

## Assignment No. 6

PRN: 2020BTECS00025

Name: Sahil Santosh Otari

Course: High Performance Computing Lab

Title of practical: Study and implementation of Open MP program.

Implement following Programs using OpenMP with C:

1. Implementation of Prefix sum.
2. Implementation of Matrix-Vector Multiplication.

1. Implementation of Prefix sum.

Code:

```
#include<stdio.h>
#include<omp.h>
#include<time.h>
void prefixSum(int * arr,int n){
    int i,sum=0;
    #pragma omp parallel for
    for(i=0;i<n;i++){
        #pragma omp critical
        {
            sum+=arr[i];
            arr[i]=sum;
        }
    }
}
int main(){
    double startTime = omp_get_wtime();
    int n=10;
    int arr[n];
    for(int i=1; i<=n; i++){
        arr[i-1] = i;
    }
    prefixSum(arr,n);
    printf("Prefix Sum:");

    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
```

```

double endTime=omp_get_wtime();
printf("Execution time: %f", endTime-startTime);
return 0;
}

```

Output:

```

PS D:\HPC> cd "d:\HPC\Assignment6\" ; if ($?) { g++ prefixSum.cpp -o prefixSum } ; if ($?) { .\prefixSum }
The number of elements in the array: The elements of the array:
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95
96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131
132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 16
6 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 2
01 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235
236 237 238 239 240 241 242 243 244 245 246 247 248 249
Parallel Prefix Sum Array:
0 1 3 6 10 15 21 28 36 45 55 66 78 91 105 120 136 153 171 190 210 231 253 276 300 325 351 378 406 435 465 496 528 561 595 630 666 703 741 7
80 820 861 903 946 990 1035 1081 1128 1176 1225 1275 1326 1378 1431 1485 1540 1596 1653 1711 1770 1830 1891 1953 2016 2080 2145 2211 2278 2
346 2415 2485 2556 2628 2701 2775 2850 2926 3003 3081 3160 3240 3321 3403 3486 3570 3655 3741 3828 3916 4005 4095 4186 4278 4371 4465 4560
4656 4753 4851 4950 5050 5151 5253 5356 5460 5565 5671 5778 5886 5995 6105 6216 6328 6441 6555 6670 6786 6903 7021 7140 7260 7381 7503 7626
7750 7875 8001 8128 8256 8385 8515 8646 8778 8911 9045 9180 9316 9453 9591 9730 9870 10011 10153 10296 10440 10585 10731 10878 11026 11175
11325 11476 11628 11781 11935 12090 12246 12403 12561 12720 12880 13041 13203 13366 13530 13695 13861 14028 14196 14365 14535 14706 14878
15051 15225 15400 15576 15753 15931 16110 16290 16471 16653 16836 17020 17205 17391 17578 17766 17955 18145 18336 18528 18721 18915 19110 1
9306 19503 19701 19900 20100 20301 20503 20706 20910 21115 21321 21528 21736 21945 22155 22366 22578 22791 23005 23220 23436 23653 23871 24
090 24310 24531 24753 24976 25200 25425 25651 25878 26106 26335 26565 26796 27028 27261 27495 27730 27966 28203 28441 28680 28920 29161 294
03 29646 29890 30135 30381 30628 30876 31125
Size of Vector: 250
Number of threads: 4
Time taken = 0.023000 seconds.
Speedup = 2.043478
PS D:\HPC\Assignment6>

```

## 2. Implementation of Matrix-Vector Multiplication.

Code:

```

#include <stdio.h>
#include <omp.h>
#include <time.h>
#define N 4
void matrixVectorMult(int mat[N][N],int vec[N],int result[N]){
    #pragma omp parallel for
    for(int i=0;i<N;i++){
        result[i]=0;
        for(int j=0;j<N;j++){
            result[i]+=mat[i][j]*vec[j];
        }
    }
}
int main(){
    double startTime = omp_get_wtime();
    int mat[N][N];
    int c=1;
    for(int i=0; i<N; i++){
        for(int j=0; j<N; j++){
            mat[i][j] = c++;
        }
    }
}

```

```

    }
    int vec[N];
    for(int i=1; i<=N; i++){
        vec[i-1] = i;
    }
    int result[N];
    matrixVectorMult(mat,vec,result);
    printf("Matrix-Vector Multiplication Result:\n");
    for(int i=0;i<N;i++){
        printf("%d",result[i]);
    }
    printf("\n");
    double endTime = omp_get_wtime();
    printf("Execution time:%f",endTime-startTime);
    return 0;
}

```

## Output:

```

PS D:\HPC> cd "d:\HPC\Assignment3\" ; if ($?) { g++ 2DMatrixAddition.cpp -o 2DMatrixAddition } ; if ($?) { .\2DMatrixAddition }
The size of matrix: 200
The elements of the matrix:
The elements of the vector:
Result vector:
20100 60100 100100 140100 180100 220100 260100 300100 340100 380100 420100 460100 500100 540100 580100 620100 660100 700100 740100 780100 8
20100 860100 900100 940100 980100 1020100 1060100 1100100 1140100 1180100 1220100 1260100 1300100 1340100 1380100 1420100 1460100 1500100 1
540100 1580100 1620100 1660100 1700100 1740100 1780100 1820100 1860100 1900100 1940100 1980100 2020100 2060100 2100100 2140100 2180100 2220
100 2260100 2300100 2340100 2380100 2420100 2460100 2500100 2540100 2580100 2620100 2660100 2700100 2740100 2780100 2820100 2860100 2900100
2940100 2980100 3020100 3060100 3100100 3140100 3180100 3220100 3260100 3300100 3340100 3380100 3420100 3460100 3500100 3540100 3580100 36
20100 3660100 3700100 3740100 3780100 3820100 3860100 3900100 3940100 3980100 4020100 4060100 4100100 4140100 4180100 4220100 4260100 43001
00 4340100 4380100 4420100 4460100 4500100 4540100 4580100 4620100 4660100 4700100 4740100 4780100 4820100 4860100 4900100 4940100 4980100
5020100 5060100 5100100 5140100 5180100 5220100 5260100 5300100 5340100 5380100 5420100 5460100 5500100 5540100 5580100 5620100 5660100 570
0100 5740100 5780100 5820100 5860100 5900100 5940100 5980100 6020100 6060100 6100100 6140100 6180100 6220100 6260100 6300100 6340100 638010
0 6420100 6460100 6500100 6540100 6580100 6620100 6660100 6700100 6740100 6780100 6820100 6860100 6900100 6940100 6980100 7020100 7060100 7
100100 7140100 7180100 7220100 7260100 7300100 7340100 7380100 7420100 7460100 7500100 7540100 7580100 7620100 7660100 7700100 7740100 7780
100 7820100 7860100 7900100 7940100 7980100
Size of Vector: 200
Number of threads: 4
Time taken = 0.010000 seconds.
Speedup = 2.100000
PS D:\HPC\Assignment3>

```