, "EOF\n")==0) {  
                break;  
            }  
            fprintf(file, "%s", input);  
        }  
  
        fclose(file);  
        printf("Text saved to %s\n", filename);  
        break;  
}
```

Procesarea optiunii de scriere in fisier cu oprire la "EOF"

Aceasta sectiune a codului gestioneaza functionalitatea de scriere in fisier atunci cand utilizatorul alege optiunea W sau w. Programul deschide fisierul indicat in modul de scriere ("w"), ceea ce va rescrie complet continutul daca fisierul exista deja sau il va crea daca nu exista. Daca deschiderea fisierului esueaza, se afiseaza un mesaj de eroare si executia programului se opreste. In cazul in care fisierul este accesibil, utilizatorul este invitat sa introduca textul dorit. Acesta poate scrie mai multe linii, iar introducerea continua pana cand o linie contine exact cuvantul EOF urmat de

un newline (\n), semn care indica oprirea scrierii. Fiecare linie introdusa este scrisa imediat in fisier prin fprintf.

Aceasta metoda ofera utilizatorului control asupra duratei introducerii si previne trimiterea accidentala a semnalului de sfarsit de fisier (Ctrl+D sau Ctrl+Z). Sectiunea nu include inca inchiderea fisierului (fclose), ceea ce sugereaza ca urmatoarea parte a codului finalizeaza aceasta operatie.

D. Procesarea optiunii de citire a continutului unui fisier

```
//citit
case 'R':
case 'r':
    file=fopen(filename, "r");
    if(file == NULL){
        perror("Error");
        return 2;
    }

    printf("Contents of %s:\n", filename);
    while (fgets(input, max_size, file)!=NULL){
        printf("%s", input);
    }

    fclose(file);
    break;
```

Aceasta parte a programului este responsabila pentru afisarea continutului unui fisier text atunci cand utilizatorul selecteaza optiunea R sau r. Programul deschide fisierul specificat anterior in modul de citire ("r"), iar daca fisierul nu poate fi deschis (de exemplu, nu exista sau permisiunile nu permit accesul), este afisat un mesaj de eroare, iar programul se opreste cu un cod de eroare corespunzator.

În cazul în care fișierul este deschis cu succes, se afișează un mesaj informativ cu numele fișierului și, folosind o buclă cu `fgets`, sunt citite și afișate pe ecran, linie cu linie, toate datele conținute în fișier. Această metodă permite afișarea întregului conținut într-un mod clar și lizibil, fără a încărca inutil memoria. La final, fișierul este închis folosind `fclose` pentru a elibera resursele asociate.

E. Procesarea opțiunii de ștergere a unui fișier

```
//stergere
case 'D':
case 'd':
    printf("Enter the file name to delete it: ");
    scanf("%s", filename);
    getchar();

    if (remove(filename)==0){
        printf("File %s deleted successfully.\n", filename);
    }
    else{
        perror("Error");
        return 3;
    }
    break;
```

Această parte a programului gestionează funcționalitatea de ștergere a unui fișier atunci când utilizatorul alege opțiunea D sau d. Programul solicită utilizatorului să introducă numele fișierului care urmează să fie șters. După ce primește acest input, încearcă să elimine fișierul folosind funcția standard `remove`.

Dacă operația are succes, utilizatorul este informat că fișierul a fost șters cu succes. În cazul în care ștergerea eșuează (de exemplu, dacă fișierul nu există sau nu există permisiuni suficiente), este afișat un mesaj de eroare care descrie problema. Astfel, această secțiune asigură o metodă simplă și

clară pentru eliminarea fișierelor din sistem, cu feedback imediat pentru utilizator.

F. Tratarea optiunilor invalide si incheierea programului

```
//default-ul  
default:  
    printf("Invalid option.\n");  
    return 4;  
}  
return 0;  
}
```

Aceasta sectiune reprezinta blocul implicit al instructiunii `switch`, care se executa atunci cand utilizatorul introduce o optiune nevalida, adica un caracter care nu corespunde niciunei comenzi disponibile (W, R sau D). In acest caz, programul afiseaza un mesaj clar de eroare, notificand utilizatorul ca optiunea aleasa nu este recunoscuta. Pentru a indica acest tip de eroare la nivelul sistemului, programul returneaza un cod de iesire specific (4), semnalizand astfel ca executia s-a incheiat din cauza unei optiuni gresite.

Dupa finalizarea blocului `switch`, programul se incheie normal cu `return 0`, indicand terminarea corecta a procesului in cazul in care una dintre optiunile valide a fost executata fara probleme.

4. Concluzie

Concluzii

Prin dezvoltarea acestui editor de text simplu, am atins mai multe obiective propuse in etapa de planificare a proiectului. Primul si cel mai important obiectiv a fost dezvoltarea unui program functional care permite utilizatorilor sa scrie, sa citeasca si sa stearga fisiere text direct din consola. Utilizatorii pot introduce text, pot citi continutul fisierelor existente si pot sterge fisierele care nu mai sunt necesare. Aceasta functionalitate de baza este esentiala pentru gestionarea eficienta a fisierelor text.

Un alt obiectiv important a fost asigurarea unei interfete de utilizator simple si intuitive, care sa permita utilizatorilor sa interactioneze usor cu programul. Prin utilizarea unui meniu clar si a instructiunilor concise, am reusit sa creez un mediu prietenos pentru utilizatori, chiar si in cadrul unei interfete de consola.

De asemenea, am considerat important sa asigur ca programul functioneaza eficient si fara erori, gestionand corect resursele si validand operatiunile cu fisiere. Aceasta abordare a permis dezvoltarea unui program robust si fiabil, care poate fi utilizat in diverse situatii practice.

Consider ca prin dezvoltarea acestui editor de text, am adus un aport la tehnologia de manipulare a fisierelor text in limbajul C, oferind o solutie simpla si eficienta pentru gestionarea fisierelor. Un aspect important al programului este utilizarea functiilor standard de manipulare a fisierelor, cum ar fi `fopen`, `fgets`, `fprintf` si `remove`, care permit o gestionare eficienta a operatiunilor cu fisiere.

Contributii Personale

Pe parcursul dezvoltarii acestui proiect, am avut oportunitatea de a imi imbunatati abilitatile de programare in limbajul C si de a intelege mai bine conceptul de manipulare a fisierelor. Am invatat cum sa gestionez erorile si sa asigur ca programul functioneaza corect in diverse situatii. De asemenea, am avut ocazia de a lucra cu instrumente de dezvoltare si depanare, cum ar fi GDB Online si Code::Blocks, care m-au ajutat sa identific si sa rezolv erorile in codul sursa.

Un alt aspect important al contributiei mele personale a fost dezvoltarea unei interfete de utilizator simple si intuitive, care sa permita utilizatorilor sa interactioneze usor cu programul. Prin utilizarea unui meniu clar si a instructiunilor concise, am reusit sa creez un mediu prietenos pentru utilizatori, chiar si in cadrul unei interfete de consola.

In concluzie, acest proiect reprezinta un pas important in dezvoltarea mea ca programator si imi ofera o baza solida pentru proiecte viitoare. Prin atingerea obiectivelor propuse si prin dezvoltarea unui program functional si eficient, consider ca am adus un aport semnificativ la tehnologia de manipulare a fisierelor text in limbajul C.

Perspective de viitor

Ca si perspective de dezvoltare ulterioara a aplicatiei, imi doresc ca pe viitor sa adaug un modul avansat de editare, care sa permita utilizatorilor sa editeze fisiere text in moduri mai detaliate. Acest modul ar putea include functionalitati precum cautarea si inlocuirea de cuvinte sau expresii, numerotarea liniilor, evidentierea modificarilor fata de o versiune anterioara, precum si posibilitatea de a salva versiuni multiple ale aceluasi fisier. De asemenea, intentionez sa implementez un sistem de autentificare care sa permita accesul diferentiat in functie de utilizator (de exemplu: utilizator obisnuit vs. administrator), astfel incat anumite operatii precum stergerea fisierelor sa fie restrictionate doar pentru administratori.

Prin aceste functionalitati, doresc sa extind capabilitatile aplicatiei si sa o transform treptat intr-un instrument util pentru gestionarea eficienta a fisierelor text, oferind in acelasi timp o experienta imbunatatita si personalizata pentru fiecare utilizator.

5. Bibliografie

<https://stackoverflow.com/questions/4358728/end-of-file-eof-in-c>

<https://www.geeksforgeeks.org/basics-file-handling-c/>

<https://www.geeksforgeeks.org/c-program-delete-file/>

https://github.com/gabi242000/AIA_LP-2025/tree/main/Laboratoare

[Curs | 2024-2025 FH-AIA I-Prog. calc. si limbaje de programare II | Microsoft Teams](#)

https://www.reddit.com/r/C_Programming/