

Ex. No.: 4

Date: 12.03.24

SIGNAL CATCHING

Aim:

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

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;

    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:

```
(kali㉿kali)-[~/os/ex4]
└─$ ./a.out
Catching SIGINT
^C      I got SIGINT, number 2

(kali㉿kali)-[~/os/ex4]
└─$ ./a.out
Catching SIGINT
No SIGINT within 3 seconds
Ignoring SIGINT
^C Sleep is over
No SIGINT within 3 seconds
```

Result:

The above program executed successfully and output got verified.