Université Paris Nanterre

Licence MIAGE

Rapport de Stage

Contribution au développement et à la distribution d'une bibliothèque de calcul scientifique

Projet réalisé du 23 mars 2020 au 22 mai 2020

Membres du groupe

EL KHOURY Steven

Remerciements

Merci ‘a monsieur Hyon, maître de conférences à Nanterre Université, de m’avoir accepté pour réaliser ce stage. Grâce à son soutien, j’ai pu solidifier l’étendue de mes compétences en algorithmique et en langage orientée objet qui étaient à la base de mes points faibles. Cela m’a permis de renforcer mon assurance quand ‘a ma volonté de poursuivre mon parcours en MIAGE dans l’optique de devenir développeur. Cette opportunité m’a été donné parce que Messieurs Hyon et Delbot ont cru en moi et m’ont encadré et soutenu, c’est pourquoi je tiens à leur témoigner toute ma reconnaissance et ma gratitude.

Table des matières

1 Introduction

J'ai fait le choix de réaliser mon stage au sein de l'université même, avec pour maître de stage Monsieur Hyon dans l'optique de progrèsser et d'avoir un stage qui serait plus centrée sur l'apprentissage que sur une remunération financiere. Déterminé, il me tient 'a cœur de continuer mon parcours en première année de master MIAGE pour atteindre mon objectif professionnel de développeur. C'est aussi pour moi le moyen de combler des lacunes d'un point de vue algorithmique, mais aussi d'asseoir mes compétences techniques acquises cette année, tout en contribuant à un réel projet enrichissant.

•	•
•	•

TABLE 1 – Titre du tableau : légende du tableau

Et ne jamais oublier que lorsqu'on place un tableau, on doit l'utiliser comme ici avec le tableau Table 1. et on fait la meme chose avec une image, comme avec la Figure 1.

FIGURE 1 – Tux, le pingouin

2 Entreprise et environnement professionnel

Mon enviromment de travail est celui de l'université Paris Nanterre. L'université Paris Nanterre , a été créée en 1965 avec l'idée qu'elle serait le lieu de l'enseignement du renouveau. De nombreux professeurs en provenance de la Sorbonne et d'ailleurs se sont lancé pour venir y enseigner à cette époque. Nanterre université accueille actuellement plus de trente-trois-mille étudiants, ainsi que plus de deux mille enseignants-chercheurs et maître de conférences, pour seulement sept-cent membres administratifs. Elle propose un large panel de formation (que ce soit en Droit ou en Lettres, en Langues ou en Economie, en gestion ou en Mathématique et informatique). J'appartiens 'a l'une des 8 Unité de Formation et de Recherche présentes sur le campus de Nanterre, elle est composée de 3 départements : Sciences economiques, Gestion, et Mathématiques-Informatique, dont mon maïtre de stage fait aussi partie. Avec l'aide de plus de trois-cent formateurs, cette unité assure la formation par an d'environ quatre mille étudiants tous niveaux confondus. Les cursus proposés sont conçus pour permettre et faciliter une insertion rapide dans l'univers professionnel. L'UFR est très impliquée dans la recherche. C'est pour cela qu'elle dispose de laboratoires de recherche constitués des enseignants-chercheurs. Ils participent à des conférences et des séminaires afin de partager le fruit de leurs recherches(Micro Big par exemple), et ils publient dans des revues de prestige des articles pour diffuser leur travaux. Dans le cadre de mon stage, je travail en télétravail du l'épidémie. Je fais des conférences web mon maître de stage

au moins une fois par semaine. Nous faisons un bilan sur ce qui a été réalisé ainsi que les difficultés rencontrées, que ce soit au niveau de mon avancement sur les missions concernant le stage et les TD de java.

3 Missions confiées

3.1

On verra

3.2 Objectifs

Problèmes des usines logicielles et de la construction automatiques

4 Outils et technologies

4.1 Makefile

Le Makefile est un fichier qui permet de réaliser des actions avec la commande make par exemple pour la compilation de projet, mais aussi d'autres actions encore, dans ce cadre, nous n'aborderons que la compilation de projet. les Makefiles ne sont pas normalisés, ils peuvent être rédigés manuellement, et il existe de nombreux utilitaires qui permettent d'en générer automatiquement. Un Makefile est composé de règles, nous allons en decrire la syntaxe. Pour commencer on nomme le fichier cible qui sera l'exécutable, puis ses dépendances (qui sont les fichiers objets et les headers' files dont l'exécutable a besoin) et enfin, un saut de ligne plus loin, la(es) commande(s) qui est précédée d'une tabulation. Cela se présente comme suit

Une désignant le compilateur utilisée nommée CC (une telle variable est typiquement nommé CC pour un compilateur C, CXX pour un compilateur C++). CFLAGS regroupant les options de compilation (Généralement cette variable est nommées CFLAGS pour une compilation en C, CXXFLAGS pour le C++). LDFLAGS regroupant les options de l'édition de liens. EXEC contenant le nom des exécutables à générer

Les bibliothèques Comment se servir d'une bibliothèque précompilées en C++ sous linux windows (mettre les chemins vers la position de la bibliothèque).

Voixi les étapes de la création d'un programme :

Figure 2 – Etapes création programme

Le préprocesseur dans un premier temps s'occupera de traiter les commandes XXX (define, ifdef. . .) pour produire un fichier source où ces directives ont été exécutées.

le compilateur dans un second temps transforme le source en un fichier intermédiaire (.o/.obj), compilé, mais qui n'est pas exécutable. Le langage utilisé est spécifique au compilateur. Pour valider le code, le compilateur a besoin de connaître tous les symboles

que vous utilisez. Le nom d'une fonction, autant que le nom d'une variable sont des symboles. Si on utilise une fonction qui est inconnue, il faut le faire savoir avec un message d'erreur. En effet, pour lui, si la fonction est inconnue, il est incapable de traiter le symbole et donc, incapable de compiler le code. En C et C++, les fichiers .h/.hpp permettent de déclarer les symboles au compilateur et donc de lui dire : " cette fonction existe, voici les paramètres qu'elles acceptent, le type de valeur qu'elle retourne ".

Enfin l'editeur de liens prends les fichier objet du programme pour les assembler et produire un exécutable. Si le code des fonctions que vous utilisez n'est pas dans vos fichiers objets, il va les chercher dans les bibliothèques que vous lui aurez indiquées.

4.1.1 différence entre bibliothèque statique et dynamique

La bibliothèque statique sera compilée avec le programme et directement intégrée dans l'exécutable final. Il n'y aura donc pas besoin de .so/.dll lors de l'exécution.L'avantage de la compilation statique est de rendre l'exécutable indépendant (il peut fonctionner sur n'importe quelle plateforme compatible, sans pour autant nécessiter l'installation de la bibliothèque

La bibliothèque dynamique fera que l'exécutable ira chercher les fonctions à exécuter dans des fichiers séparés (.so/.dll).

4.1.2 les convention de nommage

FIGURE 3 – Convention de nommage des bibliothèques

4.2 Git

Git est logiciel de gestion de versions décentralisé. Cela permet avoir une tracabilité de fichier pour savoir qui la modifié, quand à t'il été modifié, qu'est ce qui a été modifié et pourquoi via une zone dite de dépot.

Git fonctionne par branche, par défault il y a la branche master qui contiendra le projet final,d'autre branche viendront s'ajouter a celle du master qui en seront des copies au moment de leur création, ces branches permettent de definir différentes tache du projet sur lequel on va travaillé et de les implémenté sur le master une fois le travail réalisé.

Le depot distant marche à travers une interface web conviviale, GitHub.

Voici quelque commande utile à Git :

- git remote add origin <url à placer donné par github> - git push -u prigin master ces 2 commandes permettent la creation du projet sur git par rapport aux fichier sur lequel on se trouve

git add <nom du fichier à mettre à jour sur github> <*> pour tout mettre à jour git commit -m "commentaire sur ce que l'on met à jour" git push origin <nom de la branche que l'on veut mettre à jour>

git branch pour la liste des branches.
git branch <nom de la branche> permet de crée la branche.
git checkout <nom de la branche> permet d'acceder à la branche.

4.3 Compilateur mingw

Pour le projet le compilateur Mingw64 est nécessaire à la création du Makefile permettant de faire compiler un projet utilisant la bibliothèque marmotecore sous windows. Vous trouverez comment l'installer dans les annexes

Construction d'exemple
exemple construit
modélisé avant de coder dans le code
description de l'exemples description de la bibliotheque que vous allez utilisé esquisse du code exemple de MDP

5 Travail réalisé

$\mathrm{Mdp}_{j}ouetSteven$

6 Difficultés rencontrées

7 Bilan

7.1 Conclusion

7.2 Perspectives

8 Bibliographie

[label] Auteur, TITRE, editeur, annee

[LAM94] L. LAMPORT, *LaTeX : A Document preparation system, Addison-Wesley, 1994*

9 Webographie

[CAT] `savoircoder.fr/cat`

10 Annexes

Annexe A : Exemple d'exécution du projet

// ne pas mettre java
usine logiciel et la construction automatique
insertion de bibliothèque pour windows vers le logiciel précompilé en C++ sur Aide