OpenMP - worksharing constructs

A **work-sharing construct** divides the execution of the enclosed code region among the members of the team that encounter it, but do not launch new threads. There is no implied barrier upon entry to a work-sharing construct, however there is an implied barrier at the end of a work sharing construct.

There are three type of constructs:

Do/for-loop

Shares iterations of a loop across the team. Represents a type of "data parallelism".

```
#pragma omp for [clause ...]  newline
                schedule (type [,chunk])
                ordered
                private (list)
                firstprivate (list)
                lastprivate (list)
                shared (list)
                reduction (operator: list)
                collapse (n)
                nowait

    for_loop
```

Sections

Breaks work into separate, discrete sections. Each section is executed by a thread. Can be used to implement a type of "functional parallelism".

```
#pragma omp sections [clause ...]  newline
                private (list)
                firstprivate (list)
                lastprivate (list)
```

```
                    reduction (operator: list)
                    nowait

  {

  #pragma omp section    newline

      structured_block

  #pragma omp section    newline

      structured_block


  }
```

Single

Serializes a section of code.

```
#pragma omp single [clause ...]  newline
                 private (list)
                 firstprivate (list)
                 nowait

     structured_block
```