Projet programmation système W01

Introduction

Bienvenue dans ce « cours-projet » !
Il sera composé de **deux parties indépendantes** :

1. une prise en main, sur trois semaines, de l'environnement et des outils, composée de 3 rendus individuels ;
2. le développement, sur dix semaines, d'une table de hashage distribuée sur un réseau de serveurs, composée de 3 rendus de groupe (de deux étudiants).

(Et une semaine supplémentaire pour tout finir correctement.)

Le détail hebdomadaire se trouve ici : http://progos.epfl.ch/projet/

Les objectifs de ce cours-projet sont de vous permettre :

1. de développer une application concrète en C incluant des pointeurs et une gestion simple de fichiers ;
2. d'avoir un premier contact expérimental avec plusieurs outils usuels de « développement système », tels que le contrôle de versions (`git`), les pages de documentation (manpages), les Makefiles et le débogueur ;
3. d'installer et utiliser des bibliothèques logicielles externes (« *libraries* » en anglais) ;
4. de « *refactoriser* » du code au fur et à mesure des contraintes.

A partir de la 4e semaine, vous devrez travailler par groupes de deux. Chaque groupe aura son propre dépôt (« *github repository* »). Vous aurez chaque semaine de nouvelles instructions sur ce site et recevrez parfois du nouveau code via `git`, un outil permettant la gestion du travail en commun (« outil de gestion de versions »).

Les inscriptions pour les groupes sont déjà ouvertes et devront être terminées au plus tard le 12 mars au soir. Utilisez le lien dans le menu à gauche ci-contre pour inscrire votre groupe (**Note :** votre inscription est validée manuellement ; cela prend donc « *un peu* » de temps avant que vous ne receviez une confirmation et un dépôt GitHub pour votre groupe).

Concrètement, cette semaine

Objectifs

Cette semaine, nous allons procéder à la mise en place et la découverte des outils qui vous seront utiles dans votre projet :

1. mise en place de GitHub ;
2. découverte de Git ;

3. découverte des manpages Unix ;
4. compilation, modification et soumission d'un code C ;
5. documentation du code avec Doxygen.

Environnement de travail

Vous pouvez travailler : * sur les machines du CO (Ubuntu 14.04 ; si vous ne connaissez pas encore bien les possibilités de connexions à cet environnement, voir ici) ; * ou sur votre propre machine si elle est sous Linux ; * ou via un container Docker (voir cette page) ; * ou sur une machine virtuelle Ubuntu que vous aurez créée (voir cette page pour VirtualBox ou cette page pour VMWare).

Aucun autre système ne sera supporté (ni accepté).

En plus d'un environnement de développement C usuel (compilateur, éditeur), il vous faudra les outils suivants (`sudo apt-get install <package>` pour les installer sur une Debian/Ubuntu):

- un compilateur C : `gcc` ou `clang` (ou les deux si vous voulez comparer) ;
- `git` pour la gestion du projet (cf ci-dessous) ;
- un client SSH comme par exemple `openssh-client` pour communiquer avec GitHub (voir ci-dessous) ;
- les pages de documentation (*manpages*) : `manpages` et `manpages-dev` ;
- `libssl-dev` une bibliothèque de cryptographie que nous utiliserons pour calculer des « clés » (semaines 3 et 11) ;
- `doxygen` pour produire automatiquement de la documentation à partir de votre code.

GitHub

GitHub est un dépôt Git public accessible sur le Web. Chaque étudiant va recevoir au départ un dépôt personnel sur GitHub pour ce cours (inscription individuelle ci-dessous). Plus tard, une fois que vous serez également inscrit en groupe (de 2 étudiants, au plus tard le lundi 12 mars), vous recevrez un dépot GiHub supplémentaire pour votre groupe.

La première chose dont vous avez besoin est donc un compte GitHub. Si vous n'en avez pas encore, veuillez en créer un en vous enregistrant ici. Un compte gratuit suffit pour ce cours, il n'est pas du tout nécessaire d'avoir un compte payant.
(et si vous avez déjà un compte GitHub, utilisez le pour ce cours).

Ensuite, pour avoir un dépôt, procédez comme suit :

1. remplissez cette page avec votre email EPFL et votre username GitHub ;
2. attendez environ 15 minutes d'avoir un email de confirmation ;

3. vous devriez recevevoir un message de GitHub vous invitant à rejoindre un nouveau dépôt dans l'organisation projprogsys-epfl ; si vous n'avez **pas** reçu d'invitation après, disons, 15 minutes, allez directement à cet URL et acceptez l'invitation (cliquez sur le bouton près du haut) ;
   ce sera votre dépôt personnel pour ce cours.

Votre `REPO_URI` devrait ressembler à ceci :

```
git@github.com:projprogsys-epfl/pps18-student-GASPAR.git
```

Pour que cet URI fonctionne, vous devez enregistrer votre clé publique SSH dans GitHub. Vous pouvez le faire ici.

Si vous n'avez pas encore de clé SSH, vous pouvez en générer une sous Linux en faisant

```
ssh-keygen
```

puis copiez le contenu du fichier `~/.ssh/id_rsa.pub` dans GitHub à l'endroit mentionné ci-dessus.
Demandez de l'aide aux assistants si vous en avez besoin sur ce point (SSH).

NOTE : vous pouvez aussi utiliser `https` pour votre `REPO_URI` :

```
https://github.com/projprogsys-epfl/pps18-student-GASPAR.git
```

mais vous aurez à vous authentifier à chaque fois.

Git

Ce n'est pas le but de ce cours-projet que de vous *enseigner* les gestionnaire de versions ni tous les détails de `git` ; cela sera fait dans votre cours de « Software Engineering » l'an prochain. Le but ici est de vous faire prendre un premier contact avec ce genre d'outils et vous permettre de travailler efficacement en utilisant une partie miminale de ce qu'il offre.

Pour ce cours, vous ne devriez pas avoir besoin de connaître plus de quelques commandes simple de `git`:

- `git clone [REPO_URI]`
- `git pull`
- `git add [FILE]`
- `git commit -m "Commit message"`
- `git push`
- `git status`
- `git tag`

Pour chaque commande, vous pouvez obtenir de l'aide de la part de `git` en tapant:

```
git help <COMMAND>
```

Pour découvrir git : voyez cette page complémentaire.

Et ensuite...

Allez maintenant récupérer la suite de ce TP dans votre repo GitHub (si tant est que vous ayez reçu l'email de confirmation !). Pour cela, exécutez la commande suivante :

```
git clone REPO_URI
```

Cela doit créer un répertoire local sur votre machine, de même nom : `pps18-student-GASPAR` (avec *votre* GASPAR à la place !).

Allez dans ce répertoire :

```
cd pps18-student-YOURGASPAR
```

Vous devriez y trouver un répertoire `provided` contenant dans le sous-répertoire `week01` : * un fichier MarkDown `what_to_do_next.md` ; * deux fichiers C `hello_world.c` et `to_be_debugged.c`.

Ouvrez le `what_to_do_next.md` (soit dans un éditeur de texte, soit dans un viewer markdown ; dans les salles CO vous pouvez utilisez ReText qui fait les deux) et continuez les exercices de cette semaine depuis là-bas.

NOTE : pour voir du MarkDown directement dans Firefox, vous pouvez ajouter un plugin comme ceux-ci : * https://addons.mozilla.org/fr/firefox/addon/markdown-viewer/ * https://addons.mozilla.org/fr/firefox/addon/markdown-viewer-webext/ * https://github.com/Thiht/markdown-viewer

REMARQUE IMPORTANTE : Vous ne devez **jamais** modifier le contenu du répertoire `provided` de votre dépôt GitHub !

Conclusion

Nous reproduisons ici la conclusion que vous devez trouver en fin du fichier `what_to_do_next.md` (**ATTENTION !!** Ici, ce **N'EST PAS** la fin de la série ; vous **devez** continuer par le fichier `what_to_do_next.md` ; nous ne reproduisons ici sa conclusion uniquement pour information et rappel) :

Voilà, c'est tout pour cette semaine où nous avons voulu mettre en place et vous faire découvrir les outils de développement nécessaires à notre projet et très souvent utiles dans les projets de programmation système.
L'essentiel du travail de cette semaine était donc pour vous, pour votre développement personnel.

La seule chose que vous ayez à rendre, d'ici le dimanche 04 mars 23:59, est le fichier `debugged.c` (correction du fichier `to_be_debugged.c` fourni), à mettre dans votre dépôt GitHub dans le répertoire `done/week01` et à « étiquetter » du

tag `week01`. N'oubliez pas de « pousser » (`git push --tags`) le tout (avec le tag).

Par ailleurs, n'oubliez pas de vous inscrire par groupes de deux avant lundi 12 mars au soir (après il sera trop tard) au moyen du lien dans le menu à gauche ci-contre (**Note :** votre inscription est validée manuellement ; cela prend donc « *un peu* » de temps avant que vous ne receviez une confirmation et un dépôt GitHub pour votre groupe).