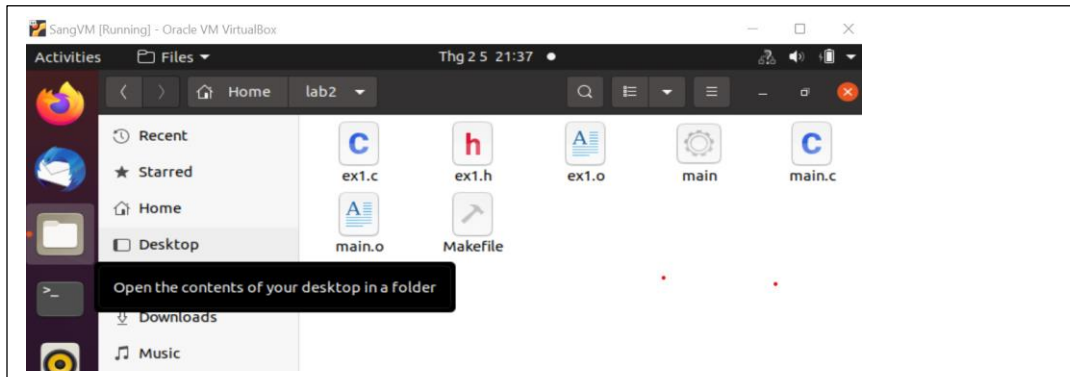
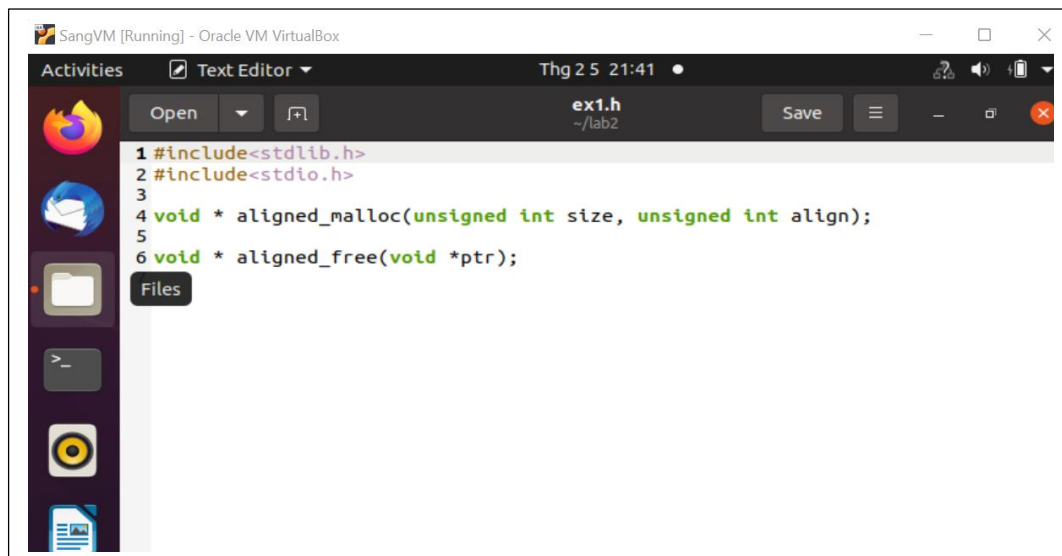


Problem 1:

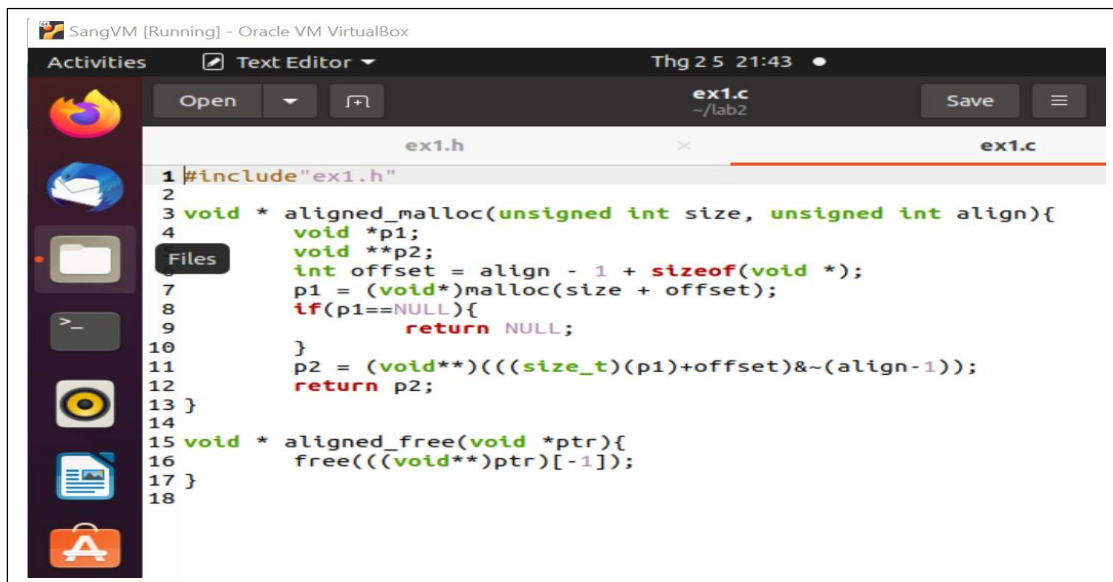
Folder prob1:



Code file ex1.h:

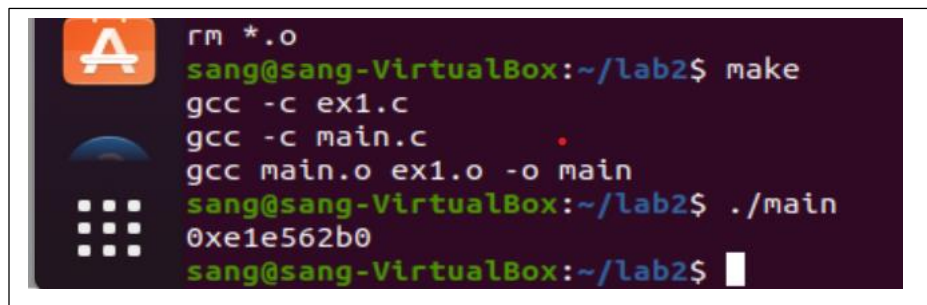


Code file ex1.c:



```
1 #include "ex1.h"
2
3 void * aligned_malloc(unsigned int size, unsigned int align){
4     void *p1;
5     void **p2;
6     int offset = align - 1 + sizeof(void *);
7     p1 = (void*)malloc(size + offset);
8     if(p1==NULL){
9         return NULL;
10    }
11    p2 = (void**)(((size_t)(p1)+offset)&~(align-1));
12    return p2;
13 }
14
15 void * aligned_free(void *ptr){
16     free(((void**)ptr)[-1]);
17 }
18
```

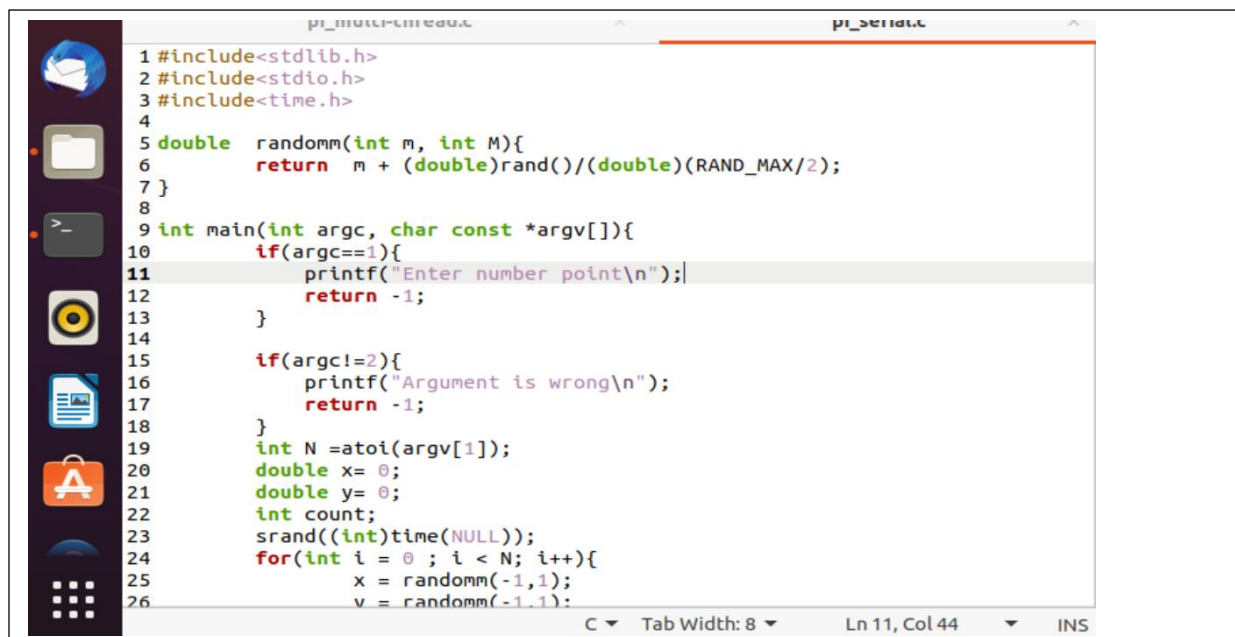
Kết quả chạy thử:



```
rm *.o
sang@sang-VirtualBox:~/lab2$ make
gcc -c ex1.c
gcc -c main.c
gcc main.o ex1.o -o main
sang@sang-VirtualBox:~/lab2$ ./main
0xe1e562b0
sang@sang-VirtualBox:~/lab2$
```

Problem 2:

Source code hàm serial.c



```
1 #include<stdlib.h>
2 #include<stdio.h>
3 #include<time.h>
4
5 double randomm(int m, int M){
6     return m + (double)rand()/((double)(RAND_MAX/2));
7 }
8
9 int main(int argc, char const *argv[]){
10     if(argc==1){
11         printf("Enter number point\n");
12         return -1;
13     }
14
15     if(argc!=2){
16         printf("Argument is wrong\n");
17         return -1;
18     }
19     int N = atoi(argv[1]);
20     double x= 0;
21     double y= 0;
22     int count;
23     srand((int)time(NULL));
24     for(int i = 0 ; i < N; i++){
25         x = randomm(-1,1);
26         y = randomm(-1,1);

```

```
25         x = randomm(-1,1);
26         y = randomm(-1,1);
27         if(x*x+y*y<=1)count++;
28     }
29     double result =(double) 4.0*count/(double)N;
30     printf("PI: %lf\n",result);
31
32 }
```

Source code hàm pi_multi-thread.c

```
SangVM [Running] - Oracle VM VirtualBox
Activities Text Editor Thg 2 12 22:30
pi_multi-thread.c Save
~/lab2/problem2
pi_multi-thread.c pi_serial.c
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4 #include <math.h>
5 #include <pthread.h>
6
7 #define NUM_THREAD 4
8
9
10 pthread_t tid[NUM_THREAD]={0};
11 int count[NUM_THREAD]={0};
12 clock_t start_time, end_time;
13 static long int total_point;
14 static long int count_circle=0;
15
16 void* circle_point(void *param){
17     int *pcount= (int*)param;
18     int i;
19     for(i=0; i<total_point;i++){
20         double x= (double)rand()/((double)RAND_MAX);
21         double y=(double)rand()/((double)RAND_MAX);
22         double r= x*x+y*y;
23         if(r<=1) *pcount=*pcount+1;
24     }
25     pthread_exit(0);
26 }
27
28
29 int main(int argc, char const *argv[]){
30     if(argc==1){
31         printf("Enter number point\n");
32         return -1;
33     }
34
35     if(argc!=2){
36         printf("Argument is wrong\n");
37         return -1;
38     }
39     total_point=atoll(argv[1])/NUM_THREAD;
40
41     srand(time(NULL));
42     static int i;
43     for(i=0; i<NUM_THREAD;i++)
44         pthread_create(&tid[i],NULL,circle_point,&count[i]);
45     for(i=0;i<NUM_THREAD;i++){
46         pthread_join(tid[i],NULL);
47         count_circle+=count[i];
48     }
49     double pi=4.0*(double)count_circle/((double)total_point/-
50 (double)NUM_THREAD;
51     printf("PI = %17.15f\n",pi);
52     return 0;
53 }
```

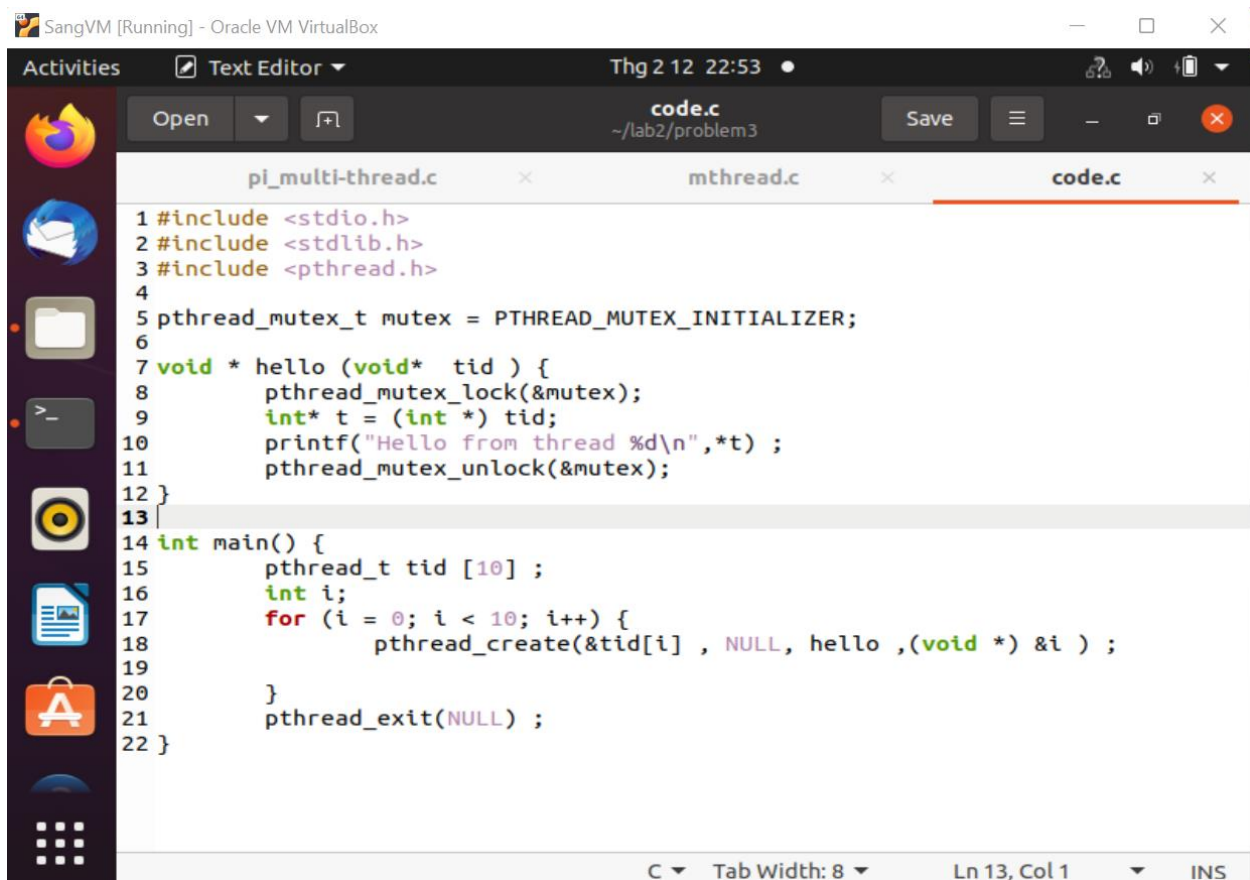
Kết quả chạy thử:

```
sang@sang-VirtualBox:~/lab2/problem2$ make exec
time ./pi_s 10000000
PI: 3.150150
0.23user 0.00system 0:00.25elapsed 95%CPU (0avgtext+0avgdata 1492maxresident)k
0inputs+0outputs (0major+65minor)pagefaults 0swaps
time ./pi_m 10000000
PI = 3.141122400000000
0.33user 0.00system 0:00.35elapsed 96%CPU (0avgtext+0avgdata 2052maxresident)k
0inputs+0outputs (0major+99minor)pagefaults 0swaps
sang@sang-VirtualBox:~/lab2/problem2$
```

Speed up: pi_serial nhanh hơn khoảng 1.4 lần

Problem 3:

Source code sau khi update:



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <pthread.h>
4
5 pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;
6
7 void * hello (void* tid ) {
8     pthread_mutex_lock(&mutex);
9     int* t = (int *) tid;
10    printf("Hello from thread %d\n",*t) ;
11    pthread_mutex_unlock(&mutex);
12 }
13
14 int main() {
15     pthread_t tid [10] ;
16     int i;
17     for (i = 0; i < 10; i++) {
18         pthread_create(&tid[i] , NULL, hello ,(void *) &i ) ;
19     }
20
21     pthread_exit(NULL) ;
22 }
```

Kết quả chạy thử:

SangVM [Running] - Oracle VM VirtualBox

Thg 2 12 22:58

Activities Terminal

sang@sang-VirtualBox: ~/lab2/problem3

```
Hello from thread 4
Hello from thread 5
Hello from thread 6
Hello from thread 7
Hello from thread 8
Hello from thread 9
sang@sang-VirtualBox:~/lab2/problem3$ ./code
Hello from thread 0
Hello from thread 1
Hello from thread 2
Hello from thread 3
Hello from thread 4
Hello from thread 5
Hello from thread 6
Hello from thread 7
Hello from thread 8
Hello from thread 9
sang@sang-VirtualBox:~/lab2/problem3$ ./code
Hello from thread 0
Hello from thread 1
Hello from thread 2
Hello from thread 3
Hello from thread 4
Hello from thread 5
Hello from thread 6
Hello from thread 7
Hello from thread 8
Hello from thread 9
sang@sang-VirtualBox:~/lab2/problem3$
```