

## OS-Experiment 11

Name : Rahul Goel

Reg No: RA1911030010094

Batch: CSE-O2

```
aryan@LAPTOP-SSBAGLQR: ~  
#include<stdio.h>  
#include<unistd.h>  
#define MSGSIZE 20  
char *msg1 = "HELLO - 1";  
char *msg2 = "HELLO - 2";  
char *msg3 = "HELLO - 3";  
int main()  
{  
    char inbuf[MSGSIZE];  
    int p[2],i;  
    if(pipe(p)<0)  
        exit(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;  
}
```

proj1.c" 22L, 383C 22,1 All

```
aryan@LAPTOP-SSBAGLQR: ~  
aryan@LAPTOP-SSBAGLQR:~$ vi proj1.c  
aryan@LAPTOP-SSBAGLQR:~$ gcc proj1.c  
proj1.c: in function 'main':  
proj1.c:12:3: warning: implicit declaration of function 'exit' [-Wimplicit-function-declaration]  
12 |     exit(1);  
    |     ^~~~~~  
proj1.c:12:3: warning: incompatible implicit declaration of built-in function 'exit'  
proj1.c:3:1: note: include '<stdlib.h>' or provide a declaration of 'exit'  
2 | #include<unistd.h>  
+++ *+include <stdlib.h>  
3 | #define MSGSIZE 20  
aryan@LAPTOP-SSBAGLQR:~$ ./a.out  
HELLO - 1  
HELLO - 2  
HELLO - 3  
aryan@LAPTOP-SSBAGLQR:~$
```

```
aryan@LAPTOP-SSBAGLQR:~  
#include <stdio.h>  
#include <unistd.h>  
int main()  
{  
    int pipefds[2];  
    int returnstatus;  
    char writemessages[2][20]={"Hi", "Hello"};  
    char readmessage[20];  
    returnstatus = pipe(pipefds);  
    if (returnstatus == -1) {  
        printf("Unable to create pipe\n");  
        return 1;  
    }  
    printf("Writing to pipe - Message 1 is %s\n", writemessages[0]);  
    write(pipefds[1], writemessages[0], sizeof(writemessages[0]));  
    read(pipefds[0], readmessage, sizeof(readmessage));  
    printf("Reading from pipe - Message 1 is %s\n", readmessage);  
    printf("Writing to pipe - Message 2 is %s\n", writemessages[1]);  
    write(pipefds[1], writemessages[1], sizeof(writemessages[1]));  
    read(pipefds[0], readmessage, sizeof(readmessage));  
    printf("Reading from pipe - Message 2 is %s\n", readmessage);  
    return 0;  
}
```

"proj2.c" 23L, 777C

22,1-8 All

```
aryan@LAPTOP-SSBAGLQR:~  
aryan@LAPTOP-SSBAGLQR:~$ vi proj1.c  
aryan@LAPTOP-SSBAGLQR:~$ gcc proj1.c  
proj1.c: In function 'main':  
proj1.c:12:3: warning: implicit declaration of function 'exit' [-Wimplicit-function-declaration]  
12 |     exit(1);  
    |     ^~~~~  
proj1.c:12:3: warning: incompatible implicit declaration of built-in function 'exit'  
proj1.c:3:1: note: include '<stdlib.h>' or provide a declaration of 'exit'  
2 | #include <unistd.h>  
+++ + #include <stdlib.h>  
3 | #define MSGSIZE 20  
aryan@LAPTOP-SSBAGLQR:~$ ./a.out  
HELLO - 1  
HELLO - 2  
HELLO - 3  
aryan@LAPTOP-SSBAGLQR:~$ vi proj2.c  
aryan@LAPTOP-SSBAGLQR:~$ gcc proj2.c  
aryan@LAPTOP-SSBAGLQR:~$ ./a.out  
Writing to pipe - Message 1 is Hi  
Reading from pipe - Message 1 is Hi  
Writing to pipe - Message 2 is Hello  
Reading from pipe - Message 2 is Hello  
aryan@LAPTOP-SSBAGLQR:~$
```

```

aryan@LAPTOP-SSBAGLQR:~
#include <stdio.h>
#include <unistd.h>

int main()
{
    int pipefds1[2], pipefds2[2];
    int returnstatus1, returnstatus2;
    int pid;
    char pipelwritemessage[20] = "Hi";
    char pipe2writemessage[20] = "Hello";
    char readmessage[20];
    returnstatus1 = pipe(pipefds1);
    if (returnstatus1 == -1)
    {
        printf("Unable to create pipe 1 \n");
        return 1;
    }
    returnstatus2 = pipe(pipefds2);
    if (returnstatus2 == -1)
    {
        printf("Unable to create pipe 2 \n");
        return 1;
    }
    pid = fork();
    if (pid != 0)
    {
        close(pipefds1[0]);
        close(pipefds2[1]);
        printf("In Parent: Writing to pipe 1 - Message is %s\n", pipelwritemessage);
        write(pipefds1[1], pipelwritemessage, sizeof(pipelwritemessage));
        read(pipefds2[0], readmessage, sizeof(readmessage));
        printf("In Parent: Reading from pipe 2 - Message is %s\n", readmessage);
    }
    else
    {
        close(pipefds1[1]); // Close the unwanted pipe1 write side
        close(pipefds2[0]); // Close the unwanted pipe2 read side
        read(pipefds1[0], readmessage, sizeof(readmessage));
        printf("In Child: Reading from pipe 1 - Message is %s\n", readmessage);
        printf("In Child: Writing to pipe 2 - Message is %s\n", pipe2writemessage);
        write(pipefds2[1], pipe2writemessage, sizeof(pipe2writemessage));
    }
    return 0;
}

```

proj3.c" 43L, 1214C

```

aryan@LAPTOP-SSBAGLQR:~
aryan@LAPTOP-SSBAGLQR:~$ vi proj1.c
aryan@LAPTOP-SSBAGLQR:~$ gcc proj1.c
proj1.c: In function 'main':
proj1.c:12:3: warning: implicit declaration of function 'exit' [-Wimplicit-function-declaration]
   12 |     exit(1);
      |     ^~~~~
proj1.c:12:3: warning: incompatible implicit declaration of built-in function 'exit'
proj1.c:3:1: note: include 'stdlib.h' or provide a declaration of 'exit'
     3 | #include <unistd.h>
       | ~~~~+~~~~
     3 | #define MSGSIZE 20
       | ~~~~~
aryan@LAPTOP-SSBAGLQR:~$ ./a.out
HELLO - 1
HELLO - 2
HELLO - 3
aryan@LAPTOP-SSBAGLQR:~$ vi proj2.c
aryan@LAPTOP-SSBAGLQR:~$ gcc proj2.c
aryan@LAPTOP-SSBAGLQR:~$ ./a.out
Writing to pipe - Message 1 is Hi
Reading from pipe - Message 1 is Hi
Writing to pipe - Message 2 is Hello
Reading from pipe - Message 2 is Hello
aryan@LAPTOP-SSBAGLQR:~$ vi proj3.c
aryan@LAPTOP-SSBAGLQR:~$ gcc proj3.c
aryan@LAPTOP-SSBAGLQR:~$ ./a.out
Writing to pipe - Message 1 is Hi
Reading from pipe - Message 1 is Hi
Writing to pipe - Message 2 is Hello
Reading from pipe - Message 2 is Hello
aryan@LAPTOP-SSBAGLQR:~$ gcc proj3.c
aryan@LAPTOP-SSBAGLQR:~$ ./a.out
In Parent: Writing to pipe 1 - Message is Hi
In Child: Reading from pipe 1 - Message is Hi
In Child: Writing to pipe 2 - Message is Hello
In Parent: Reading from pipe 2 - Message is Hello
aryan@LAPTOP-SSBAGLQR:~$

```