Laborator 12

Fişiere

Un fişier este o colecţie de date omogene stocate pe suport extern, identificabilă după nume.

În limbajul C se lucrează cu două tipuri de fişiere: *fişiere text* şi *fişiere binare*.

Înaintea oricărui acces la informaţiile dintr-un fişier trebuie executată o operaţie de deschidere. După terminarea prelucrărilor informaţiilor din fişier trebuie executată o operaţie de închidere logică a fişierului.

1.Deschiderea unui fişier

Deschiderea unui fişier se realizează prin apelul funcţiei predefinite `fopen`. Forma generală a apelului acestei funcţii este următoarea:

```
f = fopen(nume, mod);
```

unde:

`f` – pointer la fişier

`nume` – numele fişierului care va fi prelucrat; nume este un şir de caractere ce identifică numele fişierului conform sintaxei prevăzute de sistemul de operare sub care se lucrează.

`mod` - un şir de caractere ce indică modul în care se va deschide fişierul.

Funcţia fopen deschide un fişier nou sau unul existent într-unul din următoarele moduri:

w	scriere (vechiul conţinut al fişierului se pierde)
a	adăugare de înregistrări la fişierul existent (append)
r	citire secvenţială din fişier
+	se permit citiri şi scrieri
t	fişierul deschis este un fişier text
b	fişierul deschis este un fişier binar

Pointerul `f` identifică în continuare fişierul deschis prin `fopen` şi el se utilizează în continuare ca parametru în toate funcţiile de prelucrare a fişierului respectiv. Când fişierul nu poate fi deschis, funcţia `fopen` returnează `NULL`.

Exemplu

```
FILE *f;              // declarare pointer la fişier
// . . .
f = fopen("Date.txt",”rt”); //deschidere fişier text pentru
citire
if (f == NULL) //test
     printf(“\nEroare la deschiderea fisierului !”);
else
{
     // prelucrari
}
// . . .
```

2.Închiderea unui fişier

După ce toate operaţiile asupra unui fişier s-au terminat, fişierul trebuie închis. Acest lucru se face printr-un apel al funcţiei `fclose`:

```
fclose(f);
```

unde `f` este pointerul la fişierul care va fi închis. Funcţia `fclose` returnează o valoare egală cu zero dacă închiderea a avut loc sau EOF dacă s-au detectat erori.

Închiderea unui fişier cere sistemului de operare să golească toate bufferele discului asociate fişierului şi să elibereze resursele de sistem consumate de fişier, cum ar fi datele pointerului de fişier. În general, programele în C nu testează valoarea returnată de funcţia `fclose`. În cele mai multe cazuri, dacă la închiderea unui fişier apare o eroare, programul nu poate să facă prea multe pentru a corecta situaţia.

3.Fişiere text

Într-un fişier text datele sunt memorate sub forma unei succesiuni de caractere, organizate pe linii. Fiecare caracter este memorat prin utilizarea codului ASCII. Astfel, caracterul 'a' ocupă în fişierul text un octet - codul ASCII al caracterului 'a'. Pentru o variabilă de tip int ce are valoarea 1234, fişierul text va reţine codurile a patru caractere: '1', '2', '3' şi '4'.
Un fişier text se termină întotdeauna cu caracterul EOF.

3.1.Funcții de citire dintr-un fişier text

`fgetc(f)`	Funcția citeşte caracterul curent din fişier. Dacă pointerul de fişier a ajuns la sfârşitul fişierului, funcția returnează constanta EOF.
`fscanf(f, format, arg)`	Funcția permite citirea cu format cu deosebirea că datele sunt preluate din fişierul `f`. Argumentele `format`, `arg` sunt identice cu argumentele funcției `scanf`.
`fgets(buf, lungime, f)`	Funcția fgets citeşte un şir în zona `buf`.Această funcție întoarce în mod normal adresa zonei `buf` sau o valoare nulă dacă s-a detectat EOF sau s-a produs o eroare.

3.2.Funcții pentru scriere în fişiere text

`fputc(c ,f)`	Funcția scrie caracterul `c` în locația curentă a fişierului pointat de `f` ce a fost deschis doar pentru scriere. Dacă apare o eroare la scriere, funcția returnează constanta EOF.
`fprintf(f, format, arg)`	Funcția permite scrierea datelor sub controlul formatului, în fişierul indicat de argumentul `f`. Argumentele `format` şi `arg` sunt identice cu argumentele funcției `printf`.
`fputs(buf,f)`	Funcția scrie şirul de la adresa `buf` în fişierul `f`. Întoarce ultimul caracter scris în fişier dacă nu s-a detectat nici o eroare sau EOF în caz contrar.

3.3.Alte funcții

`feof(f)`	Funcția este utilizată pentru a testa detectarea sfârşitului de fişier (EOF). Are ca argument pointerul fişierului prelucrat. Dacă s-a detectat sfârşitul de fişier `feof` returnează valoarea 1, în caz contrar valoarea zero.
`fseek(f, deplasament, origine)`	Funcția `fseek` permite poziționarea pe oricare octet din fişier, conform argumentelor efective. Următoarea

	operație de intrare/ieşire în fişier după apelul funcției `fseek` va fi executată asupra înregistrării poziționate pe aceasta. Returnează 0 la poziționare corectă şi valoare diferită de 0 (-1) în caz de eşec.
`ftell(f)`	Funcția returnează o valoare de tip long care defineşte poziția curentă a indicatorului de înregistrare şi anume, ea reprezintă deplasamentul, în octeți, a poziției indicatorului de înregistrare față de începutul fişierului.

3.4.Exemple

1.Să se scrie un program care citeşte conținutul unui fişier text şi îl afişează pe ecran.

```
#include <stdio.h>
#include <conio.h>

int main()
{
    FILE *f;
    f = fopen("Date.txt", "rt");
    if (f==NULL)
        printf("Eroare la deschiderea fisierului\n");
    else
    {
        while(!feof(f))
            putch(fgetc(f));
        fclose(f);
    }
    return 0;
}
```

2.Să se realizeze un program care scrie ***n*** numere întregi într-un fişier text.

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

int main()
{
  FILE  *f;
```

```
  int n,x,i;
  if ((f = fopen("Numere.txt","wt")) == NULL)
    {
      printf("\nEroare la crearea fisierului!");
      exit(0);
    }
  printf("n=");
  do
  {
    scanf("%d", &n);
  } while (n<=0 || n>30);
  fprintf(f, "%d ",  n);
  for (i=0; i<n; i++)
  {
    printf("Introduceti un numar:");
    scanf("%d", &x);
    fprintf(f,"%d ",x);
  }
  fclose(f);
  return 0;
}
```

4.Probleme propuse

1.Să se scrie programul C care determină numărul de linii dintr-un fişier text şi crează un alt fisier, cu liniile din primul aparând în ordine inversă.
2.Să se scrie programul C care determină numărul de cuvinte de pe fiecare linie a unui fişier text.
3.Să se scrie programul C care copie dintr-un fişier text într-un alt fişier text doar liniile care conţ in un anumit cuvânt, introdus de la tastatură.
4. Fie un vector cu maxim 50 de elemente întregi. Să se scrie un program care afişează şi prelucrează meniul:

C.Citire
A.Afisare
S.Scriere in fisier
F.Citire din fisier
P.Scriere numere pare in fisierul *pare.txt*
M.Media numerelor din *pare.txt*
I.Info autor
X.Exit