

Informatique1 CM08 :

Le C : Entrées/Sorties avec fichiers

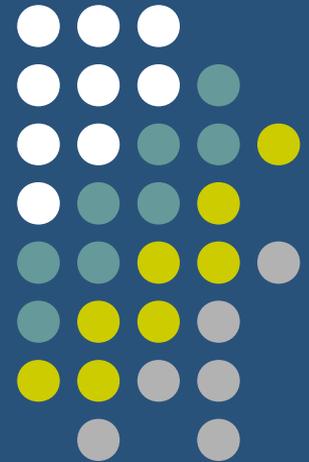
Éric CABRET

email = info@dv2000.net

Institut Supérieur de l'Automobile et des Transports

Nevers – Université de Bourgogne

1ère Année



5.10.0 - Le C : Entrées/Sorties : introduction

Rappel : fichier d'entête important : `stdio.h`



- Le chapitre qui suit explique comment utiliser des fonctions permettant de communiquer avec des fichiers.
- Pour que ces fonctions puissent être utilisées dans un fichier source quelconque, il est obligatoire de déclarer le fichier d'entête standard **`stdio.h`** par l'instruction suivante à placer en début de fichier source :
 - `#include <stdio.h>`



5.12.1 - Le C : Entrées/Sorties avec fichiers

Fonctions de base : ouverture, fermeture de fichier

- Ouverture d'un fichier :

FILE* fopen(char *Nom_du_fichier, char *Mode)

- **Nom_du_fichier** et **Mode** sont des chaînes de caractères
- Renvoie un pointeur sur le descripteur du fichier (de type **FILE***) ou **NULL** si échec
- **Mode** : "r" ouvre un fichier texte existant en entrée
 - "w" créé un nouveau fichier texte en sortie
 - "a" créé un nouveau fichier texte s'il n'existe pas déjà ou ouvre un fichier texte existant et se place à la fin
 - "rb", "wb", "ab" idem mais le fichier est **binaire**

- Exemple :

```
FILE* f;  
f = fopen("mon_fichier.txt", "r");  
if (f == NULL)  
{  
    printf("Problème à l'ouverture de f\n");  
    exit(2);  
}
```

- Fermeture d'un fichier **f** : **int fclose(FILE* f)**

- Renvoie **EOF** en cas d'échec (rarement testé)

- Exemple :

```
fclose(f)
```



5.12.2 - Le C : Entrées/Sorties avec fichiers

Fonctions de base : écriture, lecture formatées

- Accès séquentiel formaté : Écriture
 - `int putc(int car, FILE* f) ou int fputc(...)`
 - Écrit un caractère `car` dans le fichier `f`
 - Renvoie le caractère ou EOF en cas d'échec
 - `fprintf(FILE*, contrôle, E1, E2, ..., En)`
 - Écriture formatée dans le fichier (idem à `printf()`)
- Accès séquentiel formaté : Lecture
 - `int getc(FILE* f) ou int fgetc(...)`
 - Renvoie un caractère lu dans le fichier `f` ou EOF en cas d'échec
 - `fscanf(FILE* f, contrôle, av1, av2, ..., avn)`
 - Lecture formatée dans le fichier `f` (idem à `scanf()`)
 - TRES IMPORTANT : le `\n` en fin de ligne d'un fichier `f` ne sera lu/purgé qu'à l'aide d'un appel à `getc(f)`

5.12.3 - Le C : Entrées/Sorties avec fichiers

Fonctions de base : fin de fichier



- `int feof(FILE* f)`
 - Permet de détecter une fin de fichier `f`
 - Renvoie une valeur non nulle si la fin de fichier est atteinte, 0 sinon
 - **TRES IMPORTANT** : `feof()` bascule à une valeur positive si et seulement si une fin de fichier a interrompu prématurément une lecture. En particulier, la lecture des derniers octets/caractères d'un fichier ne permet pas d'en détecter la fin : il faut tenter de lire au-delà, c'est à dire au moins un octet/caractère inexistant de plus.
- Exemple très important d'utilisation de `feof()` :

```
...
while (1) {
    fscanf(fentree, ...);          /* Tenter de lire le fichier */
    if (feof(fentree) != 0) break; /* Quitter si fin de fichier */
    printf(...);                 /* Afficher les données lues dans le fichier */
}
...
```

5.12.4 - Le C : Entrées/Sorties avec fichiers

Fichiers standards : stdin et stdout : explication



- Au clavier et à l'écran correspondent des fichiers standards : **stdin** et **stdout**. Ils sont de type **FILE*** . Ces fichiers sont ouverts automatiquement au début de l'exécution d'un programme
- Dans la commande d'exécution, l'opérateur **<** redirige l'entrée standard vers un fichier
 - Exemple : **Prog < Entree**
 - Lance l'exécution de **Prog** qui lit toutes ses données **scanf()**, **getchar()** dans le fichier **Entree**
- L'opérateur **>** redirige la sortie standard
 - Exemple : **Prog > Sortie**
 - **Prog** affiche tous ses résultats **printf()**, **putchar()** dans le fichier **Sortie**
- Exemple : **Prog < Entree > Sortie** redirige les deux

5.12.5 - Le C : Entrées/Sorties avec fichiers

Fichiers standards : `stdin` et `stdout` : équivalences



- `printf(...)` est équivalent à `fprintf(stdout,...)`
- `putchar(c)` est équivalent à `putc(c, stdout)`

- `scanf(...)` est équivalent à `fscanf(stdin,...)`
- `getchar()` est équivalent à `getc(stdin)`

5.12.6 - Le C : Entrées/Sorties avec fichiers

Exemple 1/6



- **Exemple** : ce programme permet :
 - soit de lire le fichier texte "cercles.txt" et d'en afficher le contenu,
 - soit de saisir au clavier un nouveau cercle et de l'ajouter à la fin du fichier texte.

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

#define NOM_FICHIER "cercles.txt"

struct Cercle_s {
    double x, y, r;
};

typedef struct Cercle_s Cercle;
```

5.12.6 - Le C : Entrées/Sorties avec fichiers

Exemple 2/6



- La fonction **AfficherCercle()** permet d'afficher un cercle dans un fichier **f**.
- Lorsque **f** est égal à **stdout**, l'affichage se produit à l'écran.

```
void AfficherCercle(FILE *f, Cercle C)
{
    fprintf(f, "Cercle[("%.21f,%.21f),%.21f]\n", C.x, C.y, C.r);
}
```

5.12.6 - Le C : Entrées/Sorties avec fichiers

Exemple 3/6



- La fonction **SaisirCercle()** permet de saisir un cercle à partir du clavier (quand **f** vaut **stdin**) ou de le lire dans un fichier. A la différence de la lecture dans le fichier, la saisie au clavier est accompagnée d'un dialogue avec l'utilisateur.

```
Cercle  Cglobal;          /* La variable Cglobal est globale */

Cercle  SaisirCercle(FILE* f)
{
    if (f == stdin) {
        printf("Entrez le centre du cercle (x,y) : ");
        fscanf(f, "(%lf,%lf)", &Cglobal.x, &Cglobal.y);
        getc(f); /* Elimine un éventuel \n contenu dans f */
        printf("Entrez le rayon du cercle : ");
        fscanf(f, "%lf", &Cglobal.r);
        getc(f); /* Elimine un éventuel \n contenu dans f */
    } else {
        fscanf(f, "Cercle[(%lf,%lf),%lf]", &Cglobal.x, &Cglobal.y, &Cglobal.r);
        getc(f); /* Elimine un éventuel \n contenu dans f */
    }

    return Cglobal;
}
```

5.12.6 - Le C : Entrées/Sorties avec fichiers

Exemple 4/6



- La fonction **AfficherFichierCercles()** ouvre le fichier en mode lecture. Si l'ouverture s'est bien passée, les cercles sont lus dans le fichier **f** puis affichés à l'écran (**stdout**) et ceci séquentiellement jusqu'à la rencontre de la fin de fichier.

```
void AfficherFichierCercles(void)
{
    FILE *f;
    Cercle C;

    f = fopen(NOM_FICHIER,"r");
    if (f == NULL)
        printf("Le fichier %s n'a pu être ouvert en lecture\n", NOM_FICHIER);
    else
    {
        while (1) {
            C = SaisirCercle(f);
            if (feof(f) != 0) break; /* Quitter car fin de fichier */
            AfficherCercle(stdout, C);
        }
        fclose(f);
    }
}
```

5.12.6 - Le C : Entrées/Sorties avec fichiers

Exemple 5/6



- La fonction **MiseAJourFichierCercles()** ouvre le fichier en mode de mise à jour ("a"). Si l'ouverture s'est bien passée, un cercle est saisi au clavier **stdin** puis est écrit dans le fichier **f**.

```
void MiseAJourFichierCercles(void)
{
    FILE *f;

    f = fopen(NOM_FICHIER, "a");
    if (f == NULL)
        printf("Le fichier %s n'a pu être ouvert en mise à jour\n", NOM_FICHIER);
    else
    {
        AfficherCercle(f, SaisirCercle(stdin));
        fclose(f);
    }
}
```

5.12.6 - Le C : Entrées/Sorties avec fichiers

Exemple 6/6



```
void AfficherMenu(void)
{
    printf("Menu (taper le chiffre du choix puis la touche ENTREE)\n");
    printf("  1 : MiseAJourFichierCercles()\n");
    printf("  2 : AfficherFichierCercles()\n");
    printf("  0 : Quitter\n");
}

main()
{
    char c, cv;

    AfficherMenu();
    scanf("%c",&c); /* Lire puis purger les caractères de fin de saisie */
    do {
        cv = getchar(); /* Purger les caractères de fin de saisie */
    } while ((cv != '\n') && (cv != EOF));
    while (c != '0') {
        switch (c) {
            case '1': MiseAJourFichierCercles(); break;
            case '2': AfficherFichierCercles(); break;
            default: printf("Ne repondez pas n'importe quoi !\n");
        }
        AfficherMenu();
        scanf("%c",&c); /* Lire puis purger caractères de fin de saisie */
        do {
            cv = getchar(); /* Purger les caractères de fin de saisie */
        } while ((cv != '\n') && (cv != EOF));
    }
}
```