

NOM : _____

PRENOM : _____

Contrôle Continu n°3 Systèmes HLIN 504

- Comment dans un processus peut-on déclencher de manière asynchrone une fonction temporisée par un certain délai alors que l'on est en train d'effectuer des instructions ? Donner un exemple !

alarm(5); permet de demander un timer qui envoie le signal SIGALRM au bout de 5s.

```
int main() {
    struct sigaction action;
    action.sa_handler = gst;
    sigaction(SIGALRM, &action, NULL);
    alarm(5); char s[1023];
    sleep(5);
    munf("FIN DU PROG !\n");
    return 0;
}
```

- Comment créer un fichier de type d, l, - ? Vous donnerez une commande permettant de créer un fichier de chaque type en expliquant en une phrase à quoi servent ces fichiers.

où d : mkdir mondir repertoire, dossier
l : ln -s ancien nouveau lien symbolique
- : touch monfic.txt fichier régulier

NOM : _____

PRENOM : _____

3. Qu'est-ce qu'un tube simple (non nommé) (en 1 phrase) ? Peut-on faire communiquer 2 utilisateurs à travers un tube simple, si oui à quelle condition ? Des informations structurées (flottant, entier, chaîne de longueur variable) peuvent-ils transiter dans un tube ? A quelle condition ?

file d'attente de char synchronisée.

NON

On peut écrire des données structurées à condition qu'elles soient toutes à la longueur, longueur connue par les(s) lecteur(s).
Ex cube d'Eratostène - transmission d'entiers (Sockets)

L'écriture est atonique en déçà d'un certain seuil.

4. Un processus écrivain doit écrire une chaîne de caractères passée comme argument à la ligne de commande dans un tube simple ou un processus lecteur enfant devra les lire et n'afficher sur la sortie standard que les caractères numériques [0-9]. Programmer cet exemple en C.

int main (int argc, char *argv[2]) {
 if (argc != 2) {
 fprintf (stderr, "longueur\n");
 exit(1);
 }
 int tube[2]; int n = pipe(tube);
 if (n == -1) {
 fprintf (stderr, "Innommable tube\n");
 exit(2);
 }
 int f = fork();
 switch (f) {
 case -1: fprintf (stderr, "Impossible fork\n"); exit(3);
 case 0: close(tube[1]); char c;
 while (0 != read(tube[0], &c, 1)) {
 if (c >='0' && c <='9')
 munif("%c", c);
 }
 close(tube[0]);
 exit(0);
 }
 default: close(tube[0]); for (int i=0; i < strlen(argv[1]); i++) write(tube[1], argv[1][i], 1);
 close(tube[1]);
 wait();
 exit(0);
 }
}

*stolen
on bien write(tube[1], argv[1], 1);*