**Lab Exercise 2**

**Private and Shared Constructs**

**Name:** Ayush Sharma

**Reg. No:** 15BCE1335

**Faculty:** Prof. Gayatri. R

1. **Comparison between Parallel and Serial computation for summation of an array.**

**Code:**

int main()

{

clock\_t t1;

int a[1000000];

int sum=0;

for (int i=0;i<100000;i++)

a[i+1]=2\*a[i];

t1 = clock();

for (int i=0;i<100000;i++)

sum=sum+a[i];

t1 = clock() - t1;

double time\_taken1 = ((double)t1)/CLOCKS\_PER\_SEC;

printf("%f seconds for serial\n", time\_taken1);

clock\_t t2;

sum=0;

t2 = clock();

#pragma omp parallel

{

for (int j=0;j<100000;j++)

sum=sum+a[j];

t2 = clock() - t2;

double time\_taken2 = ((double)t2)/CLOCKS\_PER\_SEC;

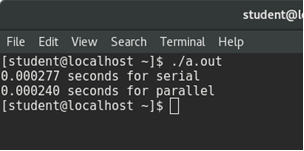
printf("%f seconds for parallel\n", time\_taken2);

}

return 0;

}

**Screenshot:**

****

1. **Choose between parallel and serial implementation using IF condition**

**Code:**

int main()

{

int choice;

printf("Enter 1 for parallel computing else anything for serial computing: ");

scanf("%d",&choice);

#pragma omp parallel if (choice==1)

if (omp\_in\_parallel())

{

#pragma omp single

printf("Parallel\n");

}

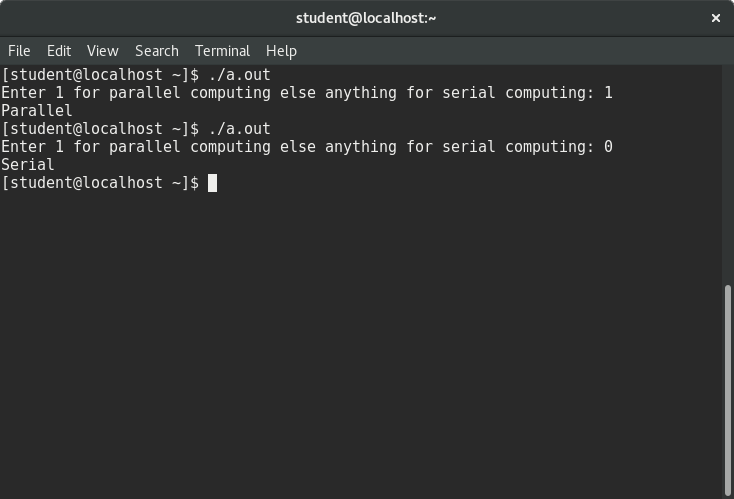
else

printf("Serial\n");

return 0;

}

**Output:**

****

1. **Matrix multiplication using serial and parallel computation.**

**Code:**int main(){

int n = 100;

int a[n][n];

int b[n][n];

int c[n][n];

clock\_t start, end;

double cpu\_time\_used;

for(int i = 0; i &lt; n; i++)

{

for(int j = 0; j &lt; n; j++)

{

a[i][j] = i + j;

   b[i][j] = i - j;

   }

}

start = clock();

for(int i = 0; i &lt; n; i++)

{

for(int j = 0; j &lt; n; j++)

c[i][j] = 0;

   for(int k = 0; k &lt; n; k++)

c[i][j]+=a[i][k]\*b[k][j];

}

end = clock();

cpu\_time\_used = ((double) (end - start)) / CLOCKS\_PER\_SEC;

printf(%f seconds for serial\n" ,cpu\_time\_used);

start = clock();

#pragma omps parallel

for(int i = 0; i &lt; n; i++)

{

for(int j = 0; j &lt; n; j++)

c[i][j] = 0;

for(int k = 0; k &lt; n; k++)

c[i][j]+=a[i][k]\*b[k][j];

}

end = clock();

cpu\_time\_used = ((double) (end - start)) / CLOCKS\_PER\_SEC;

printf(%f seconds for parallel\n",cpu\_time\_used);

}

