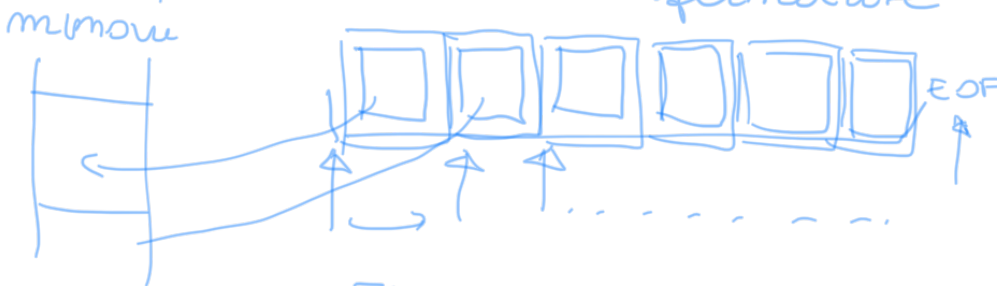


principe d'utilisation des fichiers séquentiels

Consultation

"r"
"rb"

ouverture
lectures successives
fermeture



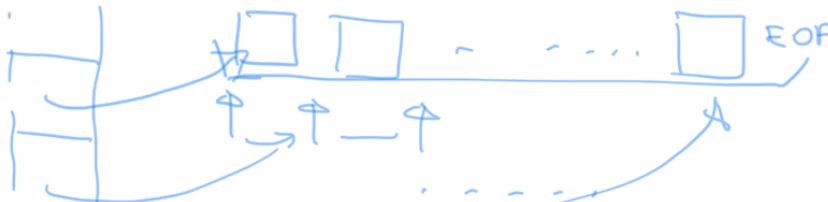
fichiers
binaires

```
FILE *fp;  
fp = fopen (".....", "rb")  
if (fp == NULL) printf ("erreur d'ouverture  
de fichier");  
else { while (!feof (fp)) {  
    fread (....., sizeof (.....), ....., fp);  
    /* traite l'element */  
}  
fclose (fp);  
}
```

Création (écriture)

"w"
"wb"

ouverture
écritures successives
fermeture



ex: fichier binaire structuré

```
FILE *fp;  
:  
if ((fp = fopen (".....", "wb")) == NULL) printf ("erreur...");  
else { while (elements à écrire) {  
    /* traite l'element */  
    fwrite (... , sizeof (...), ....., fp);  
}
```

```
} fclose(fp);
```

Acces en mode direct

Déplacement : fseek()

Permet de se déplacer à l'intérieur du fichier.

```
int fseek(FILE *fp, int déplacement, int repere)
```

fp : pointeur de fichier

déplacement : longueur en octets du déplacement

repere :

- SEEK_CUR => le déplacement est par rapport à la position courante
- SEEK_END => par rapport à la fin du fichier
- SEEK_SET => par rapport au début

du la
sète de
lecture

retour = 0 => OK pour le déplacement

« différent » 0 déplacement en dehors du fichier

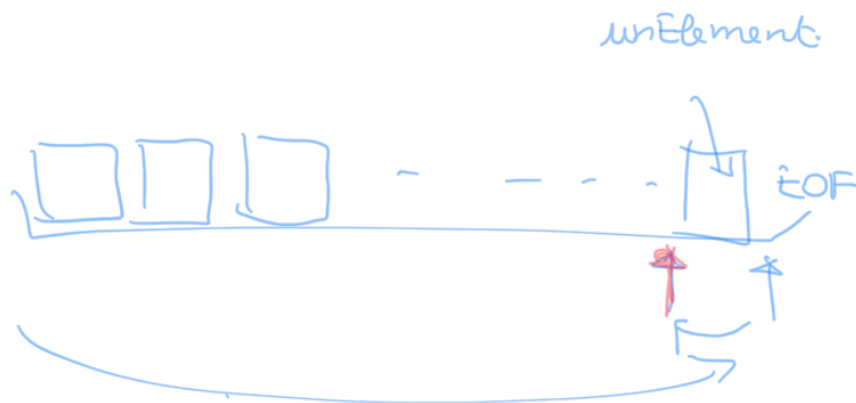
Position dans le fichier : ftell

Cette fonction renvoie la position courante (en octets) du pointeur à l'intérieur du fichier.

```
int ftell(FILE *fp)
```

aller au dernier élément d'un fichier.

```
fseek(fp, -sizeof(unElement), SEEK_END)
```



au début du fichier

```
fseek(fp, 0, SEEK_SET);
```



nbElements = ftell (fp) ←



nbElements est égale

$2 * \text{sizeof}(\text{unElement})$

pour connaître le nb d'éléments dans un fichier

fseek (fp, 0, SEEK_END)

nb = ftell (fp)

nb / sizeof (unElement).

```

//
// main.c
// ex1Fichier
//
// Created by Veronique Cherfaoui on 12/01/2017.
// Copyright © 2017 Veronique Cherfaoui. All rights reserved.
//

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

int main(int argc, const char * argv[]) {

    float x;
    FILE *fps, *fpt;
    char nomFichText[]= "/Users/vberge/Documents/Programme/L001/
ex1Fichier/nombres.txt";
    char nomFichStruct[]= "/Users/vberge/Documents/Programme/
L001/ex1Fichier/nombres.dat";
    int nb, i;

    fpt = fopen(nomFichText,"w");
    fps = fopen(nomFichStruct,"wb");
    if(fpt == NULL ||fps == NULL ){
        printf("Impossible d'ouvrir le fichier %s ou %s\n",
nomFichText, nomFichStruct);
    }
    else {
        srand(time(NULL));
        nb= rand()%100;
        for (i=0; i<nb; i++){
            x=(-2*(rand()%2)+1) * (float)rand()/(float)1000;
            printf("valeur = %f\n", x);           // affichage à
l'écran

            fwrite(&x, sizeof(float),1,fps);     // ecriture dans
le fichier structure
            fprintf(fpt, "%f\n", x);             // ecriture dans
le fichier texte, un element par ligne
        }
        fclose(fpt);
        fclose(fps);
        printf("nombre d'element = %d\n",nb);
    }
}

```

```

/
// main.c
// ex2fichier
//
// Created by Veronique Cherfaoui on 12/01/2017.
// Copyright © 2017 Veronique Cherfaoui. All rights reserved.
//

#include <stdio.h>

typedef struct {
    char nom[20];
    int age;
    float median, final, TP;
}unEtudiant;

int main(int argc, const char * argv[]) {

    unEtudiant E;
    FILE *fps, *fpt;
    char nomFichText[] = "/Users/vberge/Documents/Programme/L001/
ex2Fichier/etudiant.txt";
    char nomFichStruct[] = "/Users/vberge/Documents/Programme/
L001/ex2Fichier/etudiant.dat";
    int nb=0;

    fpt = fopen(nomFichText,"w");
    fps = fopen(nomFichStruct,"wb");
    if(fpt == NULL ||fps == NULL ){
        printf("Impossible d'ouvrir le fichier %s ou %s\n",
nomFichText, nomFichStruct);
    }
    else {
        printf ("entrer un nom d'étudiant suivi de son age et de
ses notes de TP médian et final (mettre age à 0 pour finir):
\n");
        scanf("%s %d %f %f %f", E.nom, &E.age, &E.TP, &E.median,
&E.final);
        while(E.age!=0){
            fwrite(&E, sizeof(unEtudiant),1,fps);

            fprintf(fpt, "%s %d %f %f %f\n", E.nom, E.age, E.TP,
E.median, E.final);

            printf ("entrer un nom d'étudiant suivi de son age
et des ses notes (mettre age à 0 pour finir: \n");
            scanf("%s %d %f %f %f", E.nom, &E.age, &E.TP,
&E.median, &E.final);
            nb++;
        }
    }
}

```

```

    fclose(fpt);
    fclose(fps);
    printf("nombre d'elements = %d\n",nb);
}

//
// main.c
// exfichier3
//
// Created by Véronique Cherfaoui on 04/06/2020.
// Copyright © 2020 Véronique Cherfaoui. All rights reserved.
// acces fichier structure

#include <stdio.h>

typedef struct {
    char nom[20];
    int age;
    float median, final, TP;
}unEtudiant;

int main(int argc, const char * argv[]) {

    unEtudiant E;
    FILE *fps, *fpt;

    char nomFichStruct[] = "/Users/vberge/Documents/Programme/
L001/ex2Fichier/etudiant.dat";
    int nb=0;
    float mTP=0, mmedian=0, mfinal=0;

    fps = fopen(nomFichStruct,"rb");
    if (fps == NULL ){
        printf("Impossible d'ouvrir le fichier %s
\n",nomFichStruct);
    }
    else {
        // while(!feof(fps)){ // si la fonction feof() marche
        bien ce n'est pas toujours le cas.

        while(fread(&E, sizeof(unEtudiant),1,fps) !=0) {
            mTP+=E.TP;
            mmedian+=E.median;
            mfinal+=E.final;
            nb++;
        }
        fclose(fps);
    }
    if (nb!=0) {

```

```
        printf("nombre d'elements = %d\n",nb);
        printf("les moyennes sont TP : %f, median : %f, final :
%f", mTP/(float)nb, mmedian/(float)nb, mfinal/(float)nb);
    }
    else printf("pas d'elements dans le fichier");
    return 0;
}
```