

Pipe

Program 1

```
#include <stdio.h>
#include <unistd.h>
#define MSGSIZE 19
char* msg1 = "osw";
char* msg2 = "class";
char* msg3 = "Anikt";
int main() {
    char inbuf[MSGSIZE];
    int p[2], i;
    if (pipe(p) < 0)
        return 1;
    write(p[1], msg1, MSGSIZE);
    write(p[1], msg2, MSGSIZE);
    write(p[1], msg3, MSGSIZE);

    for (i = 0; i < 3; i++) {
        read(p[0], inbuf, MSGSIZE);
        printf("%s\n", inbuf);
    }
    return 0; }
```

Program 2

```
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>

#define BUFSIZE 10

int main(void) {
    char bufin[BUFSIZE] = "empty";
    pid_t childpid;

    int fd[2]; //creating a pipe
    pipe(fd);

    childpid = fork();
    if (childpid)
        { wait(NULL);
          read(fd[0], bufin, 5);
        }
    else
        write(fd[1], "Sent by Child", 5); //write to pipe
```

```
fprintf(stderr, "Message is %s",bufin);
return 0;}
```

Program 3

```
#include <errno.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/wait.h>
#include <unistd.h>

int main()
{
    int a[2];
    char buff[11];
    if (pipe(a) == -1)
    {
        perror("error in pipe");
        return 1;
    }
    write(a[1], "Blockchain", 11);
    read(a[0], buff, 11);

    printf("The content inside buffer is :%s\n", buff);
    return 0;
}
```

program4

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
int main(void)
{
    int fd[2], nbytes;
    pid_t childpid;
    char string[] = "Hello, world!\n";
    char readbuffer[80];

    pipe(fd);
    if((childpid = fork()) == -1)
    {
        perror("fork");
        return 1;
    }
}
```

```
if(childpid == 0)
{
    write(fd[1], string, sizeof(string));
    return 1;}
else
{
    nbytes = read(fd[0], readbuffer, sizeof(readbuffer));
    printf("Received string: %s", readbuffer);
}
return 0; }
```