

# Langage C avancé : Séance 5

## Lectures/écritures de fichiers

---

Marc TAJCHMAN

@-mail : [marc.tajchman@cea.fr](mailto:marc.tajchman@cea.fr)

CEA - DES/ISAS/DM2S/STMF/LDEI

Lecture de fichiers

## Fichiers

Un fichier est un ensemble de données stockées dans un système informatique, avec un nom (chemin d'accès) permettant de le distinguer des autres fichiers. Dans le cadre de ce cours, un fichier servira à initialiser les données d'un code et/ou à enregistrer les résultats de ce code.

On peut avoir plusieurs types de fichiers:

- Des fichiers contenant du texte, lisible par la machine/l'être humain
- Des fichiers contenant du binaire, lisible seulement par la machine
- Des fichiers placés sur disque dur/externe, qui continuent à exister quand on éteint la machine et peuvent être relus à volonté
- Des fichiers placés en mémoire qui existent le temps que le code s'exécute mais disparaissent quand le code se termine.

## Fichiers (2)

Dans ce cours, on s'intéressera uniquement aux fichiers texte (les fichiers binaires sont intéressants parce que plus rapides mais plus délicats à manipuler)

Les opérations qu'on peut effectuer sur un fichier, entre autres:

- Création/destruction d'un fichier
- Ouverture/fermeture d'un fichier
- Positionnement dans un fichier
- Ecriture à partir d'une position dans un fichier
- Lecture à partir d'une position dans un fichier

## Utilisation des commandes sur fichier

Les commandes pour l'utilisation des fichiers sont disponibles seulement si on inclut le fichier `stdio.h`

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

## Ouverture d'un fichier sur disque

La commande est

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

Où

- filename est une (variable de type) chaîne de caractères
- mode est une chaîne de caractères qui indique si le fichier est créé à l'ouverture ou si le fichier doit exister, si c'est un fichier texte ou binaire, à quel endroit on se positionne dans le fichier
- le résultat de fopen est un pointeur dans le fichier ou NULL si quelque chose s'est mal passé

## Ouverture d'un fichier sur disque (2)

Le paramètre mode peut prendre plusieurs valeurs, les plus courantes sont:

- « r » : (mode read) le fichier est de type texte, il doit exister avant de l'ouvrir, le pointeur FILE \* désigne le début de fichier
- « w » : (mode write) le fichier est détruit s'il existe et un nouveau fichier vide est créé, le pointeur FILE \* désigne le début du fichier
- « a » : (mode append) le fichier est créé s'il n'existe pas, le pointeur FILE \* désigne la fin du fichier

Remarque: pour des fichiers binaires, on doit ajouter le caractère « b » à la chaîne de caractères mode

## Fermeture d'un fichier sur disque

La commande est

```
fclose(FILE * f);
```

Où

- `f` est le pointeur résultat d'une ouverture par `fopen`
- Après `fclose`, on ne peut plus utiliser le pointeur `f`, pour relire dans un fichier si on l'a fermé, il faut appeler à nouveau `fopen` avec le même nom en premier paramètre.
- Quand on a fini d'utiliser un fichier, il faut le fermer (sinon on risque que certaines informations soient perdues)



## Écriture dans un fichier sur disque

La commande est

```
fprintf(FILE * f, const char *s, valeur1, valeur2, ...);
```

- écrit les valeurs valeur1, valeur2 dans le fichier à l'endroit du pointeur (FILE \*) f
- le pointer f doit être valide, résultat de fopen en mode « w » ou « a »
- la chaîne s (décrit les types des valeurs à écrire et le format d'écriture), même contenu que pour l'affichage à l'écran par printf
- à la fin de la commande fprintf, le pointeur f est déplacé après ce qui est écrit par fprintf

## Lecture depuis un fichier sur disque

La commande est

```
fscanf(FILE * f, const char *s, &v1, &v2, ...);
```

- lit les variables v1, v2 depuis le fichier à l'endroit du pointeur (FILE \*) f (n'oubliez pas de passer des pointeurs sur les variables que vous voulez lire dans le fichier)
- le pointer f doit être valide, résultat de fopen en mode « w » ou « a »
- la chaîne s (décrit les types des valeurs à lire et le format de lecture), même contenu que pour la lecture au clavier par scanf

## Lecture d'une ligne depuis un fichier disque

Souvent un fichier texte est constitué de plusieurs lignes de texte séparée par le caractère `\n` (sur linux) ou `\r\n` (sur windows)

Pour lire une ligne depuis un fichier, on utilise la commande:

```
char *fgets (char *string, int n, FILE *f);
```

- Le 3<sup>ème</sup> paramètre est le pointeur renvoyé par `fopen` (en mode « `r` »)
- Le 2<sup>ème</sup> paramètre « `n` » est le nombre maximal de caractères à lire
- Le 1<sup>er</sup> paramètre « `string` » est une **chaîne de caractères de taille `n+1`** dans laquelle `fgets` place les caractères lus
- `fgets` met un caractère `'\0'` (ce qui permet de connaître le nombre de caractères réellement lus)

## Lecture d'une ligne depuis un fichier disque

- Si la ligne contient moins de `n` caractères avant la fin de ligne (`'\n'`), `fgets` s'arrête avant de lire toute la ligne
- Si la ligne contient plus de `n` caractères avant la fin de ligne (`'\n'`), `fgets` s'arrête au  $n^{\text{ème}}$  caractère lu.

En pratique, on indique comme paramètre « `n` » une valeur suffisamment grande pour pouvoir lire toutes les lignes du fichier.

## Positionnement dans un fichier

Les commandes suivantes permettent de se placer à une position donnée dans un fichier:

- Connaître la position actuelle d'un pointeur de fichier  
(ftell, fgetpos)
- Déplacer le pointeur vers une nouvelle position  
(fseek, fsetpos)
- Déplacer le pointeur en début de fichier  
(rewind)
- Déplacer le pointeur en fin de fichier

## Fichiers en mémoire

Ce qu'on appelle fichiers en mémoire, c'est une chaîne de caractères qui représente le contenu du fichier en mémoire.

Ce type de fichier peut servir à composer une chaîne de caractères et à extraire des informations (voir exemples)