Perfectionnement à la Programmation en C

Rapport du tp3

Sommaire :

Introduction :

L'objectif de ce troisième TP consistait à implémenter un chronomètre muni de diverses fonctionnalités. Par exemple, il est possible de sauvegarder la durée d'un tour. De plus, il y a un minuteur qui vérifie si chaque joueur ne dépasse pas une limite de temps. Ainsi, ce TP nous a permis de revoir et d'approfondir les notions relatives à l'interface NCURSES, la manipulation et la gestion du temps, la mise en place de pré-assertions ainsi que la consolidation et la prévention des erreurs provoquées par l'utilisateur.

Exercice 1 : (Gestion du temps)

Dans cet exercice, nous implémenterons les fonctions principales concernant la gestion du temps. Pour cela, nous utiliserons la structure « timeval » et la fonction « gettimeofday » qui sont disponibles dans le module « sys/time.h ».

1) Notre objectif consiste à écrire une fonction « int intervalle_ms(struct timeval début, struct timeval fin) ». Cette fonction calcule la différence en millisecondes entre deux durées qui sont intitulés « début » et « fin ». Le principal problème lors de l'élaboration de cette fonction était l'approximation exacte de la valeur de retour. En effet, avec l'utilisation de la division par 10, la rigueur de la valeur de retour est erronée à 10 puissances 2 chiffres. Ce qui provoque des différences de plusieurs secondes après 1 minutes. Ainsi, nous avions utilisé la fonction « rint » du module « math.h » pour éviter cette erreur.

2) Nous devions écrire quatre fonctions destinées à la conversion des millisecondes en centièmes, secondes, minutes et heures. Le principal problème de ses fonctions était un cas d'erreur. En effet, si la valeur en entrée était négative, alors la valeur de sortie était aussi négative. Or, ceci n'est pas admissible. Pour éviter cette erreur, nous avion mis en place des pré-assertions pour vérifier que la valeur d'enter soit positive.
3) Notre objectif consiste à écrire un programme « ChronoSimple.c » qui affiche sur la sortie standard un chronomètre sous le format h : m : s : c. Ainsi, nous utilisons les fonctions écrites précédemment. Puis, on a ajouté une nouvelle fonction intitulé « affiche_chrono » qui affiche sur la sortie standard le chronomètre. Dans cette fonction, on a juste appelé les fonctions écrites lors de la deuxième question. De plus, on a écrit un « main » qui prend au départ une mesure du temps avec la fonction « gettimeofday » puis dans une boucle while infinie on a suspendu très rapidement l'exécution du programme avant d'effectuer une deuxième mesure. Enfin, il suffit de calculer cette différence et d'afficher cette différence.
4) Enfin, nous devions implémenter l'algorithme proposé. Pour cela, on a copié le fichier précèdent dans un nouveau fichier. Puis, on a adapté le programme pour qu'il utilise la bibliothèque NCURSES. Enfin, on a implémenté l'algorithme.

Exercice 2 : (Etat de l'application et graphisme)

Dans cet exercice, notre objectif consistait à reprendre le programme « ChronoMoyen.c » et d'ajouter des nouvelles fonctions qui seront destinées à la création du chronomètre final.
Pour cela, on a procédé étape par étape.

1) Dans cette question, on a mis en place une structure chronomètre. Pour cela, on a écrit un nouveau fichier nommé « structure.h » qui contenait cette structure. Dans le fichier principal, on a inclus la structure avec la commande suivante « #include « structure.h » ».
2) Dans cette question, nous devions écrire une fonction qui initialisait un chronomètre. Pour cela, on a créé un chronomètre qui fait référence à cette structure. Puis on a mis l'état du chronomètre à 0 (chronomètre à l'arrêt). La durée totale du chronomètre à 0 secondes, la limite de temps de jeu à 25 secondes, puis on a supprimé chaque sauvegarde de temps pour chaque tour (Dans cette application n'est possible de n'effectuer que 30 sauvegarde maximum).
3) Dans cette question, il fallait écrire une fonction pour afficher la durée du chronomètre. On a donc juste réutilisé la fonction d'affichage du programme précédent.
4) L'objectif consistait à écrire une fonction pour l'interface graphique du chronomètre. Ainsi, nous sommes partis du principe d'afficher les options de l'application, de définir les espaces pour l'affiche du chronomètre, de l'avertissement et des tours sauvegardé. Puis cette fonction appelle les fonctions destinées à l'affichage du chronomètre, de l'avertissement et des tours.

5) Dans cette question, nous devons écrire une fonction qui avertit l'utilisateur que la limite de temps est dépassée. Pour cela, on a ajouté comme paramètres les coordonnées de l'écran où cet avertissement s'affichera ainsi que la valeur de la limite de temps. Ensuite nous avions défini la couleur du texte en rouge. Et, on a utilisé une fonction pour faire clignoter cet avertissement en rouge. Cet avertissement est composé du message suivant « Avertissement : h : m : s : c » avec la valeur de cet avertissement.
6) L'objectif était de coder une fonction pour ajouter des tours dans le tableau statique de la structure chronomètre. Donc, il suffisait de faire un parcours du tableau statique jusqu'à trouver un emplacement vierge pour enregistrer cette donnée ou alors sortir du tableau si le tableau était plein. Puis, il suffisait d'incrémenter les variables sur les nombres de tours.

De plus, on a rajouté une fonction pour l'affichage des tours. En effet, pour l'affichage des tours on a réservé trois lignes pour l'affichage. Puis, il suffisait de parcourir le tableau jusqu'à la fin ou qu'il n'est plus de valeur puis d'afficher les tours sous la forme suivante :

« Tour n : h : m : s : c »

Exercice 3 : (Chronomètre, étape par étape)

Dans cette exercice, l'objectif consistait à programmer l'application du chronomètre en utiliser les fonctions de l'exercice 2 et de l'exercice.

1) Pour cette question, il suffit de rajouter une condition dans la boucle principale. Et d'ajouter les commandes nécessaires pour sortir du programme proprement.
2) Dans cette question, consistait à réinitialiser l'ensemble de l'application. Tous d'abord, on a remis à zéro les deux chronomètres. Puis, on a suspendu l'exécution des deux chronomètres. Enfin, il suffisait de vider le tableau et remettre à jour l'affichage à l'écran.
3) Dans cette question, on a ajouté un chronomètre en parallèle qui se remet à zéro lorsque l'utilisateur ajoute une tour. Ainsi pour cette fonction, il suffit de remettre à zéro le chronomètre parallèle et d'ajouter un tour avec la fonction « ajouter_tour ».
4) Dans cette question, nous devions gérer la gestion de l'avertissement. Pour cela, on a juste utiliser une condition entre la limite de temps et la durée du chronomètre parallèle pour gérer cet avertissement.
5) Pour cette question, nous devions gérer les touches de F1 à F6. Ainsi, on a juste utilisé une condition pour chaque touche, puis modifier la valeur du chronomètre.

Conclusion :

Lors de ce TP, on a appris à mettre en place un chronomètre en temps réels avec des fonctionnalités évoluées comme le lancement et la suspension du chronomètre, la réinitialisation du chronomètre, ainsi que la sauvegarde de la durée d'un tour. D'autre part, ce TP nous a permis de revoir les notions de la bibliothèque NCRUSES ainsi que la mise en place de pré-assertions. Malgré certains petits problèmes rencontrés lors de l'avancement du projet. On a un projet complet et opérationnel.

Annexe 1 : (compilation du projet)

Pour compiler chaque programme on a utilisé un fichier bash qui permet de compiler les programme (un peu comme un makefile). Pour cela, il faut utiliser dans le terminal les commandes suivantes :

- chmod u+x nccomp (seulement au début pour créer un exécutable de nccomp)
- ./nccomp votre_fichier (pour compiler le programme choisie)
- ./votre_fichier (pour exécuter votre fichier)

Annexe 2 : (utilisation des fichiers)

Exercice 1 : Pour le premier exercice, aucune commande n'est demandée durant l'exécution sur programme. Il faut malgré tous compiler et exécuter le programme avec les commandes vue dans l'annexe 1. Pour arrêter l'exécution, utiliser la commande « ctrl + c ».

Exercice 2 : Pour le deuxième exercice, les commandes à connaitre sont les suivantes :

- Ctrl + c : pour arrêter le programme
- Touche Espace : pour suspendre le chronomètre

Pour utiliser le programme, il faut d'abord compiler et exécuter le programme.

Exercice 3 : Pour le troisième exercice, les commandes à connaitre sont les suivantes :

- Touche Espace : pour lancer et suspendre le chronomètre
- Touche r : pour réinitialiser le chronomètre
- Touche t : pour marquer un tour
- Touche F1 : pour incrémenter l'heure avertissement

- Touche F2 : pour décrémenter l'heure avertissement
- Touche F3 : pour incrémenter les minutes avertissements
- Touche F4 : pour décrémenter les minutes avertissements
- Touche F5 : pour incrémenter les seconds avertissements
- Touche F6 : pour décrémenter les seconds avertissements
- Touche q : pour quitter le programme

Pour utiliser le programme, il faut d'abord compiler et exécuter le programme.