

## Lucrul cu fisiere

### Tipuri de fisiere :

- **Text** ce contine o succesiune de linii (succesiune de caractere ASCII) separate prin caracterul \n – new line
- **Binare** ce continut o succesiune de octeți

Lucrul cu fisiere presupune alegerea tipului de canal: **I(Input)** sau **E(Output)** numit flux.

Fluxuri (Canale) predefinite (de stdio.h):

- **stdin (standard input):** Acest flux este asociat cu dispozitivul standard de intrare, care, în mod implicit, este tastatura. Când se utilizează funcții de intrare precum **scanf** sau **getchar**, acestea citesc date de la tastatură și utilizează fluxul menționat.
- **stdout (standard output):** Acest flux este asociat cu dispozitivul standard de ieșire, care, în mod implicit, este ecranul. Funcțiile de ieșire precum **printf** utilizează fluxul stdout pentru a afișa date pe ecran.
- **stderr (standard error):** Acest flux este, de asemenea, asociat cu dispozitivul standard de ieșire și este utilizat pentru a afișa mesaje de eroare.

Etapele prelucrării unui fișier:

- definirea unei variabile de tip **FILE\***. Aceasta variabilă conține informații despre:
  - Adresa;
  - Buffer size – dimensiunea blocurilor de transfer dintre RAM și fișier (locația în care este memorat);
  - Modul de utilizare al fișierului (scriere/citire...);
  - Indicator de sfârșit de fișier;
  - Indicator de poziție în fișier.

Funcții în C pentru prelucrarea fișierelor:

- **FILE \*fopen (const char \* numeFisier, const char \* mod);**

Deschide fișierul cu numele numeFisier pentru acces în modul *mod*

Modul poate fi:

- **r (read):** Deschide fișierul doar pentru citire. Fișierul trebuie să existe deja.
- **w (write):** Deschide fișierul pentru scriere. Dacă fișierul există, conținutul său va fi șters. Dacă nu există, un fișier nou va fi creat.
- **a (append):** Deschide fișierul pentru adăugare la sfârșitul fișierului. Dacă fișierul nu există, se va crea unul nou.

- + – permite scrierea și citirea din același fișier: r+ / w+ / a+ (O citire nu poate fi direct urmată de o scriere și reciproc. Întâi trebuie repositionat cursorul de acces printr-un apel la fseek.)

- t sau b specifică tipul fișierului: text sau binar – dacă lipsește – este considerat text

- **fclose(FILE\* numeFisier) :** Functia inchide fisierul numeFisier
- **fseek(FILE\* numeFISIER, long offset, int whence);**

Functia este folosită pentru a seta poziția cursorului de fișier asociat unui fișier deschis în C. Această funcție este utilă atunci când doriți să vă deplasați înainte sau înapoi în conținutul fișierului sau să vă poziționați la sfârșitul acestuia.

Parametrul whence poate fi:

- SEEK\_SET = 0 - Setează cursorul la offset octeți de la începutul fișierului.
- SEEK\_CUR = 1 - Deplasează cursorul cu offset octeți față de poziția curentă a cursorului.
- SEEK\_END = 2 - Setează cursorul la offset octeți înainte de sfârșitul fișierului.

- **ftell(FILE\* numeFISIER);**

Functia este folosită pentru a obține poziția curentă a cursorului de fișier asociat unui fișier deschis în C. Această funcție returnează un indicator la poziția curentă în octeți față de începutul fișierului.

#### **Citire/ scriere caracter cu caracter**

- **int fgetc(FILE \*f):** Această funcție citește un caracter din fișierul referit de f și returnează valoarea citită ca un caracter fără semn extins la întreg. În caz de eroare sau la sfârșitul fișierului, funcția returnează EOF.

- **int fputc(int c, FILE \*f):** Această funcție scrie caracterul c în fișierul referit de f și returnează caracterul scris sau EOF în caz de eroare.

#### **Citire/ scriere linie cu linie**

- **int fgets(char \*linie, int max, FILE \* f) -** Citește maxim max-1 caractere sau până la întâlnirea \n inclusiv și returnează adresa sirului linie sau NULL în caz de eroare

- **int fputs (char \*linie, FILE \* f);** -Scrie șirul linie în fișierul referit de f

- **int feof (FILE \* fp) ;**

Functia returneaza 0 nu s-a detectat sfârșitul fișierului si o valoare nenula altfel.

## Problema 1:

Se cere scrierea unei matrici intr-un fiser cu numele DATA.txt. Se va citi apoi din acelasi fiser matricea.

```
#include <stdio.h>
void main(void)
{FILE *fp;
  char linie[200];
  float a;
  int i,j, m, n;
  if ((fp = fopen("DATA.txt", "w")) == NULL)
    printf("Eroare la deschidere DATA.DAT pentru scriere\n");
  else
  {
    printf("Introduceti numarul de linii: ");
    scanf("%d", &n);
    printf("Introduceti numarul de coloane: ");
    scanf("%d", &m);
    for (i=0; i<n; i++)
    {
      printf("Elemente linie %d \n", i);
      for (j=0; j < m; j++)
      {
        printf("Element[%d]=", j);
        scanf("%f", &a);
        fprintf(fp, "%.2f ", a);
      }
      fprintf(fp, "\n");
    }
    fputs("Matricea a fost scrisa cu succes!", fp);
    fclose(fp);
  }
  if ((fp = fopen("DATA.txt", "r")) == NULL)
    printf("Eroare la deschidere DATA.DAT pentru citire\n");
  else
  {
    printf("Elementele matricei sunt:\n");
    for (i=0; i<n; i++)
    {
      for (j=0; j < m; j++)
      {
        fscanf(fp, "%f", &a);
        printf("%.2f\t", a);
      }
      printf("\n");
    }
    fclose(fp);
  }
}
```

## Problema 2:

Se cere numarul de bytes continuti intr-un fisier. De asemenea, se va afisa dintr-un fisier doar sirurile de caractere.

```
#include <stdio.h>

int sizeof_file(char *filename) {
    // deschid fisierul pentru a citi
    FILE *file = fopen(filename, "rt");
    // verific daca fisierul a fost deschis cu succes
    if (file == NULL) {
        return -1;
    }
    // pozitionarea la sfarsit de fisier
    fseek(file, 0, SEEK_END);
    // aflu pozitia ultimului caracter cu ajutorul ftell
    int bytes_count = ftell(file);
    // inchid fisierul
    fclose(file);
    return bytes_count;
}

void afisare_continut(char *filename) {
    FILE *file = fopen(filename, "rt");
    char s[100];

    if (file == NULL) {
        return -1;
    }
    while( fgets(s, 100, file) != NULL)
        printf("%s", s);
    fclose(file);
}

int main() {
    // numele fisierului de intrare
    char filename[] = "DATA.txt";

    int sz = sizeof_file(filename);
    if (sz < 0) {
        // Eroare, nu se poate deschide fisierul
        fprintf(stderr, "ERROR: Fisierul nu a putut fi deschis %s", filename);
        return -1;
    }
    printf("Afisez continutul\n");
    afisare_continut(filename);
    printf("\nFisierul %s are %d bytes\n", filename, sz);
    fprintf(stdout, "Fisierul %s are %d bytes\n", filename, sz);
    return 0;
}
```

### Problema 3:

Se citesc dintr-un fisier 4 note de laborator ale unui elev si nota de la teza. Se cere media sa anuala, aceasta va fi salvata in fisierul media.txt.

```
/* Se citesc dintr-un fisier 4 note de laborator ale unui elev si nota de la teza.
Se cere media sa anuala, aceasta va fi salvata in fisierul media.txt */
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    float n1,n2,n3,n4,t;
    float mo,mf;
    ifstream f("note.txt");
    ofstream g("media.txt");
    f>>n1;
    cout<<"Nota 1: "<<n1<<endl;
    f>>n2;
    cout<<"Nota 2: "<<n2<<endl;
    f>>n3;
    cout<<"Nota 3: "<<n3<<endl;
    f>>n4;
    cout<<"Nota 4: "<<n4<<endl;

    mo=(float) (n1+n2+n3+n4)/4;
    f>>t;
    cout<<"Nota la teza: "<<t<<endl;
    cout<<"Media notelor este: "<<mo<<endl;
    mf=(mo*3+t)/4;
    cout<<"Media finala este: "<<mf<<endl;
    g<<"Media finala este: "<<mf<<endl;
    f.close();
    f.close();
    return 0;
}
```