Lab 2(b): IPC: Named Pipes and Messages

In computing, a named pipe (also known as a FIFO is one of the methods for intern-process communication. It is an extension to the traditional pipe concept on Unix. A traditional pipe is “unnamed” and lasts only as long as the process.

A named pipe, however, can last as long as the system is up, beyond the life of the process. It can be deleted if no longer used. Usually a named pipe appears as a file, and generally processes attach to it for inter-process communication. A FIFO file is a special kind of file on the local storage which allows two or more processes to communicate with each other by reading/writing to/from this file.

Once we have created a FIFO special file in this way, any process can open it for reading or writing, in the same way as an ordinary file. However, it has to be open at both ends simultaneously before you can proceed to do any input or output operations on it.

```
/* server program */
#include <fcntl.h>
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
void main()
{
char buf[BUFSIZ];
int fd,n;
/* create named pipe (FIFO), set RWX rights for user */
mknod("/tmp/server", S_IFIFO | S_IRWXU,0);
/* open FIFO for reading */
fd = open("/tmp/server", O_RDONLY);
if (fd < 0 ) {
perror("Open");
exit(1);
}
/* read data from FIFO */
while ((n=read(fd,buf,BUFSIZ-1)) > 0) {
buf[n]=0;
printf("Read: %s\n",buf);
}
close (fd);
}

/* Client Program*/
#include <fcntl.h>
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
void main()
{
char buf[BUFSIZ];
int fd,n;
/* open FIFO for writing */
fd = open("/tmp/server", O_WRONLY);
```

```
if (fd < 0 ) {
perror("Open");
exit(1);
}
/* read data from standard input and write to FIFO */
while ((n=read(0,buf,BUFSIZ-1)) > 0) {
buf[n]=0;
if (write(fd,buf,n) < 0)
exit(1);
printf("Write[%d]: %s\n",getpid(),buf);
}
close (fd);
}
```

Assignments:

1. Prepare a client server program to do simple arithmetic operations, where client provides two operands and one operator and the server performs the operation.