

Parallel Construct

- Loop
-

Copyright (c) 2021 - 2020 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

Parallel region

the **parallel pragma** :
creates parallelism in OpenMP codes

a **parallel region** :

- a block preceded by the **omp parallel pragma**
- executed by a newly created team of **threads**.

```
#pragma omp parallel
{
  // this is executed by a team of threads
}
```

this is an instance of the **SPMD** model:
all threads execute the same segment of code.

<https://pages.tacc.utexas.edu/~eijkhout/pcse/html/omp-parallel.html>

Shared / Private data and thread number

the **threads** that are forked are
all **copies** of the **master thread** :

shared data :
they have access to all that was computed so far;

each thread can also have **private data**,

each thread can identify themselves:
thread number.

<https://pages.tacc.utexas.edu/~eijkhout/pcse/html/omp-parallel.html>

Thread number function

`omp_get_thread_num()` :

to find out which thread you are
and to execute work that is individual to that thread.

`omp_get_num_threads()`

to find out the total number of threads.

these functions give a number
relative to the current team

<https://pages.tacc.utexas.edu/~eijkhout/pcse/html/omp-parallel.html>

References

- [1] en.wikipedia.org
- [2] M Harris, <http://beowulf.lcs.mit.edu/18.337-2008/lectslides/scan.pdf>