PAGE NO :
DATE :

Lab 9

1) Write a C/C++ prog which demonstrate inter process communications b/w a reader Processed and a write Process use Mk Md fifo, open, read, write and close APIs in your Prog:,

```
# include <sys/types.h>
# include < sys/stat.h>
# include <string.h>
# include < stdio.h>
# include < unistd.h>
int main( int argc, char *argv [])
  {
      char buff [160];
     int fd, n;
     Mkfifo (argv [1], S_FIFO | 0777);
     if (argc == 3)
    {
        fd = open (argv [1], O-WRONLY);
        write (fd, argv [2], strlen(argv[2]));
        close fd;
    }
       if (argc ==2)
    {   fd = open(argv [1], O-RDONLY);
        n = read (fd, buf, size-of buf));
        buf [n] = '\0';
        Printf ("%s", buf);
        close(fd);
    }
  }
```

DATE :

Output
```
cc comm.c
./a.out comm "5b linear lab" &
[1] 3503
./a.out comm
5b linear lab [1]+ Done
```

2) write a Prog to emulate the unix ln Command

```
#include <unistd.h>
# include < stdio.h>
# include < string.h>
int main (int argc , char * argv [])
{  if (argc == 4)
   { symlink (argv [2], argv [3]);
   printf (" symbolic link is created ");
   return 0;
 }
    if (argc == 3)
 {     link (argv [1] , argv [2] );
   printf (" Hardlink is created ");
   return 0;
 }
}
```

output
```
cc lns.c
./a.out -s sum.sh slink
symbolic link is created
./a.out prime.sh hardlink
Hardlink is created
```

PAGE NO :
DATE :

```
3. # define _POSIX_SOURCE
# define _POSIX_C_SOURCE
# include <stdio.h>
#   include <unistd.h>
int main()
 if def _POSIX_JOB_CONTROL
 cout <<" Sys supp. Job control, feautes "<<endl;
 cout <<"Sys does not support control \n";
  end ife
 if def _ POSIX_SAVED_IDS
 cout << " System supp set_UID and saved set
else                                      _GID" << cout
 cout << "System doesn't support saved set
              _UID\n")
end if
  if def _ POSIX _NO_TRUNC
 cout << " System support Path Truncator option
                                   " <<end
else
 cout << " Sys doesnot supp Path truncation \n";
 end if
if det _ POSIX _VDISABLE
  cout << "System supports disable character
for files: " << endif ;
 else
  cout << " System doesnot supp disable char
  endif
  return 0;
 }
```

Output

```
g++ prog 3. cpp
'1 a .out
Sys Support job control feature
Sys Support Sabol set-UID and Saved
 set-GID
Sys Support Charge ownership feature
Sys Support Path Truncation option:
Sys Support Disable character for file'
```

4) Write C prog to the Output the content of it environment lists'

```
#include <stdio.h>
# include <unistd.h>
int main (int argc, char * argv[])
{
    Char ** Ptr;
    enter Char ** environ;
  for( ptr = environ; * Ptr'; ptr++)
    Printf ("%s\n", * Ptr);
    return 0;
}
```

Output

```
Shell = /bin/bash
Sizssion -manager
```