

Nom :
Matricule :

Prénom :
Groupe :

USTHB
Faculté d'Informatique

Année 2023/2024
SYS02
L3 ACAD A, B, C

Test TP (Sujet A)
(45min)

Les téléphones doivent être éteints. Un téléphone allumé est un motif d'exclusion de l'examen.
Toute documentation est autorisée. SAUF INTERNET .

Exercice 1 **14**

Compléter le code suivant :

```
/* Fichiers d'entêtes */
#include <stdio.h> 0,25
#include <sys/stat.h> 0,25
#include <fcntl.h> 0,25
#include <stdlib.h> 0,25
#include <unistd.h> 0,25
#include <pwd.h> 0,25
#include <dirent.h> 0,25
#include <time.h> 0,25
    <Code1>

    <Code2>

int main(int argc, char *argv[]) {
    /* Variables */
    int fd1, fd2; 0,25+0,25
    int opt; 0,25
    char buf; 0,25
    <Code3>

    <Code4>

    return EXIT_SUCCESS;
}
```

<Code 1> : La fonction FILE_INFO permet l'affichage des informations relatives à un fichier à partir de son nom

```
void FILE_INFO (char *file) { 0,25
    struct stat StatusBuffer; 0,25
    if (stat(file, &StatusBuffer) == -1) { 0,25
        perror(file); 0,25
        exit(1); 0,25
        printf("Nom de fichier : %s\n", file); 0,25
    }
```

Nom :
Matricule :

Prénom :
Groupe :

```
printf("Droits d'accès (mode octal) : %o\n",StatusBuffer.st_mode & 0777); 0,25
if (S_ISREG(StatusBuffer.st_mode)) { 0,25
printf("%s est un fichier régulier\n", file);
} else if (S_ISDIR(StatusBuffer.st_mode)) {0,25
printf("%s est un répertoire\n", file);
};
printf("le numéro d'inode %d\n",StatusBuffer.st_ino); 0,25
printf("Owner:      %s\n",getpwuid(StatusBuffer.st_uid)->pw_name); 0,25
printf("La date de la dernière modification:  %s", ctime(&sb.st_mtime));} }0,25
```

<Code 2> : La fonction permet le parcours d'un répertoire à partir de son nom et affiche son contenu (noms et inodes des fichiers/répertoires)

```
void DIRECTORY_CONTENTE (char *name_dir) { 0,25
    DIR *d; 0,25
    struct dirent *dir; 0,25
    d = opendir(name_dir); 0,25
    if (d) 0,25
    {
        while ((dir = readdir(d)) != NULL) 0,25
        {
            printf("%s\n", dir->d_name); 0,25
        }
        closedir(d); 0,25
    }
}
```

<Code 3> permet de :

Ouvrir le fichier 1 en lecture seule (on suppose que le fichier 1 contient des données)

Ouvrir le fichier 2 en Ecriture en écrasant son contenu, si le fichier n'existe pas il sera créé, avec les permissions 644.

```
fd1 = open("File1", O_RDONLY ); 0,25
if (fd1 == -1) { 0,25
    perror("open"); 0,25
    exit(-1); 0,25 }
mode_t mode2 = S_IRUSR | S_IWUSR | S_IXUSR | S_IRGRP | S_IWGRP |
S_IROTH | S_IWOTH; 0,25
fd2 = open("File2", O_RDWR | O_CREAT, mode2); 0,25 +0,25
if (fd2 == -1) { 0,25
    perror("open"); 0,25
    exit(-1); }
```

Nom :
Matricule :

Prénom :
Groupe :

<Code 4> : permet l'utilisation de getopt comme suit :

L'option -a : permet l'écriture du contenu du fichier1 dans fichier 2

L'option -b : permet l'affichage des informations relatives au fichier 1

L'option -c : permet l'affichage du contenu de répertoire courant

L'option -d : permet l'affichage des informations relatives au fichier 2

```
while ((opt = getopt(argc, argv, "abcd::")) != -1) { 0,25 + 0,25
    switch (opt) { 0,25
        case 'a': 0,25
            while (read(fd1, &buf, 1) != 0) 0,25 + 0,25 {write(fd2, &buf, 1);} 0,25 + 0,25
            printf("done");
            break;
        case 'b': 0,25
            FILE_INFO("File1"); 0,25
            break;
        case 'c': 0,25
            DIRECTORY_CONTENTE("../EXAMEN");0,25
            break;
        case 'd': 0,25
            FILE_INFO("File2"); 0,25
            break;
        default: /* '?' */
            fprintf(stderr, "Usage: %s [-t nsecs] [-n] name\n",
                argv[0]);
            exit(EXIT_FAILURE);
    }
}Return EXIT_SUCCESS;
}
```

Nom :
Matricule :

Prénom :
Groupe :

Exercice 2 : Choisir la ou les bonnes réponses (une réponse fausse annule la réponse juste dans une même question): **6**

Quelle(s) commande(s) permet le formatage d'une partition ? **0,5**

- a) **fdisk**
- b) **mkfs**
- c) **dumpe2fs**

'dumpe2fs' est une commande qui permet l'obtention d'informations sur n'importe quel système de fichier? **0,5**

- a) **Vrai**
- b) **Faux**

A quoi sert le répertoire lost+found ? **0,5**

- a) Stocker les fichiers temporaires utilisés par les applications
- b) **Récupérer les fichiers orphelins ou égarés en raison d'erreurs**
- c) Conserver les fichiers supprimés jusqu'à ce qu'ils soient définitivement effacés
- d) Sauvegarder les fichiers système importants

Vous disposez d'un disque de 16 Go en tant qu'administrateur système. On vous a confié la tâche de partitionner le disque comme suit :

Quelle est la séquence correcte d'étapes pour créer une partition primaire numérotée 2 de 1 Go, formatée en Fat32 ? **1,5**

- a) **fdisk /dev/sdX, n, p, 2, Enter, +1G, t, c, w**
- b) fdisk /dev/sdX, n, 2, Enter, +1G, t, 2, c, w
- c) fdisk /dev/sdX, p, 2, Enter, +1G, t, c, w
- d) fdisk /dev/sdX, n, p, 1, Enter, +1G, t, 2, c, w

Comment créer une partition étendue numérotée 3 qui prend le reste de l'espace du disque ? **1,5**

- a) fdisk /dev/sdX, n, e, +1G, t, 3, w
- b) **fdisk /dev/sdX, n, e, 3, Enter, Enter, w**
- c) fdisk /dev/sdX, n, p, 3, Enter, t, 3, w
- d) fdisk /dev/sdX, n, e, +1G, Enter, t, 3, 5, w

Quelle séquence de commandes est correcte pour créer deux partitions logiques de tailles égales, l'une formatée en ext2 et l'autre en exFAT? **1,5**

- a) fdisk /dev/sdX, n, l, Enter, +7.5G, t, 5, 83, n, l, Enter, +7.5G, t, 6, 8, w
- b) **fdisk /dev/sdX, n, Enter, +7.5G, t, 5, 83, n, Enter, +7.5G, t, 6, 7, w**
- c) fdisk /dev/sdX, n, Enter, +7.5G, t, 5, 83, n, Enter, +7.5G, t, 6, f, w
- d) fdisk /dev/sdX, n, +7.5G, t, 5, 83, n, +7.5G, t, 6, 7, w