



| | | | |
|---|---|--|---|
|  | <div>Ministère de l'enseignement supérieur et de la recherche scientifique</div> <div>****</div> <div>Institut Supérieur d'Informatique</div> | |  |
| Classe : M1 – MP2L | <div>EEL1 : Evaluation en ligne N°1</div> <div>Programmation Système et Réseaux sous Unix</div> <div>*****</div> <div>Chargé(s) du cours : Ezzeddine ZAGROUBA</div> | Durée : 1h30 | |
| Date : 01 Novembre 2024 | | Documents et calculatrices autorisés | |
| Nbre pages : 2 | | | |
| Barème : Ex1 (10) EX2 (10) | | | |

EXERCICE 1 : (10 PTS)

- Définissez les termes suivants : Instruction atomique, Section critique, Exclusion mutuelle, Synchronisation, Communication interprocessus.
- Soit le programme ci-dessous :

```
#define N 10
int main()
{
    int i = 1;
    while (fork() == 0 && i <= N) i++;
    printf("%d\n",i);
    exit(0);
}
```

 - Donnez la ligne de commande permettant de générer un exécutable de ce programme ?
 - Expliquer le résultat d'exécution de ce programme en l'exécutant à la main et dites ce qu'il réalise ?
- Donner la différence entre `fork()`, `system()` et `exec()` puis décrivez comment et quand utiliser `exec()` dans un programme.
- Complétez/modifiez le programme ci-dessous pour qu'il lance une calculatrice (`kcalc`) et un éditeur de texte (`gedit`) en utilisant l'appel système « `execvp` ».

```
#include <stdio.h>
#include <unistd.h>
int main()
{
    if (! fork()) { printf("Fils 1 : je lance une calculatrice !\n"); }
    if (! fork()) { printf("Fils 2 : je lance un éditeur de texte !\n"); }
    return 0;
}
```
- Expliquez le fonctionnement d'un tube (pipe) unidirectionnel et donnez un exemple d'utilisation en C ?
- Expliquez le rôle de la fonction `wait()` et comment elle diffère de `waitpid()` ?
- Qu'est-ce qu'un signal dans un système Unix ? Décrivez comment un processus peut gérer un signal spécifique et donner un exemple de programme simple ?

EXERCICE 2 : (10 POINTS)

Ecrire un programme qui réalise les tâches suivantes :

- Affiche son identifiant (`pid`) ;
- Traite tous les signaux en affichant le numéro du signal reçu.
- Traite le signal `SIGUSR1` par une fonction `Handler1` et le signal `SIGUSR2` par `Handler2` sachant que :

- Handler1 affiche le numéro du signal reçu et la liste des utilisateurs de la machine (appel à who par system("who"))
- Handler2 affiche le numéro du signal reçu et l'espace disque utilisé sur la machine (appel à df . par system("df ."))

./ Fin ./