Lab-10

Aim: Study of signals and handling in Linux.

Explanation:

signal system call:

```
#include <signal.h>

typedef void (*sighandler_t)(int);

sighandler_t signal(int signum, sighandler_t handler);
```

signal() sets the disposition of the signal signum to handler, which is either SIG_IGN, SIG_DFL, or the address of a programmer-defined function (a "signal handler").

SIG_IGN, then the signal is ignored.

SIG_DFL, then the default action associated with the signal

The signals SIGKILL and SIGSTOP cannot be caught or ignored.

Perform error handling while setting up signal handler. signal() returns the previous value of the signal handler, or SIG_ERR on error. In the event of an error, errno is set to indicate the cause.

Task-1:

Write a program to display all list of signlas with signal number, op code/acronym and description. Know that the similar output is generated using "kill -l" command as well. Practice the same.

- psignal()
-strsignal()
- extern char * sys_siglist[]

Task-2:

Demonstrate SIGSTOP and SIGCONT via a program on a running process of vim like editor or other software. Observe outcome of "bg" - background, "fg" - forground, "ps -ef | grep vim".

<u>Task-3:</u>

Implement sig2str or str2sig function.

int sig2str(int signo, char *str)

int str2sig(const char *str, int *signop)