

Ex. No.: 4

Date: 13 - 03 - 2024

SIGNAL CATCHING

Aim:

To write a C program to catch signals used in Linux.

Algorithm:

1. The program is initialized for catching interrupt signal(SIGINT).
2. If Cntrl+C is pressed within 3 seconds then my_handler is called
3. my_handler routine displays the signal that was caught.
4. If no interrupt received then PART-II is executed.
5. In PART-II,Cntrl+C is ignored till 3 seconds then it goes to PART-III.
6. In PART-III, the default action takes place.

Program Code:

```
// signals.c

#include <signal.h>
#include <stdio.h>

void my_handler (int sig); /* function prototype */

int main()
{
    struct sigaction my_action;
    /* Part I: Catch SIGINT */
    my_action.sa_handler = my_handler;
    my_action.sa_flags = SA_RESTART;
    sigaction (SIGINT, &my_action, NULL);
    printf ("Catching SIGINT\n");
    sleep (3);
    printf (" No SIGINT within 3 seconds\n");
```

```

/* Part II: Ignore SIGINT */

my_action.sa_handler = SIG_IGN;
my_action.sa_flags = SA_RESTART;
sigaction (SIGINT, &my_action, NULL);
printf ("Ignoring SIGINT\n");
sleep (3);
printf (" Sleep is over\n");

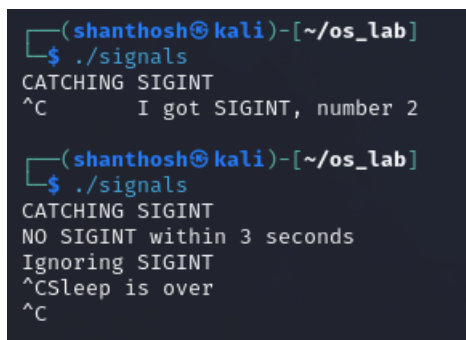

/* Part III: Default action for SIGINT */
my_action.sa_handler = SIG_DFL;

33
my_action.sa_flags = SA_RESTART;
sigaction (SIGINT, &my_action, NULL);
sleep (3);
printf ("No SIGINT within 3 seconds\n");}

void my_handler (int sig){
printf (" \t I got SIGINT, number %d\n", sig);
exit(0);
}

```

Output:



```

(shanthosh@kali)-[~/os_lab]
$ ./signals
CATCHING SIGINT
^C      I got SIGINT, number 2

(shanthosh@kali)-[~/os_lab]
$ ./signals
CATCHING SIGINT
NO SIGINT within 3 seconds
Ignoring SIGINT
^CSleep is over
^C

```

Result:

Hence the signal catching has been successfully executed and completed.