Projet Logiciel Transversal - Aide mémoire

Ensea - Option IS - Philippe-Henri Gosselin - 2016

1 Grandes étapes

Les principales étapes de préparation de votre environnement sont les suivantes :
— (Optionnel) Installation et création machine virtuelle
— Installation Ubuntu 16.04 avec packages
Puis, sur une seule machine :
— Installation kit de démarrage
— Création et envoie clefs ssh
— Création dépôt git local
— Création dépôt git distant
— "push" initial
Sur les autres machines :
— Récupération et installation clefs ssh
— Clonage local du dépôt distant
Sur toutes les machines :
— Préparation d'un environnement de développement (comme Netbeans)
— Validation via docker

2 Installation système

2.1 Système d'exploitation

Le suivi des projets a été préparé sous **Ubuntu 16.04 64-bit**. Tout autre environnement vous privera de garantie de support de la part des encadrants. Pour télécharger Ubuntu 16.04 : `http://releases.ubuntu.com/16.04/`. Choisissez la dernière version "desktop" 64-bit (AMD64).

Vous aurez besoin des paquets suivants (ici pour Ubuntu 16.04, pour les autres distributions, les noms peuvent varier) :

```
sudo apt-get install -y git cmake g++ tar xz-utils gzip bzip2 zip unzip dia
sudo apt-get install -y libpthread-stubs0-dev libsfml-dev libxml2-dev docker.io
```

2.2 Machine virtuelle

Il est possible de travailler dans une machine virtuelle si vous n'êtes pas sous Linux. Seul VirtualBox donnera lieu à une garantie de support de la part des encadrants. NB : Il n'est pas impératif de travailler sous Ubuntu 16.04 (cf Docker ci-après).

Vous pouvez télécharger virtualbox ici : `https://www.virtualbox.org/`

Créez une nouvelle machine dans virtualbox :
Page 1 du formulation de création : Nom : plt (ou autre), Type : Linux, Version Ubuntu 16.04 64-bit.
Page 2 du formulation de création : Mémoire : 2Go.
Page 3 du formulation de création : 'Créer un disque dur virtuel maintenant'.
Page 4 du formulation de création : Type de fichier disque dur : VMDK.
Page 5 du formulation de création : Allocation : Dynamiquement alloué.
Page 6 du formulation de création : Taille du fichier disque : 10Go.

Une fois la machine virtuelle créée, démarrez la. Une fenêtre s'affiche, choisissez le fichier iso d'Ubuntu 16.04 que vous avez téléchargé. Suivez la procédure d'installation.

Optionnel : installation des pilotes additionnel. Une fois la machine lancée, choisissez dans le menu périphérique[1] : "Insérer l'image CD des Additions invité". Un terminal dans la machine virtuelle va s'ouvrir, entrez le mot de passe que vous avez choisi lors de l'installation. Une fois la procédure terminée, redémarrez la machine virtuelle.

3 Dépôt versionné (à effectuer sur une seule machine)

3.1 Installation kit de démarrage

Créez un dossier pour votre projet, téléchargez et décompressez le kit de démarrage dans ce dossier :

```
mkdir <dossier de votre choix>
cd <dossier de votre choix>
curl http://perso-etis.ensea.fr/~gosselin/plt/plt-start-kit.tar.gz | tar xvz
```

Vous pouvez tester que cela fonctionne en exécutant `make` dans le dossier de votre projet (doit configurer, compiler, puis afficher à la fin "It works !").

3.2 Création clef ssh pour le projet

Créez une clef SSH spécifique pour le projet (à faire sur une seule machine) :

```
ssh-keygen -t rsa -C "<votre adresse email>" -N "" -f ~/.ssh/<nom groupe>
```

Pour `<nom groupe>`, collez vos noms de famille en utilisant que des minuscules. Vous pouvez choisir des versions plus courtes, l'important est d'avoir un id permettant aux encadrants de facilement faire le lien entre le projet et les personnes associées.

Envoyez par mail à tous les membres du projet, ainsi qu'à `gosselin@ensea.fr` les deux fichiers créés (`~/.ssh/<nom groupe>` et `~/.ssh/<nom groupe>.pub`).

3.3 (Cas Github) Création dépôt git distant pour le projet

Rendez vous sur `https://github.com/`. Créez un compte avec le nom de votre choix, puis créez un nouveau projet dans ce compte avec pour nom `<nom groupe>`.

Envoyez par mail l'url ssh du dépôt (de type `git@github.com:<compte git>/<nom groupe>.git`) à tous les membres du projet ainsi qu'à `gosselin@ensea.fr`.

Dans l'interface github, rendez vous dans les options (icône en haut à droite), menu "Settings", puis onglet à gauche "SSH and GPG keys". Cliquez sur "New SSH key", Title : ce que vous voulez, Key : contenu du fichier `~/.ssh/<nom groupe>.pub`.

3.4 Versionner dépôt local

Initialisez git localement (spécifique pour chaque machine) :

```
git config --global user.email "<email propriétaire machine actuelle>"
git config --global user.name "<nom propriétaire machine actuelle>"
```

Initialisez le dépôt local sur une machine :

```
git init
git add .gitignore CMakeLists.txt Makefile docker src
git commit -m "init"
git remote add origin <url dépôt distant>
git push -u origin master
```

1. de la fenêtre qui contient la machine virtuelle

Pour `<url dépôt distant>`, dans le cas github, vous pouvez la retrouver dans l'interface web, bouton "clone or download" (choisissez bien ssh), copiez-collez l'url affichée.

4 Dépôt versionné (à effectuer sur les autres machines)

4.1 Récupération clef ssh pour le projet

Copiez les fichiers de clefs ssh reçues par mail dans `~/.ssh`. Modifiez les droits :

```
cd
mkdir -p .ssh
chmod 700 .ssh
cd .ssh
chmod 600 <nom groupe>
chmod 644 <nom groupe>.pub
```

4.2 Clone local du dépôt distant

Initialisez git localement (spécifique pour chaque machine) :

```
git config --global user.email "<email propriétaire machine actuelle>"
git config --global user.name "<nom propriétaire machine actuelle>"
```

Initialisez le dépôt local sur les autres machines (pensez à copier les clefs ssh) :

```
cd <dossier de votre choix>
export GIT_SSH_COMMAND='ssh -i ~/.ssh/<nom groupe>'
git clone <url dépôt distant>
```

5 Logiciels de développement

5.1 Compilateurs C++

5.1.1 Linux

Installez le package g++.

5.1.2 Windows

Téléchargez Mingw à l'adresse suivante :
`https://sourceforge.net/projects/mingw/files/Installer/mingw-get-setup.exe/download`.

Exécutez le fichier téléchargé `mingw-get-setup.exe`.

Dans le choix des packages, prenez tout sauf gcc-ada, gcc-fortran et gcc-objc. Puis, dans le menu Installation, cliquez sur "Apply Changes".

Cliquez droit sur Ordinateur dans le menu démarrer, puis Paramètres Systèmes Avancés dans la liste à gauche, et enfin le bouton Variables d'environnement. Dans la liste des variables systèmes, étendez la variable PATH avec "`;C:\MinGW\bin`" (remplacez les chemins s'ils ne sont pas corrects). Pour vérifier, ouvrez une invite de commandes, tapez gcc et g++ : les commandes doivent être reconnues.

5.2 CMake

5.2.1 Linux

Installez le package cmake.

5.2.2 Windows

Téléchargez la version compatible avec votre installation de windows (téléchargez l'installeur .msi) : `https://cmake.org/download`.

Exécutez l'installeur, et choisissez l'option 'Add CMake to the system path' (pour tous les utilisateur ou uniquement celui-ci, à votre convenance).

5.3 SFML

Installez CMake et le compilateur C++ (cf sections précédentes).

5.3.1 Linux

Installez le package libsfml-dev.

5.3.2 Windows

Téléchargez les sources de SFML version 2.3.2 : `http://www.sfml-dev.org/files/SFML-2.3.2-sources.zip`.
Décompressez l'archive dans le dossier de votre choix.
Ouvrez une invite de commande, rendez vous ce dossier, puis exécutez les commandes suivantes :

```
cmake -G "MinGW Makefiles" -DCMAKE_BUILD_TYPE=RELEASE .
mingw32-make
```

Pour tous les programmes qui utiliseront SFML, pensez bien à mettre les .dll dans le même dossier que celui de votre exécutable.

5.4 Netbeans

Installez CMake et le compilateur C++ (cf sections précédentes).

5.4.1 Linux

Téléchargez netbeans 64-bit pour C++ à l'adresse suivante : `https://netbeans.org/downloads/`.
Dans le dossier où a été téléchargé le fichier, exécutez une commande du type (ici pour la version 8.1) :

```
sudo sh netbeans-8.1-cpp-linux-x64.sh
```

Netbeans sera alors disponible via le lanceur (icône Ubuntu en haut à gauche de l'écran), ainsi qu'en ligne de commande en tapant 'netbeans'. Vous pouvez supprimer le fichier que vous avez téléchargé.

5.5 Windows

Téléchargez la version de Netbeans pour C/C++ compatible avec votre installation de Windows : `https://netbeans.org/downloads`. Exécutez l'installeur.

Lancez Netbeans, allez dans le menu Tools/Options, puis onglet C/C++ : netbeans va chercher les compilateurs, et doit normalement les trouver seul. Si ce n'est pas le cas, il faut indiquer tous les éléments.

5.6 Linux et Windows

Pour créer un projet avec netbeans, suivez les étapes suivantes :
Page 1 formulaire : Créez un nouveau projet de catégorie C/C++, projet avec sources existantes.
Page 2 formulaire : Choisissez le dossier de votre projet ; Sélectionnez le mode de configuration "custom".
Page 3 formulaire : "Run in folder" : choisissez le dossier "build" de votre projet (il faudra peut être le créer).

Page 4 formulaire : (suivant).
Page 5 formulaire : Assurez vous que l'unique dossier source est le dossier "src" de votre projet (supprimer ceux proposés, ajoutez celui-ci).
Page 6 formulaire : (suivant).
Page 7 formulaire : (finir).

Si le C++11 ne semble pas être reconnu : dans les propriétés du projet (bouton droit souris sur le projet), section "Code Assistance/C++ compiler", paramètre "C++ Standard" : choisissez "C++11".

Si vous souhaitez toujours compiler avant d'exécuter : propriétés projet / section "Run" / paramètre "Build first" : cochez la case.

Lors de la première exécution, choisissez le binaire "run" dans le dossier "bin" du projet.

6 Docker

6.1 A propos de docker

Docker permet de travailler dans un environnement très précis, sans avoir à procéder à l'installation ou à l'émulation d'un système complet. Il est disponible pour la plupart des systèmes d'exploitation récent. Docker est utilisé par les encadrants pour tester les projets : à vous de reproduire ces tests afin d'assurer la validité des livrables.

Docker repose sur la notion de container et d'images. Un container est un état particulier d'un système. Chaque commande exécutée donne lieu à un nouveau container.
Commandes utiles :

`sudo docker ps` → Afficher la liste des containers actuels.

`sudo docker rm <container id>` → Supprime un container.

`sudo docker stop $(docker ps -a -q)` → Arrête tous les containers.

`sudo docker rm $(docker ps -a -q)` → Supprimer tous les containers.

Une image est un container statique. Les images ont un nom/id humainement compréhensible. Il existe de nombreuses images automatiquement téléchargées par docker. Tous les containers créés pour fabriquer l'images sont attachés à celle-ci.
Commandes utiles :

`sudo docker build -t plt-initial -f docker/plt-initial .` → Fabrique à partir du fichier docker "docker/plt-initial" une image avec le nom 'plt-initial'. Les containers à partir de cette image auront accès aux données dans le dossier courant de l'hôte.

`sudo docker images` → Affiche la liste des images actuelles.

`sudo docker run -ti plt-initial /bin/bash` → Ouvre une console à partir de l'image 'plt-initial'.

`sudo docker rmi plt-initial` → Supprime l'image nommée "plt-initial" ainsi que les containers attachés.

6.2 Procédures de validation

Si vous souhaitez faire une première pré-validation rapide, lancez tout simplement la commande `make` dans le dossier de votre projet. Celle-ci va tout nettoyer, tout reconfigurer, et tout recompiler. Puis, vous pouvez lancer votre exécutable en ligne de commande. Attention : n'utilisez pas cette commande pour travailler au quotidien, préférez celle générée par CMake (dans le dossier build), qui n'effectue que les opérations nécessaires pour la mise à jour de votre exécutable.

La procédure suivie par les encadrants du projet pour valider votre projet sont les suivantes :

— Ouvrir une session invité vierge sur une machine ;
— Cloner votre dépôt git ;
— Lancer la commande `make test` dans le dossier de votre projet. Cette fabrique une première image système initiale via docker, puis une deuxième image avec votre projet compilé, et enfin lance votre exécutable dans cette dernière image.