

```
john@john-VirtualBox:~$ ln -s file.txt softlink.txt
john@john-VirtualBox:~$ ls -l
total 228
drwxrwxr-x 2 john john 4096 Jan 11 17:30 1
-rwxrwxr-x 1 john john 15960 Jan 11 17:49 a.out
-rwxrwxr-x 1 john john 16088 Feb  1 17:03 call
-rw-rw-r-- 1 john john    0 Feb  1 16:40 child.c
drwxrwxr-x 2 john john 4096 Jan 23 07:43 cwu
drwxrwxr-x 3 john john 4096 Jan 18 18:53 CWU
drwxr-xr-x 2 john john 4096 Jan  9 18:30 Desktop
drwxrwxr-x 2 john john 4096 Jan 18 18:52 directory1
drwxrwxr-x 2 john john 4096 Jan 18 18:50 directory2
drwxr-xr-x 2 john john 4096 Jan  9 18:30 Documents
drwxr-xr-x 2 john john 4096 Jan  9 18:30 Downloads
----rw-r-- 1 john john   15 Jan 23 07:41 ex1.txt
-rwxrwxr-x 1 john john 15960 Feb  8 18:03 hello
-rwxrwxr-x 1 john john 16456 Feb 22 17:45 lab
drwxrwxr-x 2 john john 4096 Jan 11 17:30 Lab
-rwxrwxr-x 1 john john 15960 Jan 23 17:35 lab3
-rw-rw-r-- 1 john john   68 Jan 23 17:33 lab3.c
-rw-rw-r-- 1 john john    0 Jan 23 17:32 lab3.c.save
-rwxrwxr-x 1 john john 16448 Feb 22 17:30 lab4
```

```
john@john-VirtualBox: ~
Firefox Web Browser
john@john-VirtualBox:~$ ln -s lab4.c
john@john-VirtualBox:~$ ls -l
a.out
call
call.c
child.c
cwu
CWU
Desktop
directory1
directory2
Documents
Downloads
ex1.txt
hello
hello.c
lab
lab3
lab3.c
lab3.c.save
lab4.c
labs
la.c
```

```
john@john-VirtualBox:~$ gcc lab4.c -o lab
john@john-VirtualBox:~$ ./lab
The average value is 0.00
The min value is 0
The max value is 20
john@john-VirtualBox:~$
```

```
1 #include <pthread.h>
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 #define NUM_THREADS 3
6
7 int numbers[] = {2, 20, 25, 5, 70, 90, 98};
8 int num_count = sizeof(numbers) / sizeof(int);
9
10 double average;
11 int max,min;
12
13 void *calc_average (void *arg)
14 {
15     double sum = 0.0;
16     for (int i = 0; i < num_count; i++)
17     {
18         sum += numbers[i];
19     }
20     average = sum / num_count;
21     pthread_exit(NULL);
22 }
23
24 void *calc_max (void *arg)
25 {
26     // ...
27 }
```

```

24 void *calc_max (void *arg)
25 {
26     max = numbers[0];
27     for (int i = 1; i < num_count; i++)
28     {
29         if (numbers[i] > max)
30         {
31             max = numbers[i];
32         }
33         pthread_exit(NULL);
34     }
35     pthread_exit(NULL);
36 }
37
38 void *calc_min (void *arg)
39 {
40     min = numbers[0];
41     for (int i = 1; i < num_count; i++)
42     {
43         if (numbers[i] < min)
44         {
45             min = numbers[i];
46         }
47     }
48     pthread_exit(NULL);
49 }
50

```

```

1 int main (int argc, char *argv[])
2 {
3     pthread_t threads[NUM_THREADS];
4     int rc;
5
6     rc = pthread_create (&threads[0], NULL, calc_average, NULL);
7     if (rc)
8     {
9         printf("Error: Unable to create thread. \n");
10        exit(-1);
11    }
12
13    rc = pthread_create (&threads[0], NULL, calc_max, NULL);
14    if (rc)
15    {
16        printf("Error: Unable to create thread. \n");
17        exit(-1);
18    }
19
20    rc = pthread_create (&threads[0], NULL, calc_min, NULL);
21    if (rc)
22    {
23        printf("Error: Unable to create thread. \n");
24        exit(-1);
25    }
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70    rc = pthread_create (&threads[0], NULL, calc_min, NULL);
71    if (rc)
72    {
73        printf("Error: Unable to create thread. \n");
74        exit(-1);
75    }
76
77    for (int i = 1; i < NUM_THREADS; i++)
78    {
79        if (rc)
80        {
81            printf("Error: Unable to join thread. \n");
82            exit(-1);
83        }
84    }
85
86    printf("The average value is %.2f\n", average);
87    printf("The min value is %d\n", min);
88    printf("The max value is %d\n", max);
89 }
90
91

```

```
john@john-VirtualBox: ~  
john@john-VirtualBox:~$ gcc output.c -o output  
john@john-VirtualBox:~$ ./output  
john@john-VirtualBox:~$ ./output  
his is a test for opening, writing, and opening a file!john@john-Virtua  
Box:~$
```

```
int main()  
{  
    int fd;  
    char buf[100] = "Hello, OS 470 Students! This is a test for opening, writing, and closing a file....";  
    ssize_t n;  
    printf("This is a test for opening, writing, and opening a file!");  
  
    fd = open("outputchange.txt", O_WRONLY | O_CREAT, 0644);  
  
    if (fd == -1)  
    {  
        perror("open");  
        exit(EXIT_FAILURE);  
    }  
  
    n = write(fd, buf, sizeof(buf));  
  
    if (n == -1)  
    {  
        perror("write");  
        exit(EXIT_FAILURE);  
    }  
  
    if (close(fd) == -1)  
    {  
        perror("close");  
        exit(EXIT_FAILURE);  
    }  
  
    return 0;  
}
```