Part 1

Problème des machines

Changer le fichier sources.list + reboot+ Update

Dépôts Ubuntu : /etc/apt/sources.list : https://doc.ubuntu-fr.org/depots\_xenial / Us par Fr

https://debian-facile.org/doc:systeme:apt:sources.list#sourceslist-pour-debian-stable : commenter les CD (http://deb.debian.org/ )

Différence entre les types de git Repository : http://www.saintsjd.com/2011/01/what-is-a-bare-git-repository/

Clavier : sudo loadkeys || https://www.it-connect.fr/passer-le-clavier-en-azerty-sous-ubuntu-serveur%EF%BB%BF/

à installer avant la configuration des interfaces

apt-get update / upgrade !!!

sudo apt-get install build-essential

apt-get install git-all

sudo apt-get install ssh (ubuntu) // apt-get install ssh(debian)

Pour le redémarrage du système : sudo /etc/init.d/networking restart

ou service network restart|reboot

Premièrement, la configuration des interfaces .Puis, changement de l'interface (bridge) et redémarrage

Sous ubuntu

nano /etc/network/interfaces

Auto eth2

iface eth2 inet static

address 10.0.0.x

netmask 255.255.255.0

network 10.0.0.0

Puis la création des fichiers et création du repository… :

Serveur: (Debian)

mkdir RepositoryGit

cd RepositoryGit

git init --bare

chown server user name : server user name /home/ server user name /RepositoryGit -R (path repo serveur)

=> sans / à la fin du Repository !!!!!

puis sous le client

création des dossiers :

N.B : On crée le prog dans la branche master puis on crée la branche dev pour l'éxecuter

mkdir Project1

cd Project 1 => mkdir src include bin lib (puis nano main.cpp add.cpp => src et add.h => include)

http://www.learncpp.com/cpp-tutorial/19-header-files/

Puis dans le projet: sous le client

gitinit

gitadd -A

git config --global user.name "Name"

git config --global user.email "Name@hotmail.com"

git config --list

git commit -m "Création du programme"

git commit -m "firt1" --allow-empty ?

Commandes supp

pr supp une branche : git branch -D [name of branch]

pour supprimer une commit précise :

We put it on the head : git reset --hard {num commit}

And we delete it : git reset --soft HEAD^

Part 2

I) l’exécution simple se fait en trois manières  sans makefile First Commit

Créer la branche dev et se déplacer dessus : git checkout -b dev

pour voir les branches: git branch

* Méthode a:

g++ -E src/\*.cpp -I include

g++ -c src/\*.cpp -I include

g++ -s src/\*.cpp -I include

g++ \*.o

./a.out

Ls -a

* Méthodeb:

**g++ -o bin/addmake src/\*.cpp-I include**

l’exécution se fait : ./bin/addmakepat

* méthode c:

g++ -c src/add.cpp -I include

g++ -c src/main.cpp -I include

g++ main.oadd.o

./a.out

Merge

Puis on exécute via la méthode 1 le prog

git status

git add -A

git commit -m "Execution prog meth normale"

on bascule vers master : git checkout master

**{ à la première merge**

touch file.txt

git add -A

git commit -m "Test" **}**

git merge dev

git log --oneline --graph

P.S : et on revient sur la branche dev pour en continuer

N.B: les mêmes étapes pour les autres commit!

Part 3

III) méthode : Makefile Second Commit

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# addmake: src/main.cpp src/add.cpp

# g++ -o bin/addmake src/main.cpp src/add.cpp -I include

Exécution: ./bin/addmake

CC= g++

CFLAGS=

LDFLAGS=

EXEC= addmake

INC\_DIR=./include

SRC\_DIR=./src

SRC= $(wildcard $(SRC\_DIR)/\*.cpp)

OBJ=$(SRC:.cpp=.o)

all: $(EXEC)

$(EXEC): $(OBJ)

$(CC) -o $@ $^ $(LDFLAGS)

add.o: $(INC\_DIR)/add.h

%.o: %.cpp

$(CC) -c $< -o $@ -I $(INC\_DIR) $(CFLAGS)

clean:

rm $(SRC\_DIR)/\*.o

mrproper:clean

rm $(EXEC)

pour tester :

make ou make addmake pr build, afficher les erreurs....

pr éxecuter ./addmake

make mrpropergit

//////////////////////////////////////////////////

II) lib statique

Lib statique:

g++ -c src/add.cpp -I include/

ar -q libadd.a add.o

g++ src/main.cpp libadd.a -o bin/addmake -I include/

./bin/addmake

III) Lib dynamique:

**g++ -c src/add.cpp -fPIC -I include/**

**g++ -shared -fPIC add.o -o lib/libadd.socd**

g++ src/main.cpp libadd.so -o bin/addmake-I include/

./bin/addmake ? ERROR mak

création répertoire lib

**export LD\_LIBRARY\_PATH=./lib/:$LD\_LIBRARY\_P**

echo $LD\_LIBRARY\_PATH (pour vérifier la création du dossier lib : ./lib/: )

./bin/addmake

http://stackoverflow.com/questions/8096015/creating-a-simple-makefile-to-build-a-shared-library

----------------------

Part 4 PUSH

Sur la machine cliente

git remote–v

git remote add origin user-server@ipServer:/home/user/RepositoryGit => git remote -v

Pour supprimer un remote : git remote rm origin

git push origin master

Sur la machine de clonage

Config d'interface (Autre adresse : 10.0.0.3) et le nom et l'adresse de l'utilisateur pour les faire différencier ...

git clone user-server@ipServer:/home/derder/RepositoryGit ProjectClone

Après clonage: on fait des modif et on les push vers le serveur!

sur projet2 : cd Gitrepository =>nano file.txt + modif

git add -A

git commit -m "testing clonage"

git log --oneline

git push : pr renvoyer sur le serveur

Puis sur la machine cliente

git pull origin master

Et sur le premier projet,

touch testclt1.txt

gitadd -A

git push origin master

Sur le 2 eme projet : clone

cd Gitrepo

git pull ainsi de suite...