IE4a - Projet 2008

Arithmétique sur de grands entiers en C et assembleur

1 Modalités

Ce projet est prévu pour être réalisé en binôme. Il représente un volume de travail personnel d'environ 50 heures par étudiant. Une réalisation individuelle ou par groupe de trois est envisageable, mais soumise à l'accord de l'enseignant responsable (O. Bailleux) sur la base d'une demande *motivée*. Il couvre la note de travaux pratiques et la note de contrôle continu de l'unité d'enseignement IE4a.
Le travail doit être original, i.e., entièrement conçu par ses auteurs et non copié sur une réalisation tierce diffusée par exemple sur internet ou dans un livre, article, ou autre communication. Des idées ou modèles issus de travaux et résultats déjà publiés peuvent être exploités, à condition de donner les sources documentaires associées.
Chaque groupe devra livrer un rapport imprimé sur papier, l'ensemble des codes sources, également imprimés sur papier, et un programme répondant aux spécifications détaillées dans la suite du sujet.
Le rapport devra comporter notamment
- une introduction reprenant l'objectif fonctionnel et pédagogique du projet ;
- une présentation et une justification de la représentation choisie pour les grand entiers, ainsi que les structures de données correspondantes en langage C et assembleur ;
- une description et une justification des principaux algorithmes utilisés ;
- une description de la stratégie et des modalités de validation (test, essais, expérimentations) des éléments critiques du logiciel, avec les résultats les plus significatifs ;
- le détail de la répartition des tâches entre les membres du groupe ;
- un mode d'emploi succinct permettant l'utilisation par une tierce personne des ressources livrées ;
- d'éventuelles idées et perspectives d'améliorations ;
- une conclusion montrant les enseignements apportés par l'ensemble du travail.

Les codes sources devront être commentés de manière succincte et pertinente, notamment en précisant le rôle de chaque fonction ou procédures.
Le programme livré devra être compilable et exécutable sur les machines disponibles dans les salles utilisées en TP, sous environnement Linux ou windows, au choix.

2 Objectif du projet

Vous devez concevoir et programmer un ensemble de fonctions (au sens du langage C) permettant de réaliser des calculs arithmétiques sur des entiers arbitrairement grands, sans perte de précision. On distinguera deux représentations pour les grands entiers :
- la *représentation interne*, dans une *base de référence* de votre choix ;
- la *représentation utilisateur*, en base 10, sous la forme d'une chaîne de caractères.

Votre choix de la base de référence utilisée pour la représentation interne devra être motivé par des considérations de *rapidité des opérations arithmétiques* à implanter, à savoir l'addition et la multiplication de deux grands entiers.
Vous devrez fournir deux versions de chacune des fonctions réalisant ces opérations : l'une programmée en langage C, l'autre programmée en assembleur sous la forme d'inclusions `inline` dans une fonction C. Des précisions sur cette technique d'utilisation d'instructions assembleur dans un programme en langage C vous seront données en cours et en séances de travaux pratiques.
Par ailleurs, vous devrez fournir deux fonctions de conversion :
- une fonction qui accepte en paramètre un entier en représentation utilisateur et retourne le même entier en représentation interne ;
- une fonction qui accepte en paramètre un entier en représentation interne et retourne le même entier en représentation utilisateur.

Seules des versions en langage C de ses fonctions de conversion sont demandées.
Si vous faites le choix (à vous de le motiver) de la base 10 comme base de référence, vous pourriez être tenté d'utiliser une représentation interne identique à la représentation utilisateur. Rappelez vous toutefois que l'objectif est de réaliser des calculs le plus rapidement possible, et que vous devrez fournir une version C et une version assembleur de chacune des deux fonctions de calculs. Vous devrez être capable de justifier vos choix !
Vous devez également développer une fonction permettant de calculer la factorielle d'un grand entier, qui permettra d'évaluer la vitesse de calcul de vos fonctions de multiplication.

3 Organisation du travail

Le meilleur moyen de rater ce projet est de vous précipiter sur un clavier dans l'heure suivant la distribution du sujet.
Au contraire, commencez par bien identifier les différents objectifs. Vous devrez concevoir :
- une représentation interne ;
- deux variantes d'une fonction d'addition, programmées respectivement en C et assembleur ;
- deux variantes d'une fonction de multiplication, programmées respectivement en C et assembleur ;
- deux fonctions de conversion entre les représentations interne et utilisateur, programmées en C ;
- une fonction de calcul et d'affichage de la factorielle d'un entier donné en représentation utilisateur.

La représentation interne est la clé de voûte du projet. La facilité de programmation et l'efficacité de l'exécution des fonctions à réaliser sont directement liées à son choix. Donnez vous le temps de considérer plusieurs options, d'en débattre entre vous.
Ensuite, réfléchissez aux algorithmes de calcul et de conversion. Attachez vous à bien comprendre les principes de base, détaillez les de manière claire et structurée. Commencez la rédaction des parties du rapport correspondantes.
Réfléchissez également à votre stratégie de validation. Comment allez vous tester le bon fonctionnement de vos fonctions ? Les données et protocoles de test doivent être définis *avant* de commencer la programmation.
Déterminez les compétences requises pour la réalisation de chaque fonction, et appuyez vous sur ces données pour définir l'ordre dans lequel vous allez réaliser ces fonctions. Par exemple au départ, vous ne connaîtriez pas la syntaxe pour inclure du code assembleur dans du langage C, mais cela ne gêne pas pour la conception de *l'organigramme* des traitements à réaliser en assembleur. En outre, l'environnement pédagogique `Emu8086` peut s'avérer très utile pour implanter et mettre au point une première version du code assembleur qui pourra ultérieurement être adaptée.
Dans tous les cas, ayez conscience que la pire approche possible pour réaliser un logiciel est de ne commencer les tests, expérimentations, et validation, qu'après avoir tout programmé. Au contraire, il faut valider chaque étape significative de manière approfondie. Si vous n'êtes pas certain que les fondadtions sont saines au moment où vous commencez à construire les murs d'une maison, vous risquez de grosses complications, et surtout vous risquez de devoir faire un travail inutile et improductif.

4 Concours

Un petit concours est associé à ce projet. Le groupe dont le programme calculera le plus rapidement la valeur de 1000 ! (dont la représentation en base 10 comporte près de 2600 chiffres) gagnera un lot. Pour ce concours, il y a deux catégories : C et assembleur. Seuls les étudiants ayant travaillé en binôme sont autorisés à participer au concours. Le temps de calcul pris en compte inclut celui de la conversion du résultat en représentation utilisateur et de son enregistrement dans un fichier texte par redirection de la sortie standard.