Programmation Concurrente: Arguments et Fork

CPE Lyon

2026

Consignes générales:

- Vos programmes doivent être compilés avec les options: -Wall -Wextra -g
- Vos programmes ne doivent pas avoir de warnings
- Testez plusieurs fois de suite vos programmes et vérifiez que les résultats obtenus sont ceux attendus.

Arguments

Exercice 1: Miroir

Variante 1

Écrivez un programme (miroir.c) qui prend en argument une chaîne de caractères et **l'affiche** à l'envers. Exemple:

```
user@machine:~$ ./miroir trace
ecart
```

Variante 2

Modifiez le programme précédent pour qu'il traite l'ensemble des arguments que l'utilisateur lui fourni. Exemple:

```
user@machine:~$ ./miroir trace soda saper
ecart ados repas
```

Exercice 2: Moyenne

Écrivez un programme (moyenne.c) qui calcule et affiche la moyenne d'un ensemble de notes (nombres entiers) passées en arguments.

Votre programme devra vérifier qu'une note fournie est valide. C'est-à-dire qu'il s'agit bien d'un entier dont la valeur est comprise entre 0 et 20.

Exemple d'usage:

```
user@machine:~$ ./moyenne 12 18 toto
Note(s) non valide(s)
user@machine:~$ ./moyenne 12 18 3.5
Note(s) non valide(s)
user@machine:~$ ./moyenne 12 18 22
Note(s) non valide(s)
user@machine:~$ ./moyenne
Aucune note fourni, moyenne impossible à calculer
user@machine:~$ ./moyenne 12 18 3
Moyenne: 11.00
```

Aide

Conversion chaîne de caractère vers entier

```
int nb_success = sscanf(str,"%d/%d/%d",&day,&month,&year);
int nb_success = sscanf(str,"%d",&nb);
```

sscanf renvoi le nombre de conversion réussi.

- 1er cas: si nb_success est différent de 3 alors la chaine (str) fournie ne contient pas trois entiers séparés par des slashs
- 2ème cas: si nb_success est différent de 1 alors la chaine (str) fournie ne contient pas un entier

Toutes les conversions sont enregistrées dans les variables qui suivent le format. C'est pour cela que l'on passe l'adresse de ces variables. La fonction pourra modifier leur valeur, puisqu'elle connaît leur adresse.

Formatage des affichages

```
printf("%4.2f",var);
```

- % indique qu'il s'agit d'un format
- f indique qu'il s'agit d'un float
- 4 indique que la taille minimale de la chaîne affiché sera de 4 caractères
- 2 indique qu'il y aura maximum deux décimales

Fork

Exercice 3: Fork dans une boucle

Écrivez un programme (boucle.c) qui fait appel à fork dans une boucle qui fera 3 itérations. À l'intérieur de chaque itération, le programme affichera en une seule fois:

- la valeur du compteur de boucle
- le PID du processus
- le PPID (PID du processus parent)
- la valeur retournée lors de l'appel à fork.

Dessinez l'arbre des processus correspondant à l'exécution de ce programme.

⚠ Il n'y a aucune condition (if/else) à mettre que ce soit sur l'affichage des informations ou sur fork.

Exemple d'affichage:

```
0: Je suis le processus 456, mon père est le processus 123 fork a retourné 0
```

Exercice 4: Wait

Écrivez un programme (wait.c) qui:

- lit sur la ligne de commande (argc/argv) le nombre **N** de processus qu'il doit créer.
- Une fois tous les processus créés, il se met en attente de ses fils
- Dès qu'un fils se termine, il affiche l'identité de ce fils et la valeur de retour de ce fils.

Les fils quant à eux:

- affiche leur pid, le pid de leur père et se mettent en attente (sleep) pendant 2*i secondes.
- Après l'attente, ils indiquent qu'ils reprennent leur exécution avant de faire un exit avec la valeur de i.

⚠ **i** étant le numéro de l'itération qui a servi à créer le processus.

Ce programme est-il déterministe ? Pourquoi ?

Exercice 5: Analyse de code

⚠ Rappel: break sort de la boucle dans laquelle il se trouve

Cas N°1

```
1  #include <sys/types.h>
2  #include <unistd.h>
3
4  int main(void) {
5      int i, n=0;
6      pid_t fils_pid;
7      for (i=1; i< 5; i++)
8      {
9          fils_pid = fork();
10         if (fils_pid > 0)
11         {
12             n = i*2;
13             break;
14         }
15     }
16     printf("%d\n", n);
17     return 0;
18 }
```

- Dans le block d'exécution du if (ligne 12 et 13). Dans quel processus se trouve-t-on ? (père ou fils)
- Ce programme est-il déterministe ? Justifiez
- Si le programme est déterministe:
  - Indiquez exactement ce qui sera affiché lors de son exécution.
- Si le programme n'est pas déterministe:
  - Indiquez un des affichages possible
  - Faîtes en sorte de rendre ce programme déterministe
  - Indiquez ce qui sera affiché alors.
- L'appel à fork() peut-il échouer ? Pourquoi ?

Cas N°2

```
1  #include <stdio.h>
2  #include <unistd.h>
3  #include <stdlib.h>
4
5  int main(void) {
6      int i = 0;
7      for (i=0 ; i<4 ; i++)
8          if (fork())
9              break;
10     srand(getpid());
11     int delai = rand()%4;
12     sleep(delai);
13     printf("Je suis %c, j'ai dormi %d s\n"
14         ,'A'+i,delai);
15     return 0;
16 }
```

- Donnez l'arbre des processus générés par ce programme
- Ce programme est-il déterministe ?
- Sans modifier les lignes 1 à 12, modifiez ce programme pour qu'il produise un affichage dans l'ordre alphabétique inverse (EDCBA)

Cas N°3

```c
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/wait.h>

#define PROCESS 4
#define TAB_SIZE 100

int main(void) {
    int i = 0;
    int pere = 1;
    int tab[100] = {0};
    for (i = 0; i < TAB_SIZE; i++)
        tab[i] = i % 10;
    for (i=0 ; i<PROCESS ; i++)
    {
        if (fork() == 0)
        {
            pere = 0;
            break;
        }
    }
    if (pere)
    {
        for (i = 0; i < PROCESS; i++)
            wait(NULL);
        puts("Calculs terminés");
    }
    else
    {
        int somme = 0;
        for(int j=i*25; j < (i+1)*25; j++)
            somme += tab[j];
        printf("%d %d\n",i,somme);
    }
    return 0;
}
```

- Donnez l'arbre des processus
- Ce programme est-il déterministe ? Justifiez
- Si le programme est déterministe:
  - Indiquez exactement ce qui sera affiché lors de son exécution.
- Si le programme n'est pas déterministe:
  - Indiquez un des affichages possibles

Quelles conclusions pouvez-vous tirer de ces trois cas ?

Exercice 6:

Variante 1

Écrivez un programme qui crée 4 fils. Le premier fils affiche les entiers de 1 à 50, le seconde de 51 à 100, le troisième de 101 à 150 et le dernier de 151 à 200

Variante 2

Faites en sorte que l'affichage soit garanti d'être systématiquement dans l'ordre numérique croissant 1,2,3,... , 200.