

Compte rendu du Sprint 2

1) Initialisation

-Poursuite du travail sur la mémoire. Ajout d'axiome sur la pile et création d'un arbre mémoire et de noeuds mémoires pour modéliser l'implémentation du tas.

2) Axiome de la pile

-Ajout des axiomatiques (1) (page 28/76 du cours) qui n'avaient pas été réalisés au Sprint 1. Et commencement des fonctions pour la pile des pages 29 et 30 cours. (RetirerDecl, AffecterVal).

3) Arbre mémoire pour le tas

-Création d'un arbre mémoire pour le tas avec des fonctions associés. (L'arbre possède une racine qui est la base de l'arbre. Et on le parcourt depuis cette racine). Nous avons implémenté diverses fonctions, pour retourner un noeud depuis son adresse, retourner la profondeur de l'arbre et retourner la racine).

4) Noeud mémoire

-Un noeud mémoire est un noeud de l'arbre, qui représente l'adresse d'une allocation mémoire. Cette structure est composée d'une adresse, d'une taille, détermine si le bloc est disponible pour une allocation, et deux fils gauche et droite pour les noeuds suivant.
-Des méthodes ont été implémentées pour allouer et libérer l'arbre, on alloue avec la taille du bloc à allouer et la fonction allouerMemoire retourner l'adresse d'allocation du bloc. Pour libérer la mémoire, on donne l'adresse du bloc à libérer et le noeud de départ (racine) à la fonction suppressionMemoire. La fonction parcourt l'arbre et rend disponible le bloc à l'adresse donnée si elle le trouve.

5) Tests unitaires

Des tests unitaires ont été réalisés sur les parties arbres et noeuds mémoire.

6) Conclusion

Ce qui a été implémenté :

- Une structure d'arbre et noeud pour le tas
- La poursuite des axiomes de la pile (prototypages 2 et début 3 du cours page 29 et 30 du cours)
- Des tests unitaires sur les parties arbre et noeud mémoires

La prochaine étape :

- Valider et poursuivre le tas
- Finir le prototypage 3 de la pile
- Commencer les axiomes 3 de la pile
- Fonction de hachage
- Tests unitaires avec au minimum 80% de couverture de code