



Parallel
Programming:
OpenMP

Natasha S.
Sharma, PhD

Parallel Programming: OpenMP

Natasha S. Sharma, PhD



Building Blocks

Parallel
Programming:
OpenMP

Natasha S.
Sharma, PhD

OpenMP stands for Open Multi-Processing.



Building Blocks

Parallel
Programming:
OpenMP

Natasha S.
Sharma, PhD

OpenMP stands for Open Multi-Processing.

Thread: Thread of execution is defined as the smallest sequence of programmed instructions that can be managed independently.
Multiple threads of a given process may be executed concurrently.



Building Blocks

Parallel
Programming:
OpenMP

Natasha S.
Sharma, PhD

OpenMP stands for Open Multi-Processing.

Thread: Thread of execution is defined as the smallest sequence of programmed instructions that can be managed independently.

Multiple threads of a given process may be executed concurrently.

is called a directive



Building Blocks

Parallel
Programming:
OpenMP

Natasha S.
Sharma, PhD

OpenMP stands for Open Multi-Processing.

Thread: Thread of execution is defined as the smallest sequence of programmed instructions that can be managed independently.

Multiple threads of a given process may be executed concurrently.

is called a directive

OpenMP is an Application Program Interface for parallelism with Shared Memory and Multiple Threads.



Building Blocks

Parallel
Programming:
OpenMP

Natasha S.
Sharma, PhD

OpenMP stands for Open Multi-Processing.

Thread: Thread of execution is defined as the smallest sequence of programmed instructions that can be managed independently.

Multiple threads of a given process may be executed concurrently.

is called a directive

OpenMP is an Application Program Interface for parallelism with Shared Memory and Multiple Threads.

Supports C/C++ and Fortran.



OpenMP: Run-time Library Routines

Parallel
Programming:
OpenMP

Natasha S.
Sharma, PhD

```
#include<omp.h>
```

Needed for the routines to work for C/C++ functions.

What does this Library actually do?

Usage



OpenMP: Run-time Library Routines

Parallel
Programming:
OpenMP

Natasha S.
Sharma, PhD

```
#include<omp.h>
```

Needed for the routines to work for C/C++ functions.

What does this Library actually do?

- 1 Setting and querying the number of threads.

Usage



OpenMP: Run-time Library Routines

Parallel
Programming:
OpenMP

Natasha S.
Sharma, PhD

```
#include<omp.h>
```

Needed for the routines to work for C/C++ functions.

What does this Library actually do?

- 1 Setting and querying the number of threads.
- 2 Querying a thread's unique identifier (thread ID).

Usage



OpenMP: Run-time Library Routines

Parallel
Programming:
OpenMP

Natasha S.
Sharma, PhD

```
#include<omp.h>
```

Needed for the routines to work for C/C++ functions.

What does this Library actually do?

- 1 Setting and querying the number of threads.
- 2 Querying a thread's unique identifier (thread ID).

Usage

- 1 Access the total number of threads:

```
int num_threads = omp_get_num_threads();
```



OpenMP: Run-time Library Routines

Parallel
Programming:
OpenMP

Natasha S.
Sharma, PhD

```
#include<omp.h>
```

Needed for the routines to work for C/C++ functions.

What does this Library actually do?

- 1 Setting and querying the number of threads.
- 2 Querying a thread's unique identifier (thread ID).

Usage

- 1 Access the total number of threads:

```
int num_threads = omp_get_num_threads();
```

- 2 Get the thread ID for the current thread:

```
int num_threads = omp_get_num_threads();
```



OpenMP: Compiler Directives

Parallel
Programming:
OpenMP

Natasha S.
Sharma, PhD

1 `#pragma omp directive-name`

The compiler directive applies to one succeeding statement. That is, the directive-name applies to the statement that following the directive.

To enclose multiple statements we enclose the statements in braces that is { }.



OpenMP: Compiler Directives

Parallel
Programming:
OpenMP

Natasha S.
Sharma, PhD

1 `#pragma omp directive-name`

The compiler directive applies to one succeeding statement. That is, the directive-name applies to the statement that following the directive.

To enclose multiple statements we enclose the statements in braces that is { }.

2 The compiler directive is case sensitive.



OpenMP: Compiler Directives Examples

Parallel
Programming:
OpenMP

Natasha S.
Sharma, PhD

1 `#pragma omp parallel`

The code is to be executed by multiple threads in parallel.

The compile and run command:

`$ export OMP_NUM_THREADS= N, N could be any positive integer say 6 or 4 or 3.`

`$ gcc -o hello -fopenmp hello-world-open-mp.c`

`./hello`

Access class-103122/code01 folder for practice.



OpenMP: Compiler Directives Examples

Parallel
Programming:
OpenMP

Natasha S.
Sharma, PhD

1 `#pragma omp parallel`

The code is to be executed by multiple threads in parallel.

2 `#pragma omp for`

The work in a for loop to be divided among threads.

The compile and run command:

`$ export OMP_NUM_THREADS= N`, N could be any positive integer say 6 or 4 or 3.

`$ gcc -o hello -fopenmp hello-world-open-mp.c`

`./hello`

Access class-103122/code01 folder for practice.



OpenMP: Compiler Directives Examples

Parallel
Programming:
OpenMP

Natasha S.
Sharma, PhD

1 `#pragma omp parallel`

The code is to be executed by multiple threads in parallel.

2 `#pragma omp for`

The work in a for loop to be divided among threads.

3 `#pragma omp parallel for`

Shortcut combining the previous two.

The compile and run command:

\$ export OMP_NUM_THREADS= N, N could be any positive integer say 6 or 4 or 3.

\$ gcc -o hello -fopenmp hello-world-open-mp.c
./hello

Access class-103122/code01 folder for practice.