클라우드시스템 MPI 실습02

컴퓨터과학부 1515386 정혜린

- 실습 02 간단한 MPI 프로그램

  ● 소스 코드

  (소스 코드 캡처 첨부 및 간단한 설명)

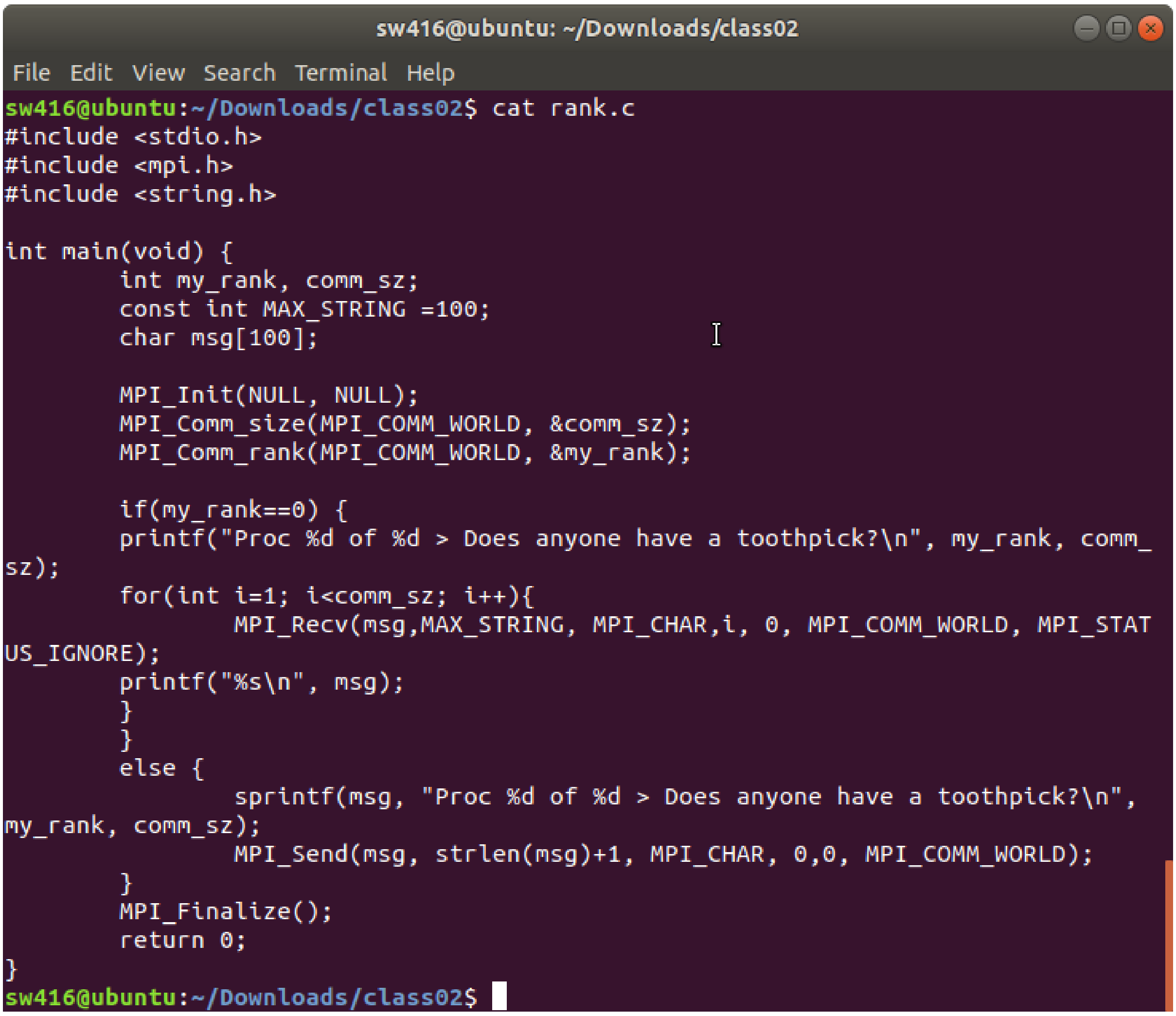
  ● 실행 결과

  (ppt에 명시 되어있는 조건들 실행한 결과 캡처 첨부)

위와 같은 양식으로 보고서 작성한 후 **학번_이름.pdf** 식으로 저장.

반드시 pdf 파일로 제출할 것

[01 소스코드]


sw416@ubuntu: ~/Downloads/class02
File Edit View Search Terminal Help

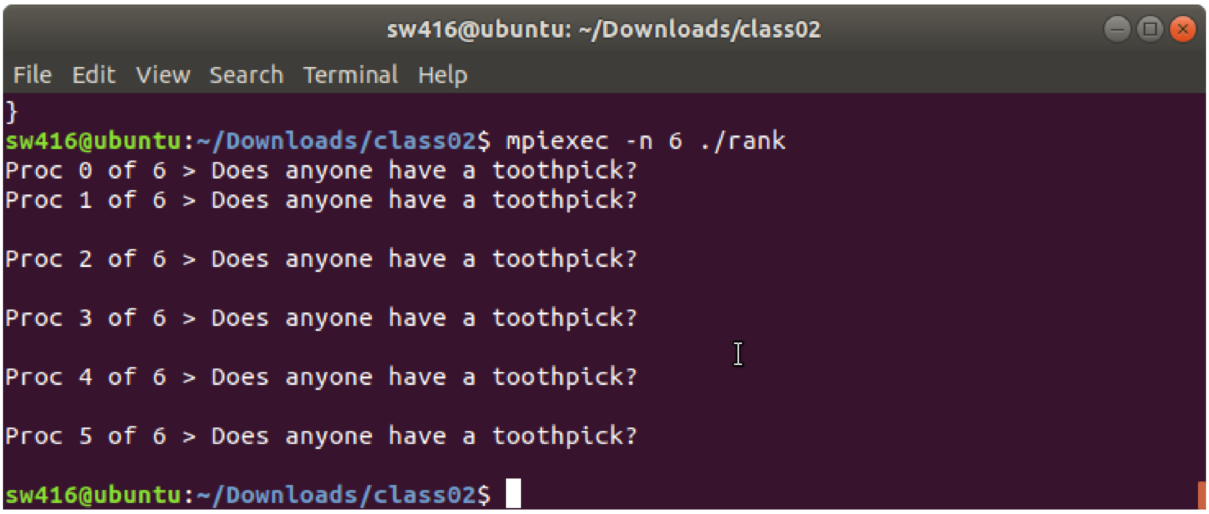
```
sw416@ubuntu:~/Downloads/class02$ cat rank.c
#include <stdio.h>
#include <mpi.h>
#include <string.h>

int main(void) {
        int my_rank, comm_sz;
        const int MAX_STRING =100;
        char msg[100];

        MPI_Init(NULL, NULL);
        MPI_Comm_size(MPI_COMM_WORLD, &comm_sz);
        MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);

        if(my_rank==0) {
        printf("Proc %d of %d > Does anyone have a toothpick?\n", my_rank, comm_
sz);
        for(int i=1; i<comm_sz; i++){
                MPI_Recv(msg,MAX_STRING, MPI_CHAR,i, 0, MPI_COMM_WORLD, MPI_STAT
US_IGNORE);
        printf("%s\n", msg);
        }
        }
        else {
                sprintf(msg, "Proc %d of %d > Does anyone have a toothpick?\n",
my_rank, comm_sz);
                MPI_Send(msg, strlen(msg)+1, MPI_CHAR, 0,0, MPI_COMM_WORLD);
        }
        MPI_Finalize();
        return 0;
}
sw416@ubuntu:~/Downloads/class02$
```

0번 프로세스는 순차적으로 다른 프로세스로부터 메시지를 받아와서 출력한다.

다른 프로세스들은 자신의 rank가 담긴 메시지를 0번 프로세스에게 보낸다.

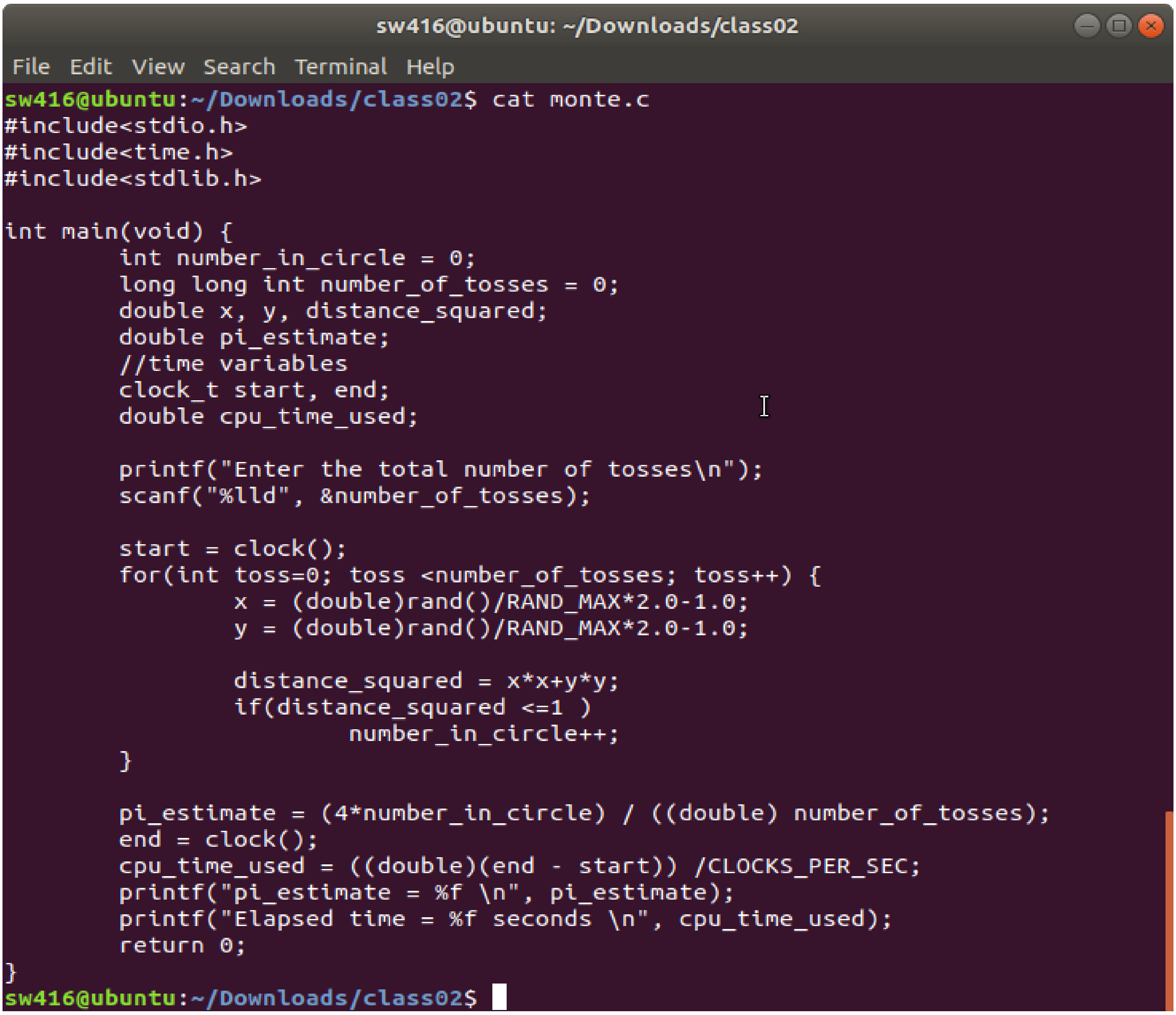
0번 프로세스는 최적화 값을 출력하고 정리한다.

[01 결과]


sw416@ubuntu: ~/Downloads/class02
File Edit View Search Terminal Help

```
}
sw416@ubuntu:~/Downloads/class02$ mpiexec -n 6 ./rank
Proc 0 of 6 > Does anyone have a toothpick?
Proc 1 of 6 > Does anyone have a toothpick?

Proc 2 of 6 > Does anyone have a toothpick?

Proc 3 of 6 > Does anyone have a toothpick?

Proc 4 of 6 > Does anyone have a toothpick?

Proc 5 of 6 > Does anyone have a toothpick?

sw416@ubuntu:~/Downloads/class02$
```

[02 코드]

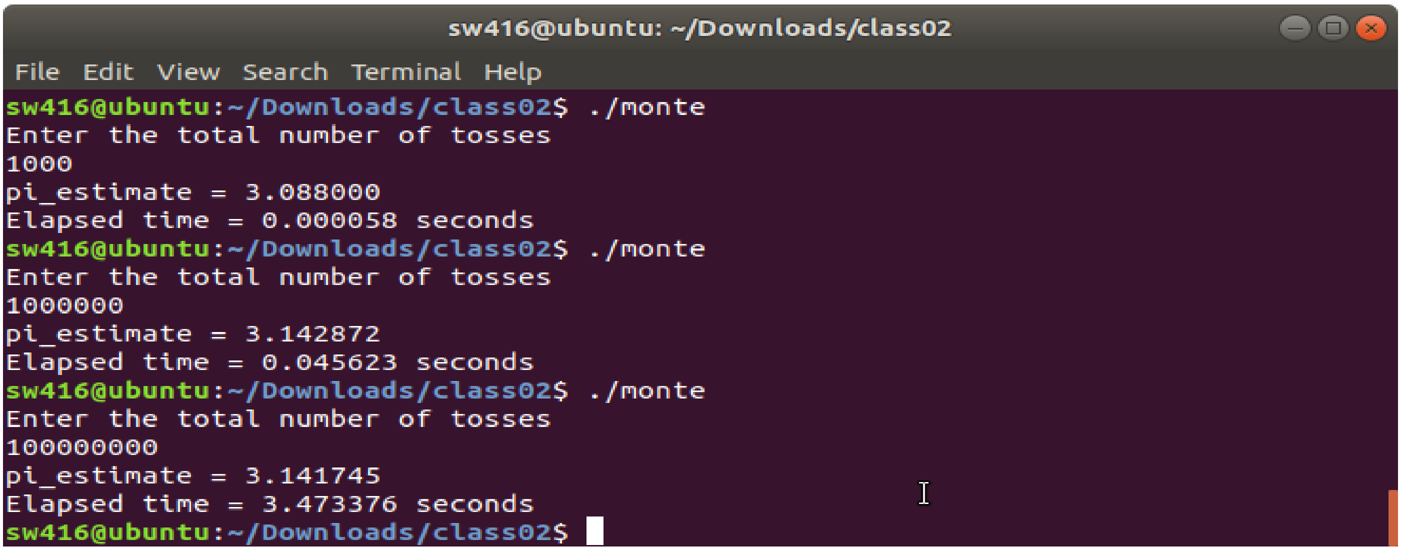

sw416@ubuntu: ~/Downloads/class02
File Edit View Search Terminal Help

```
sw416@ubuntu:~/Downloads/class02$ cat monte.c
#include<stdio.h>
#include<time.h>
#include<stdlib.h>

int main(void) {
        int number_in_circle = 0;
        long long int number_of_tosses = 0;
        double x, y, distance_squared;
        double pi_estimate;
        //time variables
        clock_t start, end;
        double cpu_time_used;

        printf("Enter the total number of tosses\n");
        scanf("%lld", &number_of_tosses);

        start = clock();
        for(int toss=0; toss <number_of_tosses; toss++) {
                x = (double)rand()/RAND_MAX*2.0-1.0;
                y = (double)rand()/RAND_MAX*2.0-1.0;

                distance_squared = x*x+y*y;
                if(distance_squared <=1 )
                        number_in_circle++;
        }

        pi_estimate = (4*number_in_circle) / ((double) number_of_tosses);
        end = clock();
        cpu_time_used = ((double)(end - start)) /CLOCKS_PER_SEC;
        printf("pi_estimate = %f \n", pi_estimate);
        printf("Elapsed time = %f seconds \n", cpu_time_used);
        return 0;
}
sw416@ubuntu:~/Downloads/class02$
```

시간을 계산 하기 위해 clock()을 사용한다.

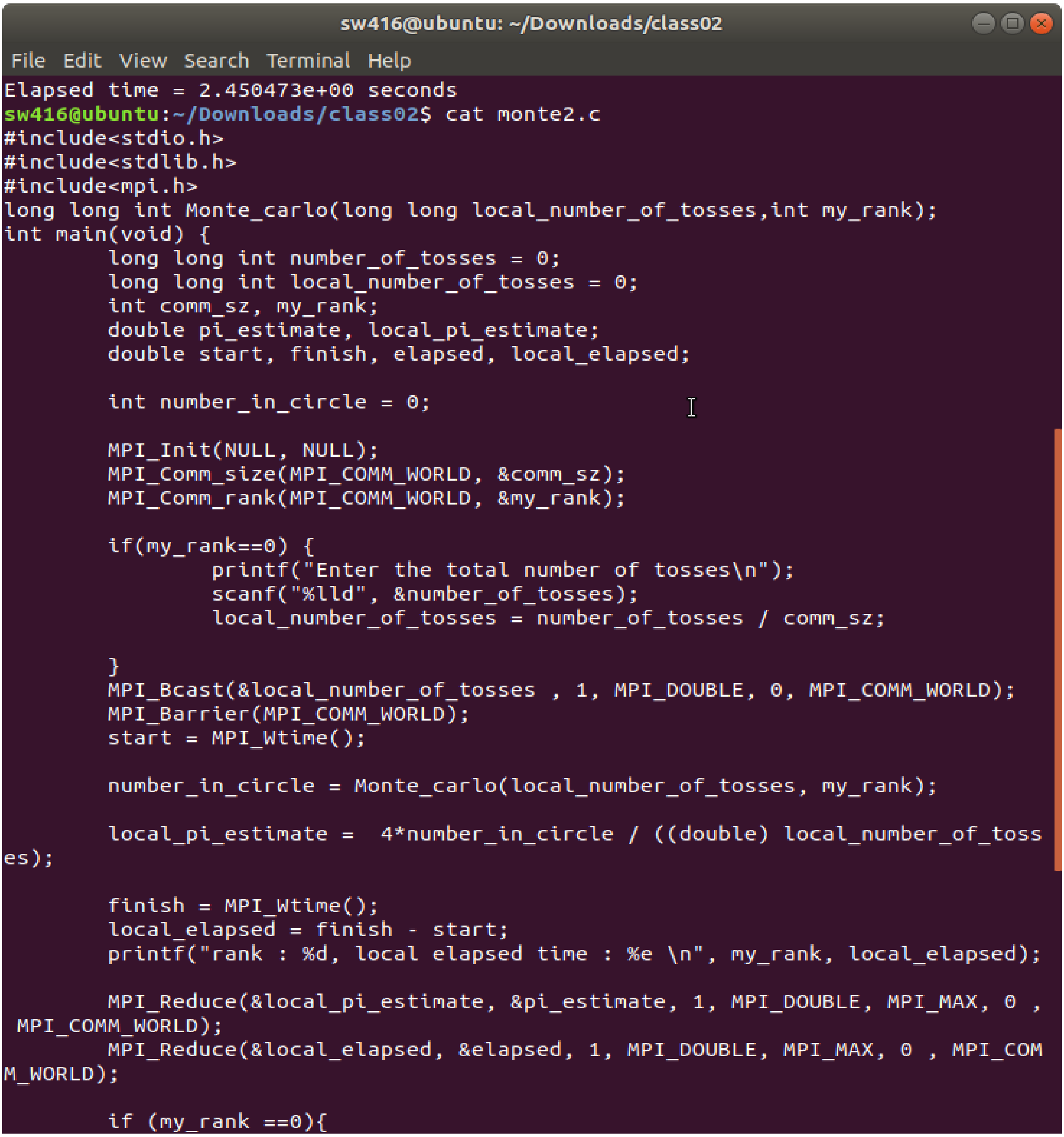
pi 값을 위해 난수 값들을 사용해서 원 안에 있을 확률을 계산하여 값을 사용한다.

[02 결과]


sw416@ubuntu: ~/Downloads/class02
File Edit View Search Terminal Help

```
sw416@ubuntu:~/Downloads/class02$ ./monte
Enter the total number of tosses
1000
pi_estimate = 3.088000
Elapsed time = 0.000058 seconds
sw416@ubuntu:~/Downloads/class02$ ./monte
Enter the total number of tosses
1000000
pi_estimate = 3.142872
Elapsed time = 0.045623 seconds
sw416@ubuntu:~/Downloads/class02$ ./monte
Enter the total number of tosses
100000000
pi_estimate = 3.141745
Elapsed time = 3.473376 seconds
sw416@ubuntu:~/Downloads/class02$
```

[03 소스코드]


sw416@ubuntu: ~/Downloads/class02
File Edit View Search Terminal Help

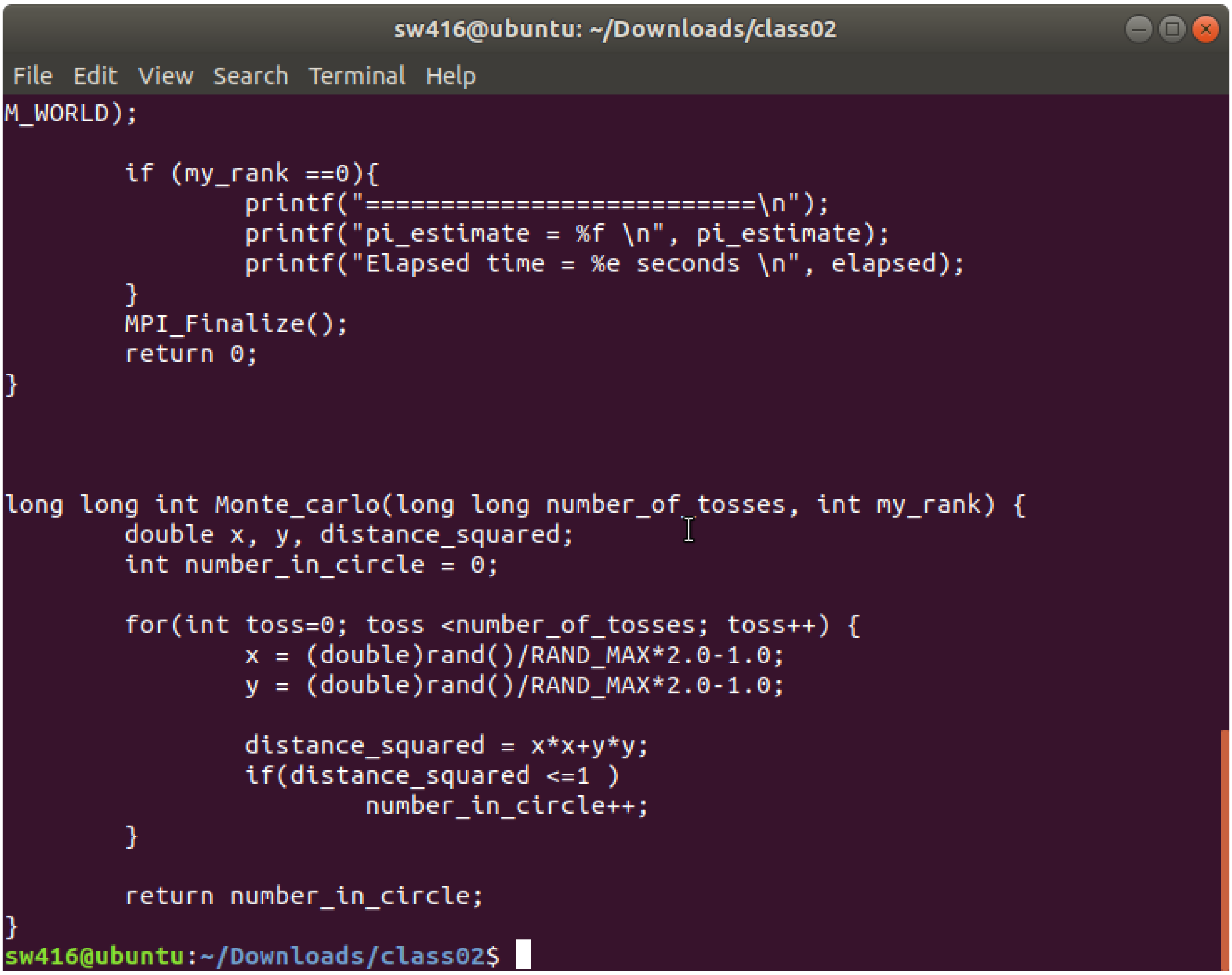
```
Elapsed time = 2.450473e+00 seconds
sw416@ubuntu:~/Downloads/class02$ cat monte2.c
#include<stdio.h>
#include<stdlib.h>
#include<mpi.h>
long long int Monte_carlo(long long local_number_of_tosses,int my_rank);
int main(void) {
        long long int number_of_tosses = 0;
        long long int local_number_of_tosses = 0;
        int comm_sz, my_rank;
        double pi_estimate, local_pi_estimate;
        double start, finish, elapsed, local_elapsed;

        int number_in_circle = 0;

        MPI_Init(NULL, NULL);
        MPI_Comm_size(MPI_COMM_WORLD, &comm_sz);
        MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);

        if(my_rank==0) {
                printf("Enter the total number of tosses\n");
                scanf("%lld", &number_of_tosses);
                local_number_of_tosses = number_of_tosses / comm_sz;

        }
        MPI_Bcast(&local_number_of_tosses , 1, MPI_DOUBLE, 0, MPI_COMM_WORLD);
        MPI_Barrier(MPI_COMM_WORLD);
        start = MPI_Wtime();

        number_in_circle = Monte_carlo(local_number_of_tosses, my_rank);

        local_pi_estimate =  4*number_in_circle / ((double) local_number_of_toss
es);

        finish = MPI_Wtime();
        local_elapsed = finish - start;
        printf("rank : %d, local elapsed time : %e \n", my_rank, local_elapsed);

        MPI_Reduce(&local_pi_estimate, &pi_estimate, 1, MPI_DOUBLE, MPI_MAX, 0 ,
 MPI_COMM_WORLD);
        MPI_Reduce(&local_elapsed, &elapsed, 1, MPI_DOUBLE, MPI_MAX, 0 , MPI_COM
M_WORLD);

        if (my_rank ==0){
```

0번 프로세서는 토스할 전체 숫자를 입력 받고, 규모를 분할 한다. MPI_Bcast를 사용해서 분할한 규모만큼 전달해준다. Barrier로 시간을 계산하기 위해 프로세스들의 sync를 맞춘다.

MPI_Wtime()을 사용해서 연사의 시작과 끝을 계산한다.

각 프로세스들은 연산이 끝나면 MPI_Reduce를 사용해서 가장 최적화된 결과 값을 찾는다 .

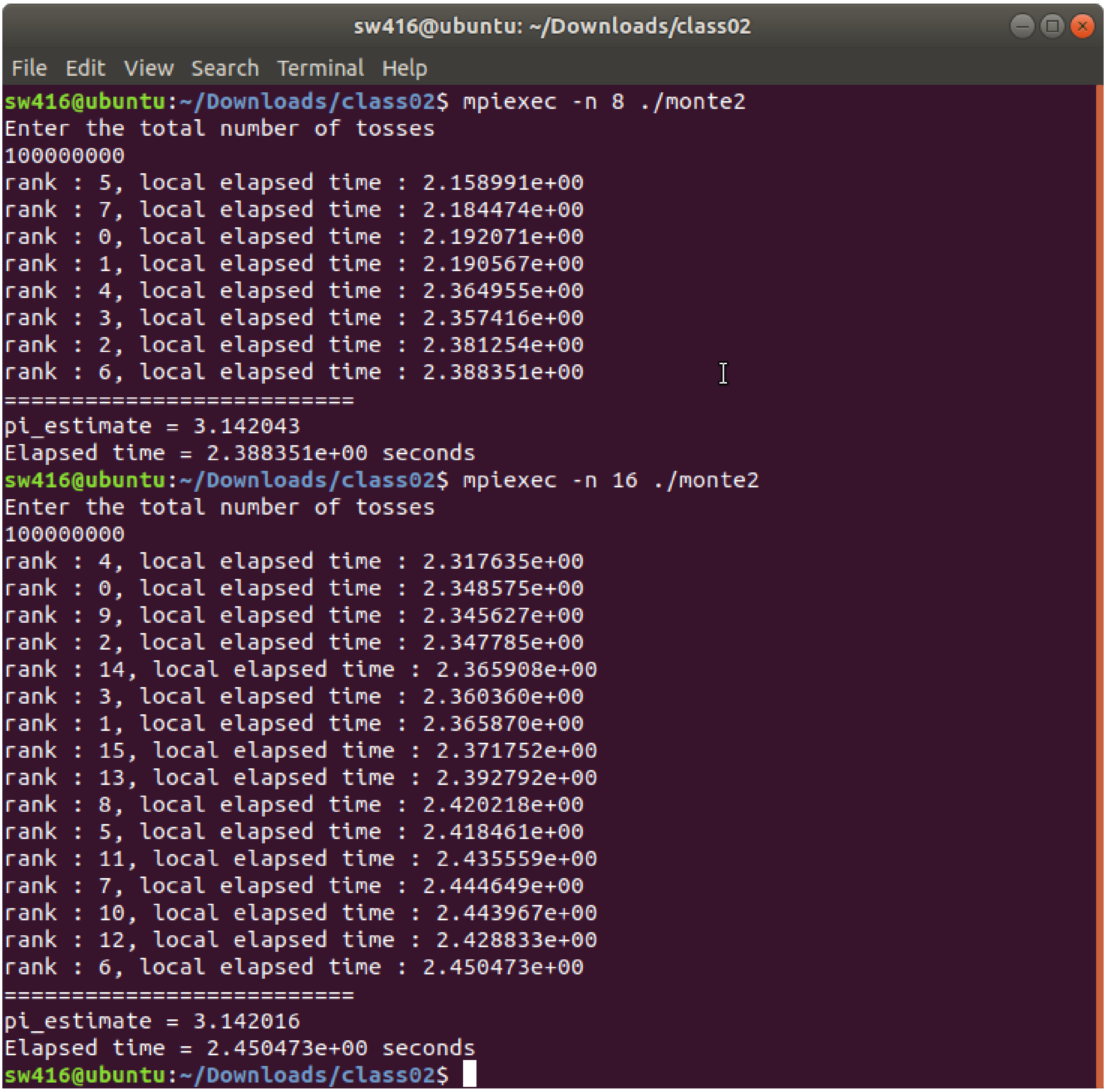
0번 프로세스는 최적화 값을 출력하고 정리한다.



sw416@ubuntu: ~/Downloads/class02
File Edit View Search Terminal Help

```
M_WORLD);

        if (my_rank ==0){
                printf("==========================\n");
                printf("pi_estimate = %f \n", pi_estimate);
                printf("Elapsed time = %e seconds \n", elapsed);
        }
        MPI_Finalize();
        return 0;
}



long long int Monte_carlo(long long number_of_tosses, int my_rank) {
        double x, y, distance_squared;
        int number_in_circle = 0;

        for(int toss=0; toss <number_of_tosses; toss++) {
                x = (double)rand()/RAND_MAX*2.0-1.0;
                y = (double)rand()/RAND_MAX*2.0-1.0;

                distance_squared = x*x+y*y;
                if(distance_squared <=1 )
                        number_in_circle++;
        }

        return number_in_circle;
}
sw416@ubuntu:~/Downloads/class02$
```

[03 결과]



sw416@ubuntu: ~/Downloads/class02
File Edit View Search Terminal Help

```
sw416@ubuntu:~/Downloads/class02$ mpiexec -n 8 ./monte2
Enter the total number of tosses
100000000
rank : 5, local elapsed time : 2.158991e+00
rank : 7, local elapsed time : 2.184474e+00
rank : 0, local elapsed time : 2.192071e+00
rank : 1, local elapsed time : 2.190567e+00
rank : 4, local elapsed time : 2.364955e+00
rank : 3, local elapsed time : 2.357416e+00
rank : 2, local elapsed time : 2.381254e+00
rank : 6, local elapsed time : 2.388351e+00
==========================
pi_estimate = 3.142043
Elapsed time = 2.388351e+00 seconds
sw416@ubuntu:~/Downloads/class02$ mpiexec -n 16 ./monte2
Enter the total number of tosses
100000000
rank : 4, local elapsed time : 2.317635e+00
rank : 0, local elapsed time : 2.348575e+00
rank : 9, local elapsed time : 2.345627e+00
rank : 2, local elapsed time : 2.347785e+00
rank : 14, local elapsed time : 2.365908e+00
rank : 3, local elapsed time : 2.360360e+00
rank : 1, local elapsed time : 2.365870e+00
rank : 15, local elapsed time : 2.371752e+00
rank : 13, local elapsed time : 2.392792e+00
rank : 8, local elapsed time : 2.420218e+00
rank : 5, local elapsed time : 2.418461e+00
rank : 11, local elapsed time : 2.435559e+00
rank : 7, local elapsed time : 2.444649e+00
rank : 10, local elapsed time : 2.443967e+00
rank : 12, local elapsed time : 2.428833e+00
rank : 6, local elapsed time : 2.450473e+00
==========================
pi_estimate = 3.142016
Elapsed time = 2.450473e+00 seconds
sw416@ubuntu:~/Downloads/class02$ 
```

[04]

04는 미완성입니다.