Sujet de test : LLVM pass

Travail à réaliser sous Linux.

- Télécharger et installer la dernière version de LLVM.
- Ecrire une pass LLVM qui :

  - Prend en paramètre un fichier f1 qui contient sur chaque ligne: nom_variable niveau_criticité. Ce dernier est un entier.
  - Prend en paramètre un fichier f2 qui représente un programme C

- Votre pass LLVM remplace dans le fichier du programme C, tous les appels une_variable=malloc(taille), par une_variable=secure_malloc(taille,niveau_criticité), lorsque nom_variable se trouve dans le fichier f1.

Approche solution

Principe

1. Récupérer toutes instructions qui contiennent les appels malloc avec leurs arguments et les variables pointant vers ces appels.
2. Ouvrir le fichier f1 et construire un dictionnaire dont la clé est le nom de variable et la valeur le niveau de criticité.
3. Pour chaque instruction retenue, vérifier si le nom de la variable fait partie des clés du dictionnaire.
   (a) Si oui on récupère la valeur qui est le niveau de criticité et on remplace dans f2 une_variable=malloc(taille) par une_variable=smalloc(taille,credicité).
   (b) Sinon ; on ne fait rien

Implémentation

Cette pass a été réalisant en C++ en incluant certaines librairies propres à LLVM. Le code se trouve dans le fichier pass.cpp du dossier pass-malloc-secure_malloc.

J'ai pu parcourir le fichier et récupérer les arguments de tous les appels malloc.

J'ai rencontré des soucis lors de la mise en pratique du principe notamment sur comment récupérer la variable pointant vers l'appel malloc, raison pour laquelle je n'ai pas pu évolue.