```
/*
File name: mathur-HW2.c
Home Assignment 2
Pratik Mathur
G no:-G00963577
Sarthak Sheth
G no:-G00979607
*/

#include<signal.h>
#include<stdio.h>
#include <string.h>
#include<unistd.h>
#include<stdlib.h>
#include<fcntl.h>
#include<sys/stat.h>
#include<signal.h>

void stopsignal_ctrlc(int n)                    //signal handler function stopsignal_Ctrlc is defined here.
{
        printf(" Cannot terminate the program with Ctrl + C. Please press Ctrl + Z to terminate\n");
}

int CS531_system(const char *comm)                      //user defined system Function
CS531_system(char*)
{
        int pid = fork();                       //creating a child process    using fork()
        char buf[BUFSIZ];
        int w,status;
        char *progname;
        char *argv[50];
        char *ps="ps -ag";
        FILE *popen(),*fin;
        progname=argv[0];
        if((fin=popen(ps,"r"))==NULL)                                   // File open in read mode in fin
        {
                fprintf(stderr,"%s:cannot open%s\n",progname,ps);    //Error handling. If the file is
empty then exit
                exit(1);
        }
        fgets(buf,sizeof buf,fin);
        while(fgets(buf,sizeof buf,fin)!=NULL)
        if (pid == 0)
        {
                execlp("sh","sh","-c",comm,(char *)0);   //Used to overlay the child process which is
created by calling fork()
                exit(127);
```

```
        }
        if(signal(SIGINT,stopsignal_ctrlc) == SIG_ERR)          //Whenever the Ctrl+c is entered, signal
handler catches the signal and prevents interuption of process
        printf("No signal caught\n");
        waitpid(pid, &status, 0);                               //Waitpid checks the wait status of child
process
        if (WIFEXITED(status))                                  //If a child process terminated
normally then non zero value is returned
                return WEXITSTATUS(status);                     //returns status as true if the
child process exits normally
        else
                return WTERMSIG(status);                        //returns status as true if the
child process is terminated due to signal

}
/*
int main ()
{
  char command[50];
  strcpy( command, "dir");
  CS531_system(command);
  return(0);
}
*/
```