

Algorithmique et structures de données

Chapitre 2 – Entrées et Sorties

Samar MOUCHAWRAB, PhD Eng

2A Cycle Préparatoire – Semestre 1
2017/2018

Les opérations d'entrée et de sortie

Un algorithme a besoin de données en entrée, et fournit un résultat en sortie. Lorsqu'on utilise un ordinateur, le clavier permet de saisir les données et l'écran d'afficher un résultat ou des textes qui donnent des directives sur les données à fournir.

Lorsqu'on voudra afficher un texte sur l'écran, on utilisera une fonction comme *printf* qui est présente dans `<stdio.h>`. Cette fonction affiche à l'écran les arguments qu'on lui demande d'afficher. Nous utiliserons également une autre fonction *scanf* pour la lecture des données saisies au clavier dans des variables.

Les opérations d'entrée et de sortie

Que fait le programme suivant ?

```
#include <stdio.h>

void main() {
    int a, b;
    printf("Entrer deux entiers : ");
    scanf("%d", &a);
    scanf("%d", &b);
    a = a + b;
    b = a - b;
    a = a - b;
    printf("a = %d\n", a);
    printf("b = %d\n", b);
}
```

Les opérations d'entrée et de sortie

Pour comprendre le programme, essayons de l'exécuter:

opération	valeur de a	valeur de b
Entrer (a)	17	
Entrer (b)		28
$a = a + b$	45	
$b = a - b$		17
$a = a - b$	28	
Afficher (a, b)	28	17

Cet algorithme échange le contenu des variables a et b

Attention, l'algorithme pose un problème : si les valeurs de a et b sont « trop grandes », la première affectation peut entraîner un débordement...

Les opérations d'entrée et de sortie

Une opération d'entrée/sortie consiste aussi à transférer des données d'un périphérique vers la mémoire (entrée) ou de la mémoire vers un périphérique (sortie). Généralement, le transfert se fait entre la mémoire et un fichier situé sur le disque dur de la machine.

Pour écrire ou lire dans un fichier, il faut auparavant l'ouvrir. Après ouverture du fichier, on dispose d'un flot. Ce flot permet de faire transiter du texte, ou des données binaires. Les fonctions d'entrées/sorties sont déclarées dans le fichier d'en-tête `stdio.h` (standard input output).

Les opérations d'entrée et de sortie

```
#include <stdio.h>

int main () {

// Déclaration d'un flot

FILE * f;

// Ouverture du fichier toto.txt en lecture et en mode texte

f = fopen("toto.txt", "r");

// Opérations d'entrée sur le flot f

...

// Fermeture du fichier

fclose(f);

//...

}
```

Les opérations d'entrée et de sortie

Le deuxième paramètre de *fopen* précise le mode dans lequel le fichier est ouvert. On distingue :

- r : ouverture en lecture, une erreur se produit si le fichier n'existe pas ;
- w : ouverture en écriture, le fichier est écrasé s'il existe, et créé sinon ;
- a : ouverture en écriture, l'écriture se fait à la suite des données du fichier s'il existe, et le fichier est créé sinon.

Plusieurs modes peuvent être précisés (r+ : lecture et écriture, w+ : créer fichier vide pour lecture et écriture, a+ ouvrir fichier pour lecture et écriture en fin de fichier).

En ajoutant b au mode, on obtient un flot binaire plutôt que texte. La lecture et l'écriture dans un fichier binaire se fait à l'aide de structures (struct).

Les opérations d'entrée et de sortie

Si l'ouverture du fichier a échoué (par exemple pour une ouverture en lecture si le fichier n'existe pas), la fonction *fopen* renvoie la valeur **NULL**. Il convient donc de tester le flot avant de continuer pour vérifier que tout s'est correctement déroulé.

```
f = fopen("toto.txt", "r");  
if (f == NULL) {  
    fprintf(stderr, "L'ouverture du fichier a échoué\n");  
} else {  
    //...  
    fclose(f);  
}
```


Les opérations d'entrée et de sortie

Les deux fonctions de lecture et d'écriture formatées en mode texte sont *fprintf* et *fscanf*. Elles s'utilisent comme les fonctions *printf* et *scanf*

```
#include <stdio.h>

int main () {

    // Déclaration d'un flot

    FILE * f;

    float a = 14.67, b = 3.14159;

    f = fopen("toto.txt", "w");

    fprintf(f, "%f + %f = %f\n", a, b, a+b);

    fclose(f);

}
```

Les opérations d'entrée et de sortie

Exemple : Lecture de valeurs d'un fichier data jusqu'à la fin du fichier (caractère End Of File EOF) et écrire la somme dans fichier out

```
#include <stdio.h>
void main () {
    FILE * in, * out;
    int sum = 0, val = 0;
    in = fopen("data.txt", "r");
    out = fopen("result.txt", "w");
    if ((in == NULL) || (out == NULL)) {
        fprintf(stderr, "L'ouverture du fichier a
        echoue\n");
    } else {
        while (fscanf(in, "%d", &val) != EOF)
            sum += val;
        fprintf(out, "%d", sum);
    }
    if (in!=NULL) fclose(in);
    if (out!=NULL) fclose(out);
}
```

Les opérations d'entrée et de sortie

Les fonctions *fgetc*, *fputc*, *fgets*, *fputs* permettent respectivement de lire et d'écrire un caractère, ou de lire et d'écrire une chaîne de caractères.

Le prototype de *fgetc* est *int fgetc(FILE* flot)*; où *flot* est le flot de type *FILE** retourné par la fonction *fopen*. Comme pour la fonction *getchar*, il est conseillé de déclarer de type *int* la variable destinée à recevoir la valeur de retour de *fgetc* pour pouvoir détecter correctement la fin de fichier.

La fonction *fputc* écrit un caractère dans le flot de données :

*int fputc(int caractere, FILE *flot)* . Elle retourne l'entier correspondant au caractère écrit (ou la constante *EOF* en cas d'erreur).

Exercice 1

Ecrire un programme qui ouvre un fichier et y enregistre des données entrées au clavier (nom de personne, son âge et son salaire).

```
#include <stdio.h>
void main() {
    FILE *fptr;
    char name[20];
    int age;
    float salary;
    /* ouverture de fichier pour écriture */
    fptr = fopen("emp.txt", "w");
    if (fptr == NULL) {
        printf("Fichier n'existe pas.\n");
        return;
    }
    printf("Entrer le nom \n"); scanf("%s", name);
    fprintf(fptr, "Nom      = %s\n", name);
    printf("Entrer l'age\n");   scanf("%d", &age);
    fprintf(fptr, "Age       = %d\n", age);
    printf("Entrer le salaire\n"); scanf("%f", &salary);
    fprintf(fptr, "Salaire  = %.2f\n", salary);
    fclose(fptr);
}
```

Exercice 2

Écrire un algorithme qui ouvre un fichier après avoir demandé le nom du fichier, lit tous les caractères qui y sont et les affiche sur l'écran