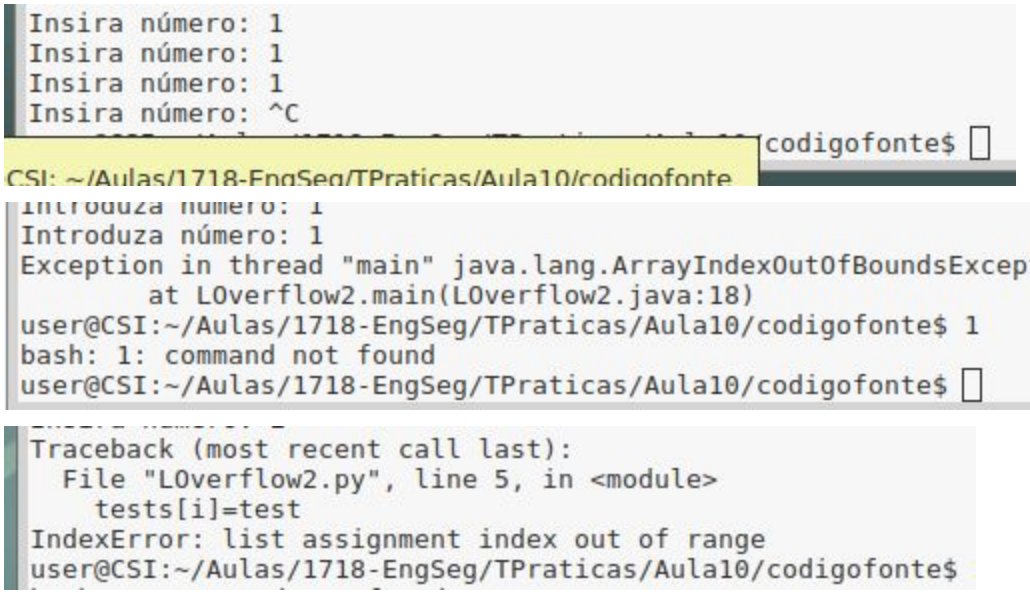
Pergunta P1.1 - Buffer overflow em várias linguagens

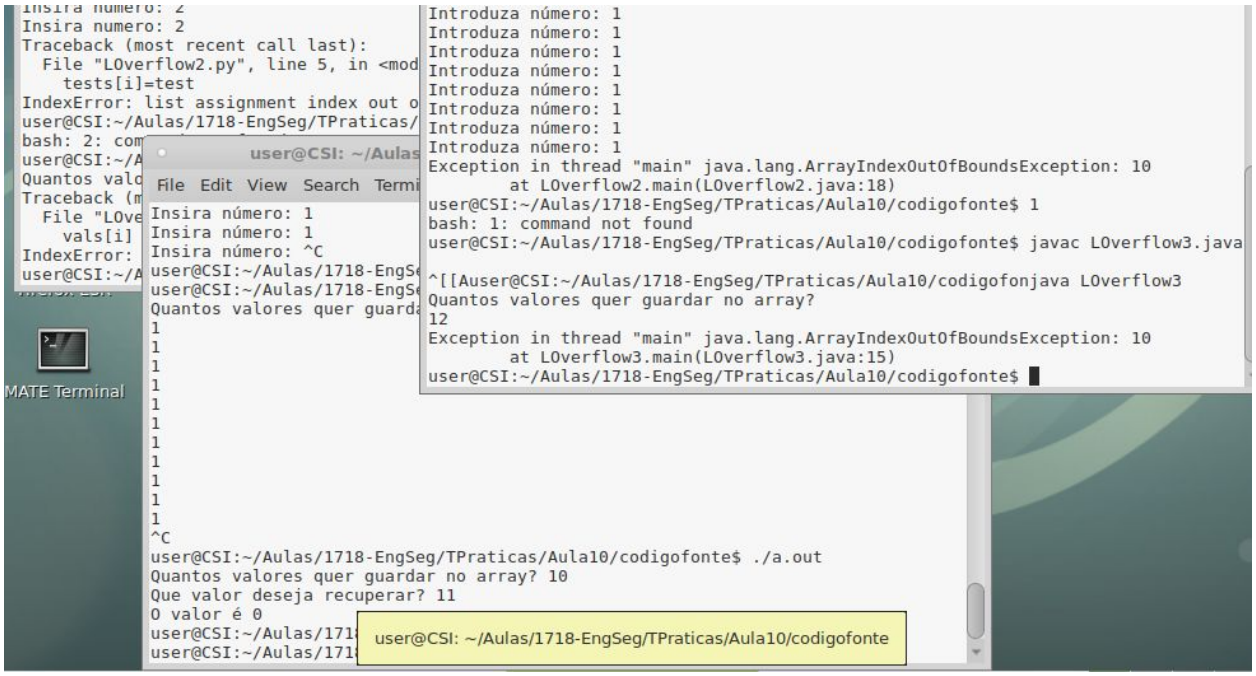


Python e Java quebram a execução assim que detectam o acesso a memória não alocada pela variável, dando o python 1 célula de segurança.
O c apenas devolve segfault quando se acede a células de memória não mapeadas ou apenas de read only.

Neste caso como a variável i é reescrita, o ciclo torna-se infinito e as células de memória escritas são restritas ao buffer e à própria variável i.

Nota: ver 12 1,1,1,....

Pergunta P1.2 - Buffer overflow em várias linguagens

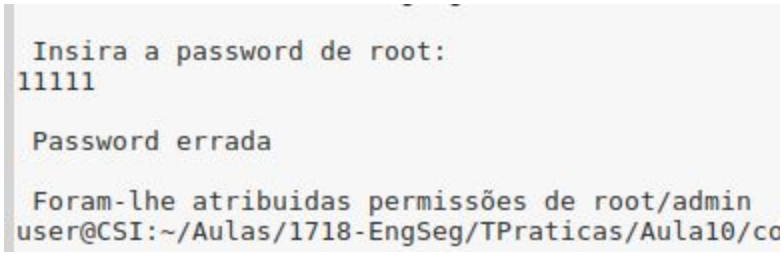


Pergunta P1.3 - Buffer overflow

Fgets não restringe o tamanho do input introduzido.

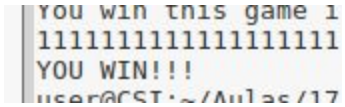
(i)

Escrever por cima da variável ‘pass’.

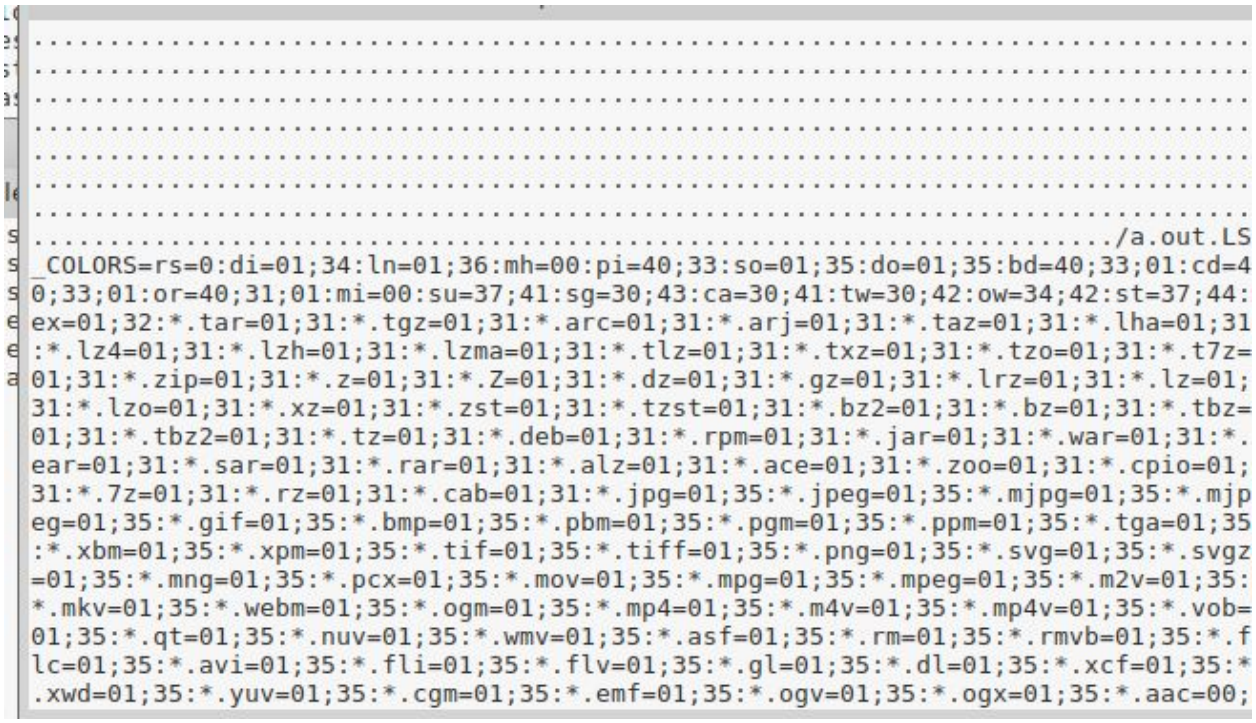


(ii)

Escrever por cima da variável control.

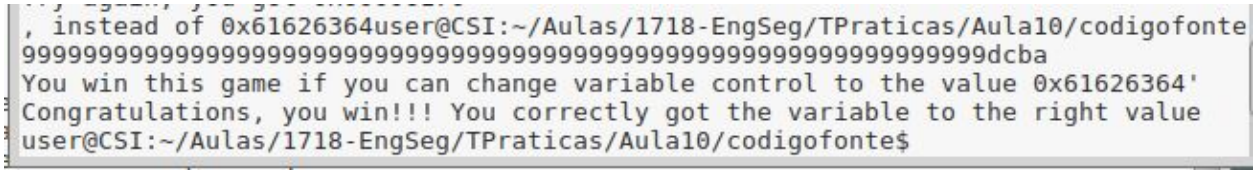


Pergunta P1.4 - Read overflow



Pode ser feito o acesso a células de memória fora do scope do programa.

Pergunta P1.5



- override do buffer todo
-tabela ascii
-inserir em ordem inversa pois estamos a inserir de baixo para cima no inteiro e ele é lido em big endian.

Pergunta P1.6

Endereço pode ser obtido através do uso do gdb disassemble win ou através de objdump -d a.out.
End:begin + 0000000000000740;
Para facilitar imprimiu-se o endereço.

```
user@CSI:~/Aulas/1718-EngSeg/TPraticas/Aula10/codigofonte$ ./a.out
func:0x562e77522740
You win this game if you are able to call the function win.'
999999999999999999999999999999999999999999999999999999999999999911111111@'Rw.V
calling function pointer, jumping to 0x562e77522740
Congratulations, you win!!! You successfully changed the code flow
user@CSI:~/Aulas/1718-EngSeg/TPraticas/Aula10/codigofonte$
```

Pergunta P1.7

todo.