

# Systèmes d'exploitation centralisés

1<sup>re</sup> année Informatique et Réseaux

18 juin 2021

Documents autorisés. Les exercices sont indépendants. Une réponse non justifiée est inutile.

## 1 Coroutines (5 points)

Qu'affiche le programme ci-dessous ? Comment se termine-t-il ?

---

```
#include <stdio.h>
#include "coroutines.h"

coroutine_t c1,c2,c3;

void foo() {
    int a = 7;
    printf("F0 %d\n", a);
    cor_transferer(c1,c2);
    a++;
    printf("F1 %d\n", a);
}

void codeA (void *unused) {
    printf("A 1\n");
    foo();
    printf("A 2\n");
    cor_transferer(c1,c2);
    printf("A fini\n");
}

void codeB (void *unused) {
    printf("C 1\n");
    cor_transferer(c2,c1);
    printf("C 2\n");
    cor_transferer(c2,c1);
    foo();
    printf("C fini\n");
}

int main() {
    c1 = cor_creer("C1", codeA, NULL);
    c2 = cor_creer("C2", codeB, NULL);
    c3 = cor_creer("C3", NULL, NULL);
    cor_transferer(c3,c2);
    printf ("main fini\n");
}
```

---

## 2 Processus (5 points)

1. On souhaite ajouter une opération `proc_commuter_avec` (`processus_t` qui) qui fonctionne comme `proc_commuter` mais en spécifiant explicitement le processus qui devient élu. Donner le code algorithmique de `proc_commuter_avec`.
2. Pourquoi l'opération `proc_commuter_avec` est-elle une mauvaise idée ?
3. On souhaite modifier `proc_continuer`(`processus_t` qui) pour que le processus débloqué devienne directement l'élu. Indiquer précisément les changements à faire dans le noyau.
4. Quel serait l'intérêt d'une telle évolution ?

## 3 Systèmes de fichiers (5 points)

1. La lecture effective sur disque se fait par bloc d'un nombre fixe d'octets. Cela est-il un inconvénient lorsqu'un programme lit séquentiellement un fichier octet par octet ?
2. Raid est une architecture de redondance pour les systèmes de fichiers, s'appuyant sur plusieurs disques physiques. Dans Raid 1, chaque fichier est dupliqué sur tous les disques. La lecture du fichier est faite depuis un des disques. Quels sont les intérêts et inconvénients de cette approche ?
3. Raid est une architecture de redondance pour les systèmes de fichiers, s'appuyant sur plusieurs disques physiques. Dans Raid 2, chaque fichier est écrit sur un disque, et un code correcteur d'erreur est écrit sur un autre disque. Quels sont les intérêts et inconvénients de cette approche ?
4. Il est possible d'associer plusieurs noms à une même inode via la commande `ln` ou l'appel système `link`. Pourtant, cette possibilité est interdite si l'inode correspond à un répertoire. Pour quelles raisons ?

## 4 Mémoire virtuelle (5 points)

### Question de cours

1. Une page réelle de la mémoire physique peut-elle être couplée dans la mémoire virtuelle de plusieurs processus ? Si oui, quelle en serait l'utilité ?
2. Un phénomène d'écroulement apparaît quand les processus en cours utilisent un nombre de pages virtuelles supérieur au nombre de pages physiques. Quels sont les symptômes et pourquoi parle-t-on d'écroulement ?

**Question de TP** On souhaite enrichir le fonctionnement de la mémoire virtuelle implantée en TP pour qu'un processus puisse indiquer qu'une de ses pages est accessible en lecture (mais pas écriture) par tous les autres processus. Pour cela, on ajoute une interface `void MV_share_page(void *adresse)` qui permet au processus courant d'indiquer que la page contenant l'adresse spécifiée est dorénavant lisible par tous les processus.

Préciser quelles sont les modifications à apporter à l'implantation actuelle, dans le cas où les appels à `MV_share_page` doivent nécessairement avoir lieu à l'initialisation, avant le moindre accès à la mémoire commune.