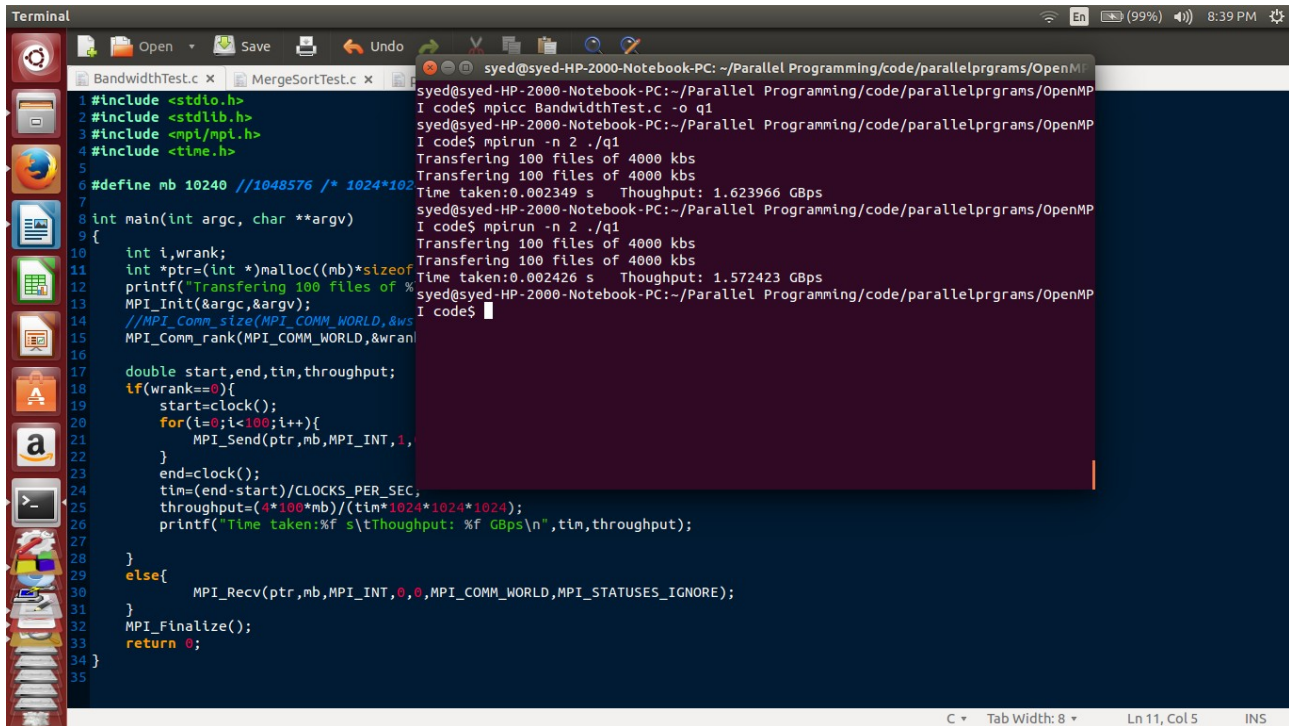


OpenMPI Assignment

CS430 – Parallel Programming

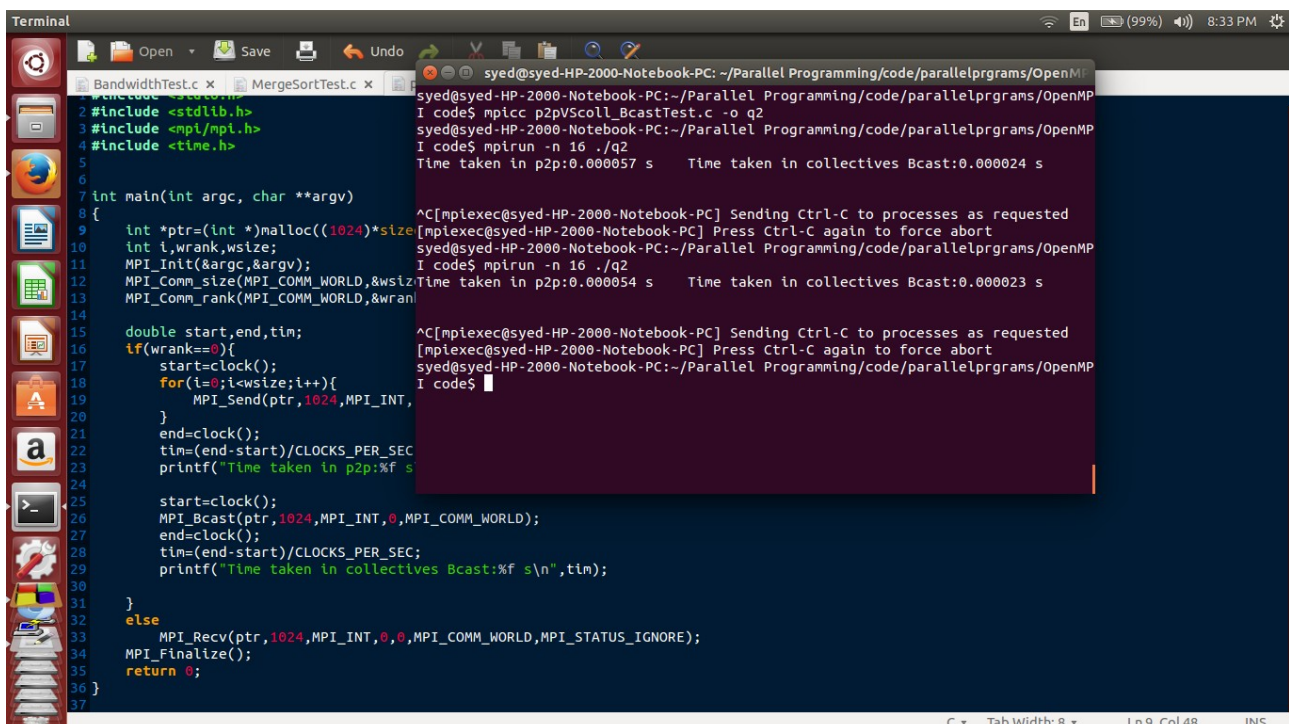
Md Syed Ahamad
Roll No.: 1301030

Q1. Tested with 4mb providing 2 processes, average speed turned out to be nearly 1.6 Gbps. It is also be tested with copying big chunk of memory.



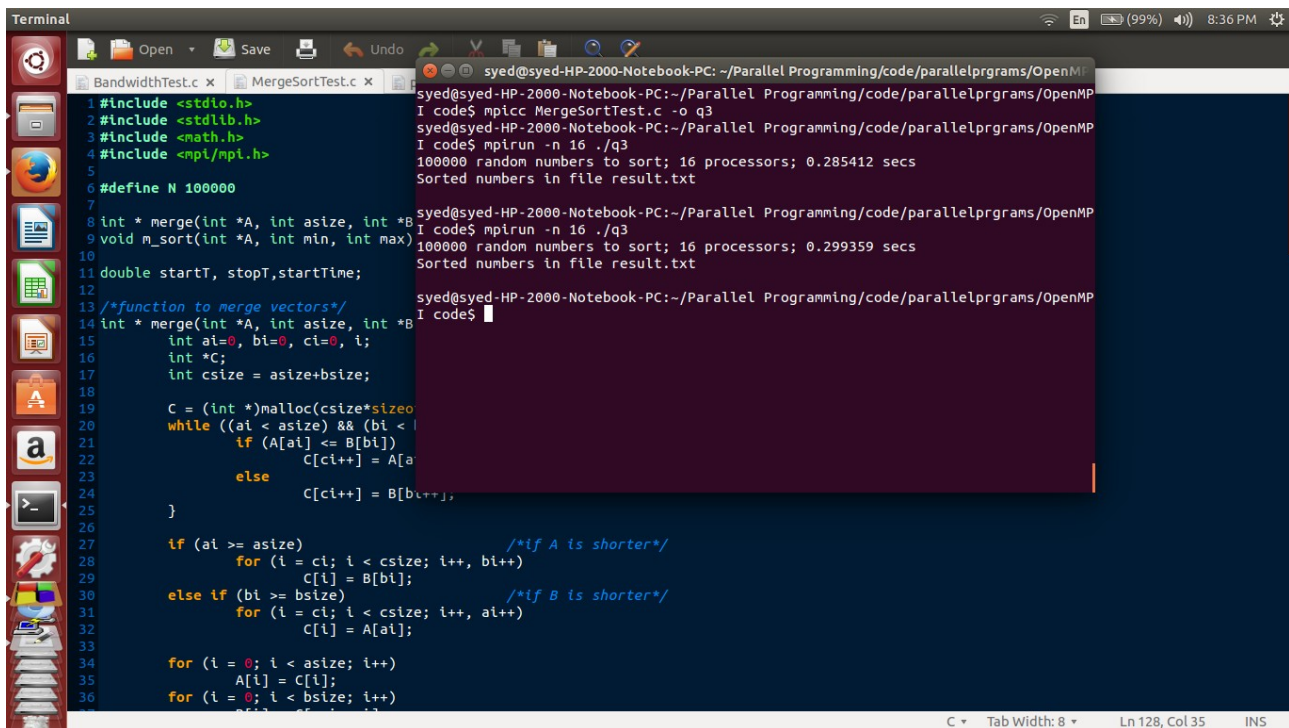
```
syed@syed-HP-2000-Notebook-PC: ~/Parallel Programming/code/parallelprgrams/OpenMPI
I code$ mpicc BandwidthTest.c -o q1
syed@syed-HP-2000-Notebook-PC: ~/Parallel Programming/code/parallelprgrams/OpenMPI
I code$ mpirun -n 2 ./q1
Transferring 100 files of 4000 kbs
Transferring 100 files of 4000 kbs
Time taken:0.002349 s   Throughput: 1.623966 GBps
syed@syed-HP-2000-Notebook-PC:~/Parallel Programming/code/parallelprgrams/OpenMPI
I code$ mpirun -n 2 ./q1
Transferring 100 files of 4000 kbs
Transferring 100 files of 4000 kbs
Time taken:0.002426 s   Throughput: 1.572423 GBps
syed@syed-HP-2000-Notebook-PC:~/Parallel Programming/code/parallelprgrams/OpenMPI
I code$
```

Q2. Providing 16 threads, point to point and collective broadcast is tested by transferring 4kb of data. Thier respective time taken are shown in the screenshot.



```
syed@syed-HP-2000-Notebook-PC: ~/Parallel Programming/code/parallelprgrams/OpenMPI
I code$ mpicc p2pVScoll_BcastTest.c -o q2
syed@syed-HP-2000-Notebook-PC:~/Parallel Programming/code/parallelprgrams/OpenMPI
I code$ mpirun -n 16 ./q2
Time taken in p2p:0.000057 s   Time taken in collectives Bcast:0.000024 s
^C[mpiexec@syed-HP-2000-Notebook-PC] Sending Ctrl-C to processes as requested
[mpiexec@syed-HP-2000-Notebook-PC] Press Ctrl-C again to force abort
syed@syed-HP-2000-Notebook-PC:~/Parallel Programming/code/parallelprgrams/OpenMPI
I code$ mpirun -n 16 ./q2
Time taken in p2p:0.000054 s   Time taken in collectives Bcast:0.000023 s
^C[mpiexec@syed-HP-2000-Notebook-PC] Sending Ctrl-C to processes as requested
[mpiexec@syed-HP-2000-Notebook-PC] Press Ctrl-C again to force abort
syed@syed-HP-2000-Notebook-PC:~/Parallel Programming/code/parallelprgrams/OpenMPI
I code$
```

Q3. Merge sorting is tested with 100000 random numbers with 16 processes. Time taken is given in the screenshot.



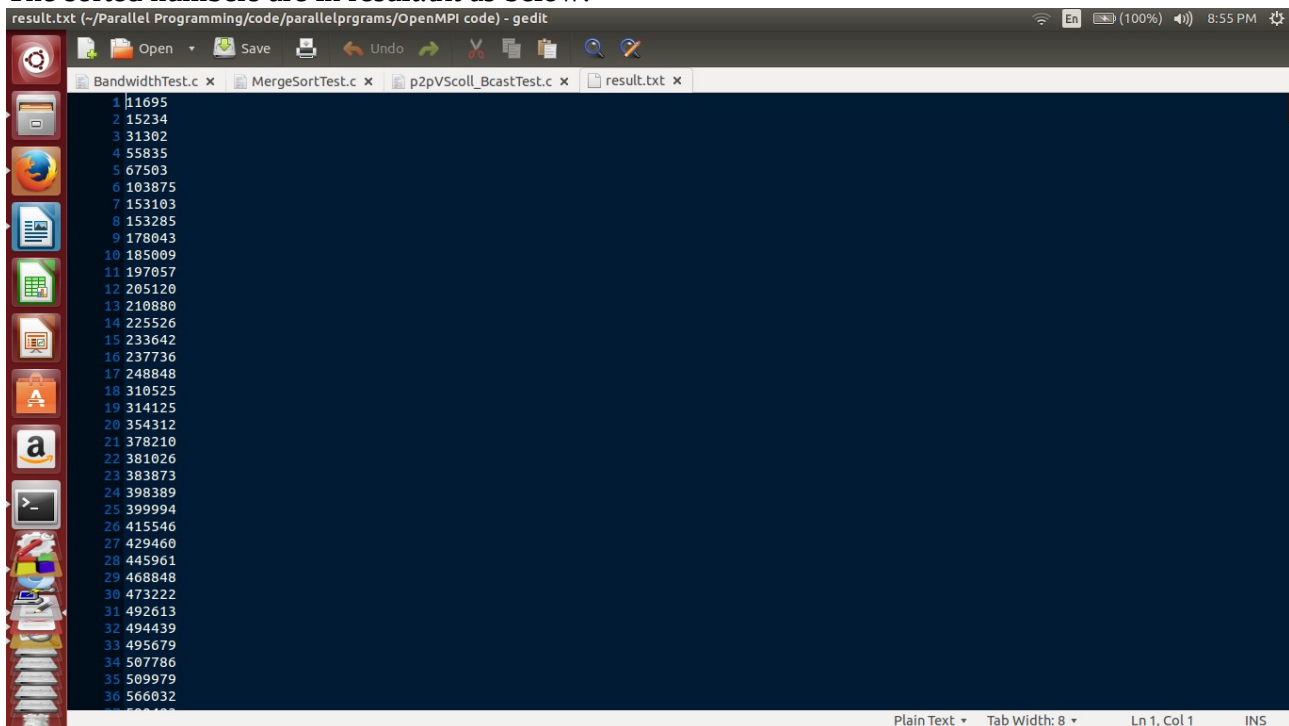
```
Terminal
syed@syed-HP-2000-Notebook-PC: ~/Parallel Programming/code/parallelprgrams/OpenMPI
I code$ mpicc MergeSortTest.c -o q3
syed@syed-HP-2000-Notebook-PC:~/Parallel Programming/code/parallelprgrams/OpenMPI
I code$ mpirun -n 16 ./q3
100000 random numbers to sort; 16 processors; 0.285412 secs
Sorted numbers in file result.txt

syed@syed-HP-2000-Notebook-PC:~/Parallel Programming/code/parallelprgrams/OpenMPI
I code$ mpirun -n 16 ./q3
100000 random numbers to sort; 16 processors; 0.299359 secs
Sorted numbers in file result.txt

syed@syed-HP-2000-Notebook-PC:~/Parallel Programming/code/parallelprgrams/OpenMPI
I code$
```

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4 #include <mpi.h>
5
6 #define N 100000
7
8 int * merge(int *A, int asize, int *B
9 void m_sort(int *A, int min, int max)
10
11 double startT, stopT, startTime;
12
13 /*function to merge vectors*/
14 int * merge(int *A, int asize, int *B
15     int ai=0, bi=0, ci=0, i;
16     int *C;
17     int csize = asize+bsize;
18
19     C = (int *)malloc(csize*sizeof(int));
20     while ((ai < asize) && (bi < bs
21         if (A[ai] <= B[bi])
22             C[ci++] = A[ai++];
23         else
24             C[ci++] = B[bi++];
25     }
26
27     if (ai >= asize) /*if A is shorter*/
28         for (i = ci; i < csize; i++, bi++)
29             C[i] = B[bi];
30     else if (bi >= bsize) /*if B is shorter*/
31         for (i = ci; i < csize; i++, ai++)
32             C[i] = A[ai];
33
34     for (i = 0; i < asize; i++)
35         A[i] = C[i];
36     for (i = 0; i < bsize; i++)
37         B[i] = C[i];
```

The sorted numbers are in result.txt as below:



```
result.txt (~/.Parallel Programming/code/parallelprgrams/OpenMPI code) - gedit
1 111695
2 15234
3 31302
4 55835
5 67503
6 103875
7 153103
8 153285
9 178043
10 185009
11 197057
12 205120
13 210880
14 225526
15 233642
16 237736
17 248848
18 310525
19 314125
20 354312
21 378210
22 381026
23 383873
24 398389
25 399994
26 415546
27 429460
28 445961
29 468848
30 473222
31 492613
32 494439
33 495679
34 507786
35 509979
36 566032
```