

# Compléments : fichiers et formats de données

# Notions complémentaires : fichiers

- ▶ Manipulés par leur nom dans l'interpréteur de commandes
- ▶ Accès en C via un descripteur : structure contenant les informations sur l'accès en cours
- ▶ Utilisation :
  - ▶ Ouverture → obtention d'un nouveau descripteur  
`FILE *fopen(char *nom, char *mode);`
  - ▶ Accès / Test  
`int fscanf(FILE *f, char *format, ...);`  
`int fprintf(FILE *f, char *format, ...);`  
`int feof(FILE *f); /* après fscanf/getc */`
  - ▶ Fermeture → libération des ressources  
`int fclose(FILE *f);`

## Exemple : lecture et affichage d'un fichier

```
#include <stdio.h>

int main(int argc, char *argv[]) {
    FILE *fichier; char c;

    if (argc > 1) {
        fichier = fopen(argv[1], "r");
        if (fichier != NULL) {
            fscanf(fichier, "%c", &c);
            while (!feof(fichier)) {
                printf("%c", c);
                fscanf(fichier, "%c", &c);
            }
        }
    }

    return 0;
}
```

# Infos utiles sur les fichiers

- ▶ Mode : "r" ou "w", mais d'autres existent
- ▶ En cas d'erreur d'ouverture, utiliser `perror`  
`void perror(char *message);`  
pour afficher la nature de l'erreur
- ▶ Descripteurs ouverts par défaut :
  - ▶ *stdin* : entrée standard (clavier)
  - ▶ *stdout* : sortie standard (écran)
  - ▶ *stderr* : sortie d'erreur standard (écran)

## Exemple : ecriture dans un fichier

```
#include <stdio.h>

int main() {
    FILE *fichier; char c;

    fichier = fopen("toto.txt", "w");
    if (fichier != NULL) {
        scanf("%c", &c);
        while (!feof(stdin)) {
            fprintf(fichier, "%c", c);
            scanf("%c", &c);
        }
    } else {
        perror("Erreur à l'ouverture");
    }
    return 0;
}
```

# Formats et caractères accentués

Nombreux formats de représentation des caractères spécifiques à chaque langue (accents et cédilles en français). Exemple : codage du caractère ê dans quatre formats répandus :

- ▶ iso-latin1 (norme de codage sur 8 bits) : 234.
- ▶ utf-8 (norme actuelle de codage sur 1 ou plusieurs octets) : 0xc3 0xaa
- ▶ HTML/WEB (commandes sous forme de séquence de caractères ASCII) : "&ecirc;"
- ▶ LaTeX (traitement de texte scientifique) : "ê"

Quelques commandes sont utiles pour adapter le format des documents à compresser ou observer le résultat de la compression.

# Commandes od, recode, djpeg

**od -t xn -c fichier.** (Octal dump) affiche le contenu par n octets.  
Octet par octet hexa, décimal, caractère : **-t x1** , **-t d1**, **-c**.

**recode Latin-1..UTF-8 liste\_fichiers** (latin → utf) Modifie en place les fichiers (pas de fichier résultat).

Exemples : "UTF-8", "Latin-1" ou "ISO\_8859-1", "h1" ou "h2" ou "h3" ou "h4" (versions de HTML) , -d "LaTeX".

**djpeg -colors 256 -bmp f.jpeg > f.bmp**

jpeg (compressé) → bitmap (forme non compressée).

Pour 1 octet/pixel : palette de couleurs sur 8 bits (-colors 256) ou niveaux de gris (-greyscale).