Les fichiers

Principe général

Soit `fich_disque` un fichier que nous ayons préalablement créé (par exemple avec un programme ou même un éditeur de texte). La question est comment relier ce nom externe (`fich_disque`) avec les instructions qui lisent les données.

Avant de pouvoir lire ou écrire dans un fichier, il faut `ouvrir` ce fichier avec la fonction de la bibliothèque standard `fopen`.

Fonction `fopen` prend un nom externe tel que `fich_disque`, négocie avec le système d'exploitation et retourne un pointeur qui servira à lire et écrire dans le fichier.

Ce pointeur (pointeur de fichier) pointe sur une structure contenant des informations sur le fichier (l'adresse du tampon, caractère courant dans le tampon, mode d'ouverture : lecture/écritures, fin atteinte ou non etc). La structure qui contient ces informations est définie dans `stdio.h` et s'appelle `FILE`.

Il faut donc déclarer un pointeur de fichier :

```
FILE * fp ;
FILE * fpb ;
```

`FILE` est un nom de type (comme `int`), il est défini avec `typedef`.

Les pointeurs `fp` et `fpb` serviront à `accéder` aux fichiers (lire/écrire). La `fermeture` des fichiers se fera aussi à travers eux.

Ouverture et fermeture de fichiers

L'ouverture de fichier en lecture se fait en appelant `fopen` :

```
fp = fopen(fich_disque, "r") ;
fpb = fopen(fich_bin_disque, "r+b") ;
```

Les autres modes d'ouverture sont

- `"w"` (écriture),
- `"a"` (ajout),
- `"w+b", "a+b"` (si fichier binaire).

Différents cas possibles à l'ouverture :

- fichier non existant au moment de l'ouverture → création,
- fichier existant ouvert en écriture → écrasement du contenu,
- fichier existant (sur lequel on a pas les droits d'accès nécessaires) → erreur d'ouverture, `fopen` retourne `NULL`.

La fermeture de fichier se fait en appelant `fclose` :

```
fclose(fp) ;
fclose(fpb) ;
```

Lecture et écriture dans un fichier

```
 FILE * fp ;
 fp = open(fich_disque,"a");
```

`getc` , `putc` → un caractère, EOF si fin de fichier

`fgets` , `fputs` → une ligne complète

`fscanf`, `fprint` → données formatées

`fread`, `fwrite` → zone quelconque

```
 char c;
 c = getc(fp);
 putc('a',fp);
```

```
 char * nom;
 fgets(fp,30,nom); /* 30 = longueur max */
 fputs(nom,fp);
```

```
 int nombre, somme;
    fscanf(fp,"%d",&nombre);
    fprintf(fp,"la somme vaut %d\n",somme);
```

```
 int tab[10];
/* fread(adr_debut,taille,nb_blocs,pfich); */

 fread(tab,sizeof(int),10,fp);
 fwrite(tab,sizeof(int),10,fp);
```

Entrée et sortie standard

Au démarrage d'un programme C, l'environnement du système d'exploitation ouvre 3 fichiers et fournit les pointeurs associés (déclarés dans `stdio.h`) :

- `stdin` – entrée standard (clavier),
- `stdout` – sortie standard (écran),
- `stderr` – erreur standard (écran).

Exemples

```
#define getchar() getc(stdin)
#define putchar(c) putc(c,stdout)
```

```
int nombre ;
...
printf("%d\n",nombre) ;  /* équivaut à */
fprintf(stdout,"%d\n",nombre) ;

scanf("%d",&nombre) ;  /* équivaut à */
fscanf(stdin,"%d",&nombre) ;
```

Un exemple plus grand

```
#include <stdio.h>

         /* copie le fichier fe dans fs */
void cpfich(FILE * fe, FILE * fs){
  int c ;
  while ((c=getc(fe)) != EOF)
    putc(c,fs) ;
}

        /* cat : concatène les fichiers */
main(int argc, char * argv[]){
  FILE * fp ;
  if (argc == 1)          /* pas d'args: */
    cpfich(stdin,stdout); /* copie l'entrée*/
  else                    /* standard */
    while (--argc > 0)
      if ((fp=fopen(*++argv,"r"))==NULL){
        printf("cat: can't open %s\n",*argv);
        return 1;
      } else {
          cpfich(fp,stdout);
          fclose(fp);
        }
  return 0;
}
```

Positionnement et accès direct

Le fichier est vu comme une suite d'octets. A tout moment il existe une position courante, elle vaut 0 si on est au début.

Chaque lecture/écriture modifie la position courante.

`fseek(pfich,nb_octets,mode)` déplace la position courante de `nb_octets` dans le fichier à partir du début si `mode=0`, de l'endroit où on est si `mode=1` ou de la fin si `mode=2`.

`ftell(pfich)` donne la position courante par rapport au début.

`rewind(pfich)` repositionne le fichier à son début.

`ungetc(car,pfich)` revient un caractère en arrière.

```
FILE * pf;
long k=33;
char c;

fseek(pf,k,0);
ftell(pf);
c = fgetc(fp);
ungetc(c,fp);
rewind(pf);
```

Un autre exemple

```
#include <stdio.h>
#define PAS_D_ERREUR     0
#define ERREUR_OUVERTURE 1
#define ERREUR_CREATION  2

FILE * srce, * dest;
main(){
  char tampon[512];
  int nombre;
  printf("source : ");
  gets(tampon);
  srce = fopen(tampon,"r+b");
  if (srce==NULL)
      return ERREUR_OUVERTURE;
  printf("destination : ");
  gets(tampon);
  dest = fopen(tampon,"w+b");
  if (dest==NULL)
      return ERREUR_CREATION;
  while ((nombre = fread(tampon,1,512,srce))>0)
      fwrite(tampon,1,nombre,dest);
  fclose(dest);
  fclose(srce);
  return PAS_D_ERREUR;
}
```