

Projet XMLight

Jubert Valentin/Melaine Gbahi

Octobre 2014

Structure list_forest :

La list_forest_struct contient 5 champs, père, fils, frère, valeur et element. Père, fils, frère, elements sont d'autres list_forest. Père, fils, frère se comportent de la même manière, Elements est différent : quand on utilise la list_forest comme un element, value sert pour stocker l'attribut, et father est le précédent de l'element actuel, brother est le suivant. C'est donc une liste doublement chaînée.

List_forest_position contient un champ, here, de type list_forest, c'est ce qui indique où on est dans la list_forest général.

Quand on crée la list_forest on a été obligé de lui créer un nœud vide en guise de racine. En effet, avant on mettait à NULL la list_forest de départ, puis quand on ajoutais le premier nœud on faisais un malloc sur pos->here. Mais ça changeait l'adresse de here donc on ne pointait plus sur la list_forest, donc problème.

Pour l'ajout d'un fils, quand il y a déjà un fils, on l'ajoute en tant que dernier fils. En vrai on fait appel à la fonction d'ajout d'un frère pour le mettre à la fin de la list.

Pour l'ajout d'un frère, on le met à la fin de manière récursive, et comme on a bougé pos pendant l'ajout on le remet là où il était.

Il y a quelques fonctions de plus par rapport au .h :

empty_elem : elle sert à supprimer la liste d'element d'un nœud

element_prev : sert à aller au suivant (enfin le suivant dans un sens logique puisqu'ici la liste d'element se prend à l'envers si j'ai bien compris)

has_element_prev : sert à vérifier si brother existe.

Répartition du travail :

A l'heure actuelle la répartition du travail a été fait comme il suit :

list_forest.c par Valentin Jubert (moi)

le reste par Melaine Gbahi

Melaine ayant un peu de mal, je vais lui filer un coup de main sur sa partie

Principal problème :

Dans list_forest.c, après avoir compris que les typedef sont des pointeurs sur les structs (donc obligé d'utiliser → et non . pour accéder aux champs), j'ai longtemps été embêté à cause des malloc dans les fonctions d'ajout de nœud, soit ils étaient refusé, soit le fils avait la même adresse que la valeur, donc toucher à l'un revenait à toucher à l'autre, donc problème. Heureusement on m'a dit qu'il fallait mettre les struct dans les malloc et non utilisé les typedef