

Rapport Travail pratique 1: GIT et Makefiles.

École polytechnique de Montréal

**Trimestre :** automne 2018

Équipier1: CYRILLE TALLA FONGANG

Équipier2: GEORGES PARFAIT *DJIMEFO KAPEN*

**Cours: LOG1000**

**Équipe: 04**

**Présenté à :**

Polytechnique Montréal

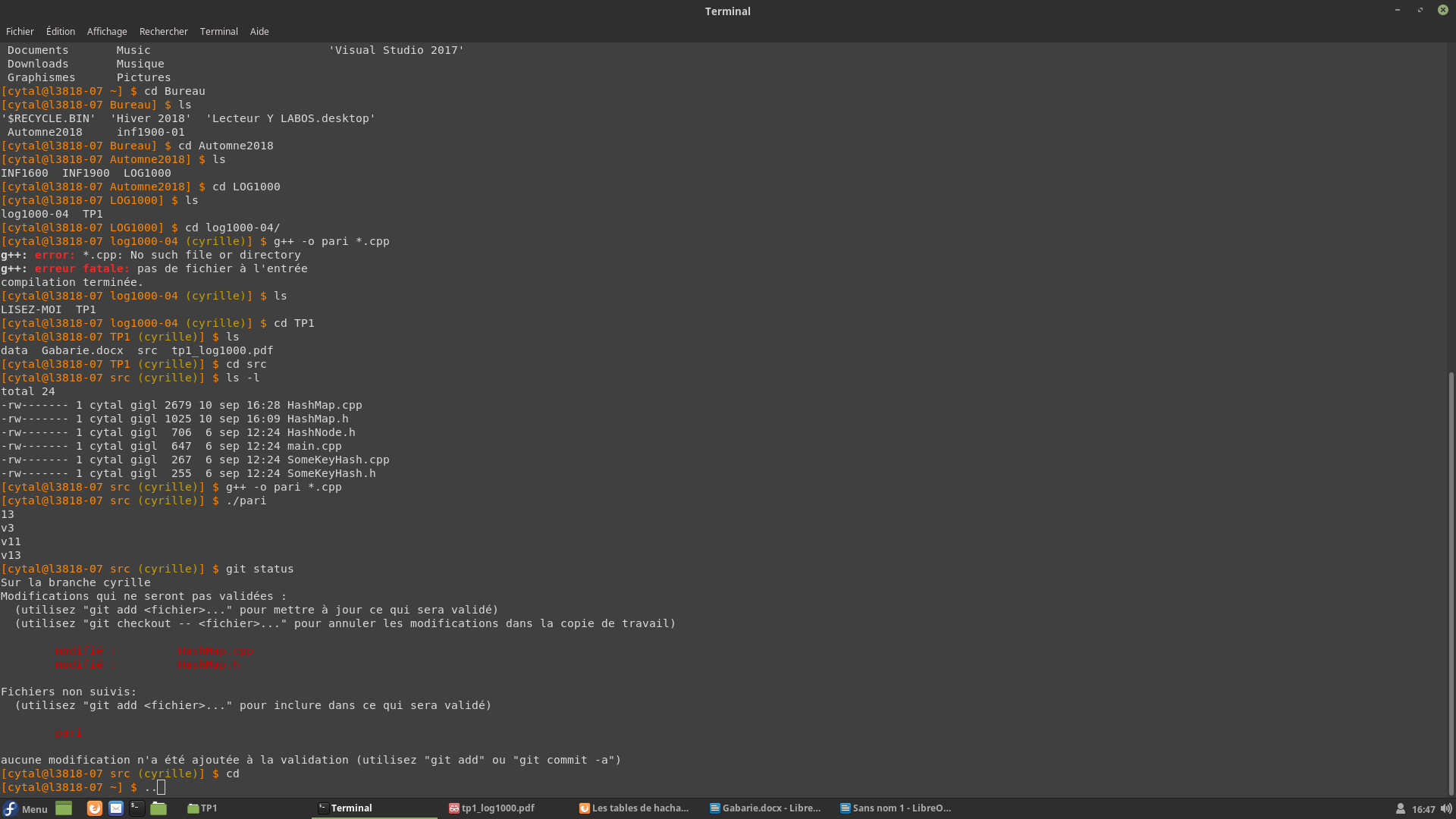
date de remise(14-09-2018)

Travail pratique #1: GIT et Makefiles

Partie 1 : GIT

E1.1 Mise en place du répertoire GIT

a) capture d’écran de la commande de la commande « git log »

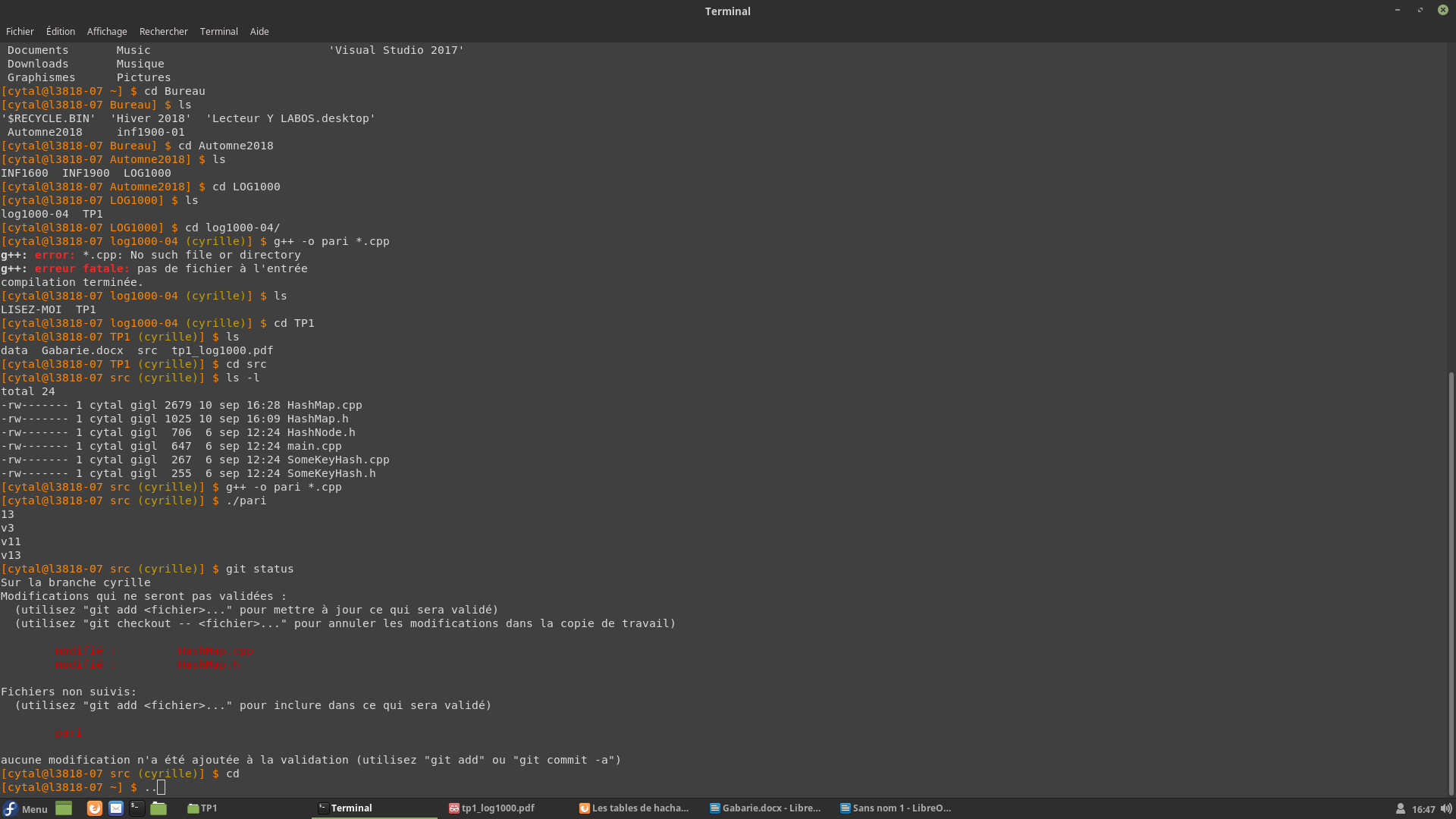


b) Différence entre « git log » et « git log -p »

La commande « git *log* » permet d’obtenir en ordre chronologique les commits effectués. Tandis que la commande « git *log - p*» permet d’avoir le détail des lignes qui ont été ajoutées et retirées dans chaque commit. (Réf. https://openclassroom.com/fr/courses/1233741-gerez-vos-codes-sources-avec-git),

E1.2

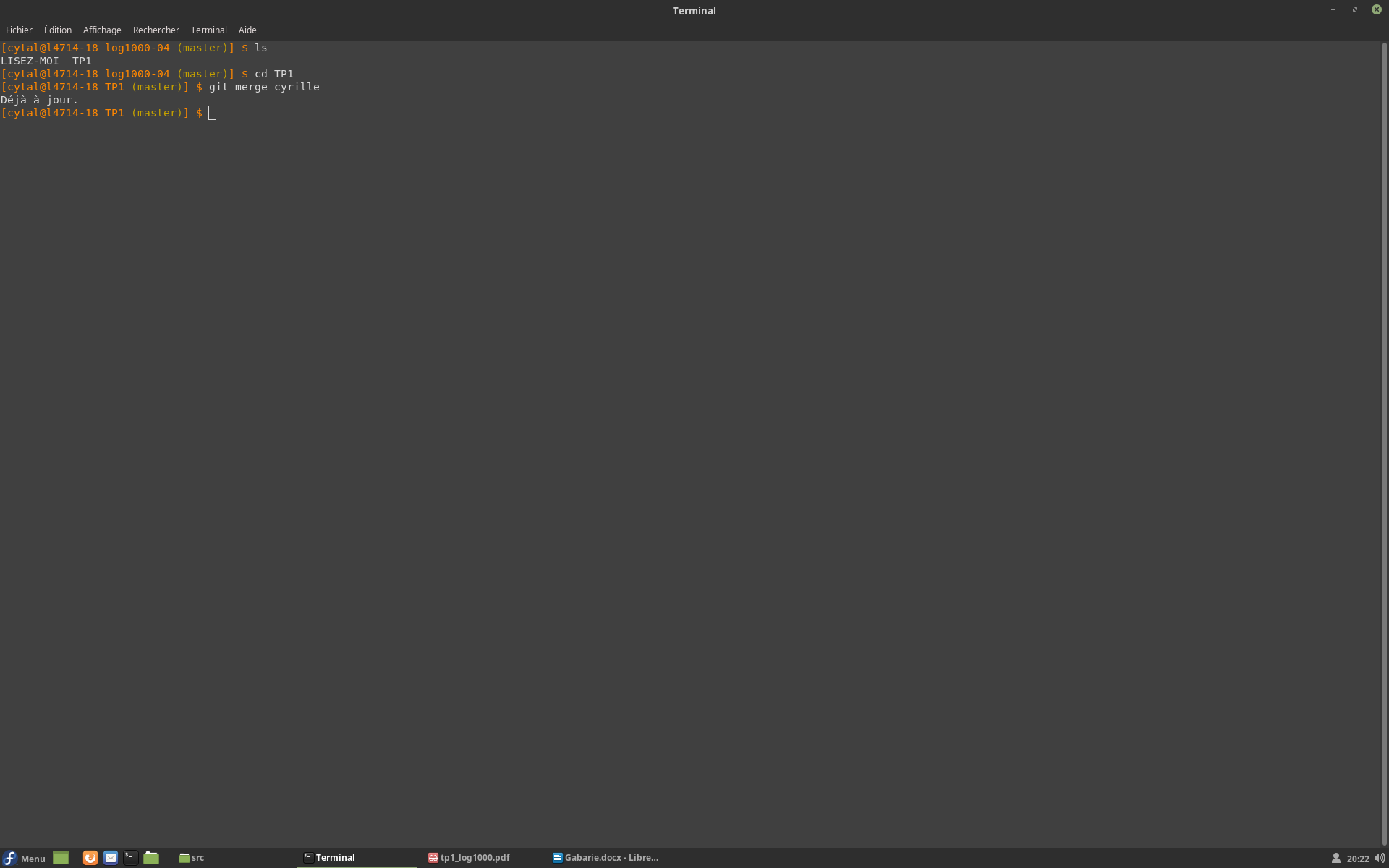
3-a) Écran de sortie



3-b)

Cette situation est normale, En effet, les modifications apportées aux fichiers  « HashMap.h » et « HashMap.cpp » ne sont effectives que dans le working directory en attendant d’être ajoutées dans le staging area (avec la commande git add) et soumises dans le répertoire local (avec la commande git commit).

5-a)



5-b)

Git ne détecte aucun conflit pour la simple raison que nous avons travaille sur 2 branches différentes et 2 fichiers distincts avant de faire un push vers le serveur.

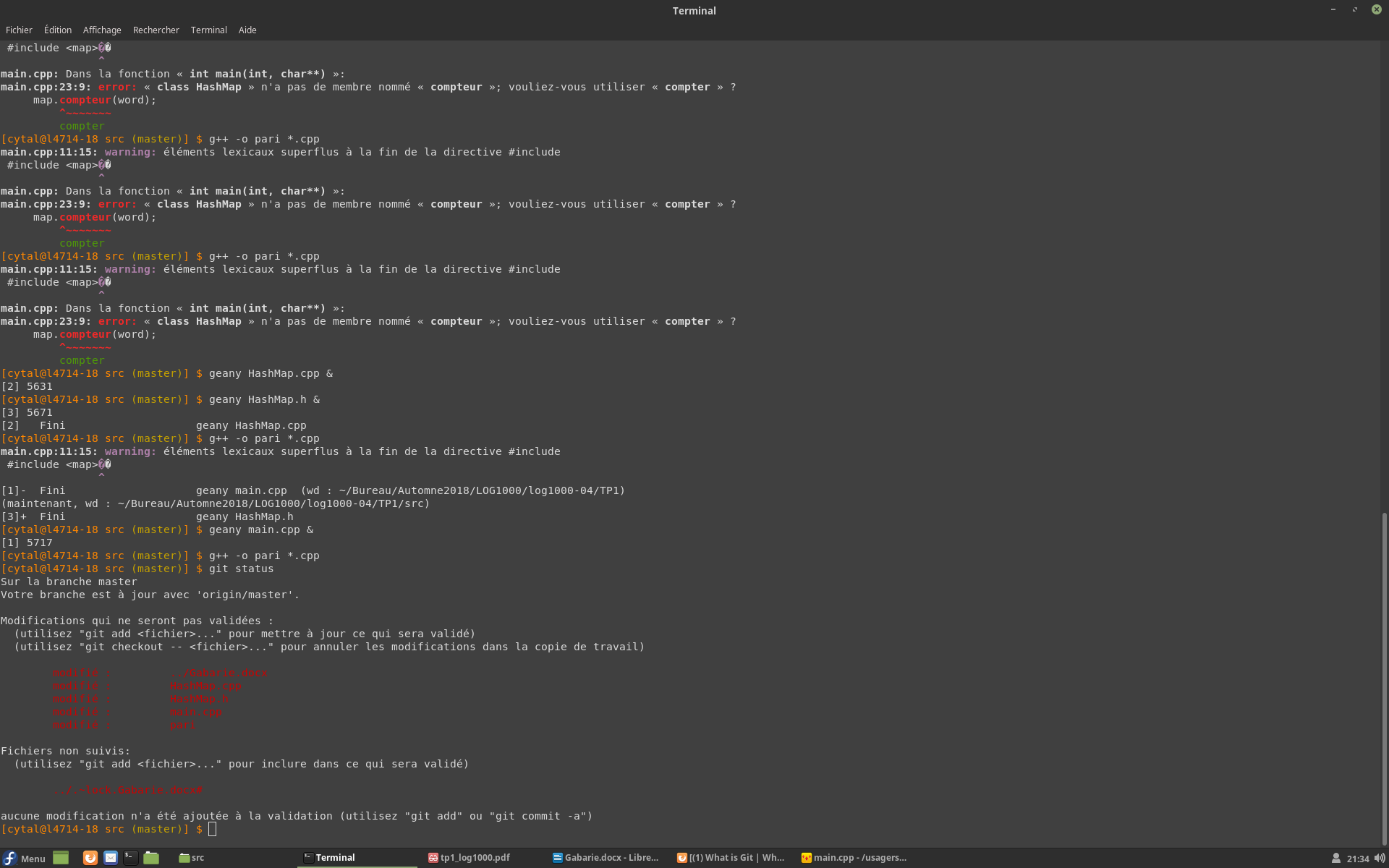
Copie du log: *Insérer la copie de log demandée ici*

E1.3

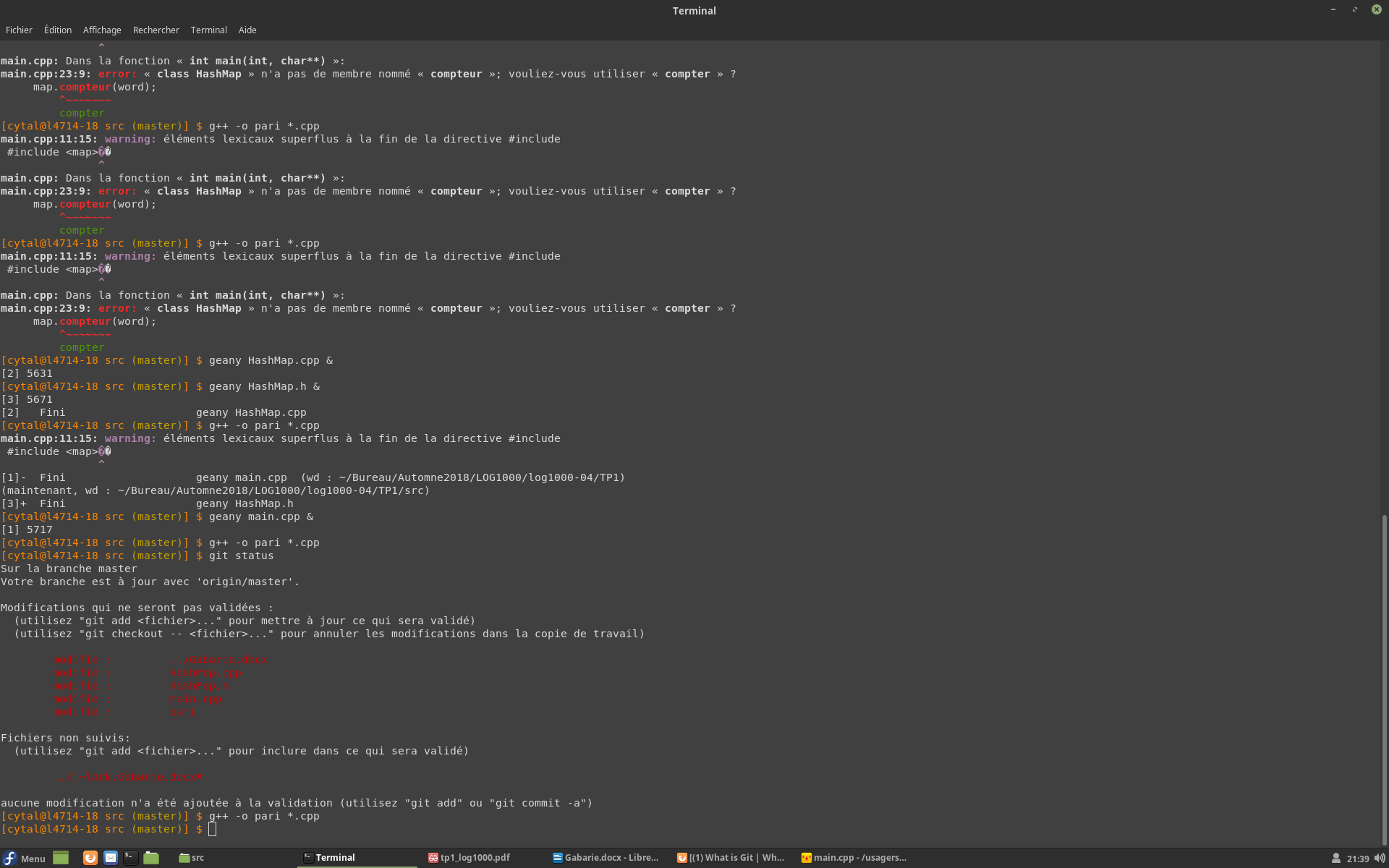
3-a)

Les noms de l’exécutable et des fichiers main, HashMap.h et HashMap.cpp sont écrits en rouge sur la console car il ont été *modifies.*

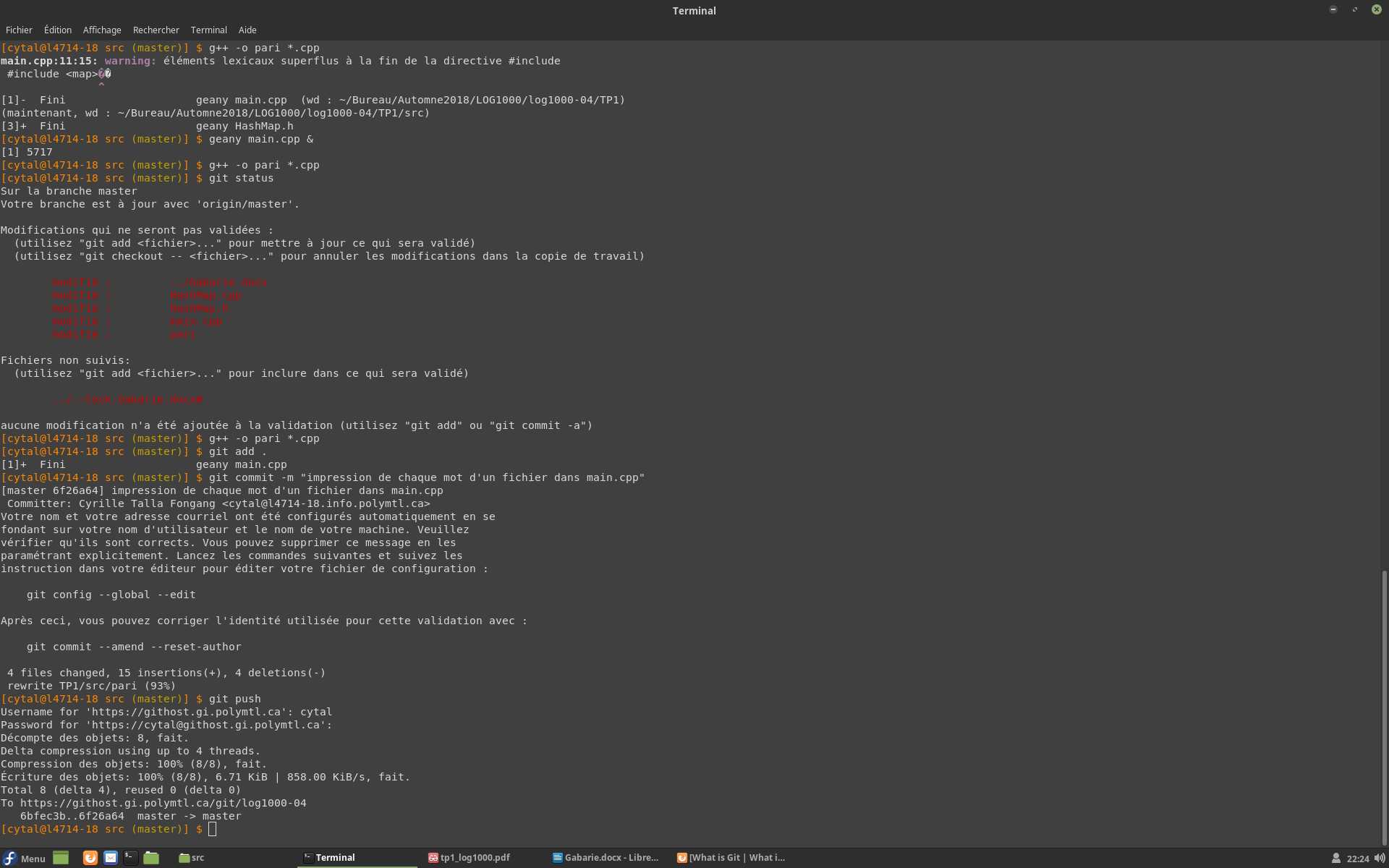
3-b)



3-c)



commentaire



5-a)

5-b)

5-c)

5-d)

Copie du log: *Insérer la copie de log demandée ici*

E1.4

5-a)

5-b)

5-c)

6)

7) BONUS *(si vous décidez de répondre à cette question)*

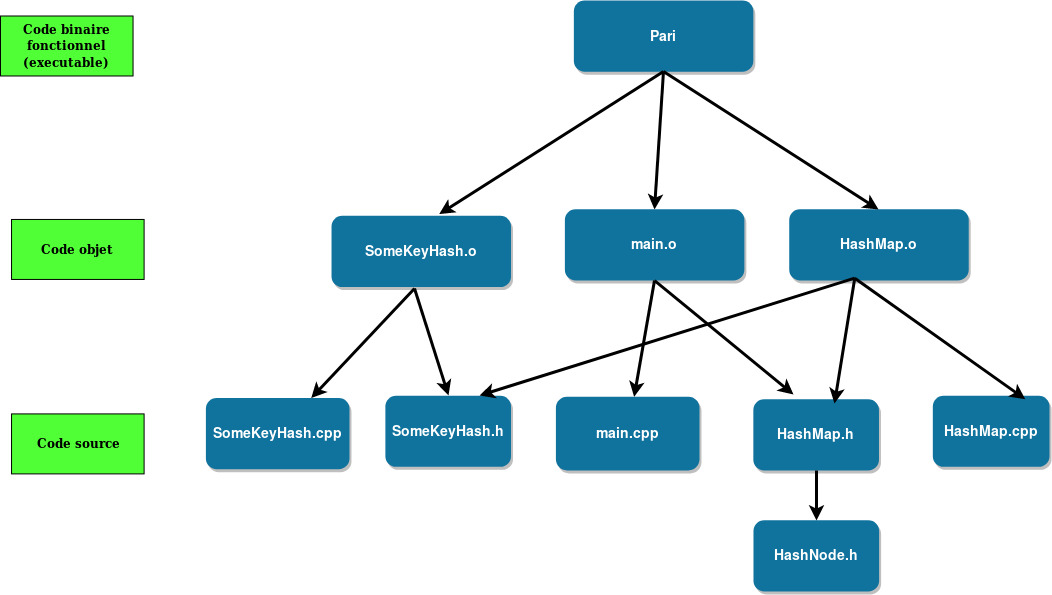
8)

Dans le développement d'un projet, il arrive souvent qu'on souhaite travailler sur une fonctionnalité en limitant les interactions avec le reste de l'équipe. Dans ce genre de situation les branches s ‘avèrent utiles. (https://ensiwiki.ensimag.fr/index.php?title=G%C3%A9rer\_des\_branches\_avec\_Git)

Copie du log:

E2.1

a) *Insérez votre graphe de dépendances ici* E2.2 *Exécutez* «make»*, puis insérez la sortie des deux commandes* «touch ... make»



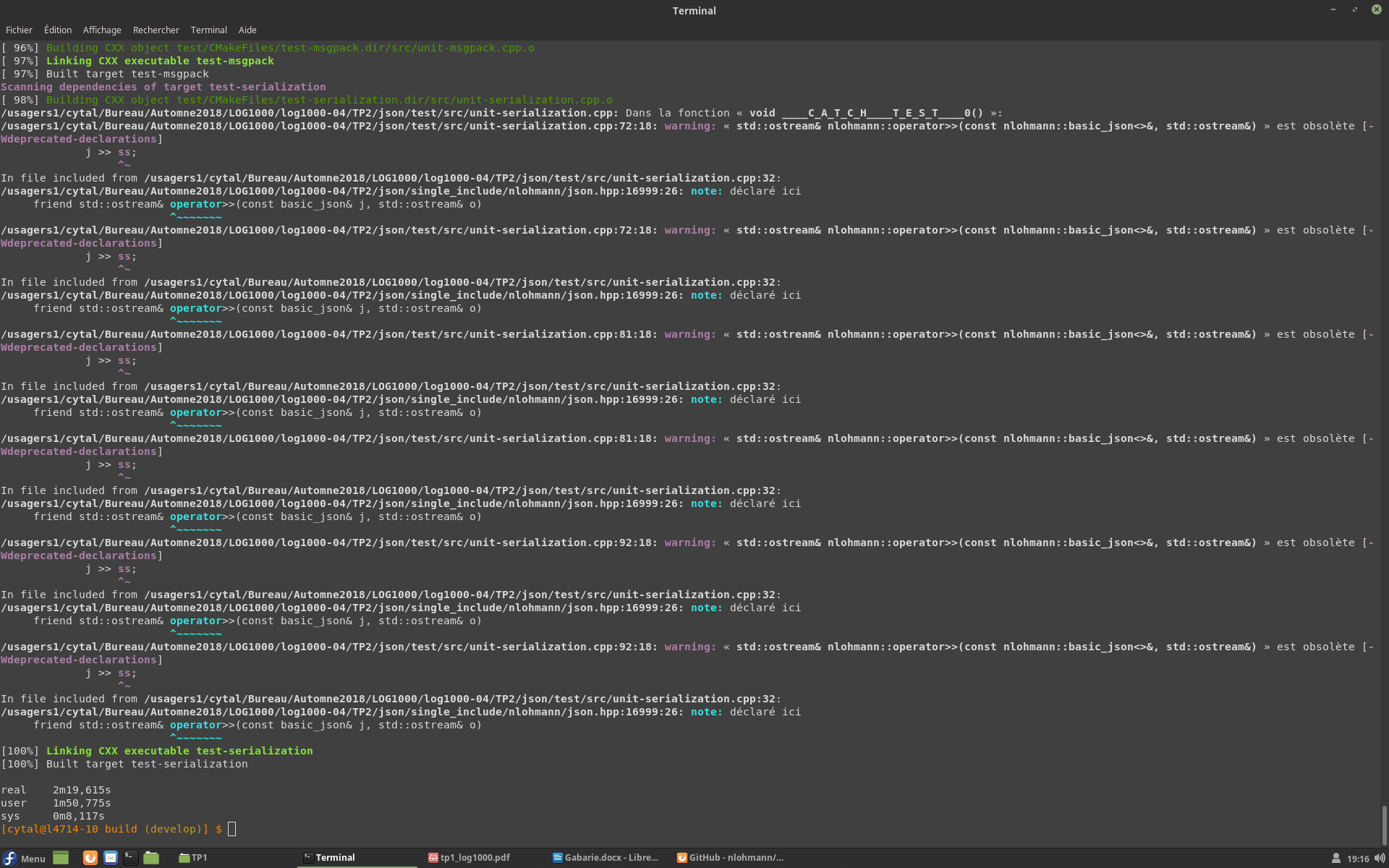
E2.2

4)

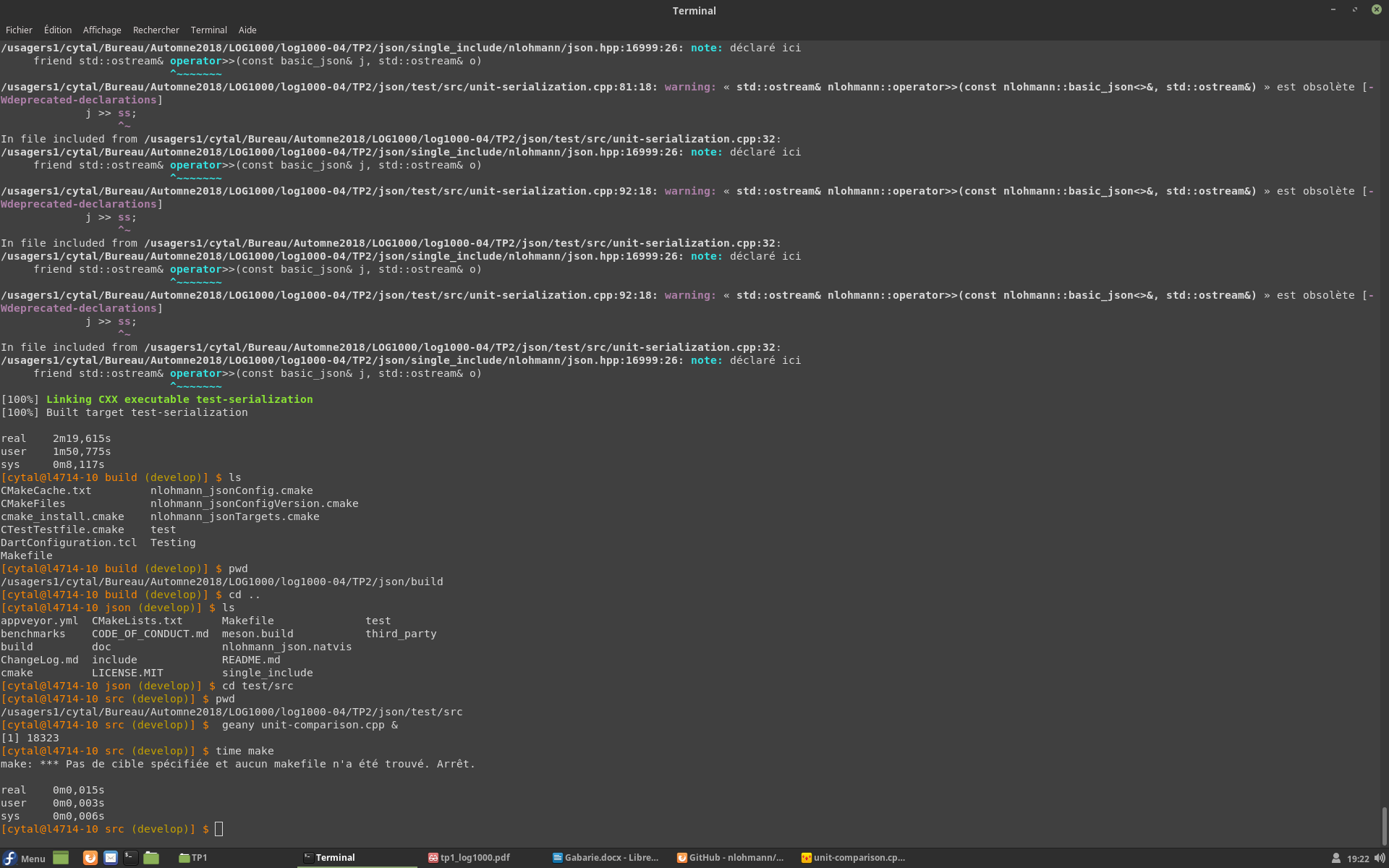
*Partie 2 :*

*2)*

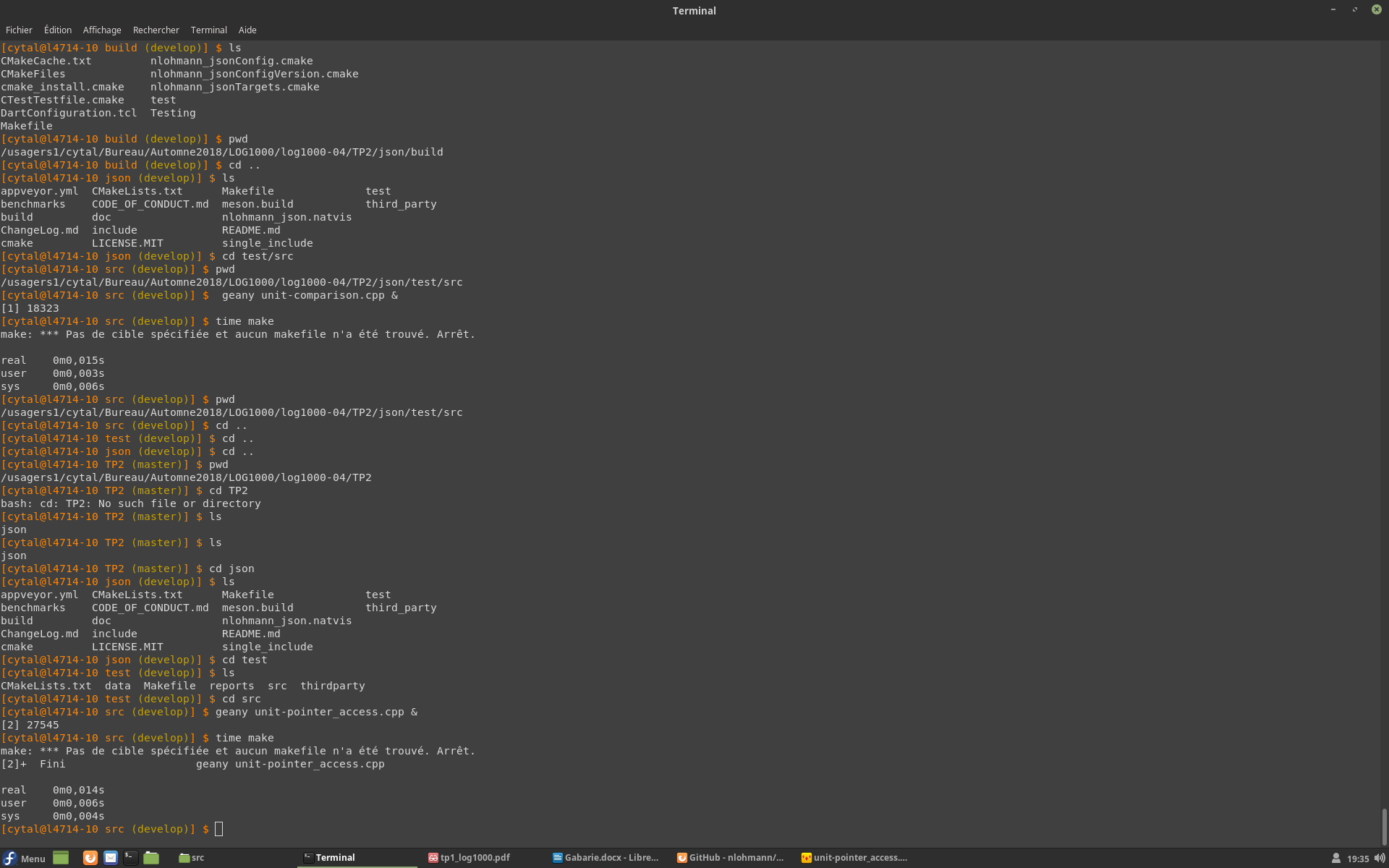
*a)*

**

*b)*

**

*c)*

**

*d)*

*Il y a une différence de temps entre les trois exécutions de la commande. En effet, la première commande a exécuté le programme avec plus de temps car il s’ agissait de la première fois qu’il était compile. Lors des autres commandes, il ne compile que les parties modifiées du programme et donc met moins de temps a le faire.*

Pondération du travail pratique

Partie 1 :

Pour E1.1 à E1.4, la qualité de vos messages pour les «commits»: [ /3]

Pour E1.1 à E1.4, la qualité du code source: [ /2]

E1.1 [ /3]

a) [ /2]

b) [ /1]

E1.2 [ /10]

2-a) [ /3]

2-b) [ /1]

5-a) [ /3]

5-b) [ /2]

Copie du log: [ /1]

E1.3 [ /12]

3-a) [ /2]

3-b) [ /1]

3-c) [ /1]

5-a) [ /2]

5-b) [ /2]

5-c) [ /2]

5-d) [ /1]

Copie du log: [ /1]

E1.4 [ /10]

5-a) [ /2]

5-b) [ /2]

5-c) [ /2]

* + 1. BONUS [/1]
    2. [ /3]

E2.1 [ /5]

*Insérez votre graphe de dépendances ici* [ /5]

E2.2 [ /10]

Fichier Makefile [ /7]

*Exécutez* «make»*, puis insérez la sortie des deux commandes* «touch ... make» [ /3]

**Partie** 2 : [ /5]

1-a)  [ /2]

1-b)  [ /1]

1-c)  [ /1]

1-d)  [ /1]

\*vous devez insérer une **capture d'écran** de votre travail remis a l'aide de la commande «git *ls-files https://githost.gi.polymtl.ca/git/log1000-XX*»

**TOTAL: [ /60]**