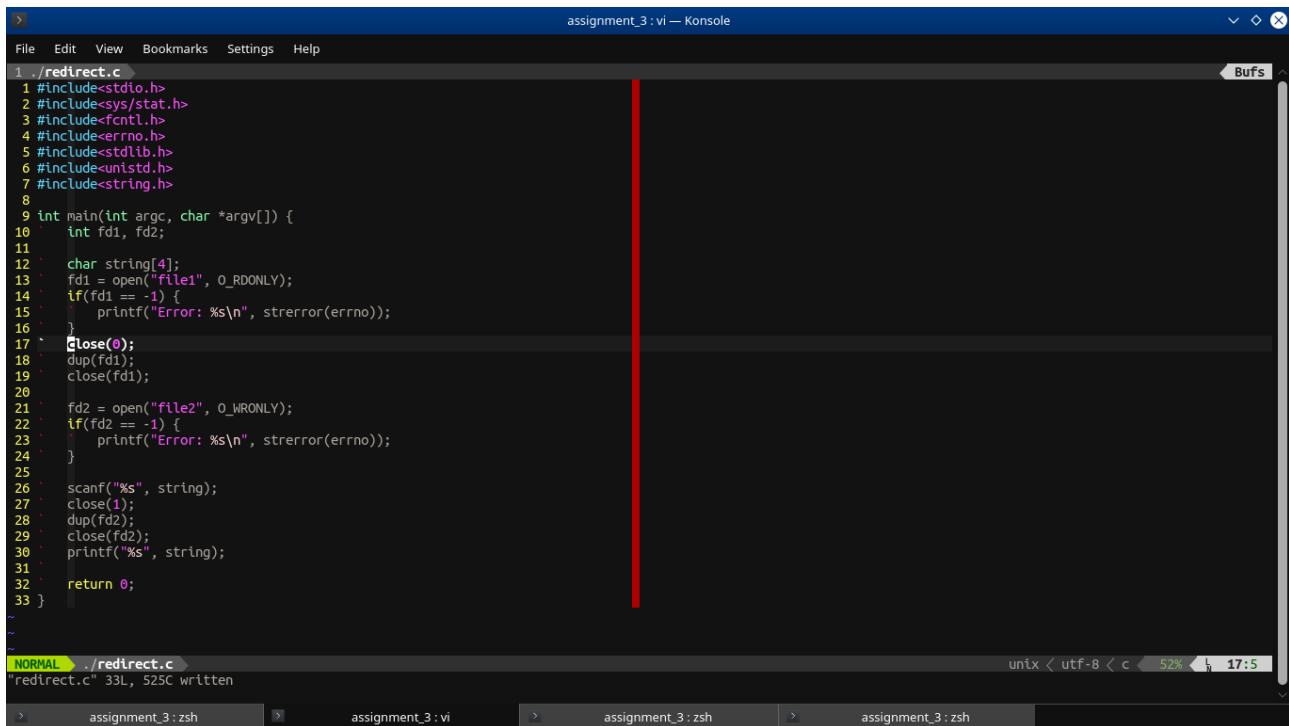


## Assignment 3

Srishti Shelke 111603056  
Niramay Vaidya 111605075

**Question 1: Using dup function redirect stdin to file1 and stdout to file2. Read a line using scanf and write the same using printf. Verify the contents of both files.**

**Code:**



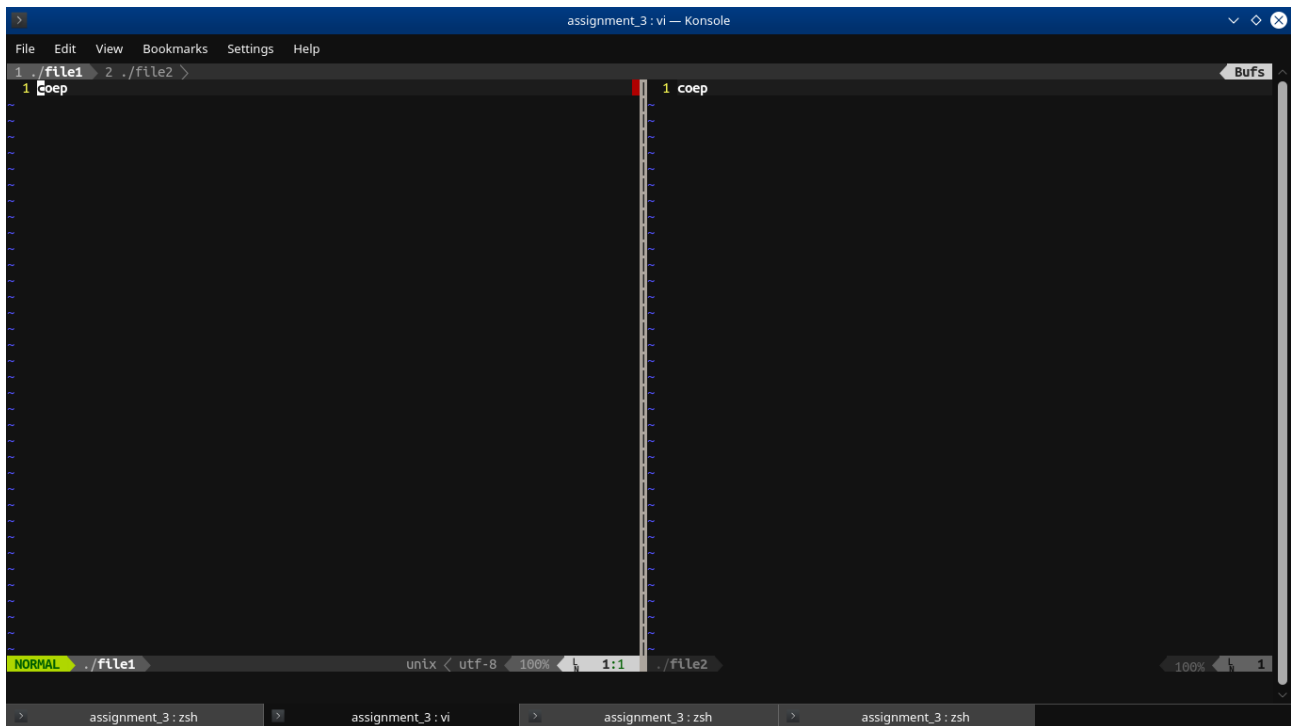
```
1 ./redirect.c
2 #include<stdio.h>
3 #include<sys/stat.h>
4 #include<fcntl.h>
5 #include<errno.h>
6 #include<unistd.h>
7 #include<string.h>
8
9 int main(int argc, char *argv[]) {
10     int fd1, fd2;
11
12     char string[4];
13     fd1 = open("file1", O_RDONLY);
14     if(fd1 == -1) {
15         printf("Error: %s\n", strerror(errno));
16     }
17     close(0);
18     dup(fd1);
19     close(fd1);
20
21     fd2 = open("file2", O_WRONLY);
22     if(fd2 == -1) {
23         printf("Error: %s\n", strerror(errno));
24     }
25
26     scanf("%s", string);
27     close(1);
28     dup(fd2);
29     close(fd2);
30     printf("%s", string);
31
32     return 0;
33 }
```

NORMAL ./redirect.c unix < utf-8 < c 52% 17:5

redirect.c" 33L, 525C written

assignment\_3: zsh assignment\_3: vi assignment\_3: zsh assignment\_3: zsh

## Output:



The screenshot shows a terminal window titled "assignment\_3 : vi — Konsole". The window is split into two panes. The left pane shows the contents of "file1" with the text "1 coep". The right pane shows the contents of "file2" with the text "1 coep". The status bar at the bottom indicates the current file is "file1" and the cursor is at line 1, column 1. The terminal also shows a menu bar with "File", "Edit", "View", "Bookmarks", "Settings", and "Help".

**Question 2: Create a file on the command prompt. Invoke functions: open, write, access and chmod on this file. Print all the time values of this file and the parent directory after each function call. Write details about your observation**

**Explanation:** For open, none of the times for the file change. Atime doesn't change since it only changes for a file while reading from it. Mtime doesn't change since it only changes for a file while writing to it. Ctime doesn't change because open does not cause any of the inode information related to the file to change.

For write to the file, mtime and ctime changes but atime does not. Mtime changes because contents of the file change and since mtime is part of inode information related to a file, ctime also changes.

For access, none of the times for the file change. The explanation is same as given for open.

For chmod, only ctime associated with the file changes but mtime and atime do not. This is because file permissions of a file are part of the inode information of a file.

For the parent directory, none of the times change for all the function calls above since atime for a directory changes when it is read, mtime changes when the contents (inode and filename pair) of the directory change and ctime changes when the inode information of the directory itself is changed.

## Code:

```
assignment_3 : vi — Konsole
File Edit View Bookmarks Settings Help
1 ./times.c
2 #include <stdio.h>
3 #include <sys/stat.h>
4 #include <fcntl.h>
5 #include <errno.h>
6 #include <stdlib.h>
7 #include <unistd.h>
8 #include <string.h>
9 #include <time.h>
10 /* atime for a file changes when read is called or when an application like cat,
11 * vi, etc. opens the file
12 */
13
14 int main(int argc, char *argv[]) {
15     int fd;
16     struct stat statbuf;
17     char buf[8];
18     // printf("Open function call:\n");
19     printf("Times for file before open:\n");
20     if (stat("before_after", &statbuf) == -1) {
21         perror("Stat failed\n");
22         return errno;
23     }
24     printf("atime: %ld mtime: %ld ctime: %ld\n", statbuf.st_atime, statbuf.st_mtime, statbuf.st_ctime);
25     printf("Times for parent directory before open:\n");
26     if (stat(".", &statbuf) == -1) {
27         perror("Stat failed\n");
28         return errno;
29     }
30     printf("atime: %ld mtime: %ld ctime: %ld\n", statbuf.st_atime, statbuf.st_mtime, statbuf.st_ctime);
31     fd = open("before_after", O_RDWR);
32     if (fd == -1) {
33         printf("Error: %s\n", strerror(errno));
34         return errno;
35     }
36     printf("Times for file after open:\n");
NORMAL ./times.c
NoExtraConfDetected: No .ycm_extra_conf.py file detected, so no compile flags are available. Thus no semanti
assignment_3 : zsh assignment_3 : vi assignment_3 : vi assignment_3 : vi
```

```
assignment_3: vi — Konsole
File Edit View Bookmarks Settings Help
1 ./times.c
37 if (stat("before_after", &statbuf) == -1) {
38     perror("Stat failed\n");
39     return errno;
40 }
41 printf("atime: %ld mtime: %ld ctime: %ld\n", statbuf.st_atime, statbuf.st_mtime, statbuf.st_ctime);
42 printf("Time for parent directory after open:\n");
43 if (stat(".", &statbuf) == -1) {
44     perror("Stat failed\n");
45     return errno;
46 }
47 printf("atime: %ld mtime: %ld ctime: %ld\n", statbuf.st_atime, statbuf.st_mtime, statbuf.st_ctime);
48 sleep(1);
49 // printf("Write function call:\n");
50 printf("Times for file before write:\n");
51 if (stat("before_after", &statbuf) == -1) {
52     perror("Stat failed\n");
53     return errno;
54 }
55 printf("atime: %ld mtime: %ld ctime: %ld\n", statbuf.st_atime, statbuf.st_mtime, statbuf.st_ctime);
56 printf("Times for parent directory before write:\n");
57 if (stat(".", &statbuf) == -1) {
58     perror("Stat failed\n");
59     return errno;
60 }
61 printf("atime: %ld mtime: %ld ctime: %ld\n", statbuf.st_atime, statbuf.st_mtime, statbuf.st_ctime);
62 if (write(fd, "coep", sizeof("coep")) == -1) {
63     printf("Error: %s\n", strerror(errno));
64     return errno;
65 }
66 printf("Times for file after write:\n");
67 if (stat("before_after", &statbuf) == -1) {
68     perror("Stat failed\n");
69     return errno;
70 }
71 printf("atime: %ld mtime: %ld ctime: %ld\n", statbuf.st_atime, statbuf.st_mtime, statbuf.st_ctime);
72 printf("Time for parent directory after write:\n");
NORMAL ./times.c
unix < utf-8 < c 52% 72:4
```

```
assignment_3: vi — Konsole
File Edit View Bookmarks Settings Help
1 ./times.c
73 if (stat(".", &statbuf) == -1) {
74     perror("Stat failed\n");
75     return errno;
76 }
77 printf("atime: %ld mtime: %ld ctime: %ld\n", statbuf.st_atime, statbuf.st_mtime, statbuf.st_ctime);
78 sleep(1);
79 // printf("\nAccess function call:\n");
80 printf("Times for file before access:\n");
81 if (stat("before_after", &statbuf) == -1) {
82     perror("Stat failed\n");
83     return errno;
84 }
85 printf("atime: %ld mtime: %ld ctime: %ld\n", statbuf.st_atime, statbuf.st_mtime, statbuf.st_ctime);
86 printf("Times for parent directory before access:\n");
87 if (stat(".", &statbuf) == -1) {
88     perror("Stat failed\n");
89     return errno;
90 }
91 printf("atime: %ld mtime: %ld ctime: %ld\n", statbuf.st_atime, statbuf.st_mtime, statbuf.st_ctime);
92 if (access("before_after", F_OK) == -1) {
93     printf("Error: %s\n", strerror(errno));
94     return errno;
95 }
96 printf("Times for file after access:\n");
97 if (stat("before_after", &statbuf) == -1) {
98     perror("Stat failed\n");
99     return errno;
100 }
101 printf("atime: %ld mtime: %ld ctime: %ld\n", statbuf.st_atime, statbuf.st_mtime, statbuf.st_ctime);
102 printf("Time for parent directory after access:\n");
103 if (stat(".", &statbuf) == -1) {
104     perror("Stat failed\n");
105     return errno;
106 }
107 printf("atime: %ld mtime: %ld ctime: %ld\n", statbuf.st_atime, statbuf.st_mtime, statbuf.st_ctime);
108 sleep(1);
NORMAL ./times.c
unix < utf-8 < c 53% 73:4
```

```
assignment_3: vi — Konsole
File Edit View Bookmarks Settings Help
1 ./times.c
109 printf("Times for file before chmod:\n");
110 if (stat("before_after", &statbuf) == -1) {
111     perror("Stat failed\n");
112     return errno;
113 }
114 printf("atime: %ld mtime: %ld ctime: %ld\n", statbuf.st_atime, statbuf.st_mtime, statbuf.st_ctime);
115 printf("Times for parent directory before chmod:\n");
116 if (stat(".", &statbuf) == -1) {
117     perror("Stat failed\n");
118     return errno;
119 }
120 printf("atime: %ld mtime: %ld ctime: %ld\n", statbuf.st_atime, statbuf.st_mtime, statbuf.st_ctime);
121 if (chmod("before_after", 0666) == -1) {
122     printf("Error: %s\n", strerror(errno));
123 }
124 printf("Times for file after chmod:\n");
125 if (stat("before_after", &statbuf) == -1) {
126     perror("Stat failed\n");
127     return errno;
128 }
129 printf("atime: %ld mtime: %ld ctime: %ld\n", statbuf.st_atime, statbuf.st_mtime, statbuf.st_ctime);
130 printf("Time for parent directory after chmod:\n");
131 if (stat(".", &statbuf) == -1) {
132     perror("Stat failed\n");
133     return errno;
134 }
135 printf("atime: %ld mtime: %ld ctime: %ld\n", statbuf.st_atime, statbuf.st_mtime, statbuf.st_ctime);
136 close(fd);
137 return 0;
138 }

NORMAL ./times.c
unix < utf-8 < c 79% 109:4
assignment_3: zsh assignment_3: vi assignment_3: vi assignment_3: zsh
```

## Output:

```
assignment
File Edit View Bookmarks Settings Help
-rwxrwxr-x 1 niramay niramay 9088 Sep  3 12:28 server*
-rw-rw-r-- 1 niramay niramay 915 Sep  3 12:26 server.c
-rw-r--r-- 1 niramay niramay 4650 Sep  3 14:27 times.c
-rw-r--r-- 1 niramay niramay 20480 Sep  3 14:43 .times.c.swp
niramay@Niramay: ~/Documents/sem7/AUP/assignment_3$ cc times.c -o times
niramay@Niramay: ~/Documents/sem7/AUP/assignment_3$ ./times
Times for file before open:
atime: 1567499820 mtime: 1567494892 ctime: 1567494892
Times for parent directory before open:
atime: 1567502083 mtime: 1567502083 ctime: 1567502083
Times for file after open:
atime: 1567499820 mtime: 1567494892 ctime: 1567494892
Time for parent directory after open:
atime: 1567502083 mtime: 1567502083 ctime: 1567502083
Times for file before write:
atime: 1567499820 mtime: 1567494892 ctime: 1567494892
Times for parent directory before write:
atime: 1567502083 mtime: 1567502083 ctime: 1567502083
Times for file after write:
atime: 1567499820 mtime: 1567502086 ctime: 1567502086
Time for parent directory after write:
atime: 1567502083 mtime: 1567502083 ctime: 1567502083
Times for file before access:
atime: 1567502087 mtime: 1567502086 ctime: 1567502086
Times for parent directory before access:
atime: 1567502083 mtime: 1567502083 ctime: 1567502083
Times for file after access:
atime: 1567502087 mtime: 1567502086 ctime: 1567502086
Time for parent directory after access:
atime: 1567502083 mtime: 1567502083 ctime: 1567502083
Times for file before chmod:
atime: 1567502087 mtime: 1567502086 ctime: 1567502086
Times for parent directory before chmod:
atime: 1567502083 mtime: 1567502083 ctime: 1567502083
Times for file after chmod:
atime: 1567502087 mtime: 1567502086 ctime: 1567502088
Time for parent directory after chmod:
atime: 1567502083 mtime: 1567502083 ctime: 1567502083
niramay@Niramay: ~/Documents/sem7/AUP/assignment_3$
```

Question 3: Write a program that prints the owner and file type of files. By inputting a directory, the program should *read* the directory and print the above information for all files in the directory.

Code:

```
> assignment_3

File Edit View Bookmarks Settings Help

1 ./print_file_info.c
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <string.h>
5 #include <sys/stat.h>
6 #include <unistd.h>
7 #include <limits.h>
8 #include <dirent.h>
9 #include <pwd.h>
10 #include <errno.h>
11
12 int main(int argc, char *argv[]) {
13     if (argc != 2) {
14         printf("Usage: ./print_file_info <directory_name>\n");
15         exit(0);
16     }
17     DIR *d;
18     struct dirent *dir;
19     struct stat st;
20     char *type, *input;
21     struct passwd *pwd;
22     int len;
23     d = opendir(argv[1]);
24     if (d == NULL) {
25         perror("Opendir failed\n");
26         return errno;
27     }
28     if (argv[1][strlen(argv[1]) - 1] != '/') {
29         len = strlen(argv[1]) + 1;
30         strcat(argv[1], "/");
31     } else {
32         len = strlen(argv[1]);
33     }
34     printf("%s : %s : %s\n", "file", "owner", "type");
35     while ((dir = readdir(d)) != NULL) {
36         input = (char *)malloc(len + strlen(dir->d_name) + 1);
37     }
38 }

NORMAL ./print_file_info.c
NoExtraConfDetected: No .ycm_extra_conf.py file detected, so no compile flags are av

> assignment_3 : zsh
> assignment_3 : vi
> assign_3 : zsh
```

File Edit View Bookmarks Settings Help

1 ./print\_file\_info.c

```
37     strcpy(input, argv[1]);
38     strcat(input, dir->d_name);
39     if (lstat(input, &st) == 0) {
40         //TODO find out why getpwnam fails
41         /* how does getpwuid know to return struct passwd corresponding to a
42          * particular file when a uid is passed, since many files can have
43          * this same uid
44          */
45         // printf("%d\n", st.st_uid);
46         pwd = getpwuid(st.st_uid);
47         if (pwd == NULL) {
48             perror("Getpwuid failed\n");
49             return 0;
50         }
51         printf("%s : %s : ", input, pwd->pw_name);
52         if ((st.st_mode & S_IFMT) == S_IFREG) {
53             type = "regular";
54         }
55         else if ((st.st_mode & S_IFMT) == S_IFDIR) {
56             type = "directory";
57         }
58         else if ((st.st_mode & S_IFMT) == S_IFCHR) {
59             type = "character special";
60         }
61         else if ((st.st_mode & S_IFMT) == S_IFBLK) {
62             type = "block special";
63         }
64         else if ((st.st_mode & S_IFMT) == S_IFLNK) {
65             type = "symbolic link";
66         }
67         else if ((st.st_mode & S_IFMT) == S_IFIFO) {
68             type = "fifo";
69         }
70         else if ((st.st_mode & S_IFMT) == S_IFSOCK) {
71             type = "socket";
72     }
```

NORMAL ./print\_file\_info.c

assignment\_3 : zsh

assignment\_3 : vi

assignment\_3 : zsh

```
assignment_3
File Edit View Bookmarks Settings Help
1 ./print_file_info.c
73     else {
74         type = "unknown";
75     }
76 }
77 else {
78     perror("Lstat failed\n");
79     return errno;
80 }
81 printf("%s\n", type);
82 free(input);
83 }
84 return 0;
85 }
```

Output:

```
assignment_3
File Edit View Bookmarks Settings Help
niramay@Niramay ~/Documents/sem7/AUP/assignment_3 ./print_file_info files/
file : owner : type
files/.. : niramay : directory
files/symlink : niramay : directory
files/temp : niramay : regular
files/character : root : character special
files/regular : temp : regular
files/link : niramay : symbolic link
files/temp.c : niramay : regular
files/. : niramay : directory
files/block : root : block special
files/fifo : test : fifo
niramay@Niramay ~/Documents/sem7/AUP/assignment_3
```

**Question4:** Create a FIFO file and write the programs for client-server communication. Print the size of the FIFO file during:

1. Before client-server starts writing to FIFO
2. After client writing a message, but before the server reading it.
3. After client writing a message and after the server reading it.

**Explanation:** Since the FIFO file does not actually exist on the file system and is just an entry made to exist for programs to reference it, and all the inter process communication is handled at the kernel level itself, the file size of FIFO is zero under all the three conditions mentioned in the question.



Code:

Server-

```
assignment_3

File Edit View Bookmarks Settings Help

1 ./server.c+
11 #define FIFO_NAME "america"
12 int main(int argc, char *argv[]) {
13     char s[100];
14     int num, fd;
15     struct stat st;
16     printf("Waiting for writers\n");
17     fd = open(FIFO_NAME, O_RDONLY);
18     if (fd == -1) {
19         perror("Open failed\n");
20         return errno;
21     }
22     printf("Got a writer\n");
23     do {
24         if (fstat(fd, &st) == -1) {
25             perror("Stat failed\n");
26             return errno;
27         }
28         if ((num = read(fd, s, 300)) == -1) {
29             perror("Read failed\n");
30             if (close(fd) == -1) {
31                 perror("Close failed\n");
32                 return errno;
33             }
34             return errno;
35         }
36         else {
37             s[num] = '\0';
38             printf("Read %d bytes: %s\n", num, s);
39             printf("Size of FIFO after server reading: %d\n", (int)st.st_size);
40         }
41     } while (num > 0);
42     if (close(fd) == -1) {
43         perror("Close failed\n");
44         return errno;
45     }
46     return 0;

INSERT ./server.c +

assignment_3 : zsh assignment_3 : zsh assignment_3 : zsh
```

## Client-

```
assignment_3: vi — Konsole
File Edit View Bookmarks Settings Help
1 ./client.c
9 #define FIFO_NAME "america"
10
11 int main(int argc, char *argv[1]) {
12     char s[300];
13     int num, fd;
14     struct stat st;
15     if (mkfifo(FIFO_NAME, S_IFIFO | 0666) == -1) {
16         perror("Mkfifo failed\n");
17         return errno;
18     }
19     printf("Waiting for readers\n");
20     fd = open(FIFO_NAME, O_WRONLY);
21     if (fd == -1) {
22         perror("Open failed\n");
23         return errno;
24     }
25     printf("Got a reader, provide text\n");
26     while (gets(s), !feof(stdin)) {
27         if (fstat(fd, &st) == -1) {
28             perror("Stat failed\n");
29             return errno;
30         }
31         printf("Size of FIFO before client writing: %d\n", (int)st.st_size);
32         if ((num = write(fd, s, strlen(s))) == -1) {
33             perror("Write failed\n");
34             if (close(fd) == -1) {
35                 perror("Close failed\n");
36                 return errno;
37             }
38             return errno;
39         }
40         else {
41             printf("Wrote %d bytes\n", num);
42             printf("Size of FIFO after client writing but before server reading: %d\n", (int)st.st_size);
43         }
44     }
45 }
NORMAL ./client.c
"client.c" 50L, 1116C written
unix < utf-8 < c 40% 20:33
assignment_3: zsh assignment_3: vi assignment_3: zsh assignment_3: zsh
```

## Output:

```
assignment_3: zsh — Konsole
File Edit View Bookmarks Settings Help
> coep
\ No newline at end of file
niranmay@niranmay: ~/Documents/sen7/AUP/assignment_3$ cc client.c -o client
client.c: In function 'main':
client.c:26:9: warning: implicit declaration of function 'gets' [-Wimplicit-function-declaration]
   while (gets(s), !feof(stdin)) {
         ^
/tmp/ccRzXvqa.o: In function 'main':
client.c:(.text+0x1a2): warning: the 'gets' function is dangerous and should not be used.
niranmay@niranmay: ~/Documents/sen7/AUP/assignment_3$ ./client
Mkfifo failed
: File exists
niranmay@niranmay: ~/Documents/sen7/AUP/assignment_3$ rm america
rm: remove fifo 'america'? y
niranmay@niranmay: ~/Documents/sen7/AUP/assignment_3$ ./client
Waiting for readers
Open failed
: No such device or address
niranmay@niranmay: ~/Documents/sen7/AUP/assignment_3$ cc client.c -o client
client.c: In function 'main':
client.c:26:9: warning: implicit declaration of function 'gets' [-Wimplicit-function-declaration]
   while (gets(s), !feof(stdin)) {
         ^
/tmp/ccyK1sGE.o: In function 'main':
client.c:(.text+0x1a2): warning: the 'gets' function is dangerous and should not be used.
niranmay@niranmay: ~/Documents/sen7/AUP/assignment_3$ ./client
Mkfifo failed
: File exists
niranmay@niranmay: ~/Documents/sen7/AUP/assignment_3$ rm america
rm: remove fifo 'america'? y
niranmay@niranmay: ~/Documents/sen7/AUP/assignment_3$ ./client
Waiting for readers
Got a reader, provide text
abcde
Size of FIFO before client writing: 0
Wrote 5 bytes
Size of FIFO after client writing but before server reading: 0
^C
niranmay@niranmay: ~/Documents/sen7/AUP/assignment_3$
```

```
assignment_3: zsh — Konsole
File Edit View Bookmarks Settings Help
file : owner : type
files/.. : niranay : directory
files/symlink : niranay : directory
files/temp : niranay : regular
files/character : root : character special
files/regular : temp : regular
files/link : niranay : symbolic link
files/temp.c : niranay : regular
files/. : niranay : directory
files/block : root : block special
files/fifo : test : fifo
niranay@niranay ~/Documents/sem7/AUP/assignment_3 ls 10179 17:15:20
111603056_111605075_AUP_assignment_3 a.out client file1 files print_file_info.c server times
america before after client.c file2 print_file_info redirect.c server.c times.c
niranay@niranay ~/Documents/sem7/AUP/assignment_3 gcc server.c 10186 17:26:07
niranay@niranay ~/Documents/sem7/AUP/assignment_3 ./a.out 10181 17:26:32
Waiting for writers
sdf
sdf^C
niranay@niranay ~/Documents/sem7/AUP/assignment_3 vi server INT(-2) + 10182 17:26:40
niranay@niranay ~/Documents/sem7/AUP/assignment_3 vi server.c 10183 17:27:28
niranay@niranay ~/Documents/sem7/AUP/assignment_3 gcc server.c 10184 17:27:50
niranay@niranay ~/Documents/sem7/AUP/assignment_3 ./a.out server 10185 17:28:23
Waiting for writers
1
1
3
^C
niranay@niranay ~/Documents/sem7/AUP/assignment_3 vi server.c INT(-2) + 10186 17:29:17
niranay@niranay ~/Documents/sem7/AUP/assignment_3 vi redirect.c 10187 17:36:53
niranay@niranay ~/Documents/sem7/AUP/assignment_3 cc server.c -o server 10189 17:49:18
niranay@niranay ~/Documents/sem7/AUP/assignment_3 ./server 10218 18:05:11
Waiting for writers
Got a writer
Read 5 bytes: abcde
Size of FIFO after server reading: 0
Read 0 bytes:
Size of FIFO after server reading: 0
niranay@niranay ~/Documents/sem7/AUP/assignment_3 10226 18:06:30
assignment_3: zsh assignment_3: vi assignment_3: zsh assignment_3: zsh
```