

Travail pratique #(1): Titre(ex :GIT et Makefiles)

École polytechnique de Montréal

**Trimestre :** automne 2018

Équipier1: *Prénom Nom*

Équipier2: *Prénom Nom*

**Équipe:**

**Présenté à :**

Polytechnique Montréal

date de remise(jj-mm-aaaa)

Travail pratique #1: GIT et Makefiles

Partie 1 :

E1.1

a)

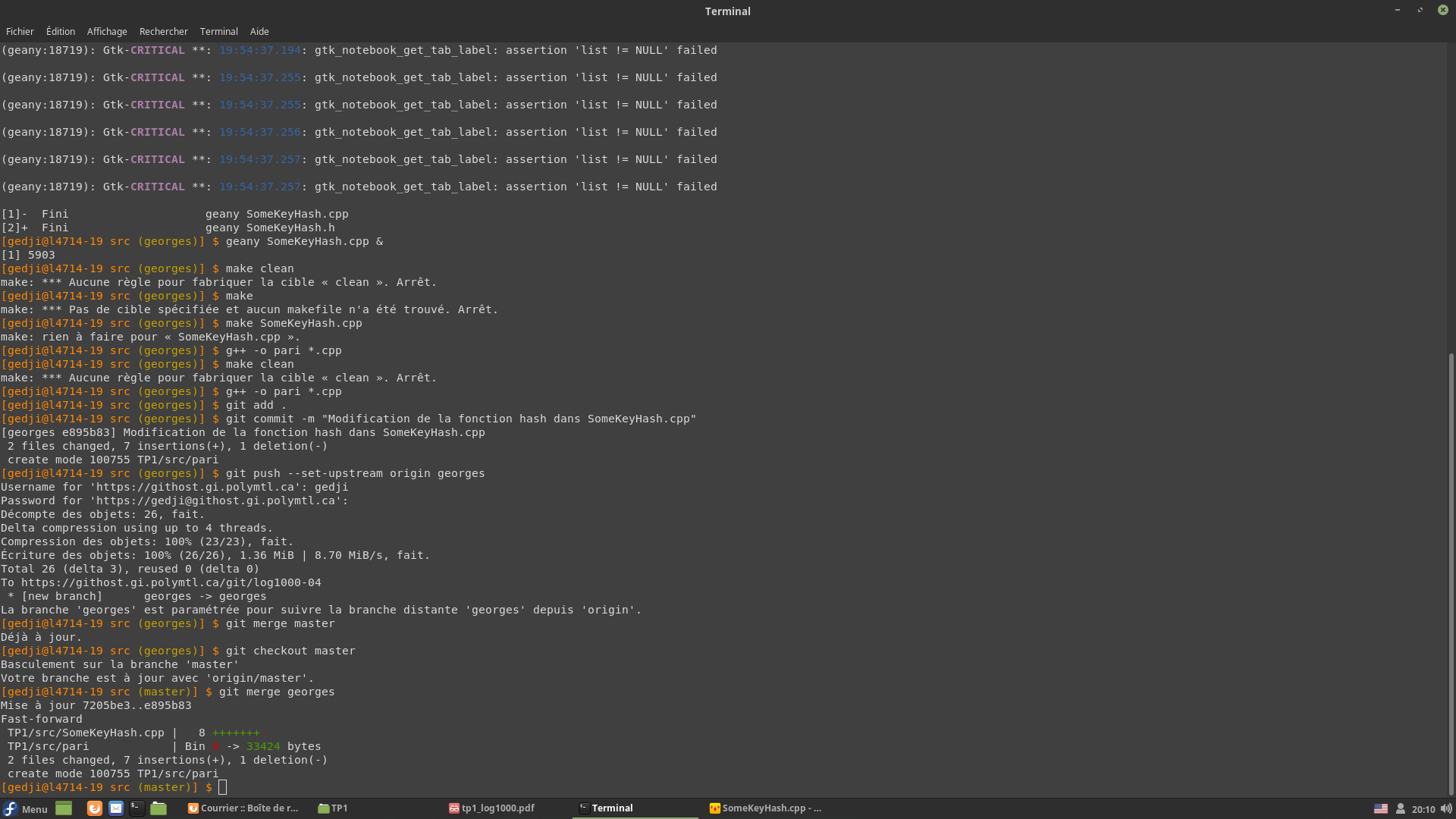
b)

E1.2

3-a)

3-b)

5-a)



5-b)

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

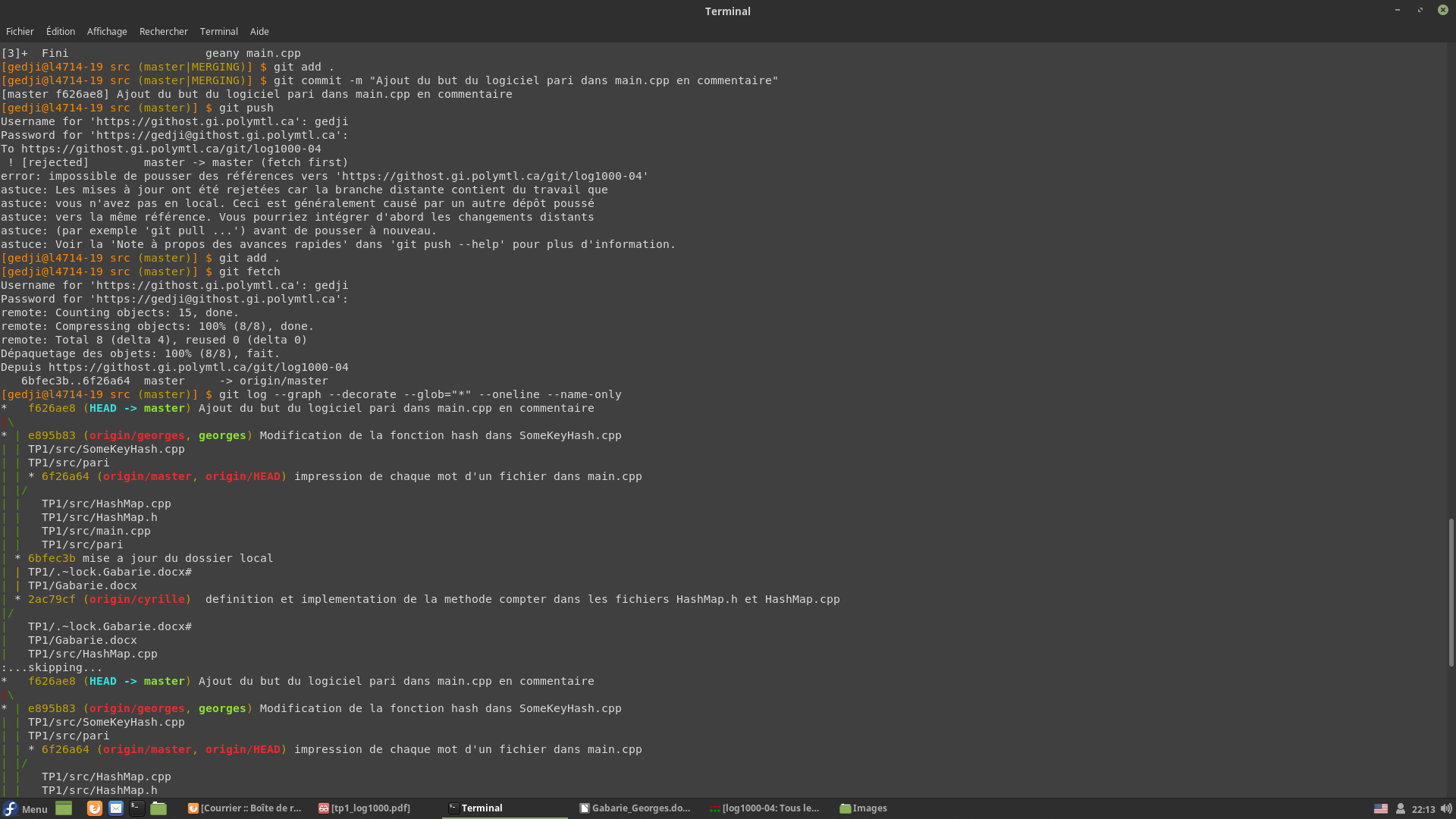
E1.3

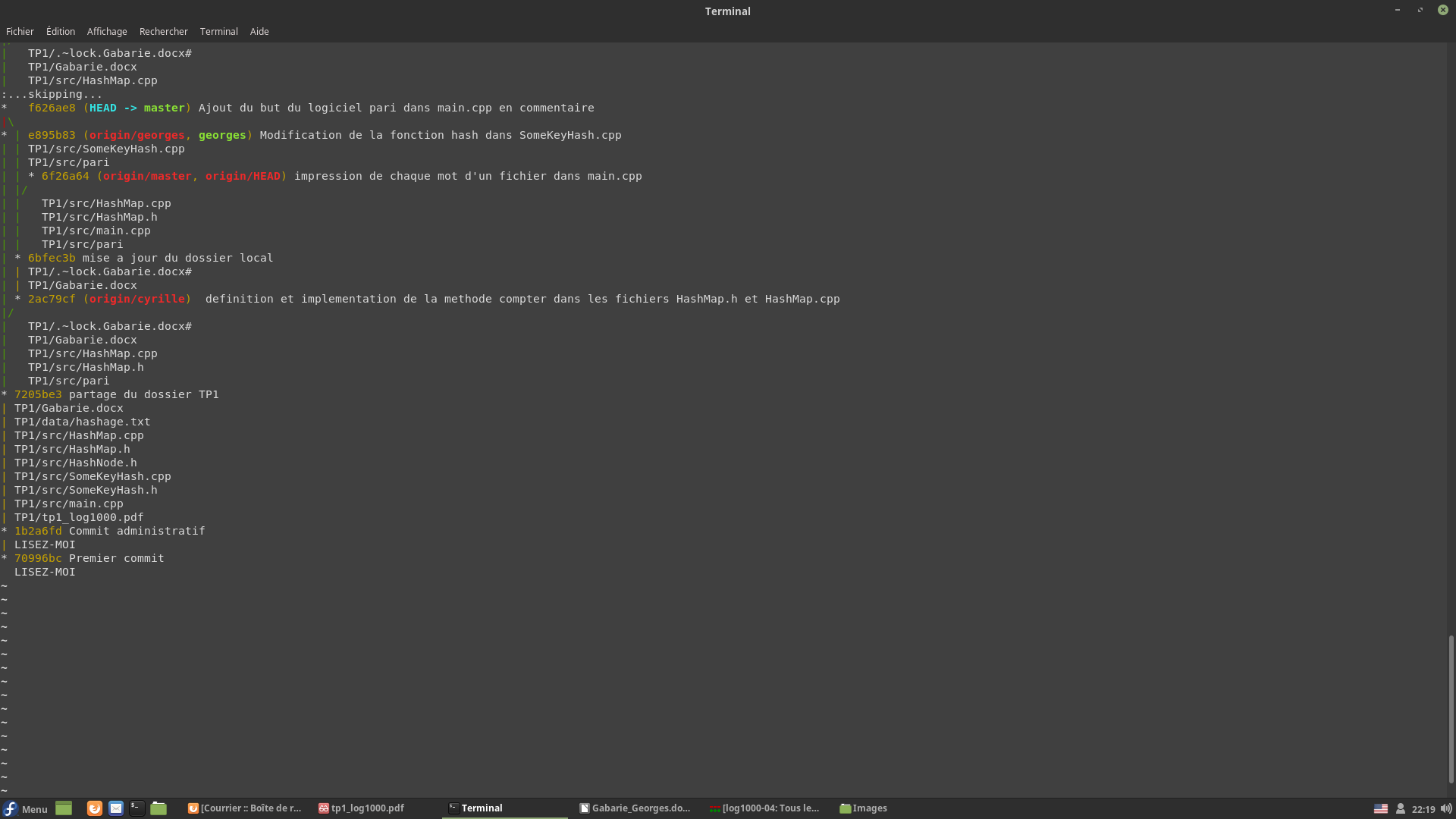
3-a)

3-b)

3-c)

5-a)

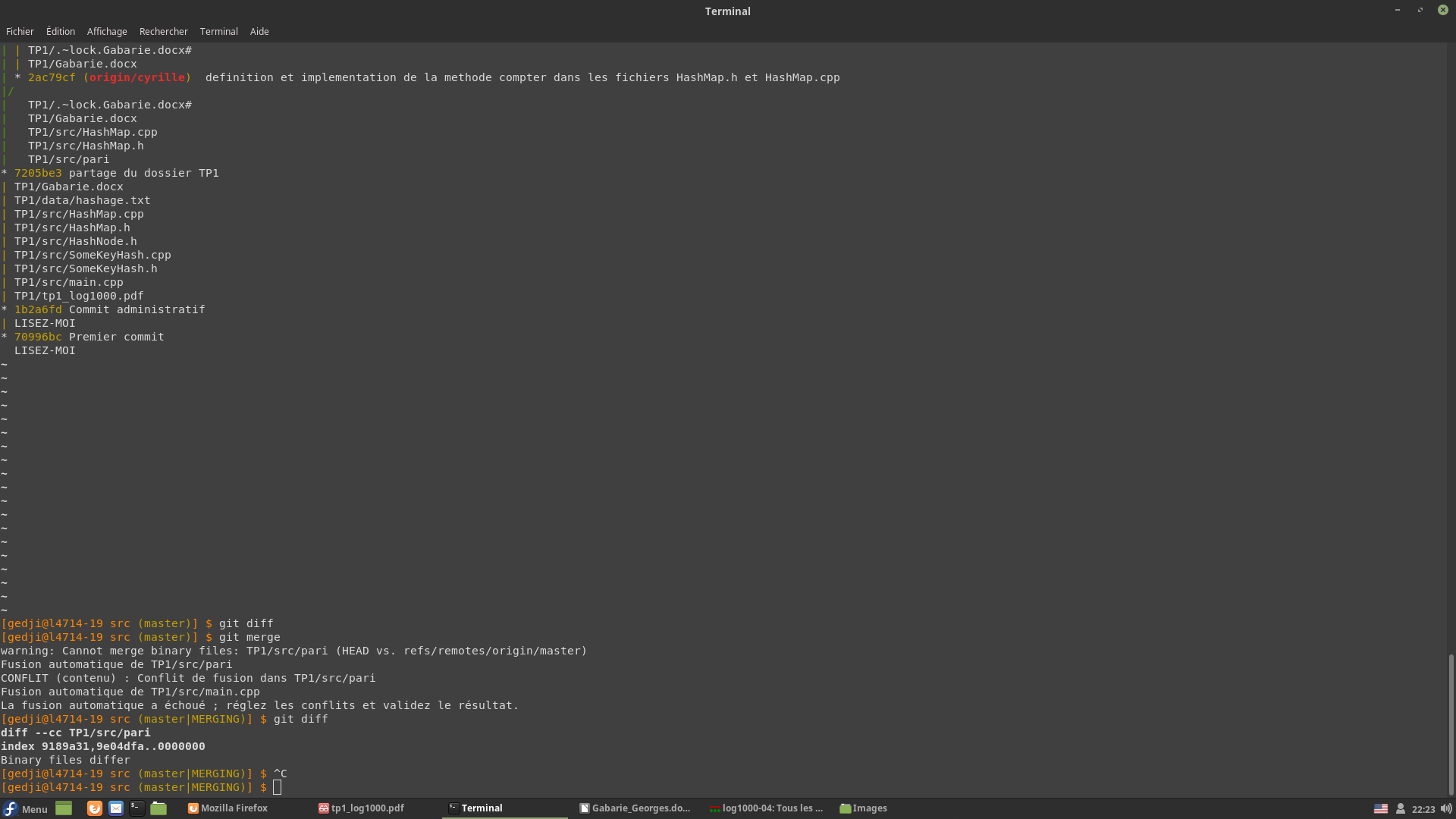




5-b)

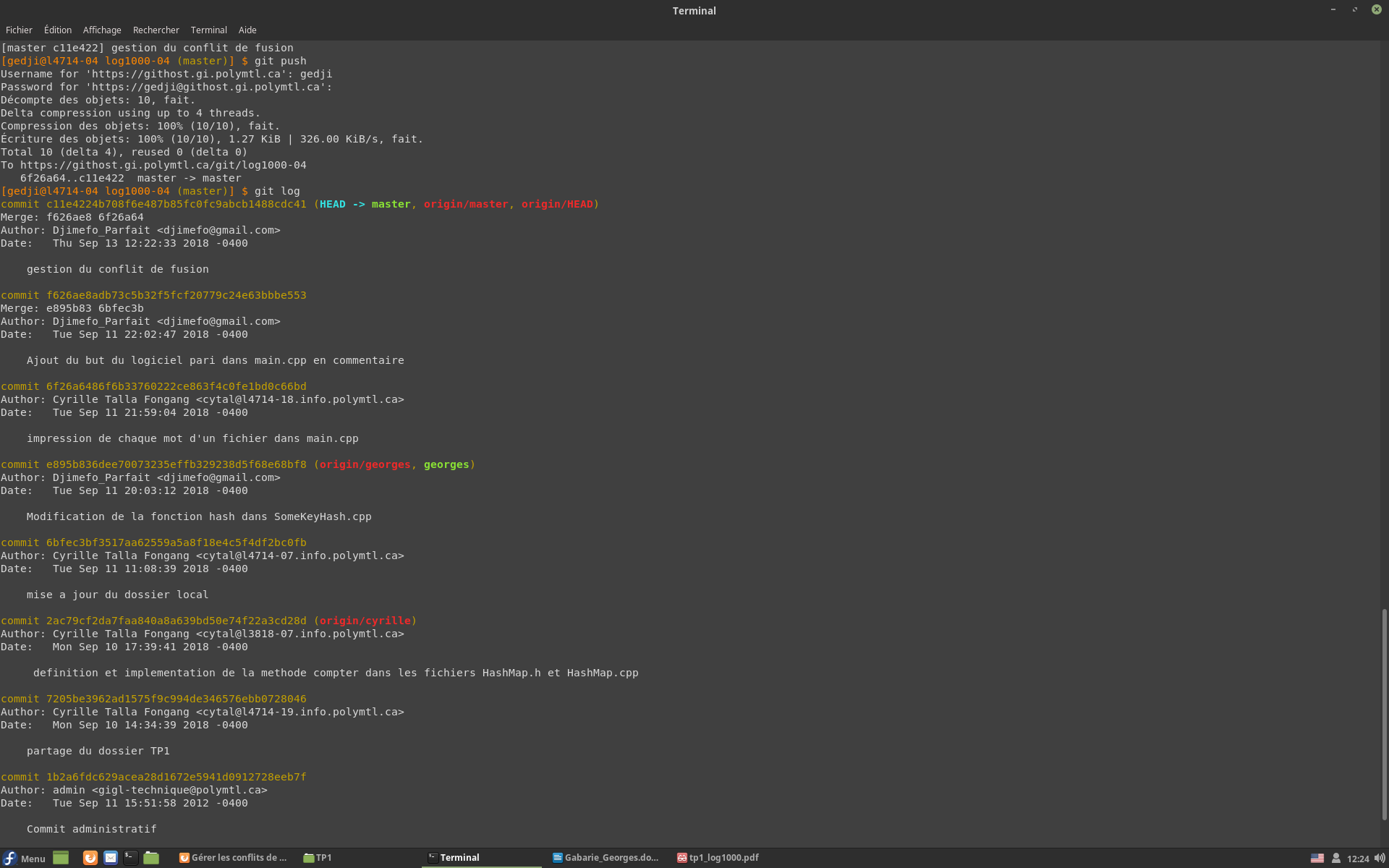
Oui il y aura un conflit du fait que la version de l’exécutable pari présent sur le serveur tient compte des modifications faites par l’équipier 1 (changement du comportement de la fonction main qui imprime maintenant chaque mot du fichier) tandis que celle qui va être poussée par l’équipier 2 n’inclut pas cette modification (puisque dernier n’a ajouté qu’un commentaire),

5-c)



5-d)

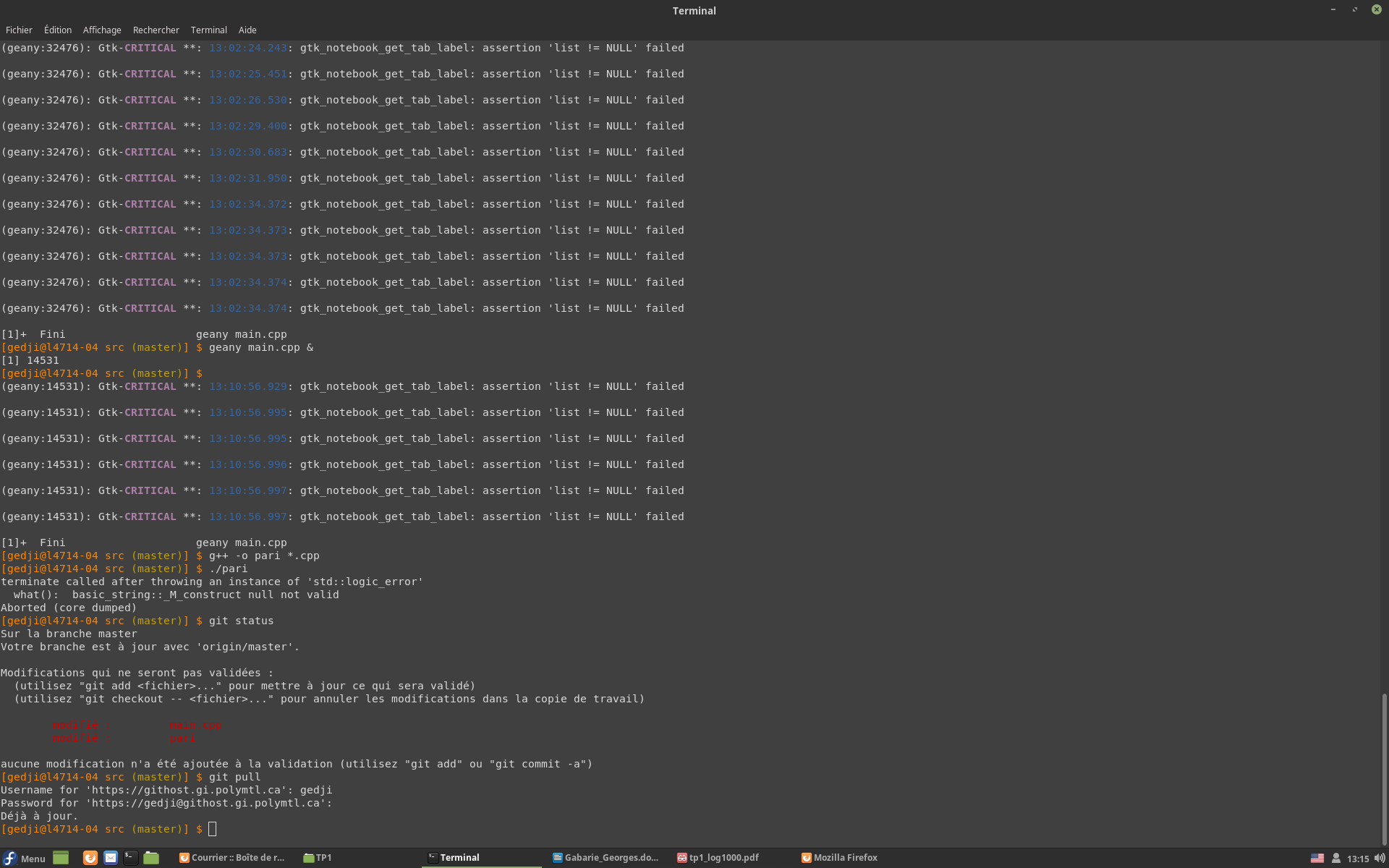
Copie du log: *Insérer la copie de log demandée ici*

**

La capture d’écran de la sortie du LOG au E1.2 ne nous avait pas été demande dans le document de référence, raison pour laquelle il est impossible de faire la comparaison demandée.

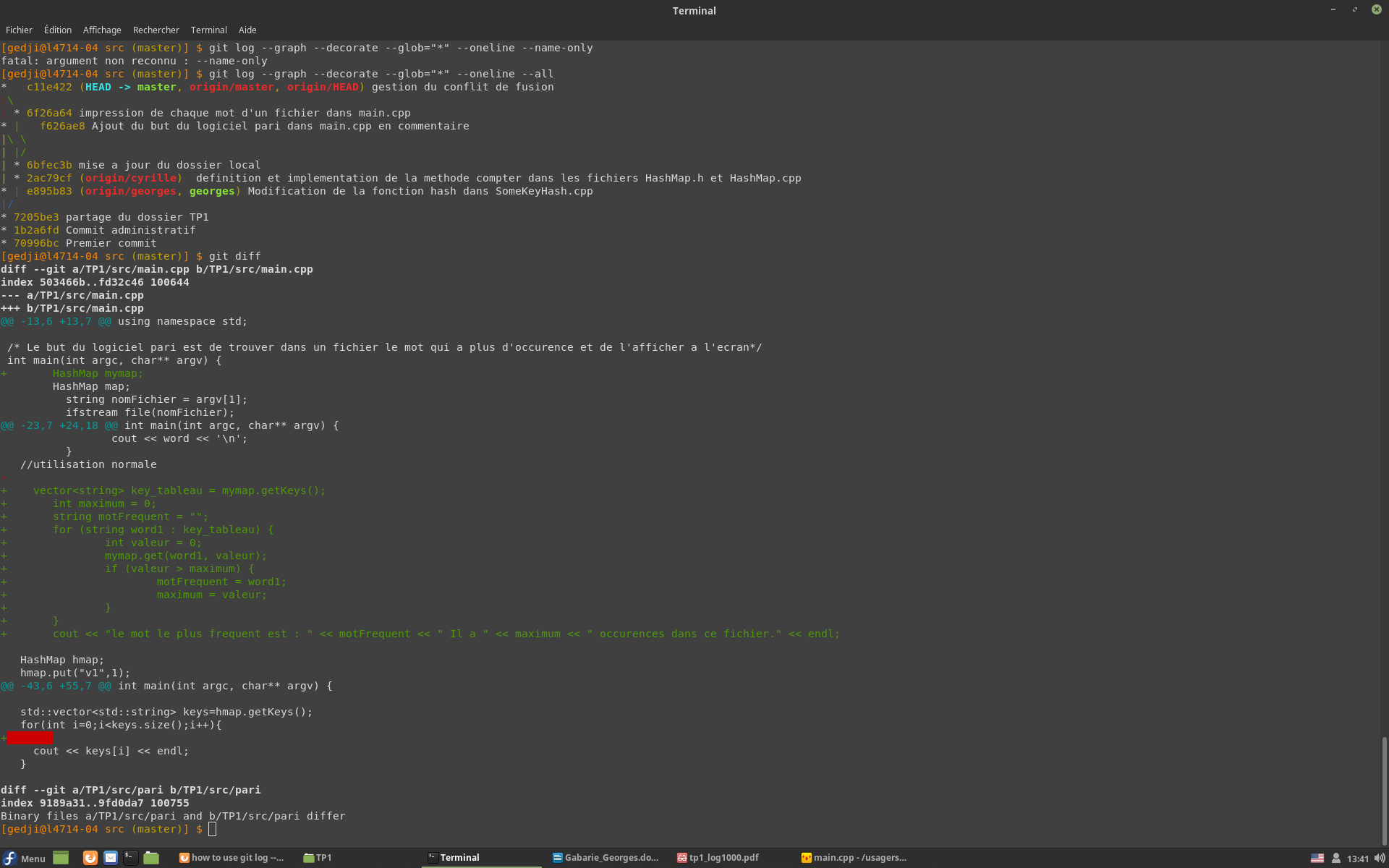
E1.4

5-a)

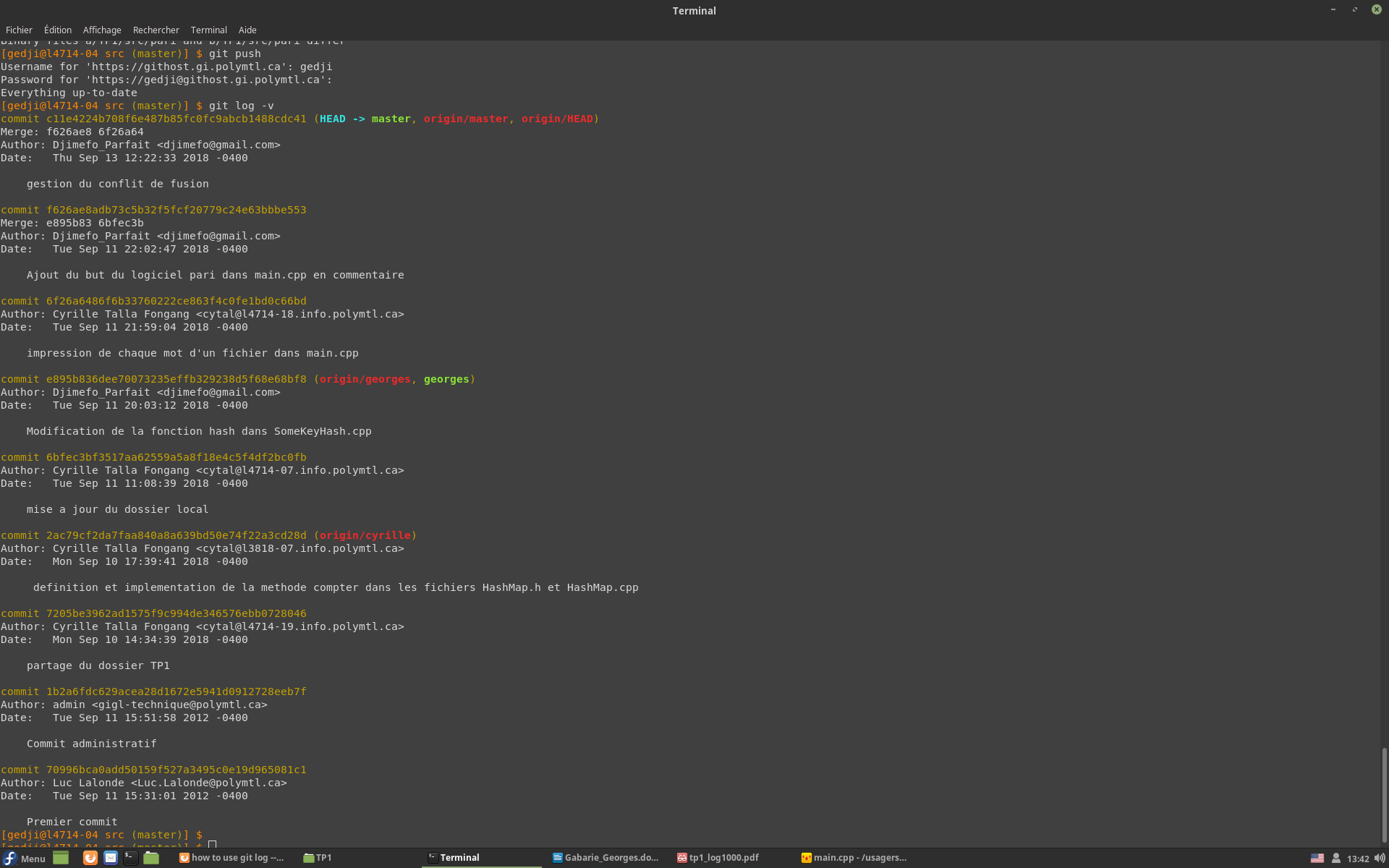


5-b)

Non il n’a pas détecté de conflit car les modifications faites dans le répertoire local de l’équipier 2 n’ont pas été validées (des lors qu’aucune commande git add, git commit n’a été exécutée depuis le dernier git pull).



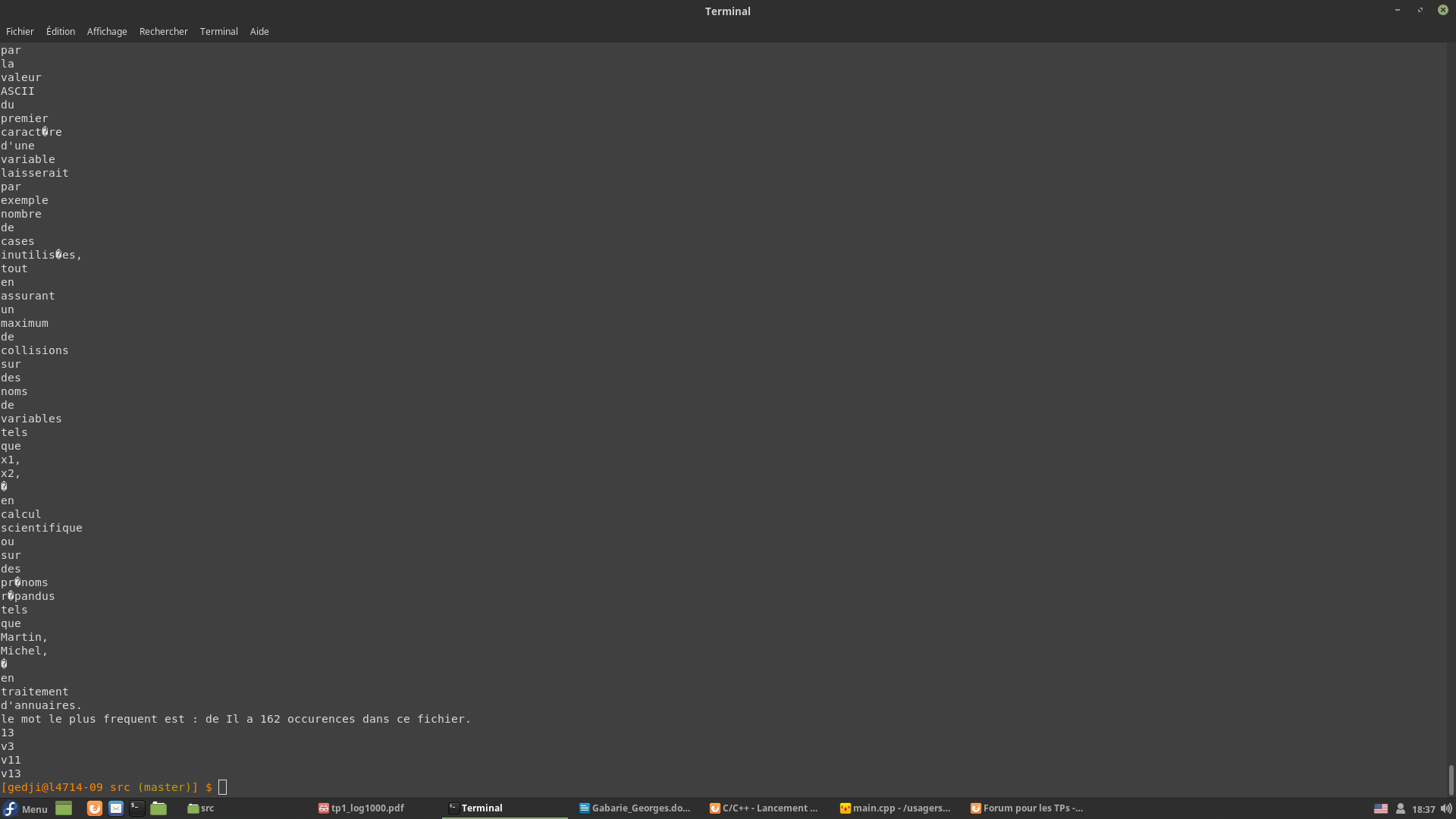
*5-c) Dans le cas où vous rencontrez une situation conflictuelle, comment pensez- vous régler cette dernière si on veut que le logiciel résultant réussit à résoudre le pari avec l’étudiant ? Est-ce que ça peut être fait automatiquement ? Pourquoi (pas) ? Si oui, résolvez le conflit.*



6)

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

*Le mot le plus populaire est «de» avec une frequence de 162 tel que montre la sortie d’ecran ci dessous :*

**

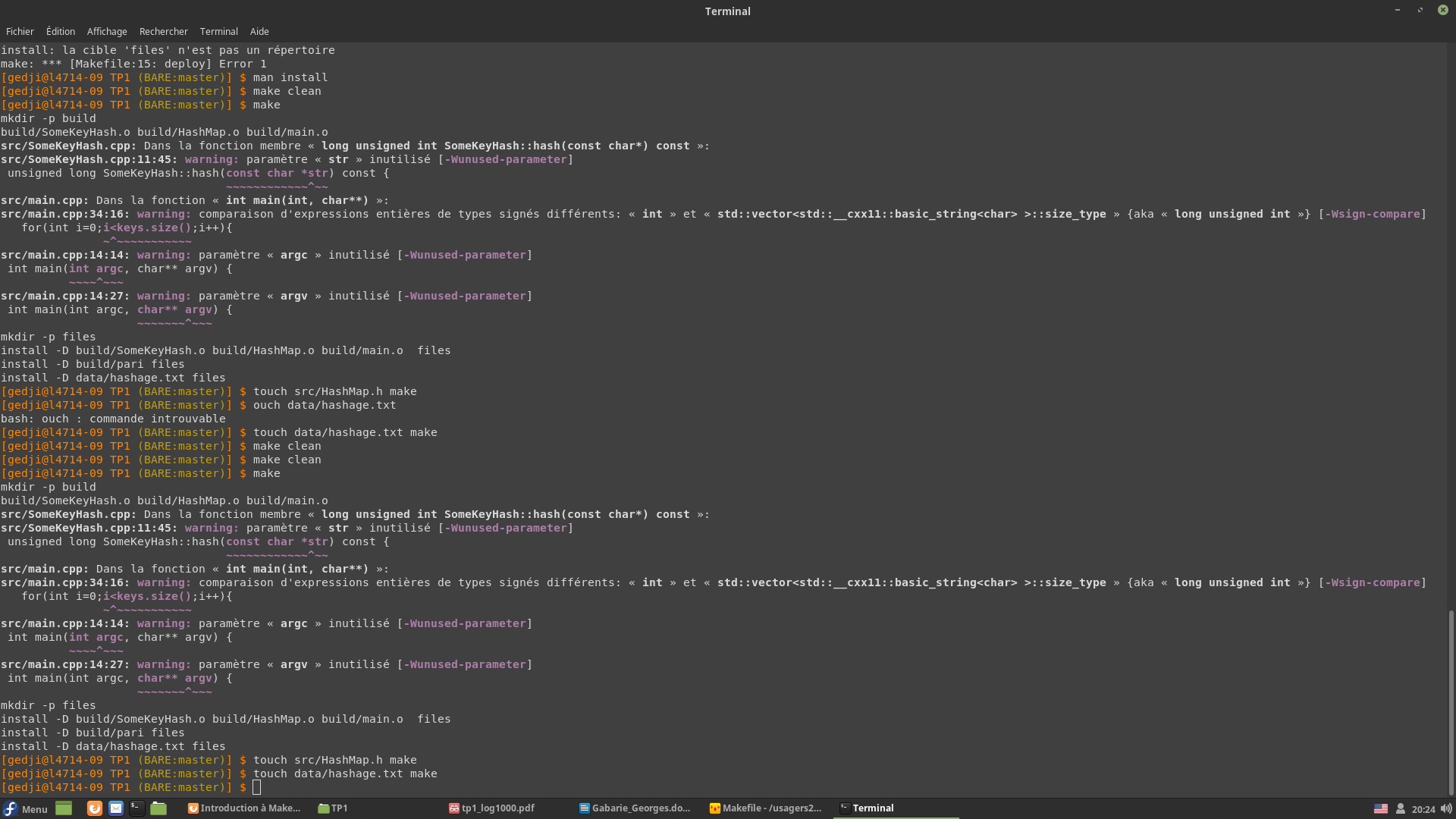
8)

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)*

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]**