

# EPL372

## Lab Exercise 8: Exercise on OpenMP

# Matrix-Matrix Multiplication

```
#define NRA 1000          /* number of rows in matrix A */
#define NCA 1000          /* number of columns in matrix A */
#define NCB 100          /* number of columns in matrix B */

int main (int argc, char *argv[])
{
    int tid, nthreads, i, j, k, chunk;
    double a[NRA][NCA], /* matrix A to be multiplied */
           b[NCA][NCB], /* matrix B to be multiplied */
           c[NRA][NCB]; /* result matrix C */

    {

        printf("Starting matrix multiple example\n",nthreads);
        printf("Initializing matrices...\n");

        for (i=0; i<NRA; i++)
            for (j=0; j<NCA; j++)
                a[i][j]= i+j;

        for (i=0; i<NCA; i++)
            for (j=0; j<NCB; j++)
                b[i][j]= i*j;

        for (i=0; i<NRA; i++)
            for (j=0; j<NCB; j++)
                c[i][j]= 0;

        for (i=0; i<NRA; i++)
        {
            for(j=0; j<NCB; j++)
                for (k=0; k<NCA; k++)
                    c[i][j] += a[i][k] * b[k][j];
        }
    }
}
```

```
printf("*****\n");
printf("Result Matrix:\n");
for (i=0; i<NRA; i++)
{
    for (j=0; j<NCB; j++)
        printf("%6.2f  ", c[i][j]);
    printf("\n");
}
printf("*****\n");
printf ("Done.\n");
}
```

Convert to OpenMP for any number of threads.  
2 Parallel Regions 1 for initialization and 1 for multiplication (needed to count the time for multiplication only.)