

La notation tiendra compte de la validité des réponses, mais aussi de la présentation et de la clarté de la rédaction.

Documents interdits, à l'exception d'une feuille A4 à rendre avec votre copie.

★ **Exercice 1: Lire des fork, des wait et des exec (5pts)**

Discutez les différents affichages possibles du programme ci-dessous (avec schémas). On suppose que le cas où `waitpid()` retourne pour une raison autre que la terminaison d'un processus n'arrive jamais (`WIFEXITED(status)` est toujours vrai).

Programme 1

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4
5 int main() {
6     pid_t pid, pid2;
7     int status;
8     if (pid = fork()) {
9         if (pid2 = fork()) {
10            waitpid(pid2, &status, 0);
11            if (WIFEXITED(status)) {
12                printf("A %d\n", WEXITSTATUS(status));
13            }
14        } else {
15            printf("B\n");
16            exit(42);
17        }
18    } else {
19        execl("/bin/echo", "echo", "D", NULL);
20        printf("C\n");
21    }
22 }
```

★ **Exercice 2: Écrire des fork, exec, et autres (6pts)**

Donner le code C (sans utiliser la fonction `system`) correspondant à ce que fait un *shell* lorsqu'on tape les lignes de commandes suivantes. Afin d'obtenir un code lisible et court, vous êtes dispensés des tests d'erreur des appels systèmes.

▷ **Question 1:** `prog1 ; prog2 < foo`

▷ **Question 2:** `prog1 arg1 arg2 | prog2`

★ **Exercice 3: Savoir utiliser les sémaphores et reconnaître les schémas de synchronisation classiques (6pts)**

Suite à une expérimentation malheureuse, un début d'incendie s'est déclaré au club robotique. Immédiatement, tous les étudiants présents sur place et au foyer se précipitent à la cafet pour remplir des carafes d'eau avant de les jeter sur les flammes. Un seul robinet, des couloirs exigus et des dizaines d'étudiants courant en tout sens pour porter de l'eau vers les flammes : le chaos est indescriptible.

▷ **Question 1:** Proposez une solution à base de sémaphore(s) pour d'éviter que les étudiants se bousculent devant le point d'eau. De quel schéma de synchronisation classique cela se rapproche-t-il ? Pourquoi ?

▷ **Question 2:** Les porteurs d'eau perdent un temps précieux à contourner le bar et attendre leur tour au robinet. Proposez une solution à base de sémaphore(s) où l'un d'entre eux seulement est en charge de remplir les carafes pour les autres, qui courent ensuite en jeter le contenu dans les flammes quand elles sont pleines. De quel schéma de synchronisation classique cela se rapproche-t-il ? Pourquoi ?

★ Exercice 4: Signaux (3pts)

Expliquez précisément le programme suivant.

Programme 2

```
1 #include <signal.h>
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <string.h>
5
6 void f2(int sig) {
7     printf("bar\n");
8     exit(0);
9 }
10
11 void f1(int sig) {
12     static compt = -1;
13
14     if (++compt == 3) {
15         kill(getpid(), SIGUSR1);
16     } else {
17         printf("foo\n");
18     }
19 }
20
21 int main() {
22     struct sigaction nvt, old;
23     memset(&nvt, 0, sizeof(nvt));
24     nvt.sa_handler = f1;
25     sigaction(SIGINT, &nvt, &old);
26     nvt.sa_handler = f2;
27     sigaction(SIGUSR1, &nvt, &old);
28     while(1);
29 }
```