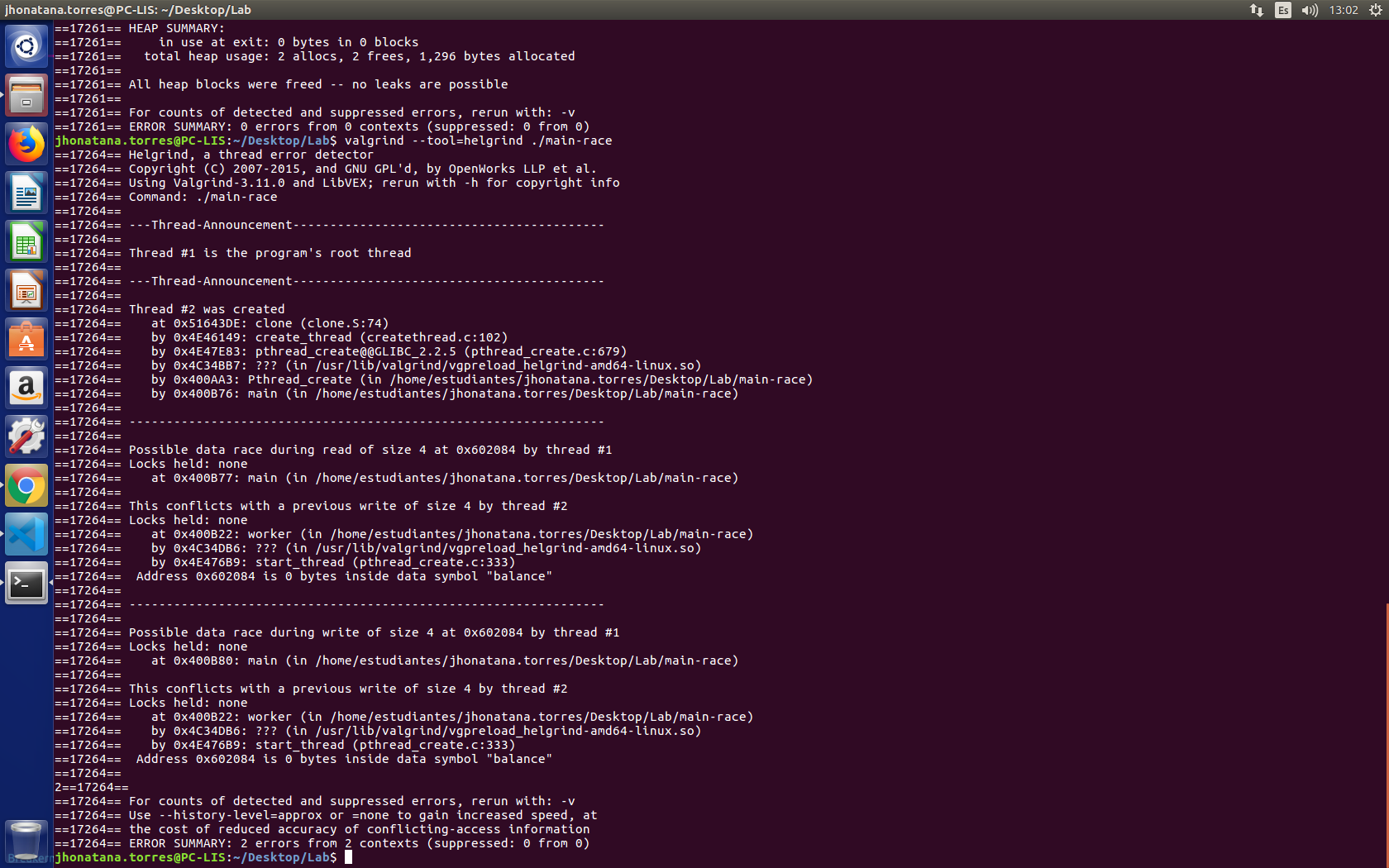
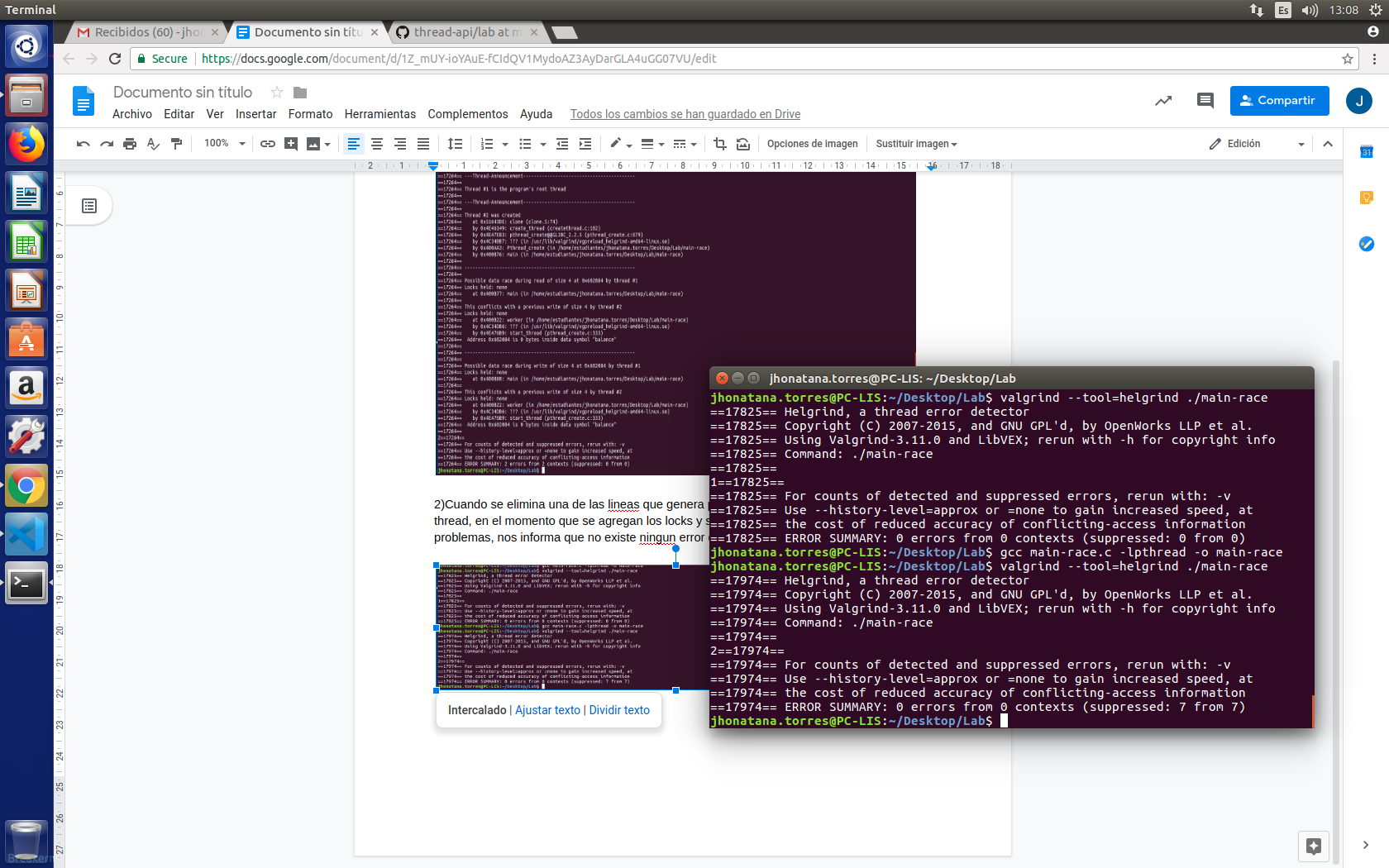
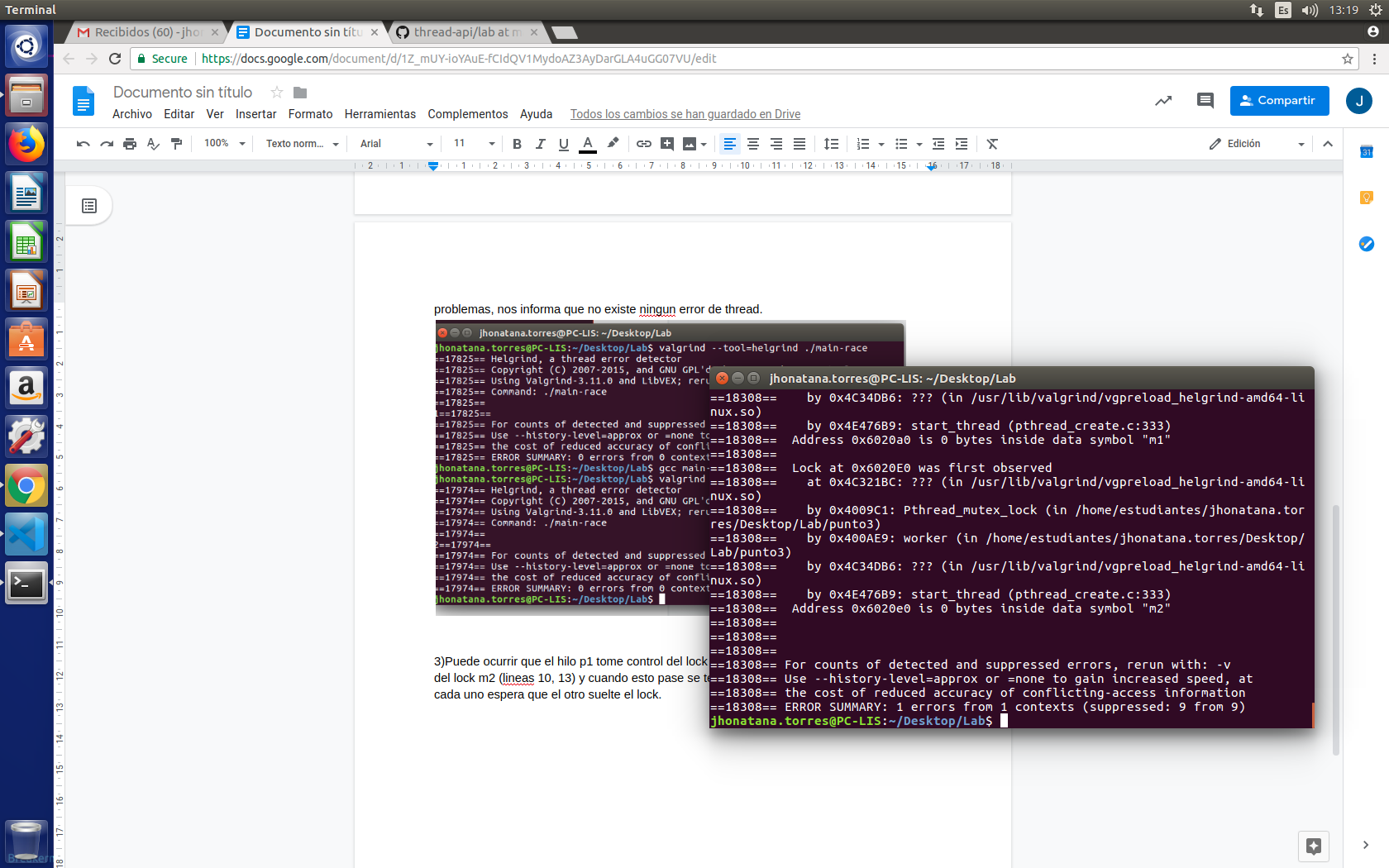
1) Se puede ver la variable que podria generar el data race, ademas entrega los hilos que esta generando este data race, ademas de una descripcion del hilo que se creo.



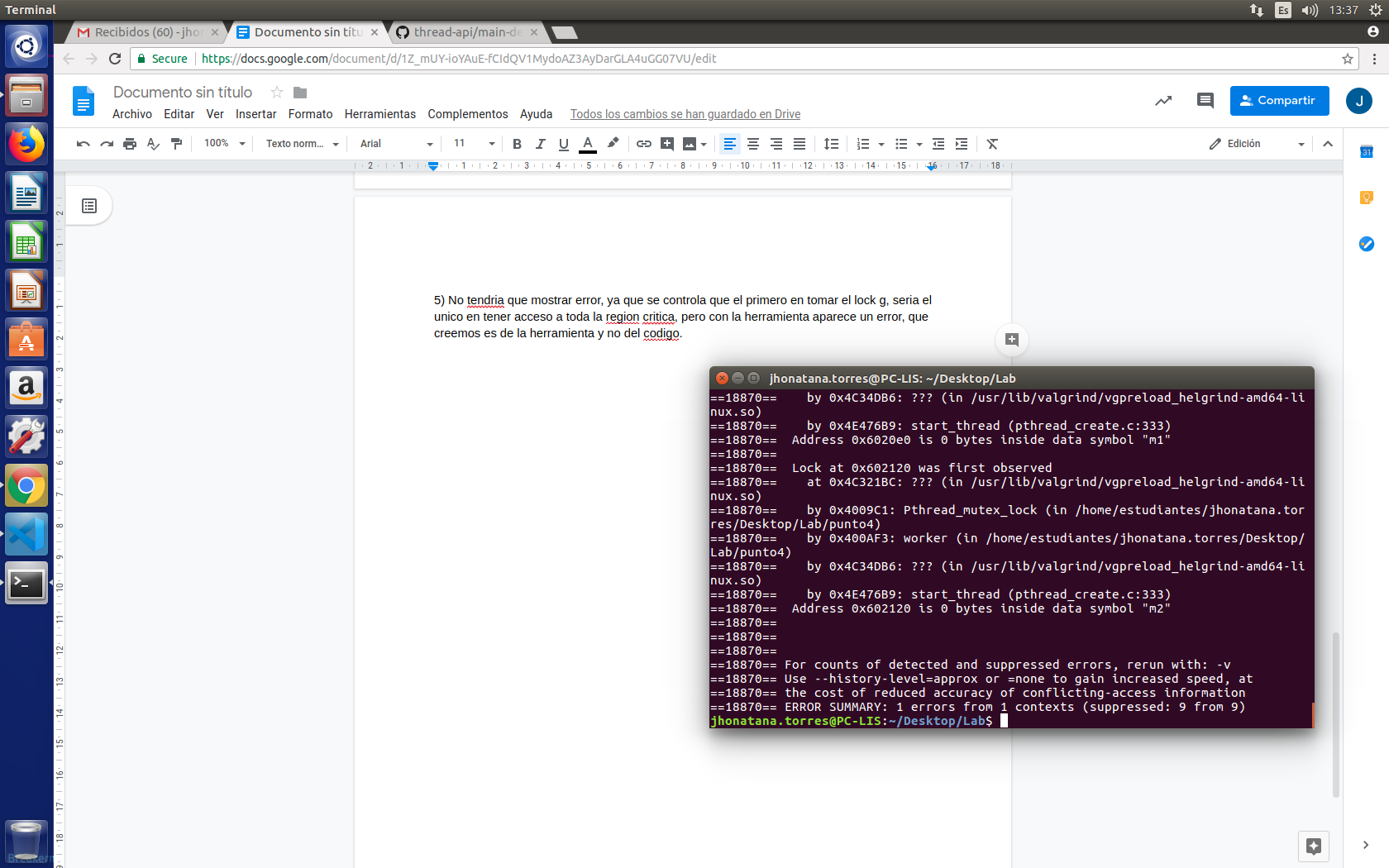
2)Cuando se elimina una de las lineas que genera problemas, no muestra ningun error de thread, en el momento que se agregan los locks y se deja la linea que podria generar problemas, nos informa que no existe ningun error de thread.

3) Puede ocurrir que el hilo p1 tome control del lock m1 y a su vez el hilo p2 toma el control del lock m2 (lineas 10, 13) y cuando esto pase se tendra un problema del filosofo en el que cada uno espera que el otro suelte el lock.

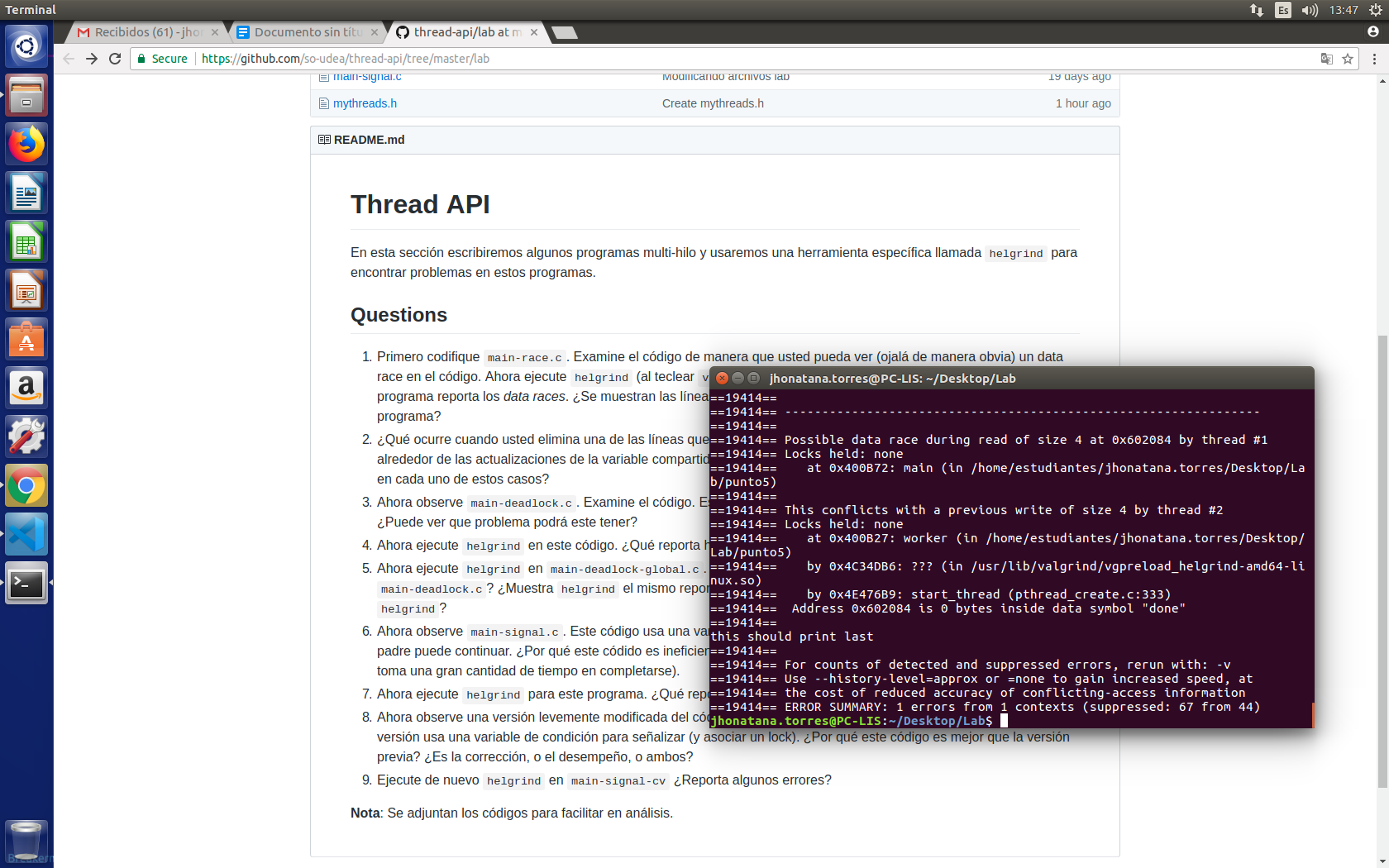
4)



5) No tendria que mostrar error, ya que se controla que el primero en tomar el lock g, seria el unico en tener acceso a toda la region critica, pero con la herramienta aparece un error, que creemos es de la herramienta y no del codigo.



6) Es ineficiente porque el hilo principal se queda quemando el quantium hasta que el otro hilo le de la señal de finalizar, el padre termina quemando su tiempo en un spin.

7) 

El codigo es ineficiente ya que desperdicia tiempo de uso de la CPU.