**Environment variable**

**2) write a program to analyze the memory mapping of environment variables**

**a) modify the program so that it adds an environment variable using setenv**

**b) analyze how the memory mappings change**

**#include <stdlib.h>**

**extern char\*\* envp[];**

**int main(int argc, char\* argv[],char\* envp[]){**

**int i =0;**

**while(envp[i] !=NULL){**

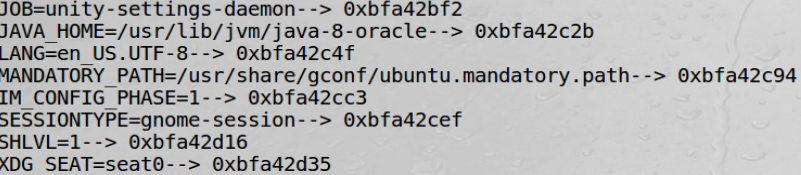
**printf("%s--> ",envp[i++]);**

**printf("%p\n",envp[i++]);**

**}**

**return 1;**

**}**

****

**Use setenv :**

**int setenv(const char \**name*, const char \**value*, int *overwrite*);**

The setenv() function adds the variable *name* to the environment with the value *value*, if *name* does not already exist. If *name* does exist in the environment, then its value is changed to *value*

if *overwrite* is nonzero; if *overwrite* is zero, then the value of *name* is not changed

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3) use the ldd program to analyze the libraries used by the previous programs

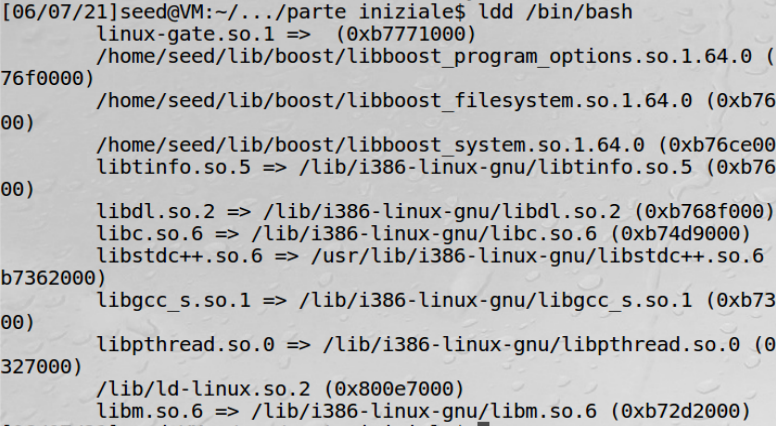
a) statically link the programs and repeat

**ldd** prints the shared objects (shared libraries) required by each program or shared object specified on the command line.

ldd es2.c



if we do ldd to bin/bash ldd /bin/bash



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4) Use the LD\_PRELOAD environment variable to force a program into using a custom function instead of a standard library one.

a) Use as a target the sleep function

b) Write a test program that uses sleep

c) create a shared library using the -shared option in gcc e.g., gcc -shared -o libmylib.so.1.0.1 sleep.o

d) insert in LD\_PRELOAD a reference to this new library

e) statically link the program and analyze the behavior in this case

5) Check system counter measures

mystest.c

#include <stdlib.h>

int main(){

sleep(1);

return 0;

}

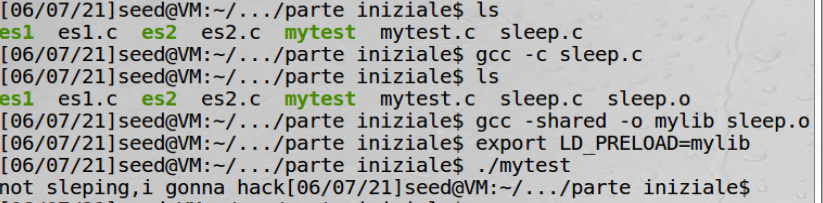
sleep.c

#include <stdio.h>

void sleep(int s){ // questa è la funzione infiltrator sleep

printf("not sleping,i gonna hack");

}

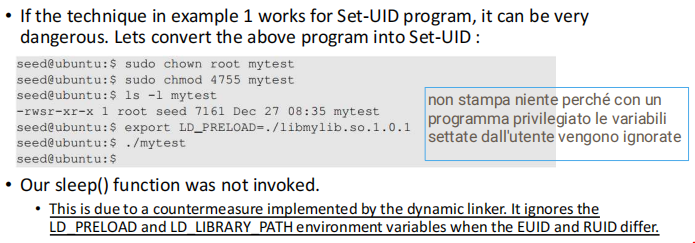


gcc -c sleep.c ----> create a object file sleep.o

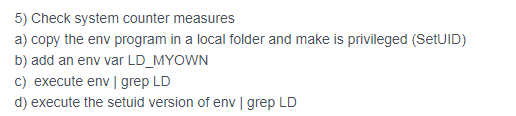
gcc -shared -o mylib sleep.o ---> create a library mylib with sleep function alterated

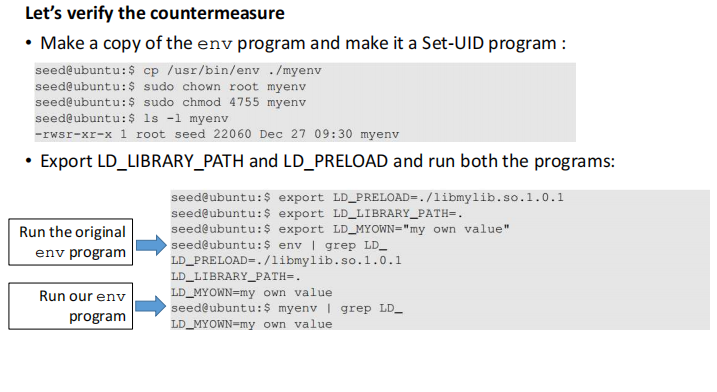
export LD\_preload=my lib (TUTTO ATTACCATO) → load my lib

**caso set user id program**

****

**non funziona piu perchè se effctive id(ROOT)e real id (SEED )sono diversi, non stampa piu nulla**





**LD\_PRELOAD E LD PATH MUOIONO PERCHè CON SETID PROGRAM DROPPA TUTTI I SET LOCALI !**