UNP Lab
Lab Exercise -2

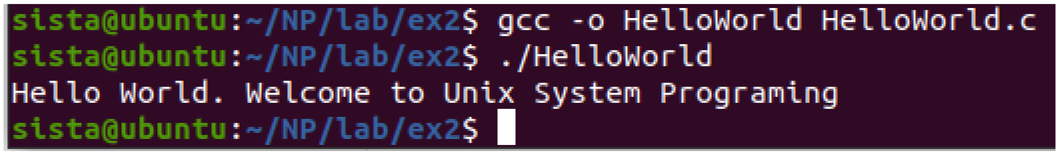
Name:Shravani SIsta
Roll No.:19MCME01

1)

a) HelloWorld.c

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

extern int errno;
void main(){
    int fd = open("Welcome.txt", O_RDWR | O_CREAT |
O_TRUNC, 0666);
    if(fd == -1){
      int errsv = errno;
      fprintf(stderr, "File Welcome.txt cannot be
created:%s\n",strerror(errsv));
      exit(1);
    }

    char *input="Hello World. Welcome to Unix System
Programing";
    if(write(fd,input,strlen(input)) == -1) {
      int errsv = errno;
      fprintf(stderr, "File Welcome.txt cannot be written
to:%s\n",strerror(errsv));
      exit(1);
    }

    if(lseek(fd,0,SEEK_SET) == -1) {
      int errsv = errno;
      fprintf(stderr, "File Welcome.txt cannot be
seeked:%s\n",strerror(errsv));
      exit(1);
    }

    char output[70];
    if(read(fd,output,100) == -1) {
      int errsv = errno;
      fprintf(stderr, "File Welcome.txt cannot be
read:%s\n",strerror(errsv));
    }
    else{
      printf("%s\n", output);
    }

    close(fd);
}
```

Output:



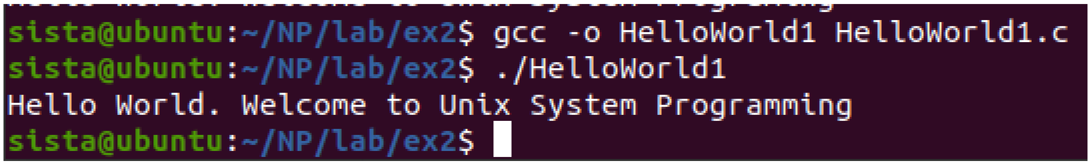
```
sista@ubuntu:~/NP/lab/ex2$ gcc -o HelloWorld HelloWorld.c
sista@ubuntu:~/NP/lab/ex2$ ./HelloWorld
Hello World. Welcome to Unix System Programing
sista@ubuntu:~/NP/lab/ex2$ 
```

b) HelloWorld1.c

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

extern int errno;
void main(){
    int fd = open("Welcome.txt", O_RDWR);
    if(fd == -1){
      int errsv = errno;
      fprintf(stderr, "File Welcome.txt cannot be
opened:%s\n",strerror(errsv));
      exit(1);
    }

    if(lseek(fd,-3,SEEK_END) == -1) {
      int errsv = errno;
      fprintf(stderr, "File Welcome.txt cannot be
seeked:%s\n",strerror(errsv));
      exit(1);
    }

    if(write(fd,"ming",4) == -1) {
      int errsv = errno;
      fprintf(stderr, "File Welcome.txt cannot be written
to:%s\n",strerror(errsv));
```

```
      exit(1);
    }

    if(lseek(fd,0,SEEK_SET) == -1) {
      int errsv = errno;
      fprintf(stderr, "File Welcome.txt cannot be
seeked:%s\n",strerror(errsv));
      exit(1);
    }

    char output[70];
    if(read(fd,output,100) == -1) {
      int errsv = errno;
      fprintf(stderr, "File Welcome.txt cannot be
read:%s\n",strerror(errsv));
    }
    else{
      printf("%s\n", output);
    }

    close(fd);
}
```

Output:



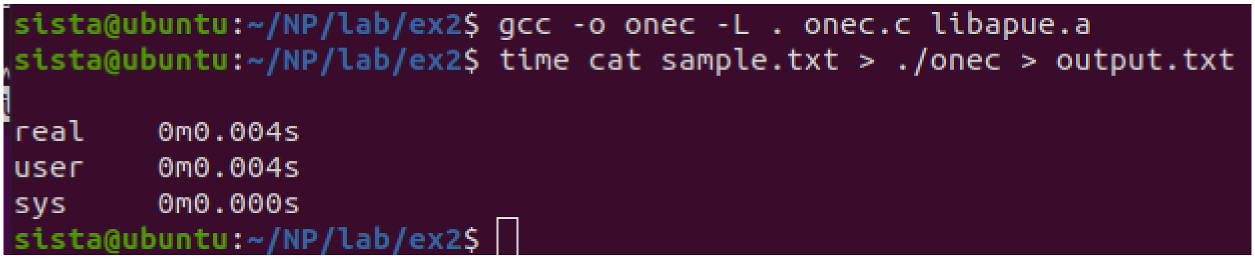
```
sista@ubuntu:~/NP/lab/ex2$ gcc -o HelloWorld1 HelloWorld1.c
sista@ubuntu:~/NP/lab/ex2$ ./HelloWorld1
Hello World. Welcome to Unix System Programming
sista@ubuntu:~/NP/lab/ex2$
```

c) onec.c(From textbook figure 3.5 from page 72)

```
#include "apue.h"

#define BUFFSIZE 4096

int main(void)
{
    int    n;
    char    buf[BUFFSIZE];

    while ((n = read(STDIN_FILENO, buf, BUFFSIZE)) > 0)
      if (write(STDOUT_FILENO, buf, n) != n)
           err_sys("write error");

    if (n < 0)
```

```
      err_sys("read error");

    exit(0);
}
```

Output:



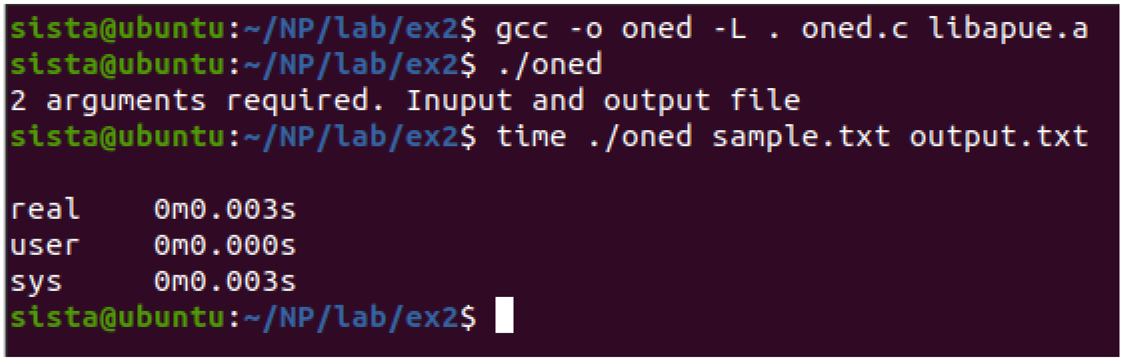
```
sista@ubuntu:~/NP/lab/ex2$ gcc -o onec -L . onec.c libapue.a
sista@ubuntu:~/NP/lab/ex2$ time cat sample.txt > ./onec > output.txt

real    0m0.004s
user    0m0.004s
sys     0m0.000s
sista@ubuntu:~/NP/lab/ex2$ 
```

Observation:
Comparing the results obtained to the results in the textbook, the user time is the only time that the time is similar. The discrepancies in the other two time values is likely due to the difference in the size of the input file and specifications of the system the program was run with/in.

d) oned.c

```
#include "apue.h"
#include<fcntl.h>
#include<errno.h>
#define BUFFSIZE 4096

extern int errno;
int main(int argc, char *argv[])
{
    if(argc != 3) {
      printf("2 arguments required. Inuput and output
file\n");
      exit(1);
    }

    int fd1 = open(argv[1],O_RDONLY);
    if(fd1 == -1){
      int errsv = errno;
      fprintf(stderr, "File %s cannot be opened:%s\n",
argv[1], strerror(errsv));
      exit(1);
    }

    int fd2 = open(argv[2],O_WRONLY);
    if(fd2 == -1){
      int errsv = errno;
      fprintf(stderr, "File %s cannot be opened:%s\n",
argv[2], strerror(errsv));
      exit(1);
    }

    int    n;
    char    buf[BUFFSIZE];

    while ((n = read(fd1, buf, BUFFSIZE)) > 0)
      if (write(fd2, buf, n) != n)
           err_sys("write error");

    if (n < 0)
      err_sys("read error");

    exit(0);
}
```

Output:



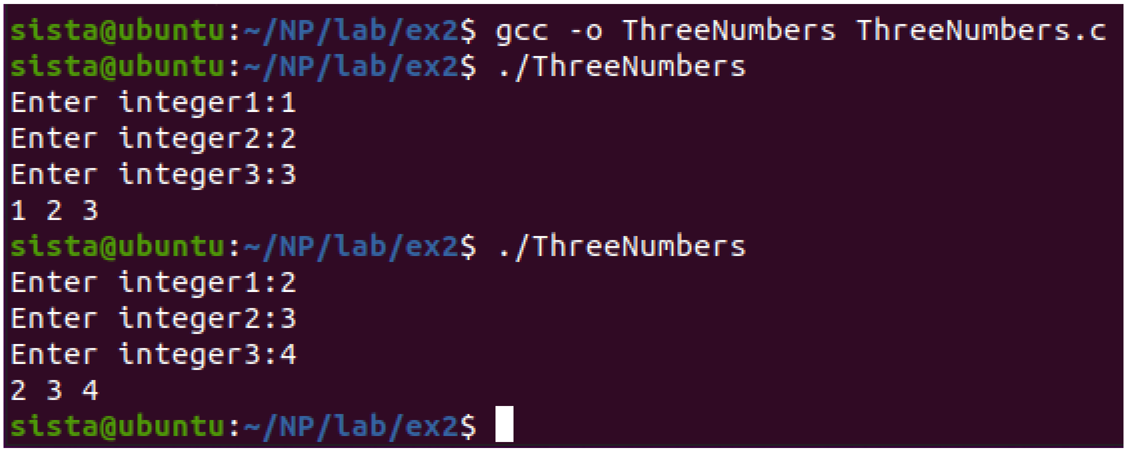
```
sista@ubuntu:~/NP/lab/ex2$ gcc -o oned -L . oned.c libapue.a
sista@ubuntu:~/NP/lab/ex2$ ./oned
2 arguments required. Inuput and output file
sista@ubuntu:~/NP/lab/ex2$ time ./oned sample.txt output.txt

real    0m0.003s
user    0m0.000s
sys     0m0.003s
sista@ubuntu:~/NP/lab/ex2$ 
```

Observations:
The results are not exactly the same for c) and d) but on running both of them multiple times they both tend to fall around the same values.

e) ThreeNumbers.c

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

void main(int argc, char* argv[]){
    int num1, num2, num3;
    printf("Enter integer1:");

    scanf("%d", &num1);

    printf("Enter integer2:");

    scanf("%d", &num2);

    printf("Enter integer3:");

    scanf("%d", &num3);

    int fd = open("threeNum.txt", O_RDWR | O_APPEND |
O_TRUNC);
    if(fd == -1){
      int errsv = errno;
      fprintf(stderr, "File threeNum.txt cannot be
opened:%s\n",strerror(errsv));
      exit(1);
    }

    //Write
    char number[10];
    sprintf(number,"%d",num1);
    if(write(fd,number,strlen(number)) == -1) {
      int errsv = errno;
      fprintf(stderr, "File threeNum.txt cannot be written
to:%s\n",strerror(errsv));
      exit(1);
    }

    if(write(fd," ",1) == -1) {
      int errsv = errno;
      fprintf(stderr, "File threeNum.txt cannot be written
to:%s\n",strerror(errsv));
      exit(1);
    }

    sprintf(number,"%d",num2);
    if(write(fd,number,strlen(number)) == -1) {
      int errsv = errno;
      fprintf(stderr, "File threeNum.txt cannot be written
to:%s\n",strerror(errsv));
      exit(1);
    }

    if(write(fd," ",1) == -1) {
      int errsv = errno;
      fprintf(stderr, "File threeNum.txt cannot be written
to:%s\n",strerror(errsv));
      exit(1);
    }

    sprintf(number,"%d",num3);
    if(write(fd,number,strlen(number)) == -1) {
      int errsv = errno;
      fprintf(stderr, "File threeNum.txt cannot be written
to:%s\n",strerror(errsv));
      exit(1);
    }

    //Set offset to beginning
    if(lseek(fd,0,SEEK_SET) == -1) {
      int errsv4 = errno;
      fprintf(stderr, "File threeNum.txt cannot be
seeked:%s\n",strerror(errsv4));
      exit(1);
    }

    //Read
    char output[70];
    if(read(fd,output,70) == -1) {
      int errsv5 = errno;
      fprintf(stderr, "File threeNum.txt cannot be
read:%s\n",strerror(errsv5));
    }
    else{
      printf("%s\n", output);
    }
    close(fd);
}
```

Output:



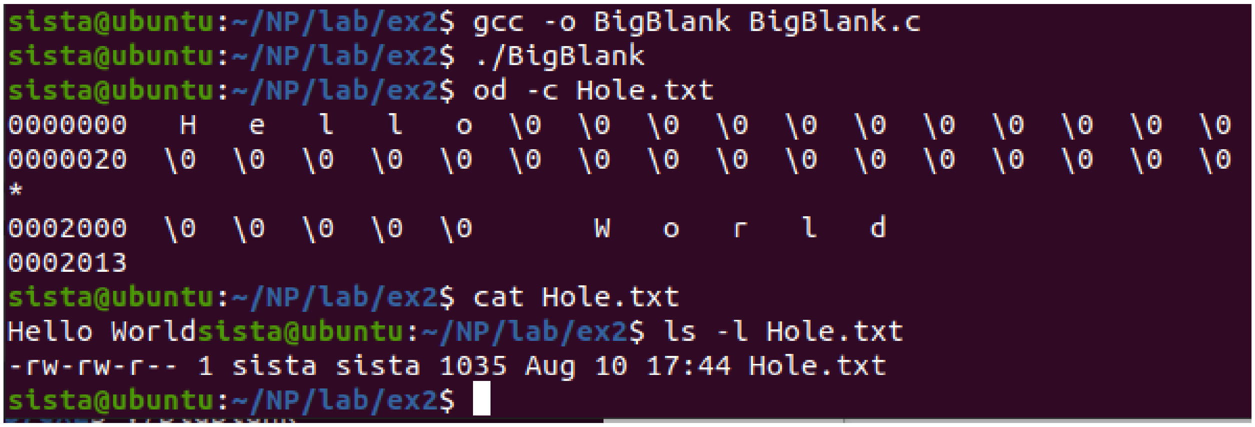
```
sista@ubuntu:~/NP/lab/ex2$ gcc -o ThreeNumbers ThreeNumbers.c
sista@ubuntu:~/NP/lab/ex2$ ./ThreeNumbers
Enter integer1:1
Enter integer2:2
Enter integer3:3
1 2 3
sista@ubuntu:~/NP/lab/ex2$ ./ThreeNumbers
Enter integer1:2
Enter integer2:3
Enter integer3:4
2 3 4
sista@ubuntu:~/NP/lab/ex2$ 
```

f) BigBlank.c

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

extern int errno;
void main(){
    int fd = open("Hole.txt", O_WRONLY | O_CREAT |
O_TRUNC, 0666);
    if(fd == -1){
      int errsv = errno;
      fprintf(stderr, "File Hole.txt cannot be
created:%s\n",strerror(errsv));
      exit(1);
    }

    if(write(fd,"Hello",5) == -1) {
      int errsv = errno;
      fprintf(stderr, "File Hole.txt cannot be written
to:%s\n",strerror(errsv));
      exit(1);
    }

    if(lseek(fd,1024,SEEK_CUR) == -1) {
      int errsv = errno;
      fprintf(stderr, "File Hole.txt cannot be
seeked:%s\n",strerror(errsv));
      exit(1);
    }

    if(write(fd," World",6) == -1) {
        int errsv = errno;
        fprintf(stderr, "File Hole.txt cannot be written
to:%s\n",strerror(errsv));
        exit(1);
     }
}
```

What is the expected size of the created file?
The expected file size is 5+1024+6=1035. This is because when the offset is set to greater than the current file contents the write function will continue writing from the newly set offset leaving a hole.

Output:

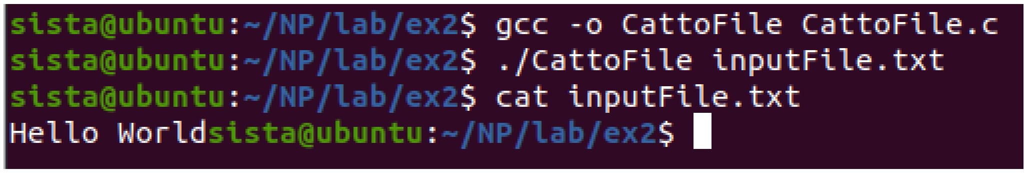


```
sista@ubuntu:~/NP/lab/ex2$ gcc -o BigBlank BigBlank.c
sista@ubuntu:~/NP/lab/ex2$ ./BigBlank
sista@ubuntu:~/NP/lab/ex2$ od -c Hole.txt
0000000   H   e   l   l   o  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0
0000020  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0
*
0002000  \0  \0  \0  \0  \0       W   o   r   l   d
0002013
sista@ubuntu:~/NP/lab/ex2$ cat Hole.txt
Hello Worldsista@ubuntu:~/NP/lab/ex2$ ls -l Hole.txt
-rw-rw-r-- 1 sista sista 1035 Aug 10 17:44 Hole.txt
sista@ubuntu:~/NP/lab/ex2$ 
```

2) CattoFile.c

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

extern int errno;
void main(int argc, char *argv[]){
    if(argc != 2) {
      printf("1 argument is required. File path\n");
      exit(1);
    }

    int fd = open(argv[1],O_WRONLY);
    if(fd == -1){
      int errsv = errno;
      fprintf(stderr, "File %s cannot be opened:%s\n", argv[1],
strerror(errsv));
      exit(1);
    }

    int fd2 = dup2(fd, 1);
    if(fd2 == -1){
      int errsv = errno;
      fprintf(stderr, "File descriptor could not be
duped:%s\n", strerror(errsv));
      exit(1);
    }

    printf("Hello World");
    close(fd);
}
```

Output:



```
sista@ubuntu:~/NP/lab/ex2$ gcc -o CattoFile CattoFile.c
sista@ubuntu:~/NP/lab/ex2$ ./CattoFile inputFile.txt
sista@ubuntu:~/NP/lab/ex2$ cat inputFile.txt
Hello Worldsista@ubuntu:~/NP/lab/ex2$ 
```

Observations:
The content written in printf is contained in the file.