Advanced Unix Programming
Lab 1

Purva Tendulkar : 111403049

Q1. 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 :

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main() {
        FILE *fp;
        char filename[15], ch, hello[5];
        int i = 0;

        printf("Enter the name of the file\n");
        scanf("%s", filename);
        fp = fopen(filename, "r+");
        if (fp == NULL) {
                printf("An error was encountered while accessing file!\n");
                return 1;
        }

        /* Reading */
        ch = fgetc(fp);
        while(ch != EOF) {
                i++;
                if (i <= 10)
                        printf("%c", ch);
                ch = fgetc(fp);
        }
        printf("\n");

        /* Writing */
        strcpy(hello, "hello");
        for (i = 0; i < 5; i++)
                fputc(hello[i], fp);

        fclose(fp);

        return 0;
}
```

Input & Output :

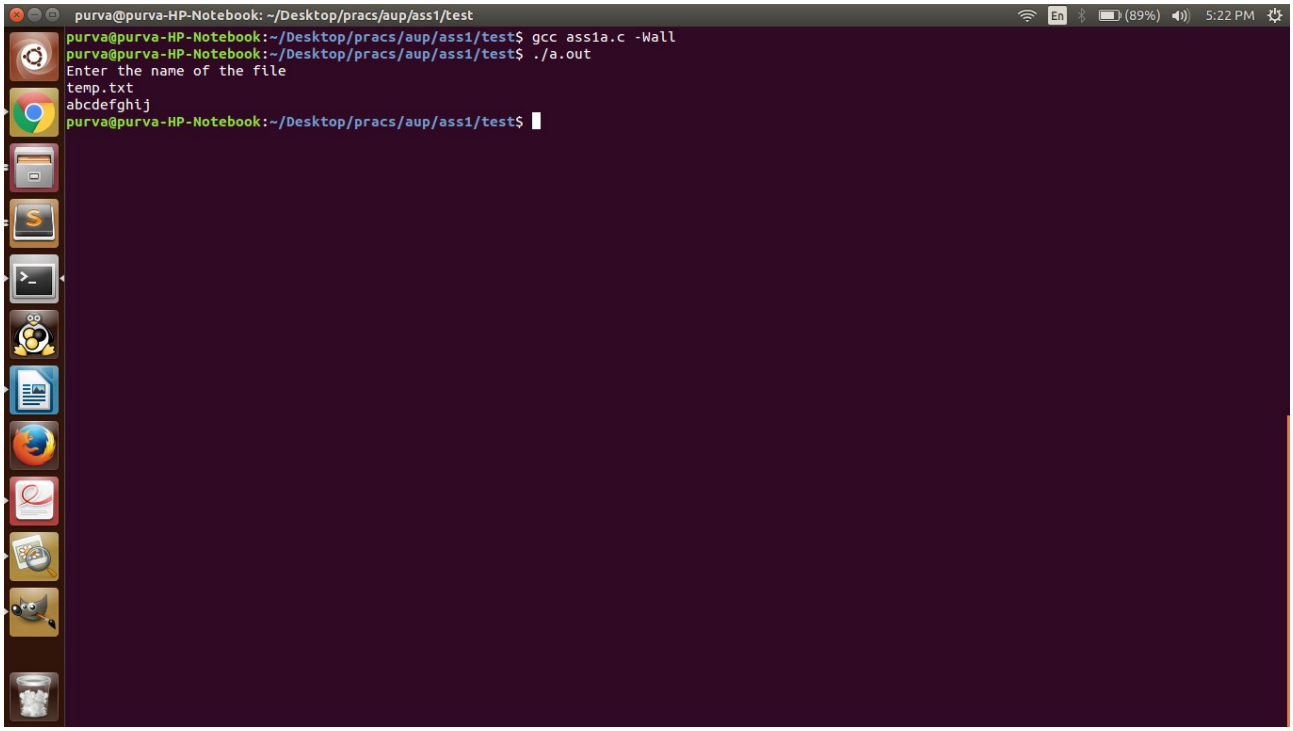
1. Contents in file temp.txt :
   abcdefghijklmnopqrstuvwxyz

2. Display on terminal :

gcc ass1a.c -Wall
./a.out
Enter the name of the file
temp.txt
abcdefghij

3. Contents in file temp.txt :
   abcdefghijklmnopqrstuvwxyzhello

Screenshot :



Q2. 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. Use cat command to demonstrate.

Code :

```
#include <stdio.h>
#include <string.h>

int truncate (const char *path, off_t len) {
        /* Return 0 on success. Return -1 on error */
        FILE *fp1, *fp2;
        char ch, filename[] = "temp.txt";
        int i;

        fp1 = fopen(path, "r");
        fp2 = fopen(filename, "w+");
        if (fp1 == NULL || fp2 == NULL) {
                printf("An error was encountered while accessing file!\n");
                return -1;
        }
```

```
        ch = fgetc(fp1);
        while (ch != EOF) {
                i++;
                if (i > len)
                        break;
                fputc(ch, fp2);
                ch = fgetc(fp1);
        }

        fclose(fp1);
        fclose(fp2);

        fp1 = fopen(path, "w+");
        fp2 = fopen(filename, "r");

        ch = fgetc(fp2);
        while(ch != EOF) {
                fputc(ch, fp1);
                ch = fgetc(fp2);
        }

        fclose(fp1);
        fclose(fp2);
        remove(filename);

        return 0;
}

int main() {
        char filename[15], path[100];
        int x;

        printf("Enter the directory of the file (eg./home/test/)\n");
        scanf("%s", path);
        printf("Enter the name of the file in above directory (eg.filename.txt)\n");
        scanf("%s", filename);
        strcat(path, filename);

        x = truncate(path, 7);

        if (x == 0)
                printf("Operation successful!\n");
        else
                printf("Oops! Error encountered in truncate operation!\n");

        return 0;
}
```

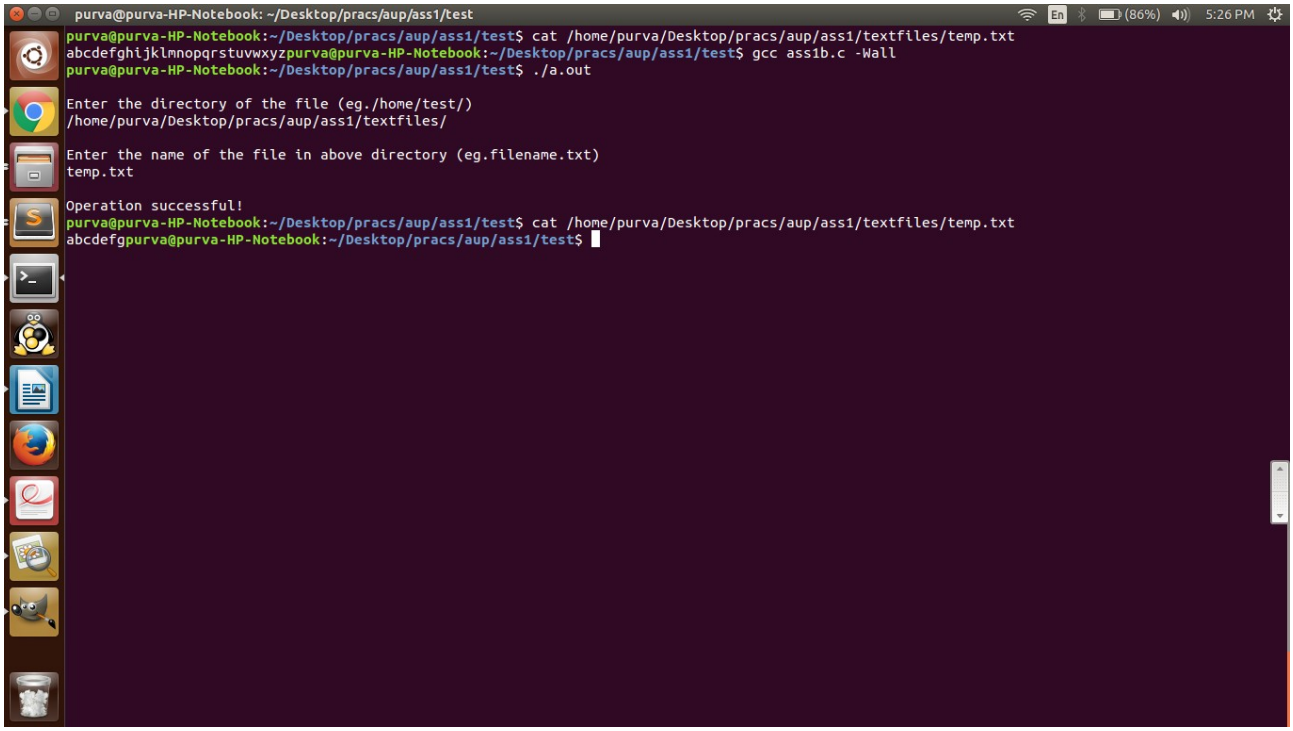
Input & Output :

1. Contents in file temp.txt :
   abcdefghijklmnopqrstuvwxyz

2. Display on terminal :
   cat /home/purva/Desktop/pracs/aup/ass1/textfiles/ emp.txt
   abcdefghijklmnopqrstuvwxyz
   gcc ass1b.c -Wall
   ./a.out
   Enter the directory of the file (eg./home/test/)
   /home/purva/Desktop/pracs/aup/ass1/textfiles/
   Enter the name of the file in above directory (eg.filename.txt)
   temp.txt
   Operation successful!
   cat /home/purva/Desktop/pracs/aup/ass1/textfiles/temp.txt
   abcdefg

3. Contents in file temp.txt :
   abcdefg

Screenshot :



Q3. 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 :

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>
```

```
int main() {
	FILE *fp;
	char filename[] = "f1", abcde[] = "abcde";
	int i, fd;

	fp = fopen(filename, "w+");
	if (fp == NULL) {
		printf("An error was encountered while accessing file!\n");
		return -1;
	}

	for (i = 0; i < 5; i++)
		fputc(abcde[i], fp);
	fclose(fp);

	fd = open("f1", O_WRONLY | O_APPEND);
	lseek(fd, 0, SEEK_SET);
	write(fd, "12345", 5);
	close(fd);

	return 0;
}
```

Input & Output :

1. Display on terminal :
   gcc ass1c.c -Wall
   ./a.out

2. Contents in f1 :
   abcde12345

Justification :

When you open a file with O_APPEND, all data gets written to the end, regardless of whatever the current file pointer is from the latest call to lseek(2) or the latest read/write operation.

Q4. 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 :

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <sys/stat.h>
#include <fcntl.h>

int main() {
	char filename1[15], filename2[15], string[] = "abcdefghij", ch;
	int x;
```

```
        FILE *fp1, *fp2;

        /* Writing to file */
        printf("Enter the filename\n");
        scanf("%s", filename1);
        fp1 = fopen(filename1, "w+");
        if (fp1 == NULL) {
                printf("An error was encountered while accessing file!\n");
                return -1;
        }

        fseek(fp1, 10, SEEK_CUR);
        fwrite(string, 1, 10, fp1);

        fseek(fp1, 10, SEEK_CUR);
        fwrite(string, 1, 10, fp1);

        fclose(fp1);

        /* Copying file contents to other file */
        printf("Enter the file to copy to\n");
        scanf("%s", filename2);
        fp1 = fopen(filename1, "r");
        fp2 = fopen(filename2, "w+");
        if (fp1 == NULL || fp2 == NULL) {
                printf("An error was encountered while accessing file!\n");
                return -1;
        }

        x = fread(&ch, 1, 1, fp1);
        while(x != 0) {
                if(ch != '\0')
                        fwrite(&ch, 1, 1, fp2);
                x = fread(&ch, 1, 1, fp1);
        }

        printf("Operation successful!\n");

        fclose(fp1);
        fclose(fp2);

        return 0;
}
```

Input & Output :

1. Display on terminal :
   gcc ass1d.c -Wall
   ./a.out
   Enter the filename
   temp1.txt
   Enter the file to copy to
   temp2.txt

Operation successful!

2. Contents in file temp1.txt :
   0000 0000 0000 0000 0000 6162 6364 6566
   6768 696a 0000 0000 0000 0000 0000 6162
   6364 6566 6768 696a

3. Contents in file temp2.txt :
   abcdefghijabcdefghij

4. Verification using od command :
   od -c temp1.txt

```
0000000 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0  a  b  c  d  e  f
0000020  g  h  i  j \0 \0 \0 \0 \0 \0 \0 \0 \0 \0  a  b
0000040  c  d  e  f  g  h  i  j
0000050
```

   od -c temp2.txt

```
0000000  a  b  c  d  e  f  g  h  i  j  a  b  c  d  e  f
0000020  g  h  i  j
0000024
```

Screenshot :