Ex. No.: 4
Date: 12/3/24

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 <stdlib.h>
#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:

```
gcc sigcat.c
./a.out

Catching SIGINT
No SIGINT within 3 seconds
Ignoring SIGINT
Sleep is over
No SIGINT within 3 seconds
./a.out
Catching SIGINT
^c     I got SIGINT, number 2
```

RESULT:

The program has been executed and the output has been veritied successfully.

12/3/24

10/10