

<p>Semestre 1</p> <p>Système 1</p>	<p>B1</p> <p>ECE Paris</p>
--	--

Projet Système 1

Consignes :

- Connexion avec un compte utilisateur quelconque ayant les droits sudoer sur votre VM Linux
- Exécuter les commandes nécessitant le pouvoir root avec cet utilisateur sudoer en les précédant par sudo
- **Rédiger un rapport de TP en binômes ou trinôme à déposer sur Boostcamp avant le 03/12/2023** (donner vos commandes et des captures d'écran sur la commande et le résultat) et faites des commentaires si nécessaire

Objectifs :

L'objectif de ce projet est, d'une part, de se familiariser avec la programmation système en langage C et d'autre part, ce projet permettra aux étudiants d'apprendre les primitives ou les fonctions de gestion de fichiers au sein SF Linux.

Travail à effectuer :

Ce projet consiste à :

- 1) Ecrire un programme C, nommé « **init.c** », qui permet de :
 - i) créer un fichier nommé **NOMBRE** en utilisant les primitive **Open, lseek, write** et **close**.
 - ii) initialiser la valeur **100** dans un fichier nommé **NOMBRE**
- 2) Ecrire un programme C nommé « **ajout.c** » qui :
 - i) **Ouvre** le fichier **NOMBRE** créé et initialisé auparavant ;
 - ii) **Effectue** une **boucle de 10 tours** permettant de :
 - a) **lire** dans le **fichier NOMBRE** et sauvegarder la valeur lue dans un compteur,
 - b) **augmenter** de 1 à chaque tour la valeur du compteur et
 - c) **écrire** la valeur du compteur dans le fichier **NOMBRE**.
 - iii) Affiche la dernière valeur sauvegardée dans **NOMBRE**.
- 3) Ecrire un programme C nommé « **supprimer.c** » qui :
 - i) **Ouvre** le fichier **NOMBRE** créé et initialisé auparavant ;
 - ii) **Effectue** une **boucle de 10 tours** permettant de :
 - a) **lire** dans le **fichier NOMBRE** et sauvegarder la valeur lue dans un compteur,
 - b) **décrémenter** de 1 à chaque tour la valeur du compteur et
 - c) **écrire** la valeur du compteur dans le fichier **NOMBRE**.
 - iii) Affiche la dernière valeur sauvegardée dans **NOMBRE**.

Remarque: Afin de réaliser le travail demandé, vous **devez** utiliser les primitives ci-dessous

Les **primitives de gestion d'un fichier** sont décrites en détail dans ce qui suit :

a. Ouvrir un fichier :

i. Syntaxe :

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
int open(const char *pathname, int flags)
int open(const char *pathname, int flags, mode_t mode) ; flags : O_RDONLY,
O_WRONLY et O_RDWR
```

ii. **Exemple** : pour ouvrir le fichier **NOMBRE** en écriture seulement et en mode d'accès 0664, nous devons taper :

int df = open ("NOMBRE",O_WRONLY|O_CREAT,0664) ; O_CREAT : si le fichier n'existe pas, il sera créé.

b. Se positionner dans un fichier :

i. Syntaxe :

```
#include <sys/types.h>
#include <unistd.h>
off_t lseek(int fd, off_t offset, int whence) □ à whence, il faut ajouter l'offset, qui peut être positif ou négatif, pour obtenir la nouvelle position. Le premier paramètre fd est le descripteur du fichier
```

ii. Exemples :

1. Pour se positionner au début du fichier : **lseek(df_n,(off_t)0,SEEK_SET)**
2. Pour se positionner à la fin du fichier : **lseek(df_n,(off_t)0,SEEK_END)**
3. Pour se positionner à la position courante : **lseek(df_n,(off_t)0,SEEK_CUR)**

c. Lire dans un fichier :

i. Syntaxe :

```
#include <unistd.h>
ssize_t read(int fd, void * buf, size_t count)
```

ii. **Exemple** : **nbo_lus=read(df_n,&nombre,sizeof(int)) ;**

d. Ecrire dans un fichier :

i. Syntaxe :

```
#include <unistd.h>
ssize_t write(int fd, void * buf, size_t count)
```

ii. **Exemple** : **nbo_ecrits=write(df_n,&nombre,sizeof(int)) ;**

e. Fermer un fichier :

i. Syntaxe :

```
#include <unistd.h>
int close(int fd)
```

ii. Exemple : `close(df_n);`