

INFO-F-201 – Systèmes d'exploitation 1  
Année académique 2019 – 2020  
Joël Goossens

Examen de Janvier, partie pratique

**Remarques préliminaires**

- On vous demande de répondre à chaque question sur des feuilles séparées (les correcteurs sont différents).
- N'oubliez pas d'inscrire vos nom et prénom sur chacune des feuilles.
- Vous disposez de trois heures.
- Vous ne pouvez pas utiliser de notes (à l'exception du formulaire fourni avec l'énoncé).
- Vous devez fournir au moins une feuille par question (même si vous n'y répondez pas).

**Question 1 – bash (10 points)**

Considérons un fichier texte contenant une liste de chemins absolus (à raison d'un chemin par ligne). Écrire un script qui étant donné en paramètre le chemin vers ce fichier, parcourt l'ensemble des chemins contenus dans le fichier et effectue le traitement suivant :

1. si le chemin identifie un fichier régulier, il en modifie l'extension en suivant le tableau suivant :

| origine | arrivée |
|---------|---------|
| mp3     | ogg     |
| mp4     | ogv     |
| wav     | flac    |
| jpg     | png     |

Si le fichier ne comporte pas d'extension ou si l'extension n'est pas une des extensions présentées dans la colonne de gauche du tableau, le script passe à l'entrée suivante de la liste sans entreprendre d'action.

2. si le chemin identifie un répertoire, le script parcourt l'ensemble des fichiers réguliers contenus dans le répertoire et effectue le traitement précédemment décrit. Notez que le script n'effectue pas un parcours récursif, il se contente d'effectuer le traitement sur les fichiers réguliers contenus à la racine du répertoire.

Il vous est demandé de structurer votre script en fonctions, de commenter votre code et de couvrir un maximum d'erreurs possibles par des tests, chaque erreur rencontrée devant être inscrite par le script dans le fichier de log : `/tmp/log_renamer`.

## Question 2 – C (10 points)

Écrivez un programme en C qui utilise fork pour créer un processus enfant.

Le processus enfant lira les entrées de l'utilisateur, qui fournira des nombres (un nombre à la fois), et les transmettra au processus parent.

Le processus parent fera la somme des nombres reçus de l'enfant et imprimera chaque résultat.

Lorsque le parent reçoit le premier numéro, il ne fait qu'imprimer ce numéro. Le processus parent et fils continuent jusqu'à ce qu'ils reçoivent le caractère x.

Votre programme devrait fonctionner lorsqu'il est appelé de la manière suivante (étant donné qu'il soit compilé avec l'option -o sum)

```
> ./sum.
```

Vous pouvez supposer que les entrées de l'utilisateur sont bien formées.

Voici un exemple d'exécution :

```
> ./sum
```

```
> 10
```

```
10
```

```
> 5
```

```
15
```

```
> x
```