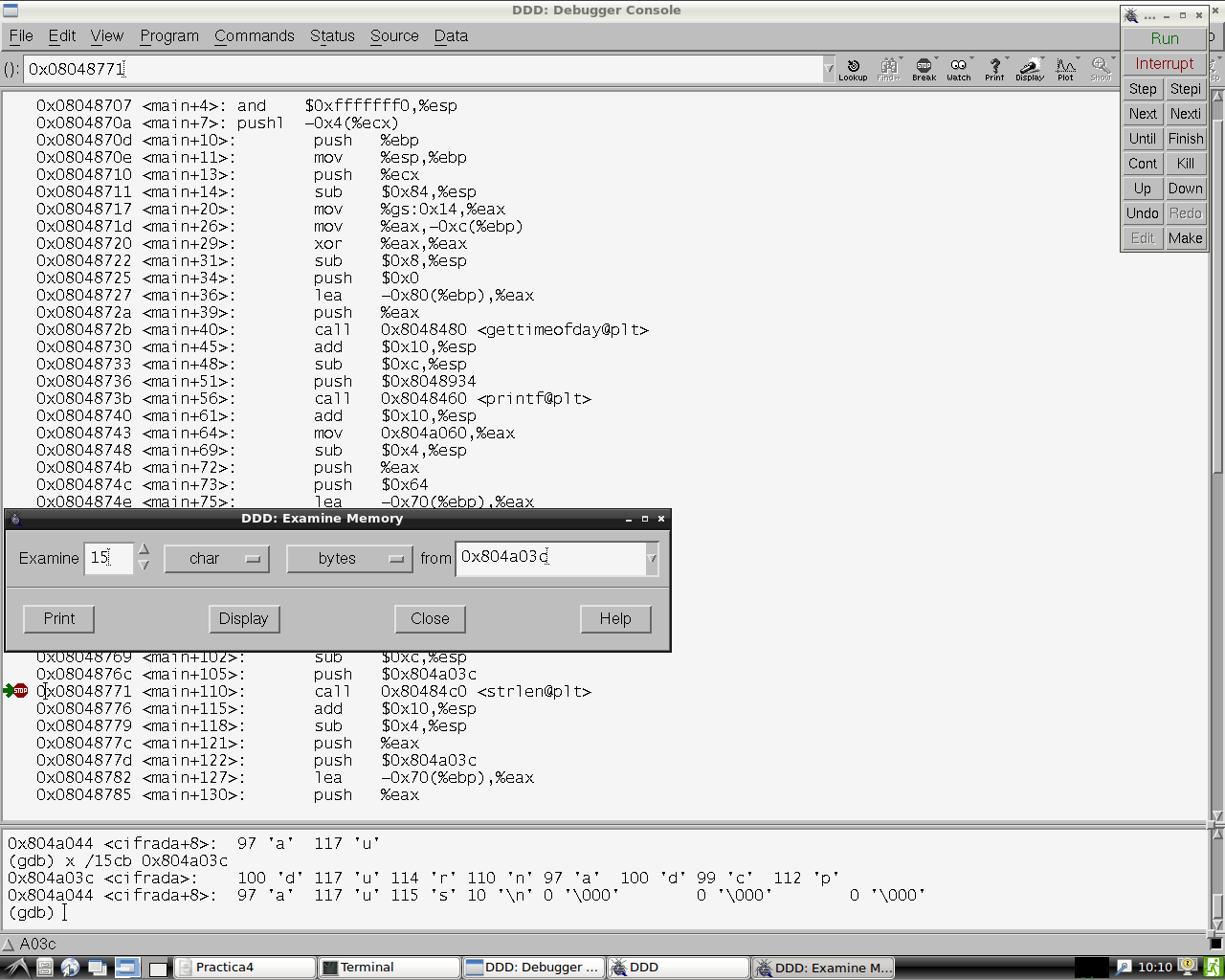
**Practica 4 Alejandro Poyatos López 2ºB(2)**

Explicación de cómo resolver la bomba:

Para resolver mi propia bomba he seguido varios pasos:

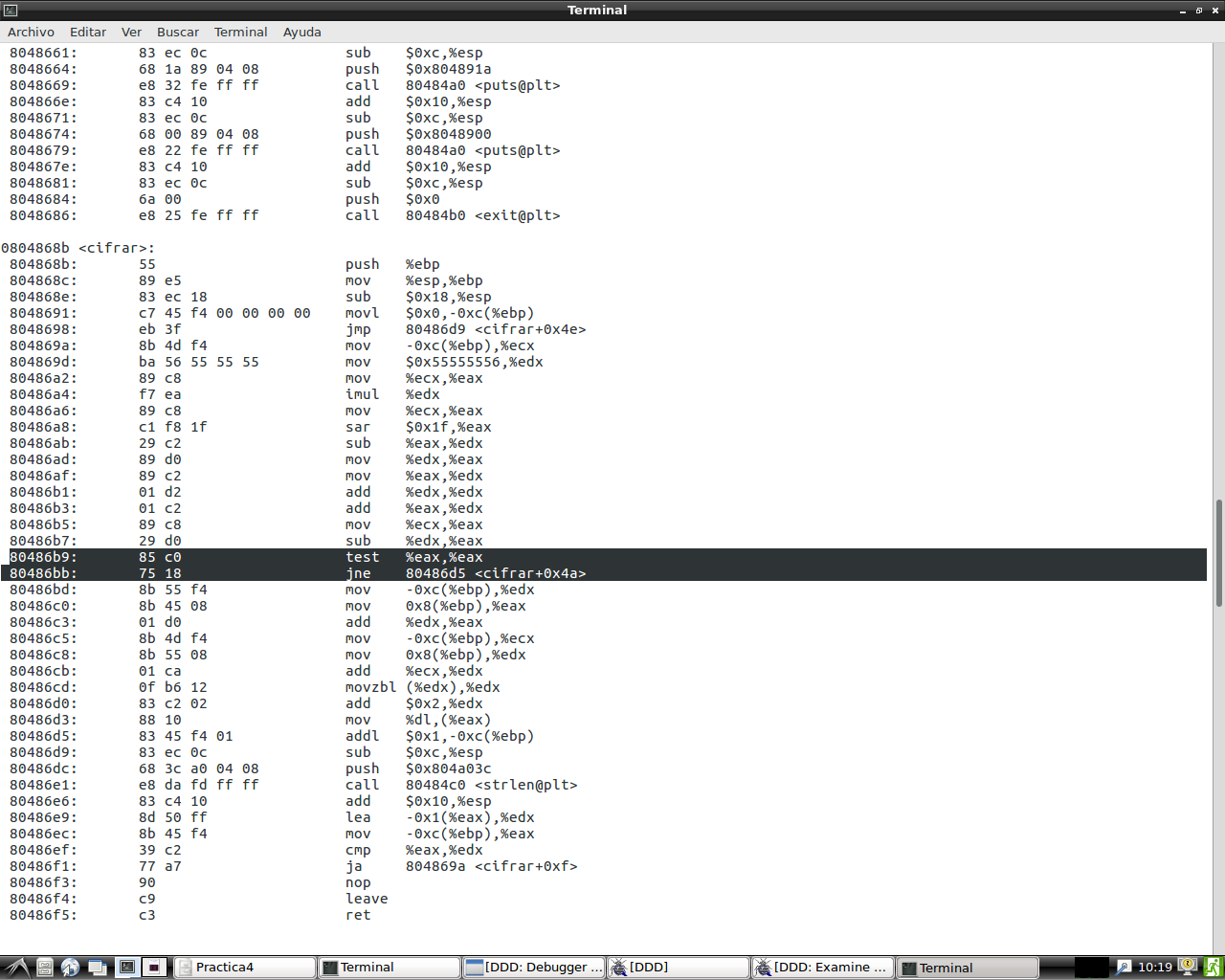
Primero he localizado y paralizado la ejecución con un breakpoint de DDD para obtener la contraseña cifrada, mediante la observación del argumento que le pasa el call a la llamada strlen:



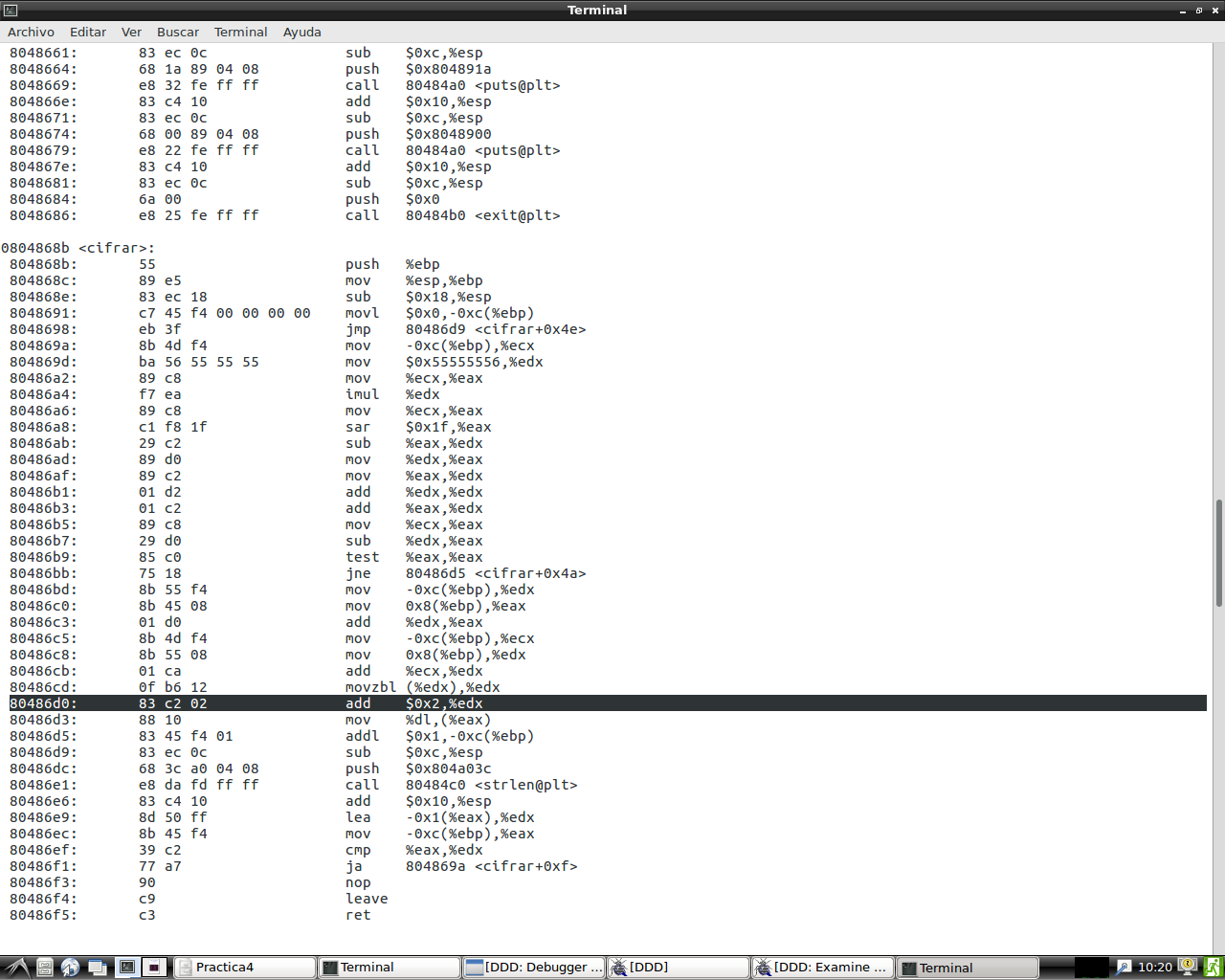
Una vez llegados ahí, examinamos los datos de la memoria obteniéndose:

Durnadcpaus

Que no parece decir nada pero si observamos el código de cifrar con objdump veremos que hay un bucle y dentro una comprobación:



Que comprobará si el elemento que se esta leyendo divide a 3, en cuyo caso hará lo siguiente:



Es decir que le suma 2 y va dando saltos, por lo que con el algoritmo que se basa en aumentar el valor del char estándar Unicode deducimos:

d - 2 = b

u = u

r = r

n – 2 = l

a = a

d= d

c -2 = a

p = p

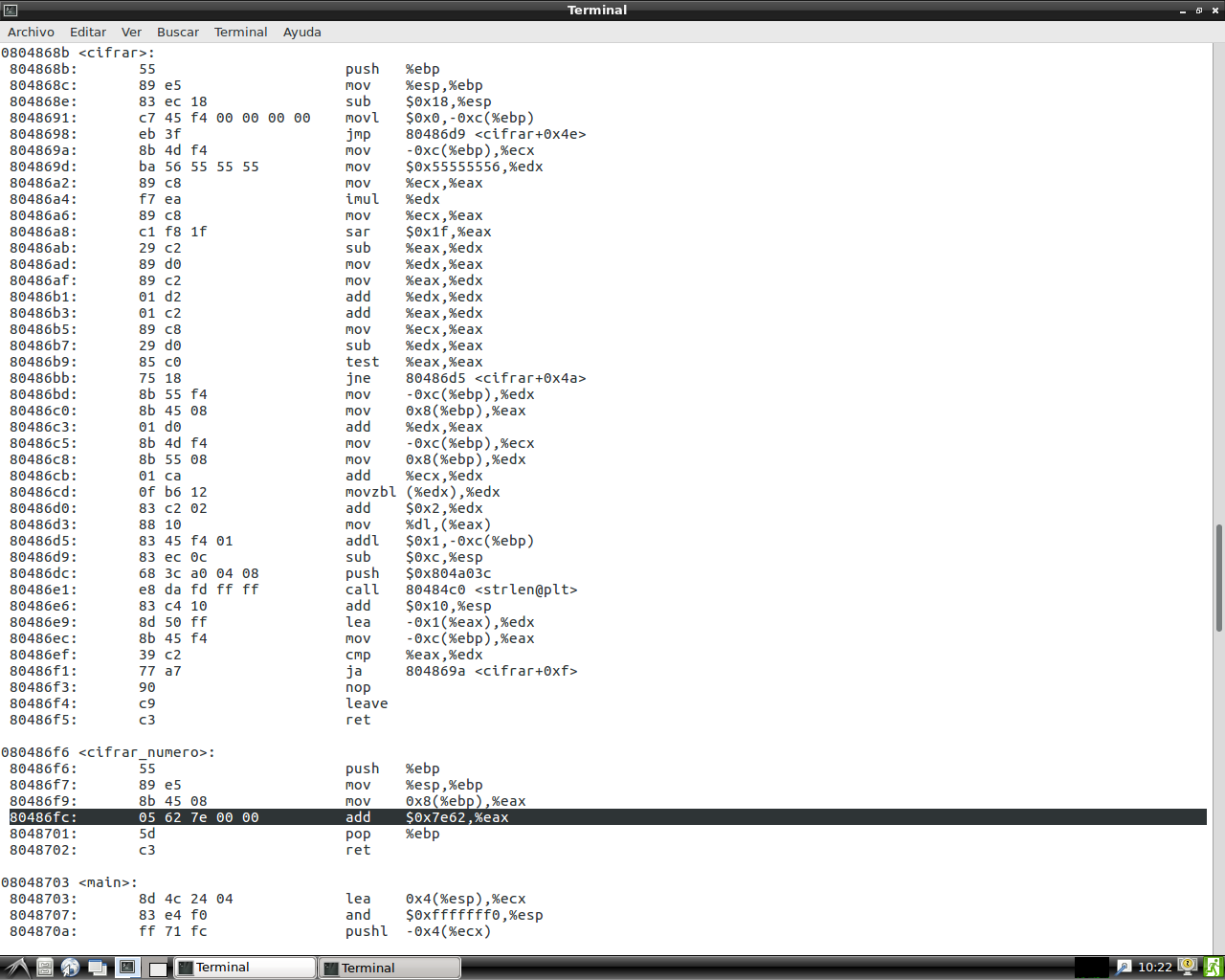
a = a

u -2= s

s = s

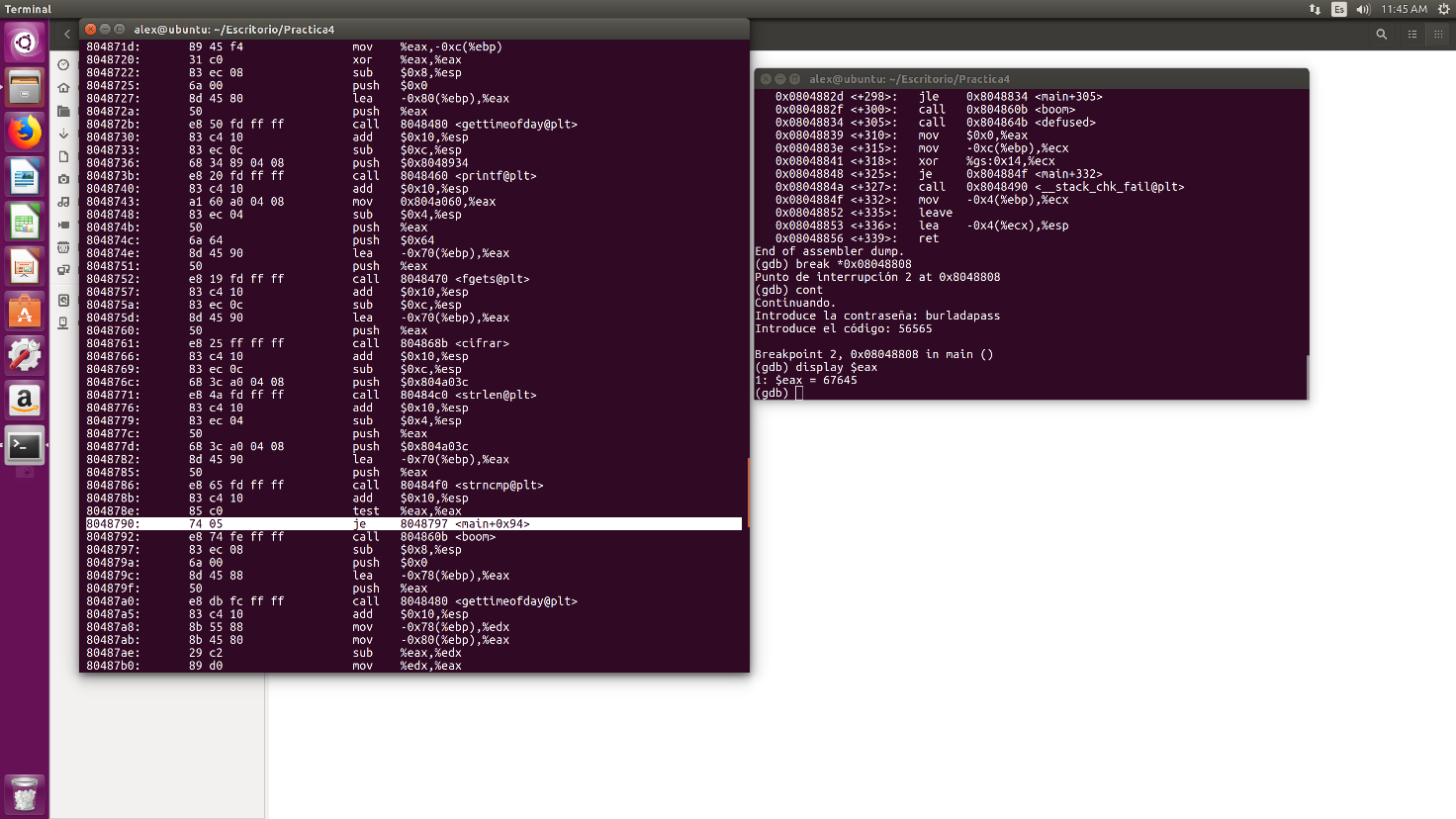
Entonces: durnadcpaus=burladapass

Para el código miraremos con objdump lo que hace el método cifrar\_numero, encargado de cifrar el código numérico:



Vemos que hace un suma del valor 7e62 (hex) que será igual a sumar 32354, por lo que ya solo queda que valor inicial tiene ese eax.

A esta altura no se muy bien el porque ddd no me permitió romper el código en el punto donde se le pasa el valor a cifrar número, por lo que opte por usar gdb:



Ahí se ven las ordenes que use, y como hago un breakpoint justo antes del salto para mirar que valor decimal contiene el registro eax = 67645

Luego teniendo en cuenta el anterior ya deducimos el código:

67645 – 32354 = 35291

Finalmente 35291 es el código correcto

Finalmente, ya tenemos las dos contraseñas descifradas:

Password= burladapass

Passcode = 35291