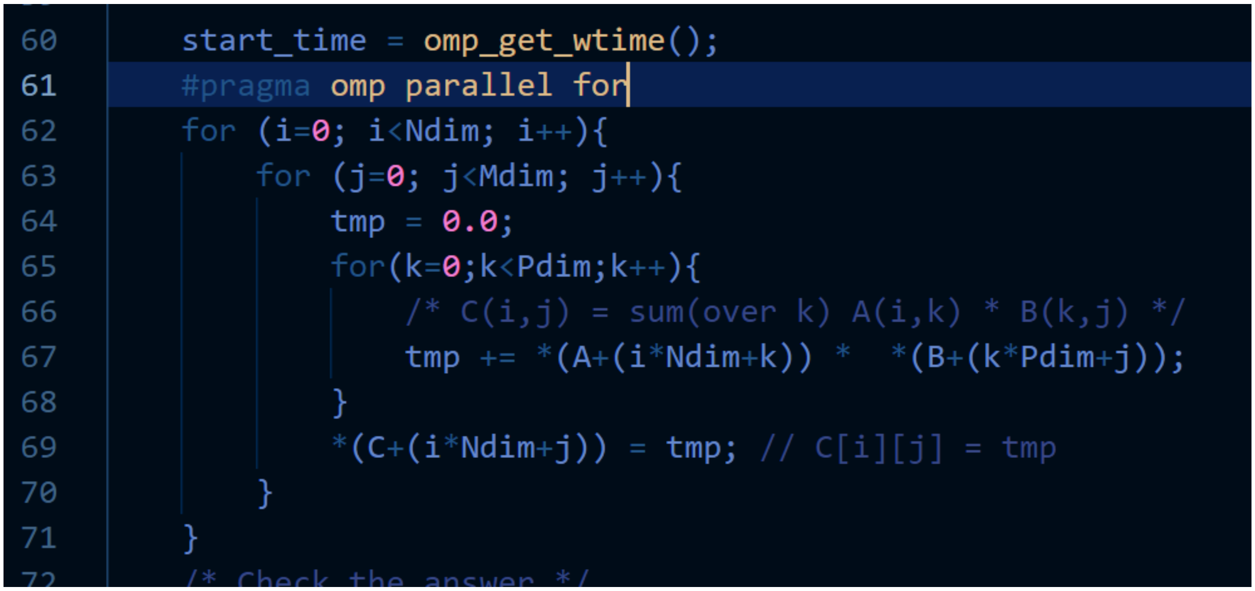
STAT_624 - Open MP - Kaushik Manikonda

Download the openmp.zip file to Terra. Use the Intel/2020b compiler.

[kaushik.manikonda@terra1 ~]$ module load intel/2020b

[kaushik.manikonda@terra1 ~]$ export OMP_NUM_THREADS=8

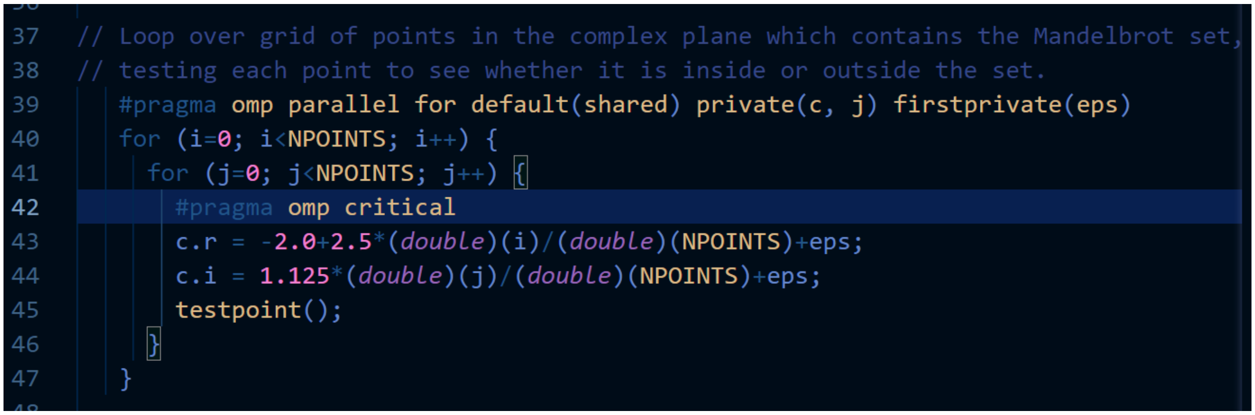
- **For matmul.c add an OpenMP directive (e.g pragma) to parallelize and speedup the code.**



```c
    start_time = omp_get_wtime();
    #pragma omp parallel for
    for (i=0; i<Ndim; i++){
        for (j=0; j<Mdim; j++){
            tmp = 0.0;
            for(k=0;k<Pdim;k++){
                /* C(i,j) = sum(over k) A(i,k) * B(k,j) */
                tmp += *(A+(i*Ndim+k)) *  *(B+(k*Pdim+j));
            }
            *(C+(i*Ndim+j)) = tmp; // C[i][j] = tmp
        }
    }
    /* Check the answer */
```

- **For mandel.c add one or more OpenMP directives (e.g. pragma) to parallelize and speedup the code.**

I added two OpenMP directives, one before the two nested for loops, and a second one inside the second for loop.

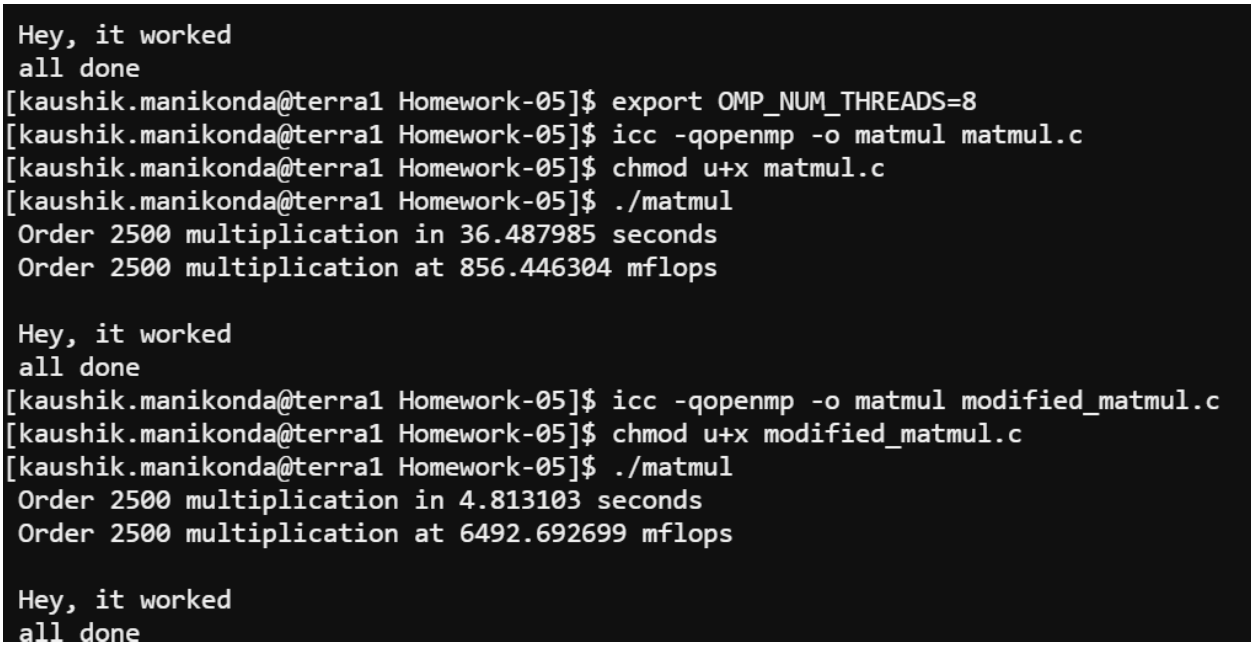


```c
// Loop over grid of points in the complex plane which contains the Mandelbrot set,
// testing each point to see whether it is inside or outside the set.
  #pragma omp parallel for default(shared) private(c, j) firstprivate(eps)
  for (i=0; i<NPOINTS; i++) {
    for (j=0; j<NPOINTS; j++) {
      #pragma omp critical
      c.r = -2.0+2.5*(double)(i)/(double)(NPOINTS)+eps;
      c.i = 1.125*(double)(j)/(double)(NPOINTS)+eps;
      testpoint();
    }
  }
```

STAT_624 - Open MP - Kaushik Manikonda

Show example runs before and after the speedup. Make sure the OpenMP program uses up to 8 CPU's. Set this using the run time environment.

[kaushik.manikonda@terra1 ~]$ export OMP_NUM_THREADS=8

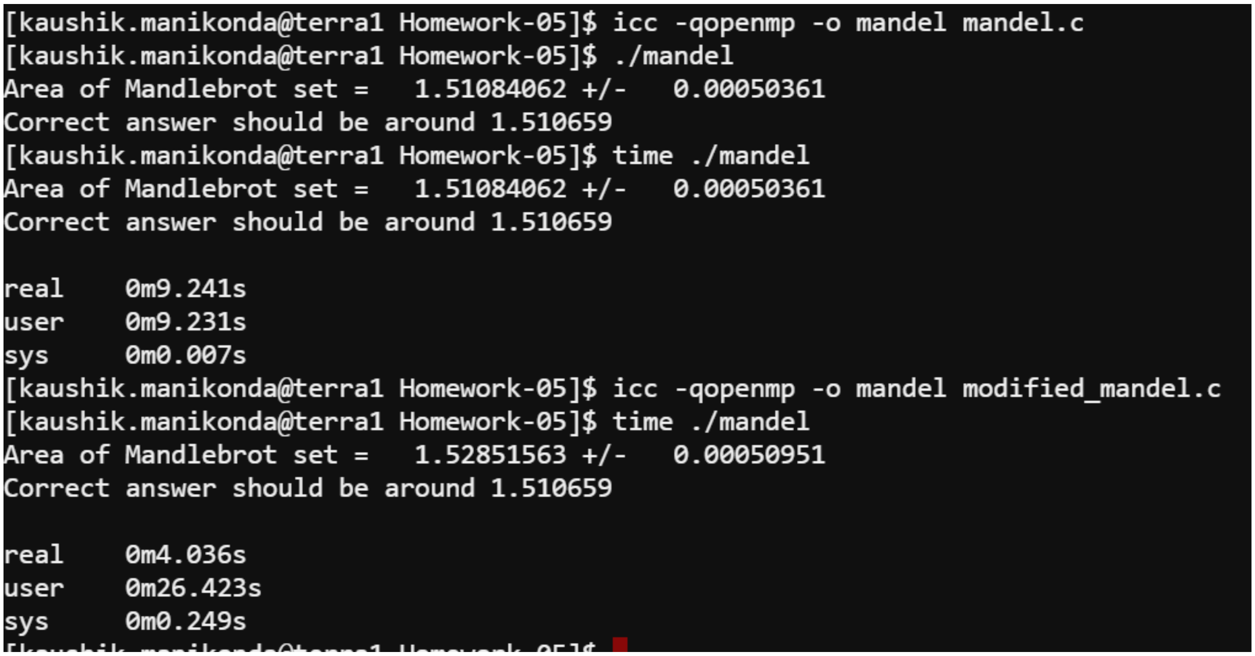


```
 Hey, it worked
 all done
[kaushik.manikonda@terra1 Homework-05]$ export OMP_NUM_THREADS=8
[kaushik.manikonda@terra1 Homework-05]$ icc -qopenmp -o matmul matmul.c
[kaushik.manikonda@terra1 Homework-05]$ chmod u+x matmul.c
[kaushik.manikonda@terra1 Homework-05]$ ./matmul
 Order 2500 multiplication in 36.487985 seconds
 Order 2500 multiplication at 856.446304 mflops

 Hey, it worked
 all done
[kaushik.manikonda@terra1 Homework-05]$ icc -qopenmp -o matmul modified_matmul.c
[kaushik.manikonda@terra1 Homework-05]$ chmod u+x modified_matmul.c
[kaushik.manikonda@terra1 Homework-05]$ ./matmul
 Order 2500 multiplication in 4.813103 seconds
 Order 2500 multiplication at 6492.692699 mflops

 Hey, it worked
 all done
```

The modified matmul with an OpenMP directive is almost 8 times faster than the original matrix multiplication file.



```
[kaushik.manikonda@terra1 Homework-05]$ icc -qopenmp -o mandel mandel.c
[kaushik.manikonda@terra1 Homework-05]$ ./mandel
Area of Mandlebrot set =    1.51084062 +/-    0.00050361
Correct answer should be around 1.510659
[kaushik.manikonda@terra1 Homework-05]$ time ./mandel
Area of Mandlebrot set =    1.51084062 +/-    0.00050361
Correct answer should be around 1.510659

real    0m9.241s
user    0m9.231s
sys     0m0.007s
[kaushik.manikonda@terra1 Homework-05]$ icc -qopenmp -o mandel modified_mandel.c
[kaushik.manikonda@terra1 Homework-05]$ time ./mandel
Area of Mandlebrot set =    1.52851563 +/-    0.00050951
Correct answer should be around 1.510659

real    0m4.036s
user    0m26.423s
sys     0m0.249s
[kaushik.manikonda@terra1 Homework-05]$
```

The modified mandel file is a little more than two times as fast as the original mandel file.