

# TP noté

## Module 1 : Programmation système

Les travaux présentés dans ce compte-rendu ont été réalisés sur Linux

### Question 1

*Pour reconnaître la séquence, une liste de entiers est créée pour enregistrer quel est la séquence secret et une variable avec la position actuel de l'utilisateur est initialisée à 0.*

*Alors chaque fois que le programme reçoit un signal, le handler vérifie si le code du signal reçu est égal à celui de la séquence dans l'état actuel. Si c'est le cas, le programme ajoute 1 à l'état actuel pour passer au prochain élément de la séquence, au contraire l'état actuel est remis à 0.*

```
(...)  
static void signalHandler (int numSig )  
{  
    if(numSig == sequence[current_state]){  
        current_state+=1;  
    }else {  
        current_state =0;  
    }  
}  
...  
...
```

*Après avoir ajouté le traitement pour les deux signaux, le thread principal exécute la fonction pause() pendant que l'état actuel est différent de 4, c'est-à-dire, pendant que l'utilisateur n'a pas tapé la séquence correctement.*

```
(...)  
while(current_state != 4) pause();
```

---

...

Exemple :

```
andre@andre-730QED:~/Documents/Evaluation Système SRE 2324/CTP_SRE_SILVA_TELLES$ ./a.out
^C^Z^C^Z
andre@andre-730QED:~/Documents/Evaluation Système SRE 2324/CTP_SRE_SILVA_TELLES$ ./a.out
^C^C^C^C^C^C^Z^Z^Z^Z^Z^Z^C^C^C^Z^Z
^C^Z^C^Z
```

## Question 2

Pour afficher la date, le programme crée un processus fils qui exécute `execv` et attends son fin.

```
if( state == NORMAL_STATE){
    pidFils = fork();
    if(pidFils == 0)
        execv("/bin/date", (char**)cmd);
    wait(NULL);
    alarm(1);
}
```

Pour sortir du programme, il utilise un état pour savoir si l'utilisateur a déjà tape `ctrl-c` dans les trois derniers seconds

```
case SIGINT : /* traitement de SIGINT */
    if(state == NORMAL_STATE){
        printf ("pour quitter, entrez à nouveau Ctrl+C dans moins de trois secondes\n");
        fflush(stdout);
        state = PREPARE_QUIT_STATE;
        alarm(STOP_TIME);
    } else if(state == PREPARE_QUIT_STATE){
        state = QUIT_STATE;
    }
    break;
```

Exemple :

```
Fri Jun 21 04:11:38 PM CEST 2024
Fri Jun 21 04:11:39 PM CEST 2024
Fri Jun 21 04:11:40 PM CEST 2024
Fri Jun 21 04:11:41 PM CEST 2024
^Cpour quitter, entrez à nouveau Ctrl+C dans moins de trois secondes
Fri Jun 21 04:11:44 PM CEST 2024
Fri Jun 21 04:11:45 PM CEST 2024
^Cpour quitter, entrez à nouveau Ctrl+C dans moins de trois secondes
^C
```

## Question 3

## Question 4

Code :

```
for(int i=0;i<DIM;i++) for(int j=0; j<DIM; j++) A[i][j] = rand()%MAX;
for(int i=0;i<DIM;i++) for(int j=0; j<DIM; j++) B[i][j] = rand()%MAX;

for(int i=0;i<DIM;i++) for(int j=0; j<DIM; j++) {
    data_threads[i][j].i = i;
    data_threads[i][j].j = j;
    pthread_create(&(pid[i][j]),NULL,calc_pos,(void*) &(data_threads[i][j]));
}

for(int i=0;i<DIM;i++){
    for(int j=0; j<DIM; j++){
        pthread_join(pid[i][j],NULL);
        C[i][j] = data_threads[i][j].res;
        printf("%d ", C[i][j]);
    }
    printf("\n");
}
```

resultat

```
andre@andre-750QED: ~/documents/
77893715 78643217
58885351 26548475
```