

POLYTECHNIQUE
MONTREAL

LE GÉNIE
EN PREMIÈRE CLASSE



Travail pratique #3 :
Proposition de solution
LOG1000 – Ingénierie logicielle

Trimestre : Automne 2019

Équipier1 : Moussa Fofana 1955968
Équipier2 : Lompo Augustin 1927045

Équipe : 40 (groupe B1)

Présenté à :
Chargé de laboratoire

École polytechnique de Montréal

1 nov. 19

Partie 1

1. Révision technique de code source

1. 1

Le logiciel Gerrit est une application Web qui est libre et gratuite pour tous. C'est un logiciel qui permet aux utilisateurs d'approuver ou de rejeter des modifications d'un code source via un navigateur web. Il est utilisé avec Git afin de pouvoir poster ces changements de code. [1]

1.2

Un patch ou révision dans un contexte de contribution à un projet est en fait une section de code ajouté à un logiciel afin d'y apporter des modifications. Le but de cette révision étant de corriger un bug trouvé dans le code du projet dans l'optique de le rendre plus fonctionnel et optimal.

1.3

- La sous-section Open est constituée de tous les projets qui sont ouverts aux utilisateurs afin d'être révisés et modifiés.
- La sous-section Merged est constituée de tous les projets qui ont été patchés et qui ont pu être soumis dans leur branche de destination.
- La sous-section Abadoned est constituée de tous les projets qui ont été abandonnés et qui ne sont plus ouverts aux changements et modifications.

1.4

L'auteur de la révision est **Vadim Yanitskiy**

1.5

Un seul fichier a été changé. Il s'agit du fichier [epan/dissectors/packet-gsmtap.c](#)

1.6

Quatre variables globales de type static int ont été créés à la ligne 68, puis à la ligne 444 une fonction void dissect_ptcch_dl() a été ajoutée, à la ligne 685 un nouveau case du switch a été ajouté, enfin à la ligne 845 des instructions afin d'afficher des messages à l'écran ont été ajoutées.

1.7

Il est indiqué un nombre relatif suivie du nom d'une personne. Le nom désigne la personne ayant fait le review c'est à une vérification du code, un test et le nombre relatif est la note attribuée au code par le reviewer (la personne qui effectue la vérification). Dans notre cas, Anders Broman et Pau Espin Pedrol ont révisé le code auquel ils ont attribué respectivement les notes de +2 et +1.

1.8

L'auteur de la révision est Pascal Quantin

1.9

Trois fichiers ont été modifiés il s'agit de hnbap.cnf, packet-hnbap-template.c et packet-hnbap.c

1.10

Dans le fichier hnbap.cnf :

A la ligne 153, une fonction #FN_BODY IMSI VAL_PTR=¶meter_tvb a été ajoutée

Dans le fichier packet-hnbap-template.c :

A la ligne 22, un include a été ajouté; à la ligne 46 une variable de type static int a été créée; et à la ligne 146 l'instruction &ett_hnbap_insi a été ajoutée.

Dans le fichier packet-hnbap.c :

A la ligne 30 un include a été ajoutée; à la ligne 1416 une instruction a été ajoutée; aux lignes 479,103,249,323,1800,1821,1842,1863,1884, ... des nombres ont été changés.

1.11

Il est indiqué un nombre relatif suivie du nom d'une personne. Le nom désigne la personne ayant fait le review c'est à une vérification du code, un test et le nombre relatif est la note attribuée au code par le reviewer (la personne qui effectue la vérification). Dans notre cas, Pascal Quantin a révisé le code auquel il a attribué la note de +2.

2. Gestion de bogues

2.12

C'est Alex Kirk qui a trouvé le bogue. Oui il s'agit d'un être humain

2.13

Ce bogue est majeur faible (Low Major en anglais)

2.14

Le niveau d'importance d'un bogue est déterminé par vote.

2.15

Non ce bogue ne risque pas de compromettre le fonctionnement du logiciel car il s'agit d'un bogue mineur.

2.16

46 personnes au total ont commenté le bogue.

2.17

Oui la communauté a confirmé la validité du bogue. Par des test et essai ils ont essayé d'exécuter le code qui posait problème et ils sont tous arrivés à la conclusion que le bogue était bien valide.

2.18

Oui le bogue a été résolu, grâce au statut du bogue qui nous indique que celui-ci a été résolu.

3. Intégration continue

3.19

Buildbot est un outil open source et multiplateforme qui permet d'envisager des scénarios et d'en tester les incidences sur l'exécution de nos projets. [2]

3.20

Les builders sont : Clang Code Analysis, Ubuntu 18.04 x64, Visual Studio Code Analysis, Windows Server 2019 x64, Windows Server 2019 x86, macOS 10.14 x64.

3.21

Le vert désigne les compilations qui ont réussi.

L'orange désigne les warnings renvoyés par le builder.

Le rouge désigne les compilations échouées.

3.22

Dans le buid 456 il y a un warning
(Image ci-dessous)

Builder Windows Server 2019 x86 Build #456

Results:

Build successful

SourceStamp:

Project	wireshark
Repository	ssh://wireshark-buildbot@code.wireshark.org:29418/wireshark
Branch	refs/heads/master
Revision	35615574e5691c7bf2e09254271ac32c9cf59d95
Got Revision	35615574e5691c7bf2e09254271ac32c9cf59d95
Changes	1 change

BuildSlave:

windows-2019-x86

Reason:

The SingleBranchScheduler scheduler named 'Gerrit' triggered this build

Steps and Logfiles:

1.	leftover processes check for leftover processes	(0 secs)
1.	stdio	
2.	rename-item-cmbuild removed old build directory	(3 secs)
1.	stdio	
3.	git update	(5 secs)
1.	stdio	
4.	make-version.pl set release information	(0 secs)
1.	stdio	
5.	shell created CMake build directory	(0 secs)
1.	stdio	
6.	compile ran CMake	(1 mins, 16 secs)
1.	stdio	
7.	compile 1 compiled with MSBuild warnings	(40 mins, 9 secs)
1.	stdio	
2.	warnings (1)	
8.	compile 2 built guides	(7 mins, 31 secs)
1.	stdio	
9.	compile 3 prepared NSIS package	(3 mins, 21 secs)
1.	stdio	
10.	compile 4 prepared WiX package	(3 mins, 13 secs)
1.	stdio	
11.	sign executables signed executables	(1 mins, 13 secs)
1.	stdio	
12.	compile 5 built NSIS package	(5 mins, 18 secs)
1.	stdio	

Build Properties:

Name	Value	Source
branch	refs/heads/master	Build
buildid	C:\buildbot\builders\wireshark-master-32\windows-2019-x86	slave
buildname	Windows Server 2019 x86	Builder
buildnumber	456	Build
codebase		Build
commit-description	v3.1.1rc0-575-g35615574e5	Git
event.eventCreatedOn	1572552182	Change
event.refUpdate.newRev	35615574e5691c7bf2e09254271ac32c9cf59d95	Change
event.refUpdate.oldRev	4759eb7a7a3aceff7d96cccc6a55c7dd9fdd17a	Change
event.refUpdate.project	wireshark	Change
event.refUpdate.refName	refs/heads/master	Change
event.submitter.email	guy@alum.mit.edu	Change
event.submitter.name	Guy Harris	Change
event.submitter.username	gharris	Change
event.type	ref-updated	Change
got_revision	35615574e5691c7bf2e09254271ac32c9cf59d95	Git
project	wireshark	Build
repository	ssh://wireshark-buildbot@code.wireshark.org:29418/wireshark	Build
revision	35615574e5691c7bf2e09254271ac32c9cf59d95	Build
scheduler	Gerrit	Scheduler
slavename	windows-2019-x86	BuildSlave
warnings-count	1	WarningCountingShellCommand
workdir	C:\buildbot\builders\wireshark-master-32\windows-2019-x86	slave (deprecated)

Forced Build Properties:

Name	Label	Value
------	-------	-------

Responsible Users:

1. Guy Harris

Timing:

Start	Thu Oct 31 20:08:03 2019
End	Thu Oct 31 21:33:57 2019
Elapsed	1 hrs, 25 mins, 54 secs

Resubmit Build:

To force a build, fill out the following fields and push the 'Force Build' button

Reason for re-running build:

Rebuild using:

Figure 1: Image du warning du build

3.23

Le build a un warning car dans le fichier lemon.c il y a un paramètre formel qui est déréférencé.

3.24

Le numéro de révision est 35615574e5691c7bf2e09254271ac32c9cf59d95. L'auteur est Guy Harris.

Partie 2

2 : Développement

2.25

Log1000_A19 correspond au nom d'un projet.

2.26

Elle sert à cloner un projet existant (créer une copie) d'un projet existant, afin de permettre d'apporter des modifications à la copie du projet sans affecter le projet initial.

2.27

(*Image ci-dessous :*)

You pushed to **Equipe40** just now

Create merge request

Equipe40 ▾

log1000_a19 / + ▾

History

🔍 Find file

Web IDE

📄 ▾



test

sergiu mihaltan authored 1 month ago

41cab359



Name	Last commit	Last update
📁 TP3-A19	ajout de fichiers	1 month ago
📄 README.md	Initial commit	1 month ago
📄 README.txt	test	1 month ago

📄 README.md

log1000_A19

Figure 2: Branche Equipe40

2.28

Nous avons utilisé la commande git checkout Equipe40 (Equipe40 est le nom de notre branche).

2.29

Nous avons utilisé la commande git add.

2.30

Un Merge Request est une demande afin de fusionner une branche à la branche master.

2.31

Le merge Request permet de faire vérifier son code par d'autres utilisateurs afin de vérifier qu'il est correct et sans bogue et lorsque le code est approuvé, il peut être fusionné à la branche master.

2.32

La branche source est Equipe40 et la branche destination est master.

3 : Révision de code

3.33

Nous avons décidé de faire la révision technique de **l'équipe 36**.

3.34

Il y a eu un commit à savoir : « compilation pour obtenir les fichiersobjets ».

3.35

Seul le fichier Makefile a été modifié.

3.36

Les modifications apportées au makefile :

Ligne 10 : moteur.o :moteur.cpp moteur.h

Ligne 11 : g++ -o\$@-c\$<

Ligne 12 :

Les modifications sont conformes à ce qu'il fallait faire; nous concluons que les modifications sont correctes.

3.37

Dans le répertoire TP3-A19 les fichiers suivants ont été créés :

Garage

Main.o

Mecanicienne.o

Moteur.o

Moteur_beepbeep.o

Moteur_putput.o

Piston.o

Voiture.o

L'exécutable garage et tous les fichiers objets ont bien été créés.

3.38

Toutes les modifications ajoutées sont correctes et conforme à ce qui est demandé; nous attribuons donc à **l'équipe 36** le score de **10** sur une échelle de dix (**10/10**).

3.39

Nous avons ajouté les commentaires demandés dans la discussion du Merge Request.

RÉFÉRENCES

[1] : <https://fr.wikipedia.org/wiki/Gerrit>

[2] : <https://www.01net.com/telecharger/linux/Programmation/fiches/143174.html>

