**Lab Exercise 3**

**Private and Shared Constructs**

**Name:** Ayush Sharma

**Reg. No:** 15BCE1335

**Faculty:** Prof. Gayatri. R

To differentiate between the working of private and shared constructs, I will be using a basic code for addition of numbers in array. (For this example Array is [0, 1, 2, 3…..99])

**Private:**

int main(){

int sumOut = 0;

int array[100];

for(int i = 0; i < 100; i++){

array[i] = i;

}

#pragma omp parallel

{

int sum = 0;

for(int i = 0; i < 100; i++){

sum += array[i];

}

printf("%d",sum);

sumOut = sum;

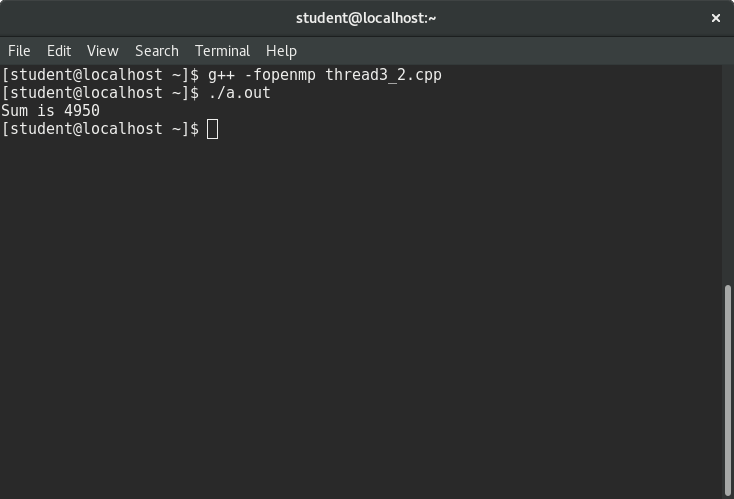
}

printf("Sum is %d",sumOut);

return 0;

}

**Output:**

****

In private mode, every thread has its own variables and the calculations are done individually.

Hence, the output is correct (i.e. 4950).

**Shared:**

int main(){

int sum = 0;

int array[100];

for(int i = 0; i < 100; i++){

array[i] = i;

}

#pragma omp parallel

{

for(int i = 0; i < 100; i++){

sum += array[i];

printf("%d=%d\n",omp\_get\_thread\_num(),sum);

}

printf("\n");

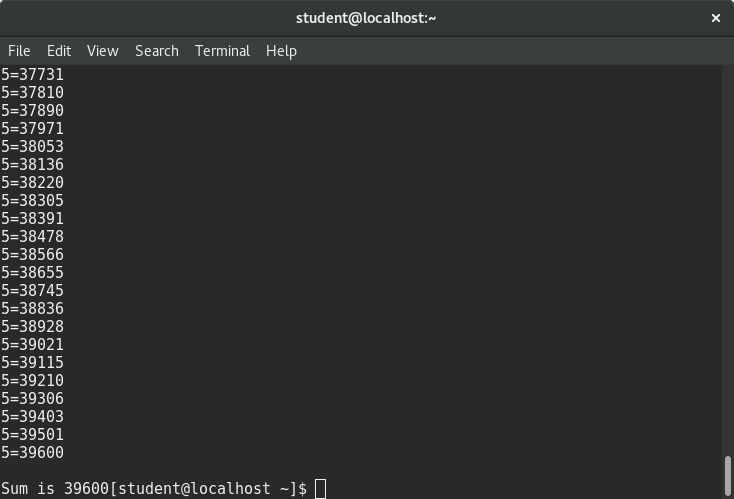
}

printf("Sum is %d",sum);

return 0;

}

**Screenshot:**

****

In shared mode, the variable in which we stored the summation was shared therefore each thread computed and added their values in the same variable. Thus, the result came out to be **4950\*8 = 39600**.