Rapport et documentation technique

1.Définition les macros

Nous écrivons deux macros dans le fichier “macron_profile.h” :

->Le premier est PROFILE,nous obtenons le temps actuel en utilisant “clock_gettime”,nous calculons et écrivons le résultat dans le fichier “profile.log”.

->Le deuxième est return,nous calculons le temps après nous retournons la fonction.

2. Création l’arbre

Dans “linkedList.h”

Nous définirons une structure qui s’appelle Node,chaque node a 4 attributs.

->char* name: le nom de la fonction

->double funcTime: le temps d’exécution de cette fonction

->struct node *son: une pointeur qui associe son fils

->struct node *brother: une pointeur qui associe son frère

Dans “linkedList.c”

D’abord nous parcourons(utilisant “fscanf”) le fichier “profile.log” dans la fonction “createList”(récursive) et utilisons la fonction

“createNode” pour construire l’arbre,mais cette étape nous sauvegardons le temps d’entrer de la fonction aulieu du temps d’exécution.

En suite pour obtenir le temps d’exécution, nous parcourons le fichier à nouveau et sauvegardons le temps où la fonction finit d’exécuter dans une list .Nous parcourons l’arbre et calculons le temps d’exécution de chaque fonction.

Maintenant nous finissons de créer l’arbre dans la fonction “actualizeTree()”.

3. Représentation graphique

D’abord nous utilisons la fonction “countDepth” pour calculer le profondeur de l’arbre.Donc nous savons le dégalage de couleur en base 255(intervalColor=255/depth).

En suite nous commençons à parcourir l’arbre:

si ce node n’a pas de frères nous dessinons directement une rectangle dans une zone limite(que nous obtenons en parametre);

sinon nous couper la zone:

->soit selon le pourcentage du temps(son temps d’exécution/le temps de son frère);

->soit selon le nombre des fils(en cas du temps d’exécution est très petit par exemple inférieur à 0,01).

Nous coupons la zone verticalement ou horizontalement selon haut et large.Si le rectangle est trop petit donc nous le remplaçons par "...",et **nous arrêtons de parcourire ses fils et frères.**

Ensuite nous dessinons son fils et son frère récursivement.

4. Tester et démarrer

Nous avons écrit une fichier de bash "**test.sh**",et nous avons met tous les codes sources à profiler et son fichier log dans une répertoire "./test".Une fois nous exécutons le fichier de bash "test.sh",nous faisons les instructions "make" et "make clean".Nous exécutons tous les fichiers (*.c) pour obtenir son propre log et en suite nous applelons l'exécutable "myprofiler" pour construire l'arbre et visualiser graphique pour chaque fichier *.log.

Quand nous voulons **passer au fichier suivant** autrement dit visualiser le **profilage graphique suivant**,nous **cliquons la souris sur la fenêtre**.

5.Attention

Pour le fichier(.log) de VIOLAINE_HUYNH_KAIS_KINGONGO,son profondeur est 27 et le temps d'exécution sont très petit,donc sur le fenêtre son profilage graphique est incomplet.