

Rapport UV Libre
Calculatrice Scientifique - Suite du projet d'Algorithmique - Promo 2001
encadré par Nicolas Delestre

Sujet : Au premier semestre, les étudiants d'ASI3 ont eu pour projet la création du calculatrice scientifique sur des nombres dont la précision est paramétrable. Malheureusement, ce projet n'a pas abouti. L'objectif de cette UV est donc de finir l'implémentation de cette calculatrice.

1) Utilisation d'un serveur CVS

La mise en place d'un serveur CVS m'a permis de développer ce projet et de me familiariser à cet outil. CVS est un système de gestion des différentes versions d'un ensemble de fichiers. Ainsi, j'ai pu développer sans me soucier des versions et faire des tests sans crainte de perdre une version qui fonctionnait.

2) Le modèle incrémental

L'UV d'algorithmique a permis à l'ensemble de la promotion d'appliquer le modèle incrémental. Ce modèle consiste en 3 phases : une phase initiale de communication entre les différents groupes du projet, une phase de développement pour chaque groupe indépendamment des autres et une phase de rassemblement pour aboutir un produit final.

Mais ce modèle a ses limites : Il était approprié pour un développement avec 30 personnes, pas avec 2. Par ailleurs, l'objectif premier du projet d'algorithmique était d'appliquer un modèle et non de le faire marcher. Le modèle incrémental a aussi permis à toute la promotion de participer à l'élaboration d'un projet informatique conséquent.

3) Le modèle en spirale

Nous avons choisi d'appliquer ce modèle parce qu'il correspondait bien à l'objectif de l'UV, au nombre de personnes inscrits et au type du projet. En effet, le développement consiste en des cycles de 4 phases répétées itérativement, chaque phase correspondant à une opération de la calculatrice. Les phases ont été défini comme suit :

- 1 Définition de l'opération à ajouter en fonction des opérations déjà validées
- 2 Retrait des commentaires autour des fonctions utiles pour cette opération, ou implémentation des fonctions
- 3 Test, correction des erreurs et détection des bugs
- 4 Validation de la phase

On voit très bien, avec la répétition de ce cycle, l'importance que représente la gestion des versions. A chaque fonctionnalité en cours d'implémentation, je pouvais toujours présenter un produit qui marchait en revenant à la version précédente. Le fichier principalement utilisé comporte, après 30 heures de travail, 19 versions différentes. A ce jour, je ne connais que CVS, qui pouvait me faciliter cette tâche.

4) Le projet à l'heure actuelle

Dans un temps défini de 30 heures, voici les opérations qui ont été validées :

- Addition sur les entiers
- Soustraction sur les entiers
- Multiplication sur les entiers
- Division sur les entiers
- Calcul de la factorielle
- Calcul d'arrangement
- Calcul de combinaison
- Addition sur les réels

et celle en cours de développement : Soustraction sur les réels (ils restent encore quelques bugs).

5) Quelques choix stratégiques de développement

Dés le début, une réflexion a été menée sur les calculs et le nombre significatif de chiffres après la virgule, qui reste la principale difficulté du projet et lui donne toute sa spécificité. La décision a été prise de passer le nombre de chiffres après la virgule au lancement de l'exécutible. Ainsi, on se restreignait à une précision constante durant une session de calcul.

Par ailleurs, pour faciliter le développement et la compilation des 35 fichiers, nous avons créé un Makefile, réalisant l'ensemble des compilations nécessaires : bison, flex et gcc; ainsi que le lancement de l'exécutible. Ce Makefile a par ailleurs été développé pour compiler aussi tous les fichiers non utilisés à ce jour : le développement de la calculatrice scientifique est donc toujours possible... En effet, seulement 14 fichiers ont été utilisés pour implémenter les opérations validées.

6) Problèmes rencontrés

Des problèmes, j'en ai rencontré à chaque cycle durant la phase de tests : une compilation qui ne marche pas, une opération qui ne donne pas le bon résultat. J'ai résolu ces types d'erreurs par la mise en place d'un affichage systématique dans chaque fonction, et ainsi détecter à quel moment le problème survenait. Dans la dernière version, on peut encore voir l'application de cette démarche : avant d'afficher le résultat, vous avez un affichage du type : "ReelInfini.c : On sort de l'addition de 2 entiers".

Un problème, que je n'ai pas résolu, c'est l'inclusion des bibliothèques après la compilation lex et bison. En effet, les bibliothèques à inclure dans le fichier "virgule.tab.h" n'était pas recopiées lors de la compilation "bison -d virgule.y". Après avoir contacté des spécialistes du Yacc et la documentation, il semble que ce soit un problème de C, ce que je ne crois pas du tout. Pour contourner ce problème, le makefile stoppe les compilations pour éditer les 2 fichiers et ainsi faire le copier/coller nécessaire.

7) Ce que j'ai appris

Cette UV m'a permis de me familiariser et d'utiliser CVS pour un cas pratique. J'ai aussi pris beaucoup de plaisir à développer selon le modèle en spirale, qui permet de se remotiver à chaque validation d'un cycle. Par ailleurs, j'ai pu approfondir mes connaissances sur les syntaxes des langages lex et yacc et sur la compilation en générale

D'un point de vue gestion de projet, j'ai pu découvrir tous les enjeux d'un choix de modèle, de la gestion du temps, des personnes et des compétences dans un projet informatique : j'ai pu travailler parce qu'avant, il y avait eu l'application du modèle incrémental par 30 personnes, qui m'ont fournit un code compréhensible et qui compilait.