Assignment 1

Srishti Shelke 111603056
Niramay Vaidya 111605075

1. Assume that you have to read 10 characters from the beginning of an existing file and then to write “hello” to the end of the file. Write a program to achieve this without using lseek function.

Code-

```
assignment_1 : vi — Konsole
File  Edit  View  Bookmarks  Settings  Help
./rdwr.c                                                                    Bufs
int main(int argc, char *argv[]) {
    if (argc != 2) {
        printf("Usage: ./rdwr <filename>\n");
        return EINVAL;
    }
    // file already exists, opening in read write flag
    int fd = open(argv[1], O_RDWR);
    if (fd == -1) {
        perror("Open failed\n");
        return errno;
    }
    int bytes;
    char chars[10];
    // assumption that 10 characters have to be read hence there will be at
    // least 10 characters in the file
    if ((bytes = read(fd, chars, 10)) == -1) {
        perror("Read failed\n");
        return errno;
    }
    if (bytes < 10) {
        printf("Less bytes read than requested\n");
        exit(0);
    }
    int ch;
    while (read(fd, &ch, 1) == 1 && ch != EOF);
    if ((bytes = write(fd, "hello", 5)) == -1) {
        perror("Write failed\n");
        return errno;
    }
    if (bytes < 5) {
        printf("Less bytes written than requested\n");
        exit(0);
    }
    if (close(fd) == -1) {
        perror("Close failed\n");
        return errno;
    }
    return 0;
}
NORMAL  ./rdwr.c                                   unix < utf-8 < c  100%  47:1
assignment_1 : vi
```

Output-

```
niramay@Niramay  ~/Documents/sem7/AUP/assignment_1  cat temp_rdwr.txt          10060  18:12:52
College of Engineering Pune
niramay@Niramay  ~/Documents/sem7/AUP/assignment_1  ./rdwr temp_rdwr.txt      10061  18:12:55
niramay@Niramay  ~/Documents/sem7/AUP/assignment_1  cat temp_rdwr.txt         10062  18:13:02
College of Engineering Pune
hello
niramay@Niramay  ~/Documents/sem7/AUP/assignment_1  ./rdwr temp_rdwr.txt      10063  18:13:07
niramay@Niramay  ~/Documents/sem7/AUP/assignment_1  cat temp_rdwr.txt         10064  18:13:09
College of Engineering Pune
hellohello
niramay@Niramay  ~/Documents/sem7/AUP/assignment_1                            10065  18:13:10
```

2. Linux provides a function as given below to truncate file to specific length.

int truncate (const char *path, off_t len); return 0 on success. On error, return -1,

Write a program to emulate this function (donot use any built-in function). Use cat command to demonstrate the contents of the truncated file.

Code-

```c
int mytruncate(int fd, int offset) {
    int size = lseek(fd, 0, SEEK_END);
    if (size == -1) {
        // perror("Lseek failed in truncate\n");
        // return errno;
        return 1;
    }
    if (size == offset) {
        return 0;
    }
    else if (size < offset) {
        if (lseek(fd, 1, SEEK_END) == -1) {
            return 1;
        }
        char ch = '\0';
        for (int i = offset - size; i > 0; i--) {
            if (write(fd, &ch, 1) != 1) {
                return 1;
            }
        }
        return 0;
    }
    else { // size > offset
        int pos;
        if ((pos = lseek(fd, offset, SEEK_SET)) == -1) {
            return 1;
        }
        char ch = '\0';
        for (int i = pos; i <= size; i++) {
            if (write(fd, &ch, 1) != 1) {
                return 1;
            }
        }
        return 0;
    }
}
```

```c

int main(int argc, char *argv[]) {
    if (argc != 3) {
        printf("Usage: ./truncate <filename> <offset>\n");
        return EINVAL;
    }
    // file already exists, opening in read write flag
    int fd = open(argv[1], O_RDWR);
    if (fd == -1) {
        perror("Open failed\n");
        return errno;
    }
    if (mytruncate(fd, atoi(argv[2])) == 1) {
        printf("Truncate failed\n");
        exit(0);
    }
    if (close(fd) == -1) {
        perror("Close failed\n");
        return errno;
    }
    return 0;
}
```

Output-

```
niramay@Niramay  ~/Documents/sem7/AUP/assignment_1  cat temp_truncate.txt
College of Engineering Pune
niramay@Niramay  ~/Documents/sem7/AUP/assignment_1  ./truncate temp_truncate.txt 20
niramay@Niramay  ~/Documents/sem7/AUP/assignment_1  cat temp_truncate.txt
College of Engineeri
niramay@Niramay  ~/Documents/sem7/AUP/assignment_1  od -c temp_truncate.txt
0000000   C   o   l   l   e   g   e       o   f       E   n   g   i   n
0000020   e   e   r   i  \0  \0  \0  \0  \0  \0  \0  \0  \0
0000035
niramay@Niramay  ~/Documents/sem7/AUP/assignment_1  ./truncate temp_truncate.txt 50
niramay@Niramay  ~/Documents/sem7/AUP/assignment_1  cat temp_truncate.txt
College of Engineeri
```

```
niramay@Niramay  ~/Documents/sem7/AUP/assignment_1  od -c temp_truncate.txt
0000000   C   o   l   l   e   g   e       o   f       E   n   g   i   n
0000020   e   e   r   i  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0
0000040  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0
0000060  \0  \0  \0  \0
0000064
niramay@Niramay  ~/Documents/sem7/AUP/assignment_1
```

3. What will be the output for the program with following operation?

a. Create a new file “f1” and write “abcde” in it and close
b. Open the file “f1” for writing with O_APPEND flag
c. lseek to the beginning of the file
d. Replace the existing data in the file with “12345”

Justify your answer.

Code-

```c
int main(int argc, char *argv[]) {
    if (argc != 2) {
        printf("Usage: ./append <filename>\n");
        return EINVAL;
    }
    int fd = open(argv[1], O_WRONLY | O_CREAT, S_IRUSR | S_IWUSR | S_IRGRP | S_IWGRP | S_IROTH | S_IWOTH);
    if (fd == -1) {
        perror("Open failed\n");
        return errno;
    }
    int bytes;
    if ((bytes = write(fd, "abcde", 5)) == -1) {
        perror("Write failed\n");
        return errno;
    }
    if (bytes < 5) {
        printf("Less bytes written than requested\n");
        exit(0);
    }
    if (close(fd) == -1) {
        perror("Close failed\n");
        return errno;
    }
    int fd_new = open(argv[1], O_WRONLY | O_APPEND, S_IRUSR | S_IWUSR | S_IRGRP | S_IWGRP | S_IROTH | S_IWOTH);
    if (fd_new == -1) {
        perror("Open failed\n");
        return errno;
    }
    if (lseek(fd, 0, SEEK_SET) == -1) {
        perror("Lseek failed\n");
        return errno;
    }
    if ((bytes = write(fd_new, "12345", 5)) == -1) {
        perror("Write failed\n");
        return errno;
    }
```

```c
    if (bytes < 5) {
        printf("Less bytes written than requested\n");
        exit(0);
    }
    if (close(fd_new) == -1) {
        perror("Close failed\n");
        return errno;
    }
    return 0;
}
```

Output-

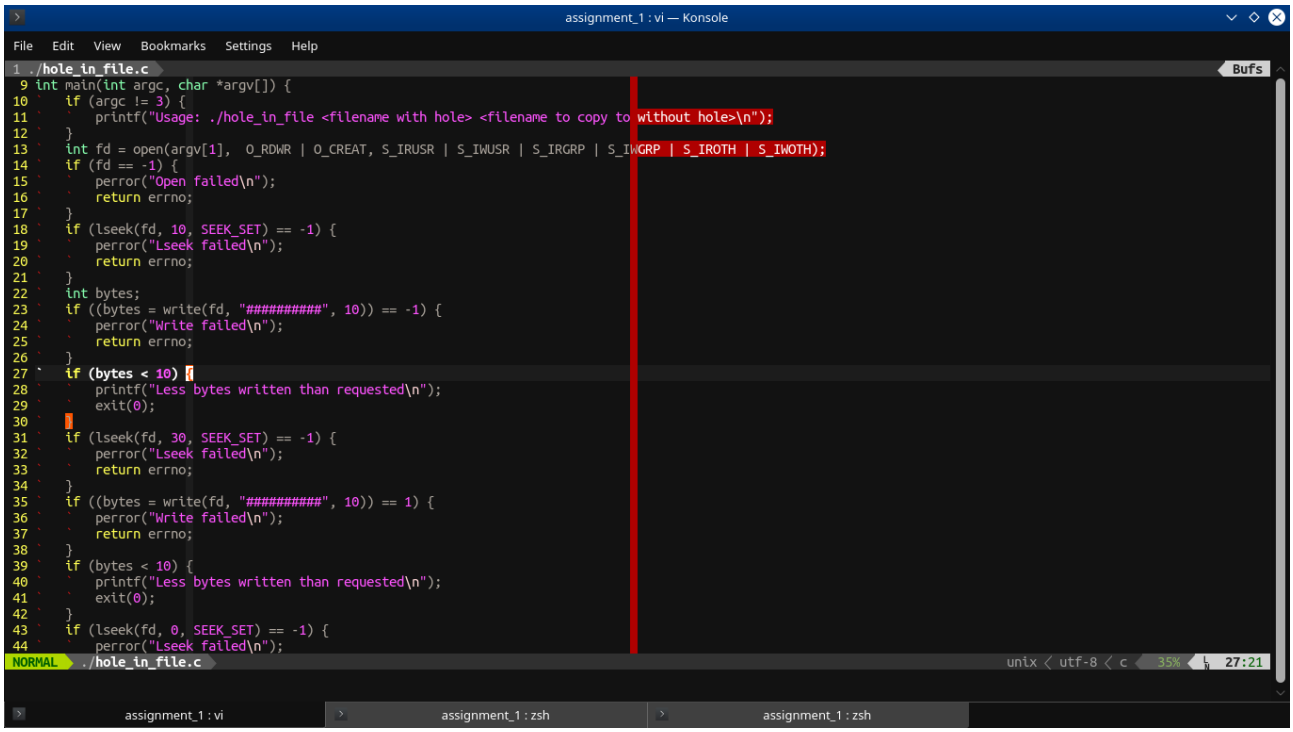
```
niramay@Niramay  ~/Documents/sem7/AUP/assignment_1  ./append temp_append.txt
niramay@Niramay  ~/Documents/sem7/AUP/assignment_1  ll temp_append.txt
-rw-rw-r-- 1 niramay niramay 10 Oct 20 18:27 temp_append.txt
niramay@Niramay  ~/Documents/sem7/AUP/assignment_1  cat temp_append.txt
abcde12345%
niramay@Niramay  ~/Documents/sem7/AUP/assignment_1  ./append temp_append.txt
niramay@Niramay  ~/Documents/sem7/AUP/assignment_1  cat temp_append.txt
abcde1234512345%
niramay@Niramay  ~/Documents/sem7/AUP/assignment_1
```

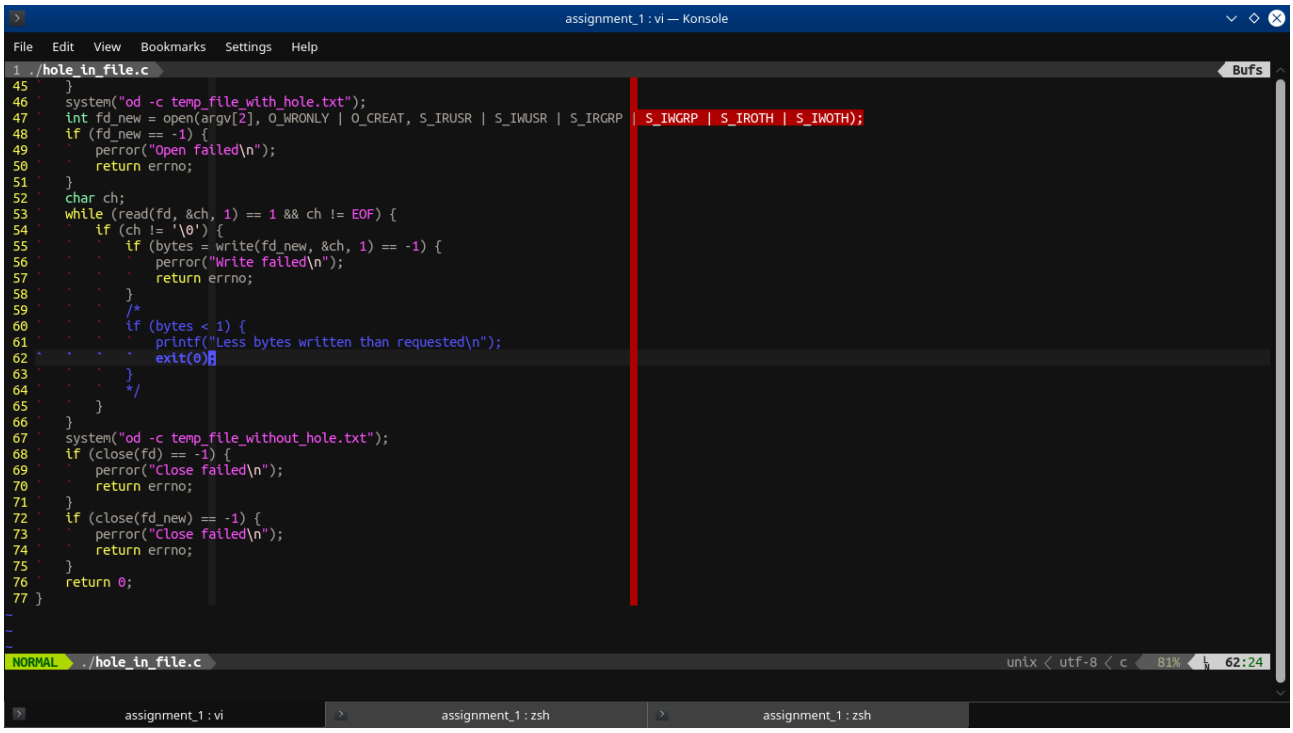
Explanation-

When the file is opened with the O_APPEND mode, calling lseek to change the file offset does not have any effect in terms of writing to the lseeked position. Irrespective of the successful call to lseek, when such a file is written into, the data always gets appended to the end of the file and does not replace existing data at the lseeked position.

4. Write a program to create a file with a hole: write any 10 bytes at an offset of 10 and another 10 bytes at an offset of 30. Using “system” function, invoke “od” command and view the contents. Later copy the contents of the file to another file without writing the bytes of 0. Once again verify the contents by invoking “system” with “od”.

Code-





Output-

```
niramay@Niramay  ~/Documents/sem7/AUP/assignment_1  ./hole_in_file temp_file_with_hole.txt temp_file_without_hole.txt        ✔  10101  18:28:11
0000000  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0   #   #   #   #   #   #
0000020   #   #   #   #  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0   #   #
0000040   #   #   #   #   #   #   #   #
0000050
0000000   #   #   #   #   #   #   #   #   #   #   #   #   #   #   #   #
0000020   #   #   #   #
0000024
niramay@Niramay  ~/Documents/sem7/AUP/assignment_1                                                                          ✔  10103  18:33:38
```