**Name: Manoj Patil**

**Assignment 1:** Write an OpenMP program such that, It should print the name of your family members, such that the names should come from different threads/cores. Also print the respective job id.

#include<stdio.h>

#include<omp.h>

#define NUMBER\_OF\_STRING 6

#define MAX\_STRING\_SIZE 40

int main(int argc, char\* argv[])

{

char arr[NUMBER\_OF\_STRING][MAX\_STRING\_SIZE] = { "Manoj", "Surekha", "Vijay",

"Harshal", "Mugdha", "Rohan" };

omp\_set\_num\_threads(6);

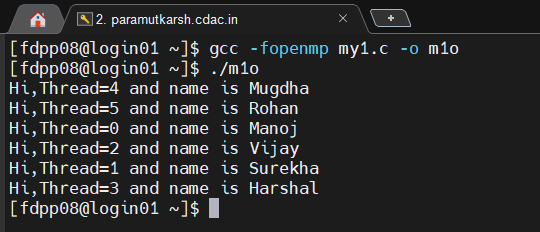
#pragma omp parallel

printf("Hi,Thread=%d and name is %s\n", omp\_get\_thread\_num(), arr[omp\_get\_thread\_num()]);

}

A screen shot of a computer code

Description automatically generated



**Assignment 2:** Write an OpenMP program such that, It should print the sum of square of the thread id’s. Also make sure that, each thread should print the square value of their thread id.

#include<stdio.h>

#include<omp.h>

#define NUMBER\_OF\_ELE 4

int main(int argc,char\* argv[])

{

int arr[4],sum=0;

omp\_set\_num\_threads(4);

#pragma omp parallel

{

arr[omp\_get\_thread\_num()]=omp\_get\_thread\_num()\*omp\_get\_thread\_num();

printf("Thread id=%d and square is %d\n", omp\_get\_thread\_num(),

arr[omp\_get\_thread\_num()]);

sum=sum+arr[omp\_get\_thread\_num()];

printf("Sum=%d \n",sum);

}

printf("Final Sum=%d \n",sum);

}

A screen shot of a computer code

Description automatically generated

A screen shot of a computer code

Description automatically generated

**Assignment 3:** Consider a variable called “Aryabhatta” declared as 10 (i.e int Arbhatta=10). Write an OpenMP program which should print the result of multiplication of thread id and value of the above variable.

#include<stdio.h>

#include<omp.h>

#define NUMBER\_OF\_ELE 4

int main(int argc,char\* argv[])

{

int mult;

int Aryabhatta=10;

omp\_set\_num\_threads(4);

#pragma omp parallel private(Aryabhatta)

{

mult=Aryabhatta\*omp\_get\_thread\_num();

printf("Thread id=%d and Aryabhatta is %d\n",omp\_get\_thread\_num(),mult);

}

return 0;

}

A screen shot of a computer

Description automatically generated

**Assignment 4:** Write an OpenMP program that calculates the sum of the first 20 natural numbers using parallelism. Each thread should compute a portion of the sum by iterating through a loop. Implement the program using the lastprivate clause to ensure that the final total sum is correctly computed and printed outside the parallel region.

#include <omp.h>

#include <stdio.h>

#include <stdlib.h>

int main (int argc, char \*argv[])

{

int sum = 0;

int i;

omp\_set\_num\_threads(2);

#pragma omp parallel for lastprivate(sum)

for(i = 1; i <= 20; i++){

sum = sum + i;

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

}

printf("The sum of 20 natural numbers is %d\n\n\n", sum);

}

A computer screen shot of a code

Description automatically generated

A screen shot of a computer

Description automatically generated