



## TD – 2

### Exercice 1

Écrire un programme C pour lire le nom et la note d'un nombre « n » d'étudiants et les stocker dans un fichier.

1. Demandez à l'utilisateur d'entrer le nombre d'étudiant.
2. Utilisez une boucle pour demander les noms et les notes des étudiants.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    char nom[50];
    int note, i, num;
    printf("Entrez le nombre des etudiants: ");
    scanf("%d", &num);
    FILE *fptr;
    fptr = (fopen("/root/ProgSys/etudiants.txt", "w"));
    if(fptr == NULL)
    {
        printf("Error!");
        exit(1);
    }
    for(i = 0; i < num; ++i)
    {
        printf("Etudiant%d\nEntrez le Nom: ", i+1);
        scanf("%s", nom);
        printf("Entrez la Note: ");
        scanf("%d", &note);
        fprintf(fptr, "\nNom: %s \nNote=%d \n", nom, note);
    }
    fclose(fptr);
    return 0;
}
```

### Exercice 2

Écrire un programme C pour lire le nom et la note d'un nombre « n » d'étudiants et les stocker dans un fichier. Si le fichier existe, ajoutez les informations au fichier.

1. Ouvrez le fichier en mode « a »
2. Demandez à l'utilisateur d'entrer le nombre d'étudiant.
3. Utilisez une boucle pour demander les noms et les notes des étudiants.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    char nom[50];
    int note, i, num;
    printf("Entrez le nombre des etudiants: ");
    scanf("%d", &num);
    FILE *fptr;
    fptr = (fopen("/root/ProgSys/etudiants.txt", "a"));
    if(fptr == NULL)
    {
```

```

        printf("Error!");
        exit(1);
    }
    for(i = 0; i < num; ++i)
    {
        printf("Etudiant%d\nNom: ", i+1);
        scanf("%s", nom);
        printf("Entrez la Note: ");
        scanf("%d", &note);
        fprintf(fptr, "\nNom: %s \nNote=%d \n", nom, note);
    }
    fclose(fptr);
    return 0;
}

```

### Exercice 3

Écrivez un programme qui ouvre un fichier existant (le fichier sera fourni en argument de l'exécutable, en cas de l'exécution sans argument le programme doit afficher un message) pour l'écriture avec l'indicateur (O\_APPEND), puis cherche le début du fichier (lseek) avant d'écrire des données (write).

Ouvrez le fichier que vous venez de modifier :

1. Où les données apparaissent-elles dans le fichier?
2. Pourquoi?

```

#include <fcntl.h>
#include <stdio.h>
#include <unistd.h>

int main(int argc, char *argv[])
{
    int fd;
    char mot = "Bonjour!"
    /* Ouvrir le fichier pour modification */
    if (argc != 2) {
        printf("Merci de fournir un fichier en argument 1\n");
        return 1;
    }
    fd = open(argv[1], O_RDWR | O_APPEND);

    /* Se déplacer vers le début du fichier */
    lseek(fd, 0, SEEK_SET);

    /* Ecrire un mot */
    write(fd, mot, sizeof(mot));
    close(fd);
    return 0;
}

```

### Exercice 4

Après chacun des appels à « write( ) » dans le code suivant, expliquez quel serait le contenu du fichier de sortie et pourquoi:

**fd1 = open(fichier, O\_RDWR | O\_CREAT | O\_TRUNC, S\_IRUSR | S\_IWUSR);**



Essaye de créer un nouveau descripteur de fichier pour le fichier "fichier" pour un accès en lecture / écriture. S'il n'existe pas, le fichier sera créé avec les permissions de lecture / écriture de l'utilisateur. Si le fichier existe déjà, le contenu est réduit.

```
fd2 = open(file, O_RDWR);
```

Créez un nouveau descripteur de fichier et ouvrez-le avec des permissions de lecture / écriture. Si le fichier n'existe pas, il ne sera pas créé.

```
write(fd1, "Bonjour,", 6);
```

À partir de la position actuelle dans le fichier, écrivez (ou écrasez) les 6 octets "Bonjou" dans le fichier. Le nombre d'octets réellement écrits sera retourné (ou -1 en cas d'échec). L'opération peut agir légèrement différemment selon les drapeaux.

```
lseek(fd2, 0, SEEK_SET);
```

Définir le début comme position du descripteur de fichier (fd2).