Ayush Goyal
190905522 CSE D Roll 62

Operating Systems Lab 1 – Working with Regular Files

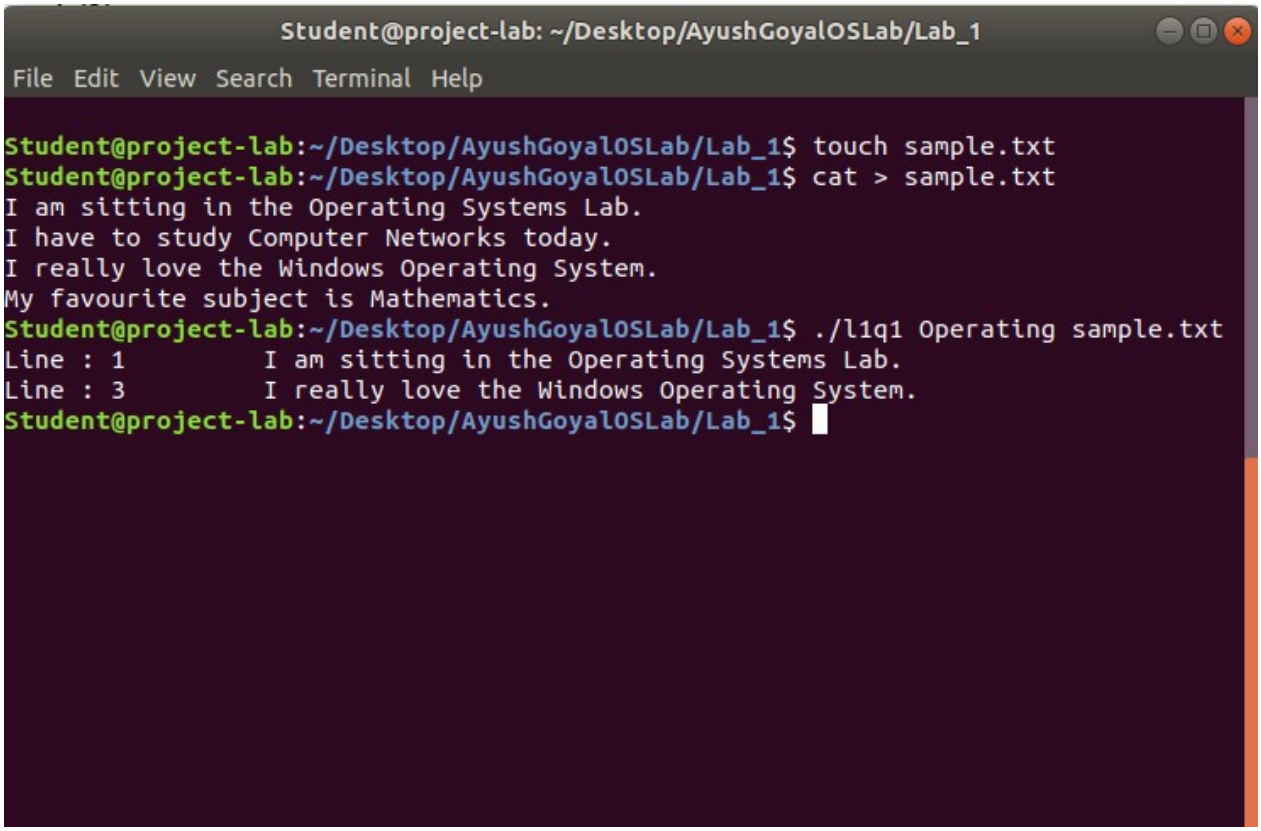
1. Write a program to print the lines of a file that contain a word given as the program argument (a simple version of grep UNIX utility).

Code:

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

int main(int argc, char const *argv[]){
	int sfd,i=0,k=0;
	char ch[100], chr;
	if(argc!=3){
		printf("\nInsufficient Arguments\n");
		exit(1);
	}
	if((sfd=open(argv[2],O_RDONLY)) == -1){
		printf("File not found.\n");
		exit(2);
	}
	while((read(sfd, &chr, 1)) > 0){
		if(chr != '\n'){
			ch[i++] = chr;
		} else{
			k++;
			ch[i] = '\0';
			i=0;
			if(strstr(ch,argv[1]) != NULL){
				printf("Line : %d\t %s\n",k,ch);
			}
		}
	}
	exit(0);
}
```

We also create a txt file: sample.txt to test out output on.

Output :



2. Write a program to list the files given as arguments, stopping every 20 lines until a key is hit. (a simple version of more UNIX utility).

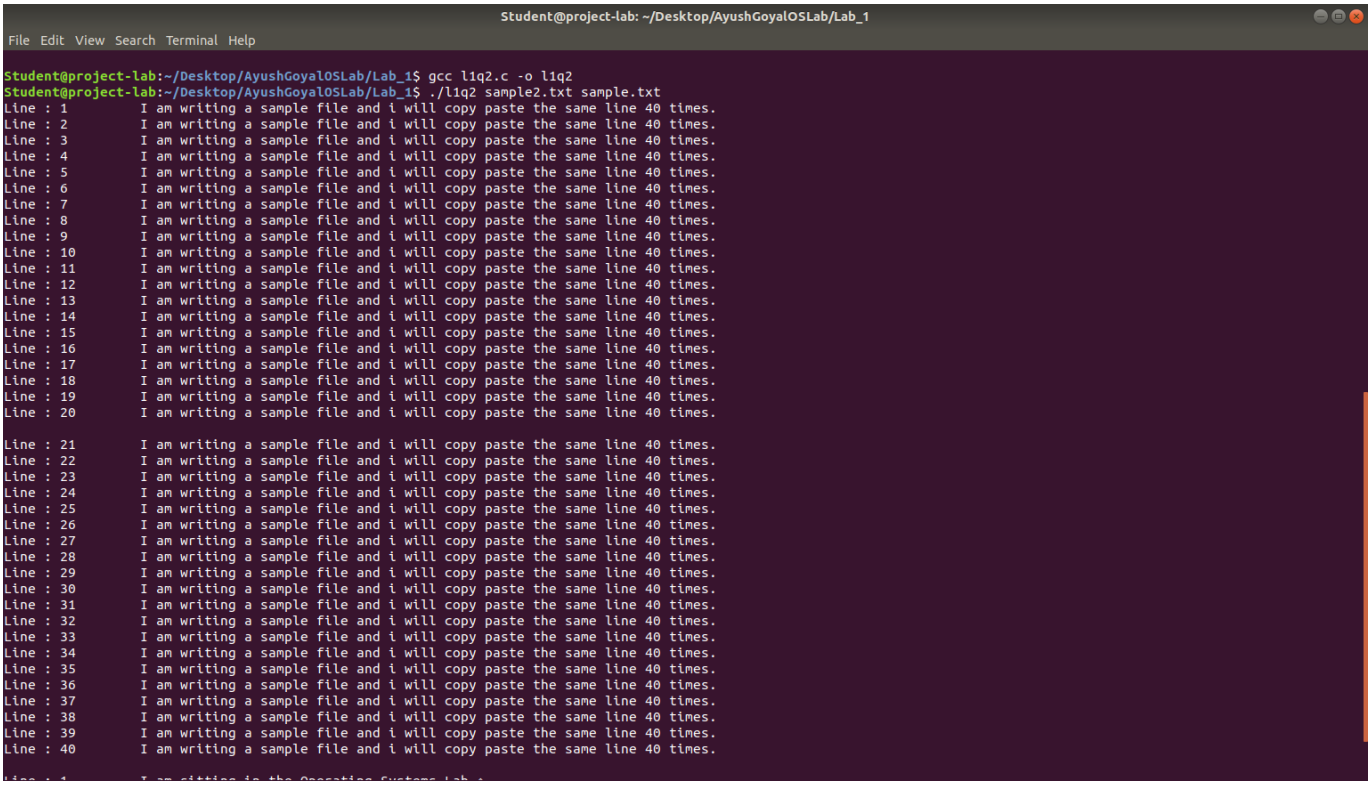
Code:

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

int main(int argc, char *argv[]){
        int sfd, sfd2, i=0, k=0, p=0;
        char ch[100], ch2[100], chr;

        if(argc != 3){
                printf("\nInsufficient Arguments\n");
                exit(1);
        }
        if((sfd = open(argv[1],O_RDONLY)) == -1){
```

```
		printf("File not found!\n");
		exit(2);
	}
	while((read(sfd, &chr, 1)) > 0){
		if(chr != '\n'){
			ch[i++] = chr;
		} else{
			k++;
			p++;
			ch[i] = '\0';
			i=0;
			printf("Line : %d\t %s\n",p,ch);
			if(k == 20){
				fgetc(stdin);
				k=0;
			}
		}
	}
	close(sfd);

	if((sfd2 = open(argv[2],O_RDONLY)) == -1){
		printf("File not found!\n");
		exit(2);
	}
	p=0;
	while((read(sfd2, &chr, 1)) > 0){
		if(chr != '\n'){
			ch2[i++] = chr;
		} else{
			k++;
			p++;
			ch[i] = '\0';
			i=0;
			printf("Line : %d\t %s\n",p,ch2);
			if(k == 20){
				fgetc(stdin);
				k=0;
			}
		}
	}
	exit(0);
}
```

We have created an additional file called “sample2.txt” where i have written 40 lines and i have also used the previously created “sample.txt” file. The output is shown as below:

Output :

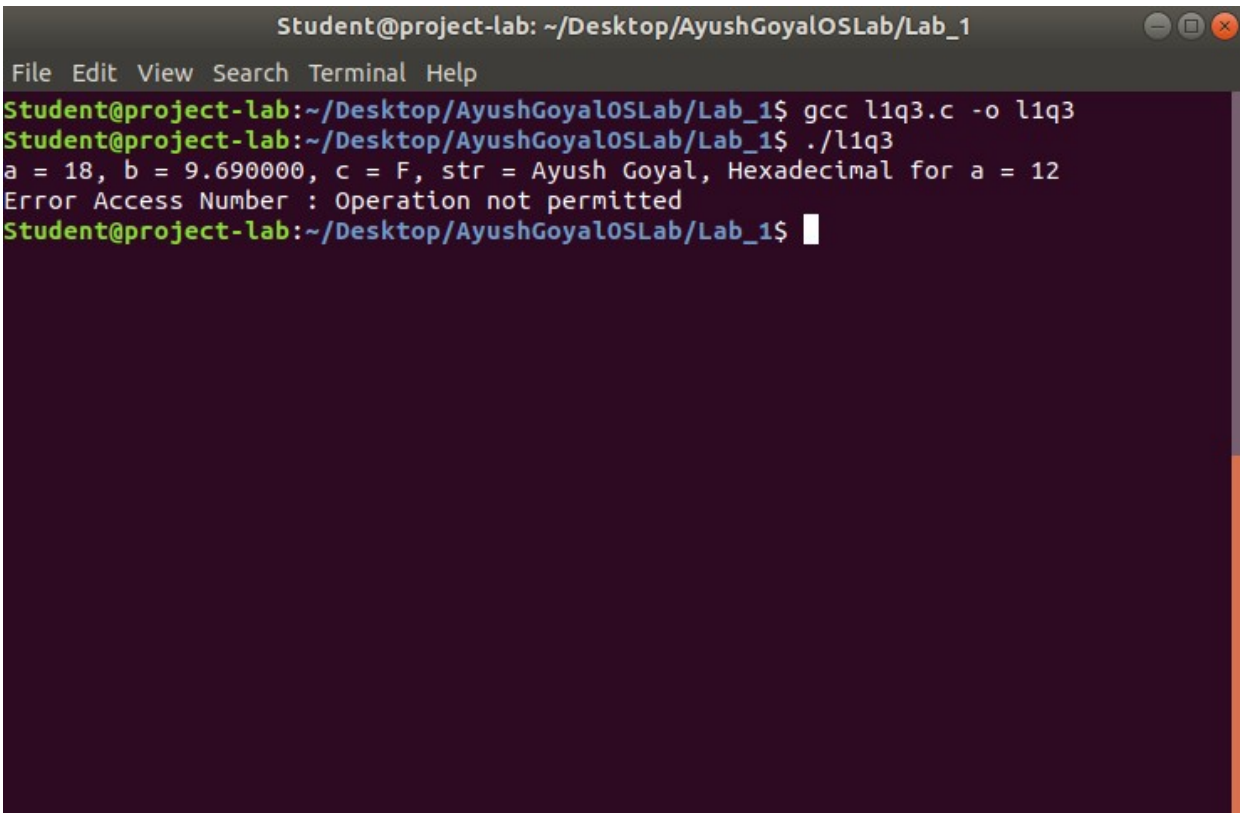


3. Demonstrate the use of different conversion specifiers and resulting output to allow the items to be printed.

Code:

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

//extern int errno

int main(){
        int a = 18;
        float b = 9.69;
        char c = 'F';
        char str[] = "Ayush Goyal";
        printf("a = %d, b = %f, c = %c, str = %s, Hexadecimal for a = %x\n",a,b,c,str,a);
        errno = EPERM;
        printf("Error Access Number : %m\n");
}
```

Output:



4. Write a program to copy character-by character copy is accomplished using calls to the functions referenced in stdio.h

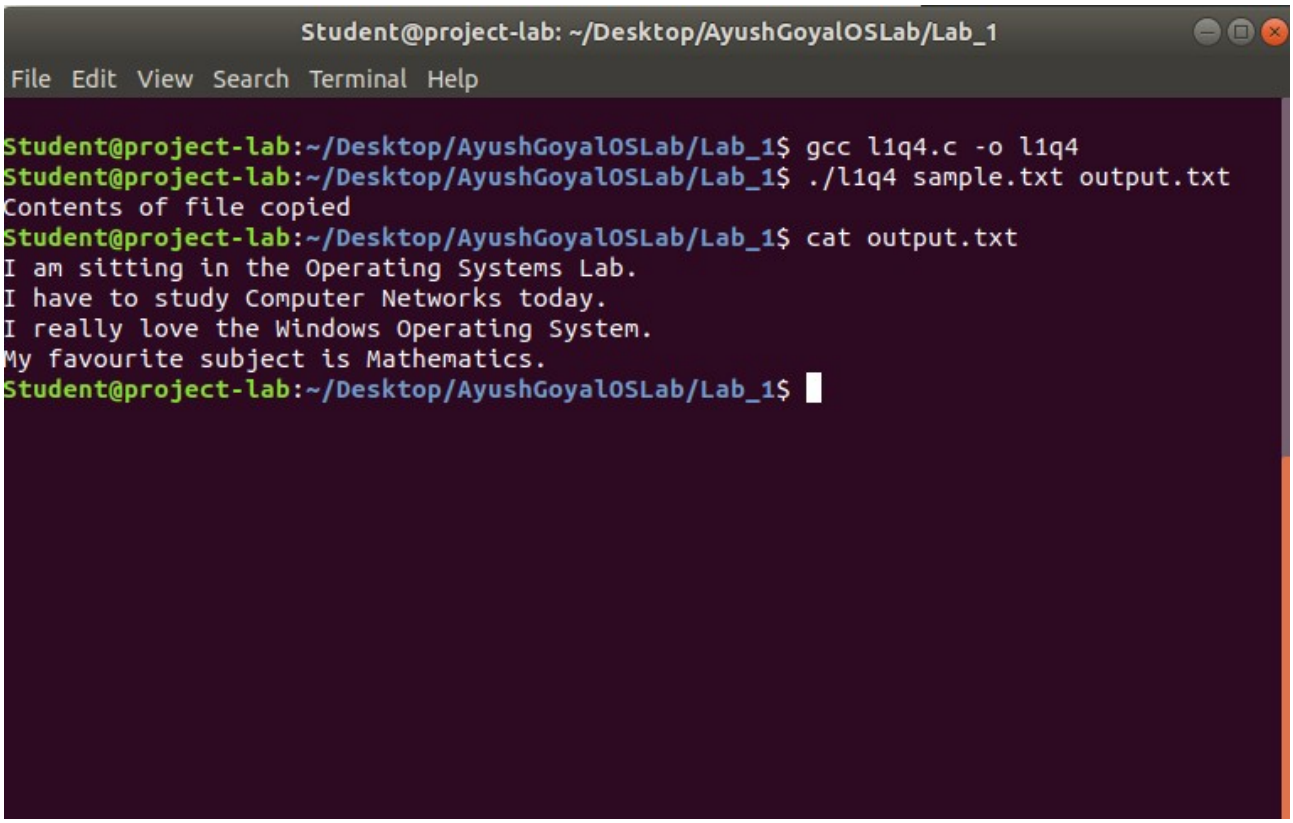
Code:

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

int main(int argc, char *argv[]){
        char c;
        int in,out;
        char buffer[128];
        int nread;
        if(argc!=3){
                printf("\nInsufficient Arguments\n");
                exit(1);
        }
        in=open(argv[1],O_RDWR);
        out=open(argv[2],O_WRONLY|O_CREAT, S_IRUSR|S_IWUSR);
        if( in==-1 || out==-1){
                printf("File not found\n");
                exit(1);
```

```
        }
        while(read(in,&c,1) == 1){
                write(out,&c,1);
        }
        printf("Contents of file copied\n");
        exit(0);
}
```

We have used the previously created file “sample.txt” and made an output file called “output.txt”.

Output:



THE END