Laborator 6 – IO cu fisiere

Deschiderea unui fisier:

FILE *fopen(const char * filename, const char * mode);

	r
1	Deschide un fisier cu drepturi de citire.
	w
2	Deschide un fisier cu drepturi de scriere. Daca acesta nu exista, atunci este creat, iar daca exista, indexul de citire va fi la inceputul fisierului.
	a
3	Deschide un fisier cu drepturi de scriere. Daca acesta nu exista, atunci este creat, iar daca exista, indexul de citire va fi la sfarsitul fisierului.
	r+
4	Deschide un fisier cu drepturi de citire si de scriere.
	w+
5	Deschide un fisier cu drepturi de citire si de scriere. Daca fisierul exista, tot continutul este sters, iar daca nu exista acesta va fi creat.
	a+
6	Deschide un fisier cu drepturi de citire si de scriere. Daca fisierul exista, scrierea incepe de la sfarsitul continutului fisierului, iar daca nu exista acesta va fi creat.

Daca avem de a face cu fisiere binare: "rb", "wb", "ab", "rb+", "r+b", "wb+", "w+b", "ab+", "a+b".

Observatii:
- Functia returneaza un descriptor de fisier care este folosit de sistemul de operare pentru a identifica fisierul deschis si drepturile cu care este deschis.
- Calea catre fisier poate sa fie relativa (adica fisierul se afla in acelasi director cu codul sursa si trebuie sa mentionam doar numele acestuia, e.g: “fisier.txt”).
- Calea catre fisier poate sa fie absoluta (adica dam toata calea catre fisier, incepand de la partitie, pe windows, sau de la root, pe sistemele Unix, ex: “C:\\Program Files\\fisier.txt”).

Inchiderea unui fisier:

int fclose(FILE *fp);

Returneaza 0 in caz de succes si EOF in caz de eroare. Constanta EOF este definita in stdio.h.

Scrierea in fisier:

int fputc(int c, FILE *fp);

Scrie caracterul dat ca argument in fisier si returneaza acelasi caracter in caz de succes, EOF in caz de esec.

==========

int fputs(const char *s, FILE *fp);

Scrie sirul de caractere dat ca argument in fisier si returneaza o valoare pozitiva in caz de succes, EOF in caz de esec.

==========

int fprintf(FILE *fp,const char *format, …);

Se poate folosi si fprintf care returneaza o valoare pozitiva in caz de succes si EOF in caz de esec, dar care poate scrie formatat in fisier, similar cu functia printf obisnuita, care scrie implicit in stdout.

Exemplu:

```
#include <stdio.h>

main()
{
  FILE *fp;

  fp = fopen("test.txt", "w+");
  fprintf(fp, "test fprintf...\n");
  fputs("test fputs...\n", fp);
  fclose(fp);
}
```

Citirea din fisier:

int fgetc(FILE * fp);

Returneaza fie caracterul citit, fie EOF in caz de eroare.

char *fgets(char *buf, int n, FILE *fp);

- Functia citeste pana la n-1 caractere din fisierul dat ca argument si copiaza sirul de caractere in argumentul buf, punand la final si caracterul de terminare a sirului.

- Daca functia intalneste un newline sau EOF inainte sa citeasca numarul de caractere specificat atunci aceasta va returna numarul de caractere citit pana in acel moment, inclusiv newline-ul.

int fscanf(FILE *fp, const char *format, ...)

Se poate folosi si fscanf, dar aceasta se va opri la primul spatiu intalnit.

Exemplu:

```
#include <stdio.h>

main() {

   FILE *fp;
   char buff[255];

   fp = fopen("test.txt", "r");
   fscanf(fp, "%s", buff);
   printf("1 : %s\n", buff );

   fgets(buff, 255, (FILE*)fp);
   printf("2: %s\n", buff );

   fgets(buff, 255, (FILE*)fp);
   printf("3: %s\n", buff );
   fclose(fp);

}
```

Scriere si citire de fisiere binare:

size_t fread(void *ptr, size_t size_of_elements, size_t number_of_elements, FILE *a_file);

size_t fwrite(const void *ptr, size_t size_of_elements, size_t number_of_elements, FILE *a_file);

Functii ce scriu blocuri de memorie (spre deosebire de valori ASCII vizibile la deschiderea fisierului,daca incercam sa deschidem un fisier binar cu un editor de text, acesta va fi interpretat ca ASCII si vom avea caractere fara sens).

Scrierea si citirea trebuie sa fie simetrice (daca scriem 3 intregi, un float si 20 de caractere, trebuie sa citim in exact aceeasi ordine).

Exercitii:

1. Se citește de pe prima linie a unui fișier o literă. Se citește apoi din fișier un text. Să se scrie textul eliminând din toate cuvintele litera citită de pe prima linie:

a) într-un fișier nou;

b) la sfârșitul fișierului din care se citește.

2. Se citesc de pe prima linie a unui fișier două litere separate prin spațiu. Se citește pe urmatoarea linie un text. Să se rescrie textul inlocuind din toate cuvintele litera citită de pe prima linie cu cea de-a doua litera. Se va suprascrie fisierul din care s-a citit.

3. Se citește dintr-un fișier un text. Să se înlocuiască fiecare cuvânt cu unul care are inversate majusculele cu minusculele și viceversa(de ex, cuvântul „infORmaTică” va fi înlocuit cu „INForMAtTICĂ”). Să se realizeze operaţia folosind un singur fișier.

4. Scrieți un program care afișează cele mai frecvente litere care apar într-un fișier. Programul nu va face distincție între literele mari și mici. Dacă există mai multe litere care apar de cel mai multe ori în fișier, programul le va afișa pe toate. Dacă nu există nici o literă în fișier, programul va afișa un mesaj corespunzător.

5. Se citește dintr-un fișier un text și de la tastatură se introduce un cuvânt. Să se scrie într-un fișier toate cuvintele care conţin cuvântul citit.