PASD - Compte Rendu TDM3

Nicolas Hiot - Darenn Keller

30 septembre 2016

1 État d'avancement du TD

Nous avons terminé le TD. Tous les tests passent et il n'y a pas d'erreur mémoire.

2 La structure arbre

Question 1 : Quelle est la représentation d'un arbre vide (En dessin) ?
Réponse sur la figure 1.

Question 2 : Les fonctions d'affichage (prefixe, infixe, postfixe) prennent en argument le flux où afficher, un pointeur vers une fonction permettant d'afficher une valeur et une valeur. **Pourquoi doit-on fournir un tel pointeur ?** On doit fournir un pointeur vers une fonction permettant d'afficher une valeur car l'arbre est incapable de savoir comment afficher une valeur sachant que celle-ci peut-être de n'importe quel type. En effet une valeur est stockée dans un **pointeur de type void (void *)**, pour nous permettre de placer au bout de ce pointeur une valeur du type que l'on souhaite (un **int** par exemple). C'est ce qui rend notre arbre **générique**.

Question 3 : Pour l'insertion, la recherche et la suppression, il faut trouver la bonne position dans l'arbre. Afin de ne pas ecrire trois fois la recherche, une fonction annexe est a faire (**arbre_chercher_position**). **Quel est le prototype de cette fonction et pourquoi utiliser celui-ci ?** Cette fonction prend en paramètres l'arbre dans lequel faire la recherche, et un pointeur vers la valeur à trouver dans l'arbre. Elle retourne un pointeur vers le noeud de l'arbre contenant la valeur que l'on recherche si la valeur est dans l'arbre, sinon vers le noeud (NULL) qui devrait contenir la valeur (on pourra y placer la valeur).

Question 4 : Quelle est la complexité de la fonction **arbre_rechercher_position** (En prenant en compte que l'arbre est équilibré) ?
La complexité de cette fonction est **O(log(n))** si l'arbre est équilibré (O(n) sinon). En effet à chaque étape de la recherche, on élimine la moitié des noeuds restants à visiter (En suivant le fils gauche on élimine tout les noeuds enfants du fils droit de la recherche et vice-versa).

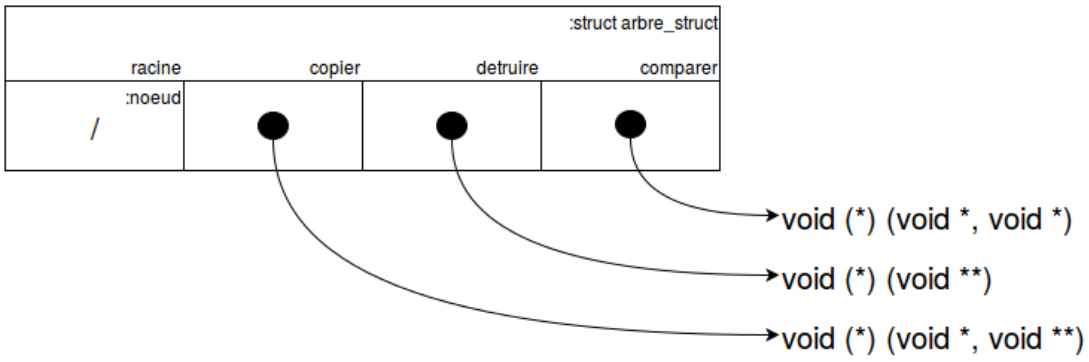

Figure 1 – Dessin présentant un arbre vide.

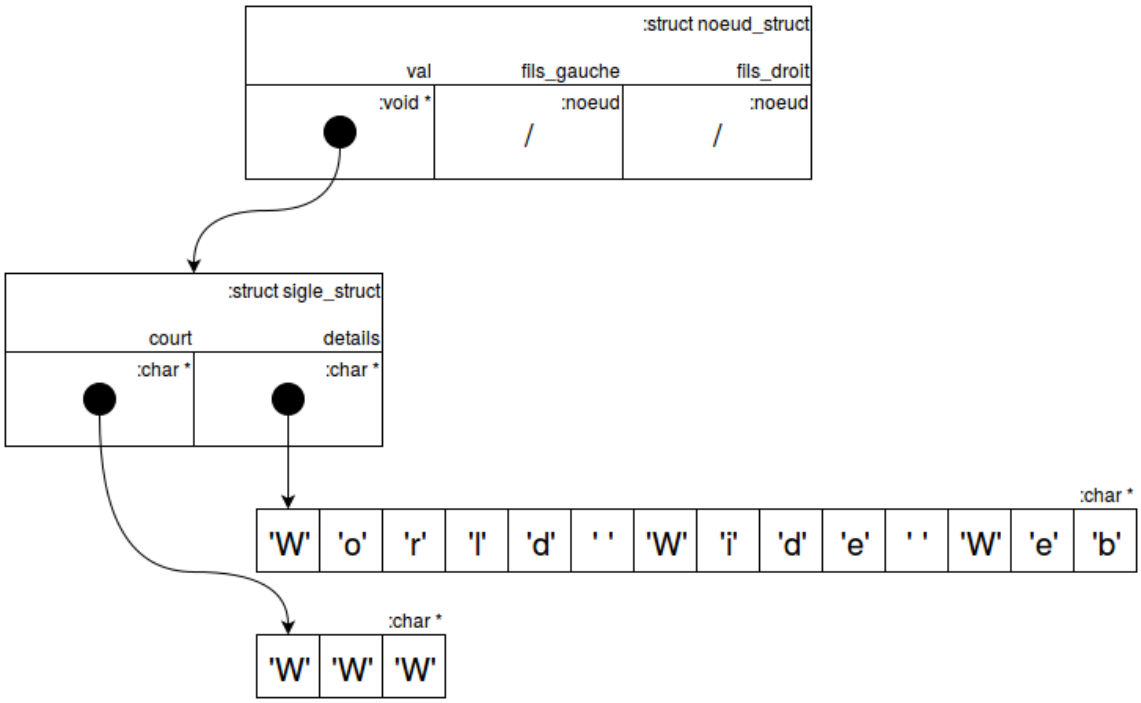

FIGURE 2 – Représentation d'un noeud enregistrant le sigle {.court = "WWW" , .detail = "World Wide Web"}

Question 5 : Quelle est la signification de la balise **\pre** dans Doxygen ?
Cette balise permet de spécifier l'ensemble des pre-conditions à l'execution de la fonction. Par exemple pour la fonction **int arbre_taille(arbre a)**, **a** doit être définie pour le bon déroulement de la fonction.

3 Le dictionnaire de sigles

Question 6 : Quelle est la représentation d'un noeud enregistrant le sigle { .court = "WWW" , .detail = "World Wide Web"} ?
Réponse en figure 2.

4 Bilan Personnel

Nous avons revu comment créer une structure générique (ici un arbre). Nous avons aussi appris à créer et utiliser la structure d'**arbre binaire**, et nous sommes rendu compte que l'arbre est plus efficace qu'une liste sur une recherche, si il est équilibré. Enfin nous avons utilisé cette structure pour créer un dictionnaire et avons appris à utiliser la balise **\pre** de Doxygen.