|  |  |
| --- | --- |
| Exp no: 3(A) | INTERPROCESS COMMUNICATION |
| Date: |

# AIM:

# ALGORITHM:

# SOURCE CODE:

#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[0]);

write(pipefds[1], writemessages[1], sizeof(writemessages[0]));

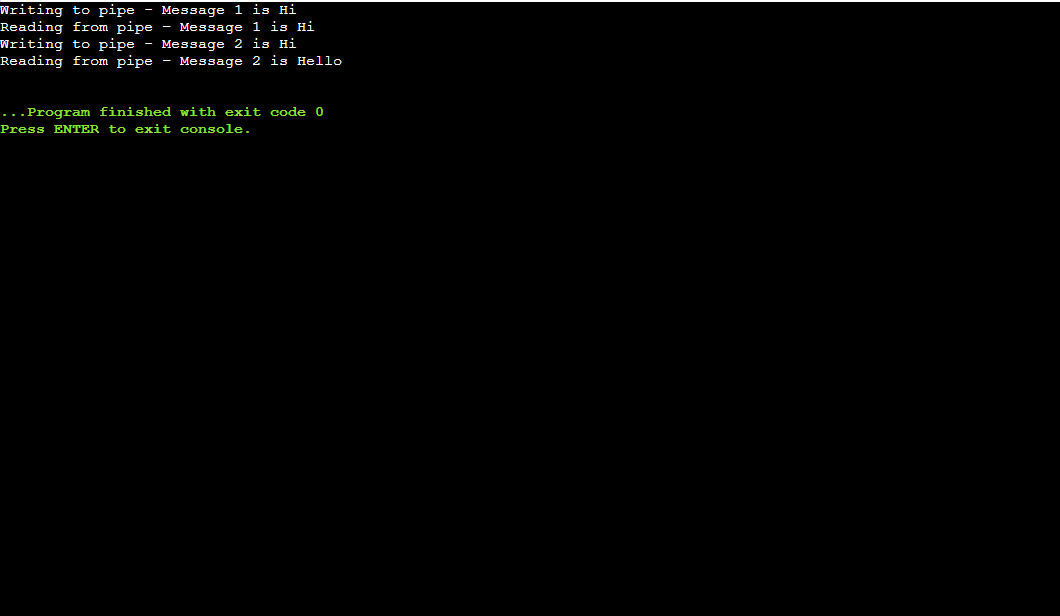
read(pipefds[0], readmessage, sizeof(readmessage));

printf("Reading from pipe – Message 2 is %s\n", readmessage);

return 0;

}

# OUTPUT:



# RESULT:

|  |  |
| --- | --- |
| EXP NO: 3(B) | SYSTEM CALLS |
| DATE: |

# AIM:

# DESCRIPTION:

|  |  |
| --- | --- |
| EXP NO: 3(B) I | OPEN AND CLOSE |
| DATE: |

# AIM:

# ALGORITHM:

# SOURCE CODE:

#include<stdio.h>

#include <fcntl.h>

int main()

{

int fd1 = open("foo.txt", O\_RDONLY);

if (fd1 < 0)

{

perror("c1");

exit(1);

}

printf("opened the fd = % d\n", fd1);

if (close(fd1) < 0)

{

perror("c1");

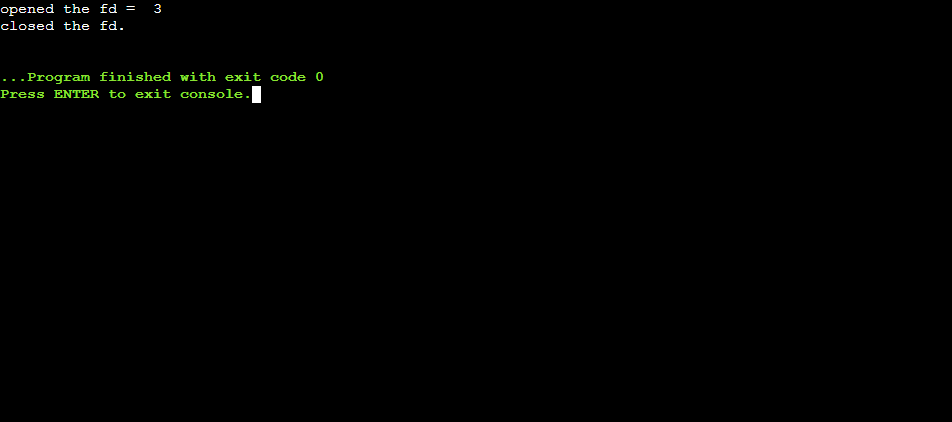
exit(1);

}

printf("closed the fd.\n");

}

# OUTPUT:



# RESULT:

|  |  |
| --- | --- |
| EXP NO 3(B) II | READ AND WRITE |
| DATE: |

# AIM:

# ALGORITHM:

# SOURCE CODE(WRITE):

#include<stdio.h>

#include <fcntl.h>

main()

{

int sz;

int fd = open("foo.txt", O\_WRONLY | O\_CREAT | O\_TRUNC, 0644);

if (fd < 0)

{

perror("r1");

exit(1);

}

sz = write(fd, "hello geeks\n", strlen("hello geeks\n"));

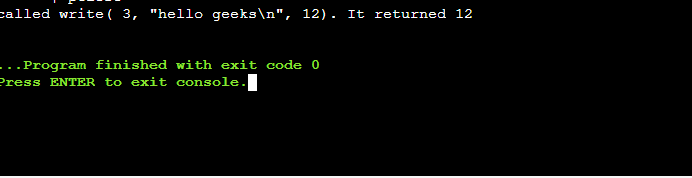
printf("called write(% d, \"hello geeks\\n\", %d)."

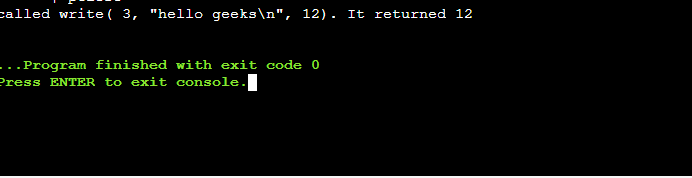
" It returned %d\n", fd, strlen("hello geeks\n"), sz);

close(fd);

}

# OUTPUT:





#include<stdio.h>

#include<unistd.h>

#include<fcntl.h>

#include<stdlib.h>

int main()

{

char c;

int fd1 = open("sample.txt", O\_RDONLY, 0);

int fd2 = open("sample.txt", O\_RDONLY, 0);

read(fd1, &c, 1);

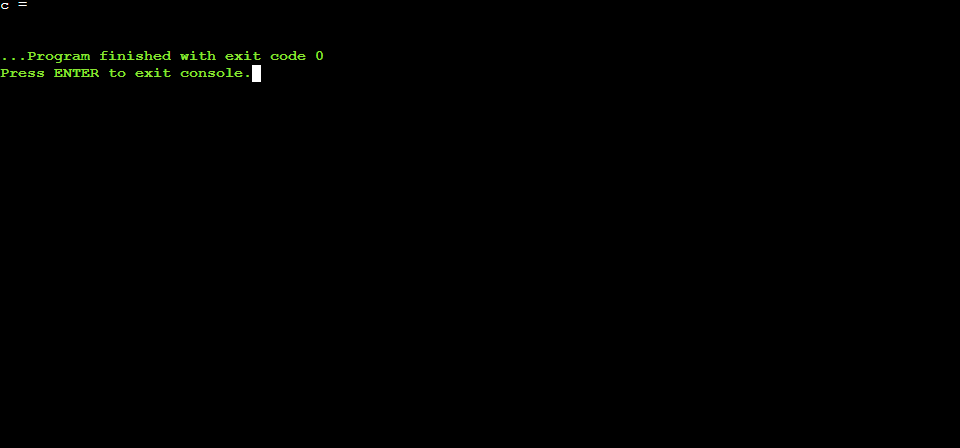
read(fd2, &c, 1);

printf("c = %c\n", c);

exit(0);

}

# OUTPUT:



# RESULT:

|  |  |
| --- | --- |
| EXP NO 3(B) III | CREATE AND FORK |
| DATE: |

# AIM:

# ALGORITHM:

# SOURCE CODE (FORK):

#include <stdio.h>

#include <string.h>

#include <sys/types.h>

#define MAX\_COUNT 200

#define BUF\_SIZE 100

void main(void)

{

pid\_t pid;

int i;

char buf[BUF\_SIZE];

fork();

pid = getpid();

for (i = 1; i <= MAX\_COUNT; i++) {

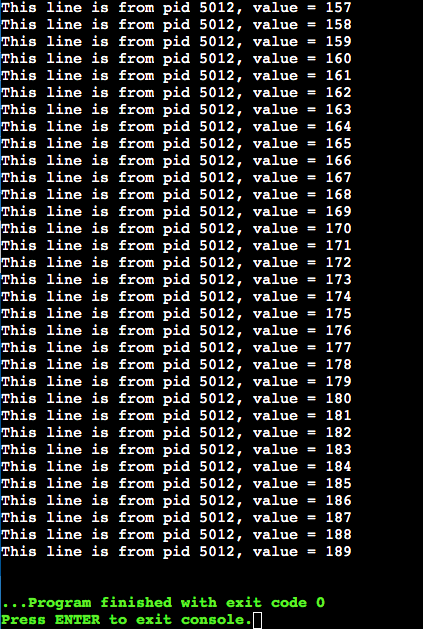
sprintf(buf, "This line is from pid %d, value = %d\n", pid, i);

write(1, buf, strlen(buf));

}

}

# OUTPUT:



# SOURCE CODE (CREATE):

#include<stdio.h>

#include<fcntl.h>

#include<errno.h>

extern int errno;

int main()

{

int fd = open("foo.txt", O\_RDONLY | O\_CREAT);

printf("fd = %d/n", fd);

if (fd ==-1)

{

printf("Error Number % d\n", errno);

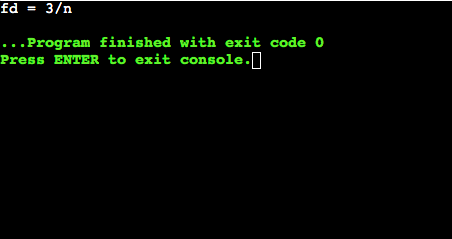
perror("Program");

}

return 0;

}

# OUTPUT:



# RESULT: