

## Programmation Système – Exercices de mise à niveau

*Exercices de niveau L1 & L2 – Voir cours de révisions dans la zone « Pré-requis » sur Moodle*

### Exercice 1 – Utiliser le langage C (L1)

1. Écrire une fonction `premier` qui détermine si un nombre entier donné en paramètre est premier ou non. Écrire un programme qui saisit une valeur entière N, puis saisit N valeurs entières et teste pour chacune d'entre elles si cette valeur est un nombre premier ou non grâce à la fonction.
2. Écrire une fonction `permute` qui permute les valeurs de ses deux paramètres entiers. Écrire un programme qui lit deux valeurs entières au clavier, les permute grâce à la fonction `permute` puis affiche leur valeur précédée de leur nom.
3. Écrire une fonction `convertirDuree` qui permet de transformer un temps donné en secondes en un temps exprimé en heures, minutes et secondes. Utiliser cette fonction pour convertir différents temps saisis au clavier jusqu'à ce que l'utilisateur saisisse la valeur 0.
4. Écrire une fonction `trier` permettant de trier, par ordre croissant, les éléments d'un tableau passé en paramètre (on passera aussi en paramètre la taille de ce tableau) selon l'algorithme de votre choix. Écrire un programme qui lit une liste d'entiers, les stocke dans un tableau, trie ce tableau et affiche le résultat.
5. Écrire une fonction qui permet de saisir une matrice de L lignes et C colonnes (L et C pouvant varier d'une exécution à l'autre). Écrire une fonction qui permet d'afficher le contenu d'une matrice de L lignes et C colonnes. Les utiliser pour écrire un programme qui saisit la dimension d'une matrice, saisit ses éléments puis les affiche.

### Exercice 2 – Main paramétré (L2)

Écrire un programme (= une commande) qui accepte en paramètre, une liste d'entiers de longueur variable, calcule la somme de ces entiers et l'affiche à l'écran. L'exécutable se nommera `somme`.

*Exemples d'utilisation :*

`somme 3 4 5 6 10` → affichera 28

`somme` → affichera 0

`somme -3 8 -5 9 2 12` → affichera 23

### Exercice 2 – Création d'un processus Unix (L2)

#### Utilisation de `fork()` et `exit()`

Écrire un programme (= une commande) dans lequel un processus père crée un processus fils.

Le père exécute le traitement suivant :

```
for(i=1; i<= NB_MAX; i++)  
    printf("%4d Je suis le pere\n", i);
```

Le processus fils exécute le traitement suivant :

```
for(i=1; i <= NB_MAX; i++)  
    printf("\t\t\t\t%4d Je suis le fils\n", i);
```

- Exécuter plusieurs fois ce programme.
- Que constatez-vous ? Si les exécutions ne sont pas significatives, faites varier la valeur de la constante `NB_MAX`.

## Exercice 3 – Création de plusieurs processus Unix (L2)

### Utilisation de `fork()` et `exit()`

Écrire un programme dans lequel le processus principal crée N processus fils (N est le paramètre de l'application). Chaque processus s'identifie en affichant son pid, celui de son père, le numéro de son groupe de processus et le numéro effectif d'utilisateur.

- Version 1 : les N processus fils doivent s'exécuter les uns après les autres.
- Version 2 : les N processus fils doivent s'exécuter en parallèle.
- Version 3 : le processus père affiche la manière dont chaque fils termine son exécution.

## Exercice 4 – Commutation d'image et attente de la terminaison d'un processus fils (L2)

### Utilisation de `fork()`, `exec()`, `exit()` et `wait()`

Écrire un programme qui effectue la combinaison : `date`; `ps -al` en respectant la séquentialité des commandes :

- un processus fils réalisera la commande `date` ;
- le processus père effectuera l'appel à la commande `ps`.

## Exercice 5 – Redirection Unix (L2)

### Utilisation de `fork()`, `dup()` ou `dup2()` – Compilation séparée

Récupérer les fichiers `afficher.h` et `afficher.c` qui définissent une fonction affichant le contenu d'un fichier, dont le nom est donné en paramètre, à l'écran (sortie standard).

Sans modifier ces fichiers et cette fonction, l'utiliser afin d'écrire la commande `moncat` qui affiche à l'écran de l'utilisateur le contenu d'une liste de fichiers, mais qui peut prendre en compte une éventuelle redirection vers un fichier autre que l'écran.

*Exemples d'utilisation :*

`moncat fichier1 fichier2` → affiche à l'écran le contenu des fichiers `fichier1` et `fichier2`

`moncat fichier1 fichier2 fichier3 ">" fichierDestination` → envoie dans `fichierDestination` le contenu des fichiers `fichier1`, `fichier2` et `fichier3`

*Attention :* Pour tester la commande, on tapera le signe de redirection encadré par des guillemets.