Pasii urmati pentru compilarea kernel-ului si adaugarea unui nou apel de sistem.

1. Am facut update

sudo apt update

sudo apt upgrade –y

1. Am instalat cateva utilitare de care voi avea nevoie mai tarziu

apt install gcc build-essential libncurses-dev libssl-dev libelf-dev bison flex –y

1. Am descarcat versiunea de kernel apropiata cu cea pe care o ruleaza masina mea virtuala, am dezarhivat apoi am dat un restart masinii virtuale

wget -P ~/ <https://mirrors.edge.kernel.org/pub/linux/kernel/v5.x/linux-5.15.52.tar.xz>

tar -xvf ~/linux-6.0.6.tar.xz -C ~/

reboot

1. Am accesat tabela apelurilor de sistem si la final am adaugat un nou apel de sistem

gedit arch/x86/entry/syscalls/syscall\_64.tbl

449 common hello sys\_hello

1. Am adaugat declaratia in syscalls.h

gedit include/linux/syscalls.h

asmlinkage long sys\_helllo(void);

1. In kernel/sys.c am adugat definitia functiei cu macro-ul SYSCALL\_DEFINE1

SYSCALL\_DEFINE1(hello, char \*, msg){

char buf[256];

long copied = strncpy\_from\_user(buf, msg, sizeof(buf));

if(copied < 0 || copied == sizeof(buf))

return -EFAULT;

printk(KERN\_INFO "hello syscall called with \"%s\"\n", buf);

return 0;

}

1. Am deschis meniul de config pentru kernel si l-am salvat fara a face modificari

make menuconfig

1. Am scris script-ul deploy.sh:
2. #!/usr/bin/bash
3. # Compile and "deploy" a new custom kernel from source on Ubuntu Linux
4. # Change this if you'd like. It has no relation
5. # to the suffix set in the kernel config.
6. SUFFIX="-ppp"
7. # This causes the script to exit if an error occurs
8. set -e
9. # Compile the kernel
10. make -j4
11. # Compile and install modules
12. make -j4 modules\_install
13. # Install kernel image
14. make -j4 install
15. # Update the bootloader
16. sudo update-grub

source: <https://github.com/armi3/custom_syscall#1-set-your-virtual-machine>

9. Am accesat fisierul .config si am facut urmatoarea modificare

CONFIG\_LOCALVERSION="-hello"

10. Am oferit drepturi de executie script-ului si l-am rulat cu drepturi de sudo

chmod 0744 deploy.sh

sudo ./deploy.sh

* Mai departe am avut tot felul de probleme, asteptam cate 20-30-40 de minute apoi mai aparea o eroare, cautam rezolvare si tot asa.
* Dupa mai multe incercari esuate, am reusit sa duc procesul la final

11. Am schimbat versiunea de kernel de pe care sa booteze sistemul de operare

12. Am testat apelul de sistem sa vad daca functioneaza

#define \_GNU\_SOURCE

#include <unistd.h>

#include <sys/syscall.h>

#include <stdio.h>

#define SYS\_hello 449 // numarul apelului de sistem

int main(int argc, char \*\*argv)

{

if (argc <= 1) {

printf("Must provide a string to give to system call.\n");

return -1;

}

char \*arg = argv[1];

printf("Making system call with \"%s\".\n", arg);

long res = syscall(SYS\_hello, arg);

printf("System call returned %ld.\n", res);

return res;

}

