

## BUFFERING

Creato un programma dove chiediamo un nome con caratteri limitati, questo lo stampa a dovere, il problema arriva quando la stringa inserita è superiore alla definita, questo evento si chiama BUFFERING OVERFLOW, il buffering è un meccanismo utilizzato per gestire in modo efficiente il flusso di dati tra componenti o processi, evitando così situazioni di overflow. L'overflow rappresenta il problema che si verifica quando i dati superano la capacità di elaborazione o di archiviazione del sistema, causando potenzialmente errori o perdita di dati.

Qui possiamo osservare come il programma crea un metodo per poter rettificare la situazione, inizialmente chiede l'input al utente e come supera il limite viene mostrato dove viene archiviato temporalmente, come è superiore al limite richiesto parte un ciclo affinché l'utente non inserisca un input adeguato

```
GNU nano 7.2      bof1.c *
#include <stdio.h>
#include <string.h>

int main () {
char buffer [30];
int dim = 0; //dim, dimensione

//do-while per correggere il buffering, cosa da far ripetere al utente di reimpostare una stringa piu corta
do{
printf("\n-----\nInserire il nome utente: ");
scanf("%s", buffer);
printf("\n-----\n");
dim = strlen(buffer); //strlen, string length

//se dim è maggiore di 30 su stampa la locazione in memoria
if (dim > 30) {
for (int i=0; i < dim; i++) {
printf("!!!Nome troppo lungo!!!\nIndirizzo di memoria del carattere %d : %d\n", i, &buffer[i]);
}
}

//se dim è minore stampa nome
else {
printf("Nome utente inserito: %s\n", buffer);
}
}while (dim > 30);

return 0;
}
```

```
(kali@kali)-[~/Desktop]
$ ./bof1

Inserire il nome utente: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

indirizzo di memoria del carattere 0 : 1306928800
indirizzo di memoria del carattere 1 : 1306928801
indirizzo di memoria del carattere 2 : 1306928802
indirizzo di memoria del carattere 3 : 1306928803
indirizzo di memoria del carattere 4 : 1306928804
indirizzo di memoria del carattere 5 : 1306928805
indirizzo di memoria del carattere 6 : 1306928806
indirizzo di memoria del carattere 7 : 1306928807
indirizzo di memoria del carattere 8 : 1306928808
indirizzo di memoria del carattere 9 : 1306928809
indirizzo di memoria del carattere 10 : 1306928810
indirizzo di memoria del carattere 11 : 1306928811
indirizzo di memoria del carattere 12 : 1306928812
indirizzo di memoria del carattere 13 : 1306928813
indirizzo di memoria del carattere 14 : 1306928814
indirizzo di memoria del carattere 15 : 1306928815
indirizzo di memoria del carattere 16 : 1306928816
indirizzo di memoria del carattere 17 : 1306928817
indirizzo di memoria del carattere 18 : 1306928818
indirizzo di memoria del carattere 19 : 1306928819
indirizzo di memoria del carattere 20 : 1306928820
indirizzo di memoria del carattere 21 : 1306928821
indirizzo di memoria del carattere 22 : 1306928822
indirizzo di memoria del carattere 23 : 1306928823
indirizzo di memoria del carattere 24 : 1306928824
indirizzo di memoria del carattere 25 : 1306928825
indirizzo di memoria del carattere 26 : 1306928826
indirizzo di memoria del carattere 27 : 1306928827
indirizzo di memoria del carattere 28 : 1306928828
indirizzo di memoria del carattere 29 : 1306928829
indirizzo di memoria del carattere 30 : 1306928830

Inserire il nome utente: sssssssssss

Nome utente inserito: sssssssssss

(kali@kali)-[~/Desktop]
```

Questa è la dimostrazione di come funziona, e possiamo osservare dove viene allocato in memoria.