HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

Alberto Dassatti, Bruno
Da Rocha Carvalho

Laboratoire de High Performance Coding

semestre printemps 2025

Laboratoire 3 : Optimisations de compilation

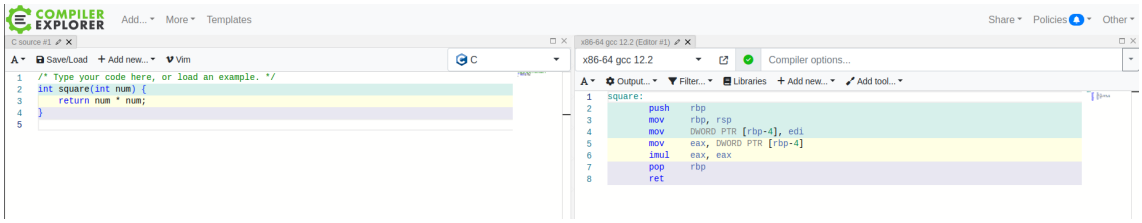
Temps à disposition: 2 périodes (1 séances de laboratoire)

1 Objectifs de ce laboratoire

Dans ce laboratoire, il vous sera demandé de mettre en évidence le comportement et les optimisations du compilateur.

2 Compiler explorer

Pour ce laboratoire, vous allez utiliser Compiler Explorer. Il s'agit d'un outil en ligne extrêmement pratique qui vous permettra d'écrire du code et de visualiser le code machine généré après compilation.



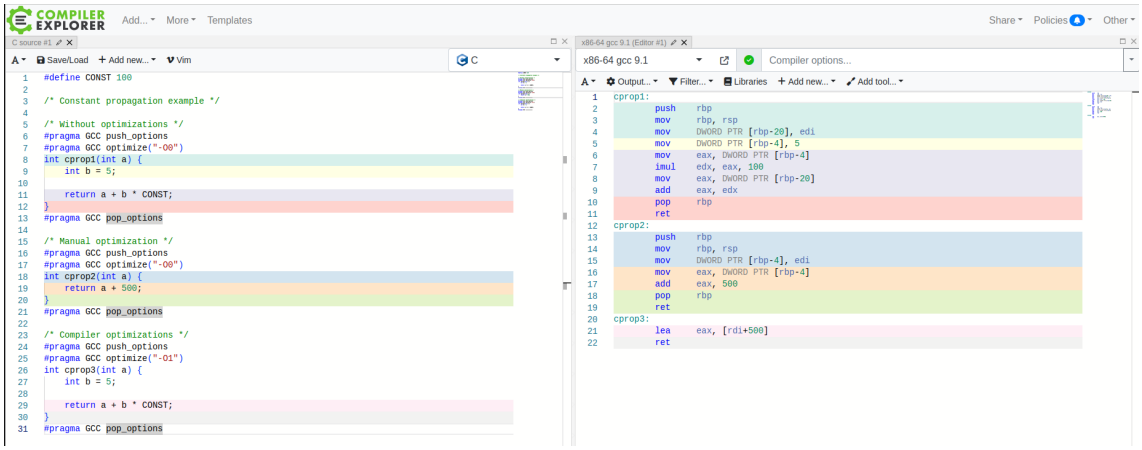
Dans l'outil, il vous est possible d'écrire n'importe quel code que vous souhaitez tant que sa syntaxe est correcte. Il vous est possible également de choisir votre compilateur, sa version ainsi que les options de compilations.

3 Travail demandé

3.1 Partie 1

Durant le cours sur les optimisations du compilateur, vous avez appris que le compilateur peut être assez conservateur. En effet, selon le code que vous écrivez, le compilateur peut omettre certaines optimisations afin d'éviter des effets de bord. En toute circonstance, le compilateur doit garantir le même comportement du programme qu'il soit optimisé ou non. Toutefois, le compilateur est souvent très efficace pour certaines optimisations que le programmeur peut, souvent, difficilement exprimer dans le langage avec lequel il développe.

Votre tâche est de trouver 3 exemples d'optimisations sur du code C. Pour chaque exemple écrit en C, vous devrez donner une version non optimisée (ni par vous ni par le compilateur), une version optimisée manuellement par vous et en dernier une version optimisée par le compilateur uniquement.



L'exemple ci-dessus est tiré du cours. Notez comment les fonctions sont optimisées différemment avec l'aide des pragmas. Exemple ici.

Nous vous encourageons à vous inspirer des options d'optimisations de GCC.

Néanmoins, vous aurez quelques contraintes pour ce laboratoire :

- Ne reprenez pas les exemples du cours.
- Écrivez des exemples simples et courts qui mettent bien en évidence l'optimisation que vous êtes en train de traiter.
- Ne donnez qu'uniquement des exemples en C.
- Vous pouvez utiliser toutes les options de compilation que vous souhaitez afin de montrer les effets qu'elles ont sur votre code.

3.2 Partie 2

Après vous être échauffé dans la première partie, vous allez maintenant analyser un code en suivant la même démarche que dans la Partie 1. Vous pouvez choisir une partie de votre propre code développé au premier labo ou, si vous préférez, explorer un autre code qui vous intéresse. L'objectif reste le même : observer un code non optimisé, l'optimiser vous-même, puis analyser les optimisations effectuées par le compilateur. N'hésitez pas à utiliser l'outil de diagnostic de compilation pour identifier des opportunités d'optimisation que vous auriez pu manquer.

4 Rendu

Vous devez rendre un rapport concis, au format md ou pdf, contenant toutes les informations nécessaires à la compréhension de vos exemples et des optimisations que vous avez testé. Expliquez bien les optimisations/problèmes que vous mettez en évidence. N'intégrez pas vos codes directement dans le rapport, mais plutôt, insérez un lien vers vos exemples de code avec la méthode de partage de Compiler Explorer.