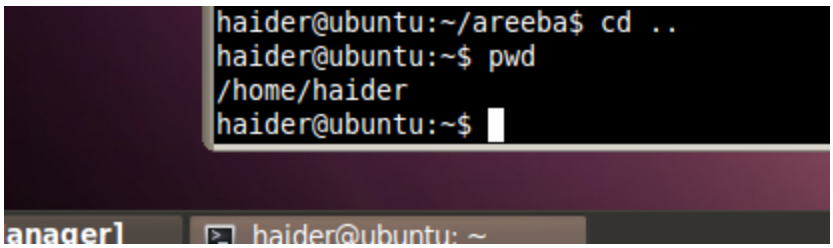
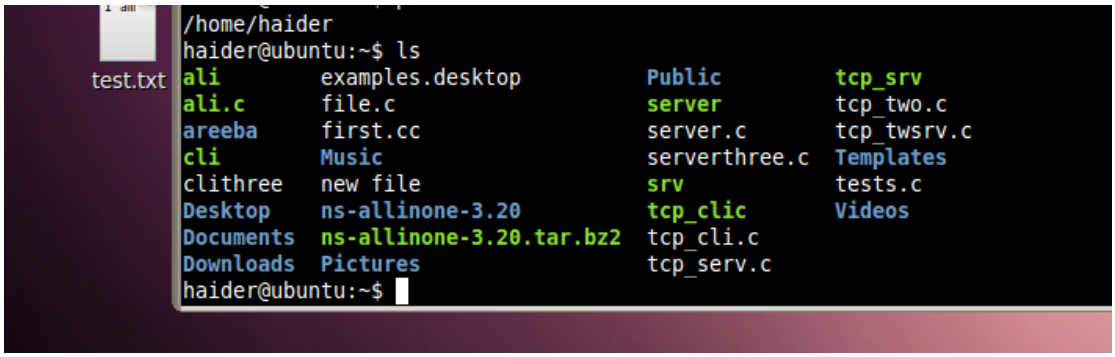
Operating System

Lab 1

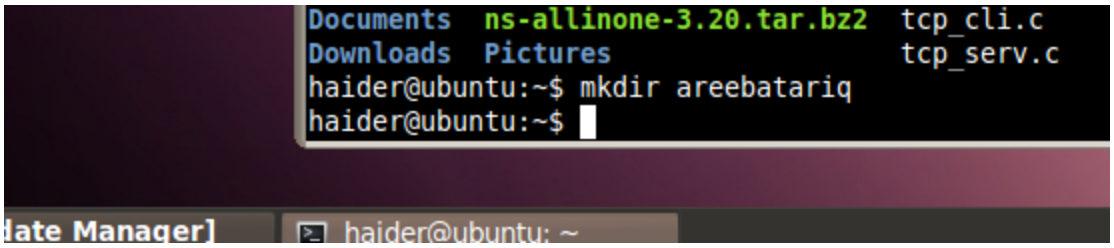
pwd



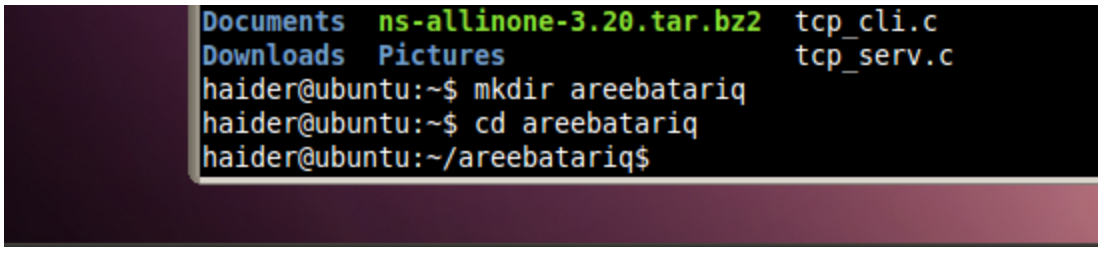
ls



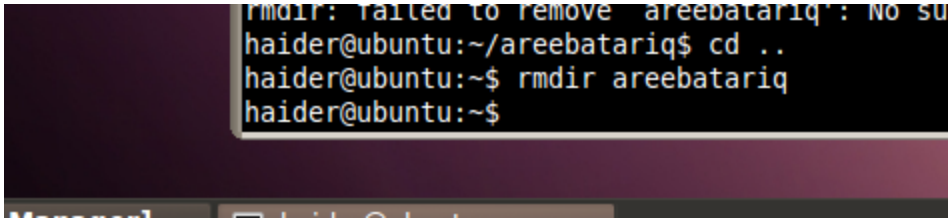
mkdir



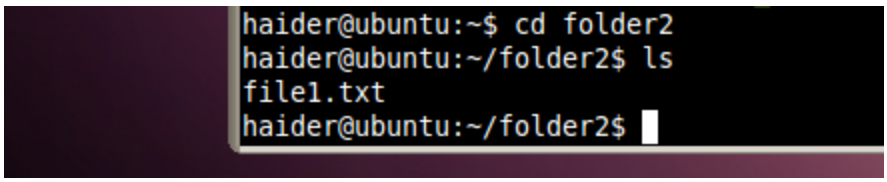
cd

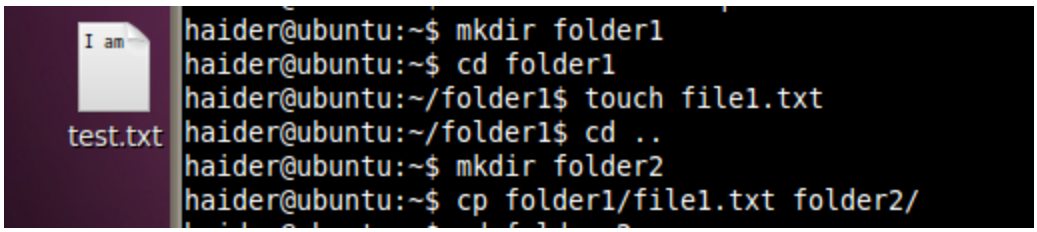


rmdir

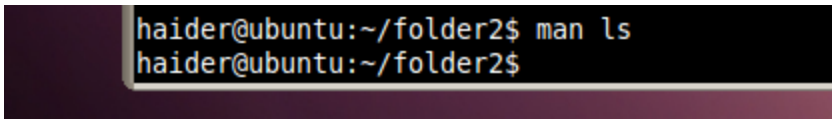

rmdir: failed to remove 'areebatariq': No su
haider@ubuntu:~/areebatariq$ cd ..
haider@ubuntu:~$ rmdir areebatariq
haider@ubuntu:~$

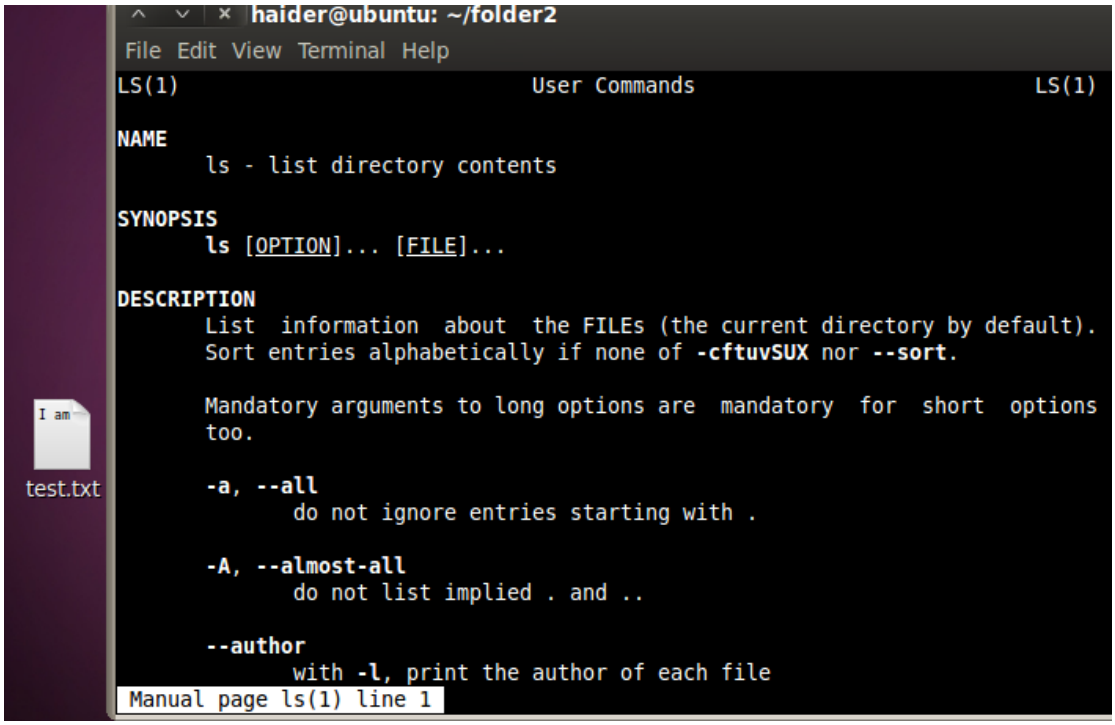
cp


haider@ubuntu:~$ cd folder2
haider@ubuntu:~/folder2$ ls
file1.txt
haider@ubuntu:~/folder2$

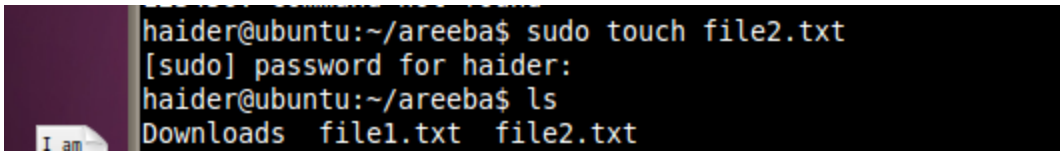

I am
test.txt
haider@ubuntu:~$ mkdir folder1
haider@ubuntu:~$ cd folder1
haider@ubuntu:~/folder1$ touch file1.txt
haider@ubuntu:~/folder1$ cd ..
haider@ubuntu:~$ mkdir folder2
haider@ubuntu:~$ cp folder1/file1.txt folder2/

man


haider@ubuntu:~/folder2$ man ls
haider@ubuntu:~/folder2$

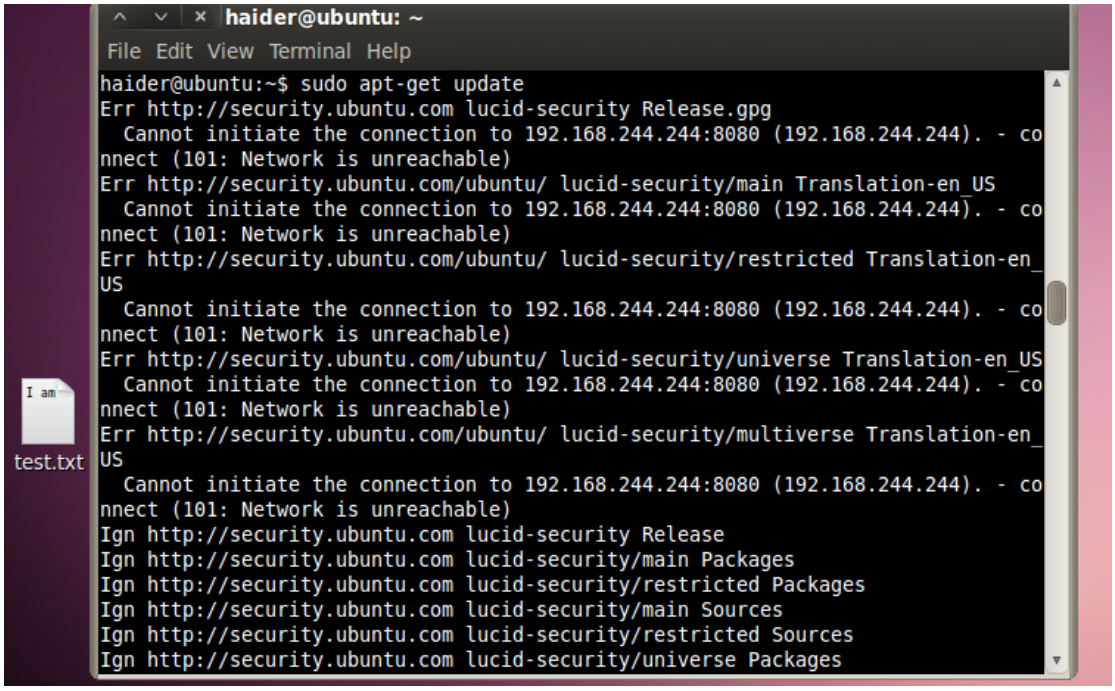

I am
test.txt
haider@ubuntu: ~/folder2
File Edit View Terminal Help
LS(1) User Commands LS(1)
NAME
ls - list directory contents
SYNOPSIS
ls [OPTION]... [FILE]...
DESCRIPTION
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort.
Mandatory arguments to long options are mandatory for short options
too.
-a, --all
do not ignore entries starting with .
-A, --almost-all
do not list implied . and ..
--author
with -l, print the author of each file
Manual page ls(1) line 1

sudo



\

apt-get



Question 2

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

int main(int argc, char *argv[]) {
    // Check if at least one command-line argument is provided (which is the course name).
```

```
    if (argc < 2) {
        printf("The %s Course Name is
missing\n", argv[0]);
        return 1; // Exit with an error code.
    }
  else{
    printf("Welcome to os lab %s",argv[1]);
  }
}
```

Output

```
~/lab1os$ gcc main.c -o main
~/lab1os$ ./main os21SE
Welcome to  Operating System Lab os21SE
~/lab1os$
```

```
~/lab1os$ gcc main.c -o main
~/lab1os$ ./main
The ./main Course Name is missing
~/lab1os$
```

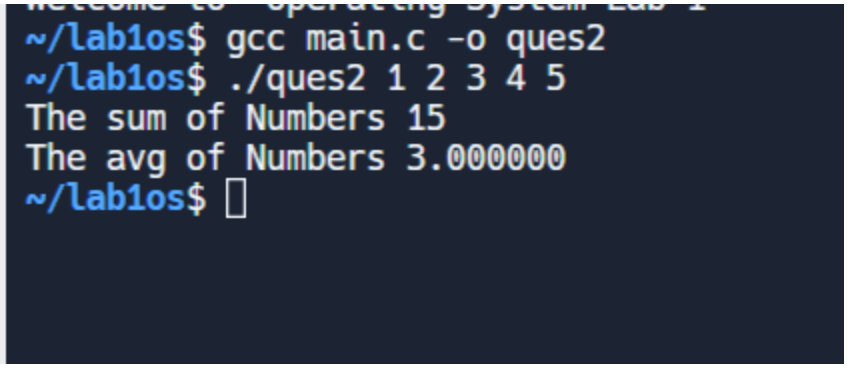
Question 3

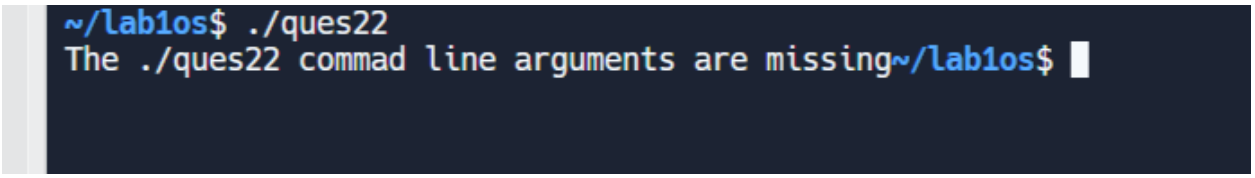
```
#include <stdio.h>
#include <stdlib.h>
```

```c
int main(int argc, char *argv[]) {

    // Check if there are at least 2 command-line arguments (including the program name).
    if (argc < 2) {
        printf("The %s command line arguments are missing\n", argv[0]);
        return 1; // Exit with an error code.
    } else {
        int sum = 0;   // Variable to store the sum of integers.
        float avg = 0; // Variable to store the average of integers.

        // Iterate through the command-line arguments, starting from the second argument (index 1).
        for (int i = 1; i < argc; i++) {
            sum += atoi(argv[i]); // Convert the argument to an integer and add it to the sum.
        }

        // Calculate the average by dividing
```

```
the sum by the number of arguments (excluding
the program name).
        avg = (float)sum / (argc - 1);

        // Print the sum and average.
        printf("The sum of Numbers: %d\n",
sum);
        printf("The avg of Numbers: %f\n",
avg);

        return 0; // Exit successfully
    }
}
```

Output





Question 4

```c
#include <ctype.h>
#include <stdio.h>

int main(int argc, char *argv[]) {

  FILE *input = fopen(argv[1], "r");
  FILE *output = fopen(argv[2], "w");
  char c = getc(input);
  while (c != EOF) {

    if (c >= 48 && c <= 57) {
      fprintf(output, "%c", c);
    }
    c = getc(input);
  }

  printf("Done.Check file2.txt\n");
  return 0;
}
```

Output

```
~/lab1os$ gcc main.c -o program
~/lab1os$ ./program "i.txt" "out.txt"
Done.Check file2.txt
~/lab1os$
```

i.txt

```
1  2 a b c 3 4 5
```

out.txt

```
1  2345
```