Private Clause (4A)

- Loop
-

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

This document was produced by using OpenOffice and Octave.

Clauses (1)

private (list)

Declares the scope of the data variables in list
to be **private** to each thread.
Data variables in list are separated by commas.

https://www.ibm.com/docs/en/xl-c-aix/13.1.2?topic=processing-pragma-omp-section-pragma-omp-sections

Clauses (2)

firstprivate (list)

Declares the scope of the data variables in list
to be **private** to each thread.
Each new private object is initialized
as if there was an implied declaration within the statement block.
Data variables in list are separated by commas.

lastprivate (list)

Declares the scope of the data variables in list
to be **private** to each thread.
The final value of each variable in list, if assigned,
will be the value assigned to that variable in the last section.
Variables not assigned a value will have an indeterminate value.
Data variables in list are separated by commas.

https://www.ibm.com/docs/en/xl-c-aix/13.1.2?topic=processing-pragma-omp-section-pragma-omp-sections

Collapse example (3)

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

int main(void)
{
    int var = 100;

    printf("Before parallelism, var's address is %p\n", &var);

    #pragma omp parallel for
    for (int i = 0; i < 4; i++)
    {
        printf("var's address in thread %d is %p\n", omp_get_thread_num(), &var);
        var = i;
    }

    printf("After parallelism, var's address is %p, and value is %d\n", &var, var);
    return 0;
}
```

https://nanxiao.gitbooks.io/openmp-little-book/content/posts/collapse-clause.html?q=

Collapse example (3)

```
# gcc -fopenmp parallel.c
# ./a.out
Before parallelism, var's address is 0x7ffe993d79fc
var's address in thread 1 is 0x7ffe993d79fc
var's address in thread 2 is 0x7ffe993d79fc
var's address in thread 3 is 0x7ffe993d79fc
var's address in thread 0 is 0x7ffe993d79fc
After parallelism, var's address is 0x7ffe993d79fc, and value is 0
# ./a.out
Before parallelism, var's address is 0x7ffc13c7cd8c
var's address in thread 3 is 0x7ffc13c7cd8c
var's address in thread 2 is 0x7ffc13c7cd8c
var's address in thread 0 is 0x7ffc13c7cd8c
var's address in thread 1 is 0x7ffc13c7cd8c
After parallelism, var's address is 0x7ffc13c7cd8c, and value is 1
# ./a.out
Before parallelism, var's address is 0x7ffd72be3bbc
var's address in thread 3 is 0x7ffd72be3bbc
var's address in thread 2 is 0x7ffd72be3bbc
var's address in thread 0 is 0x7ffd72be3bbc
var's address in thread 1 is 0x7ffd72be3bbc
After parallelism, var's address is 0x7ffd72be3bbc, and value is 1
```

https://nanxiao.gitbooks.io/openmp-little-book/content/posts/collapse-clause.html?q=

Collapse example (3)

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

int main(void)
{
    int var = 100;

    printf("Before parallelism, var's address is %p\n", &var);

    #pragma omp parallel for private(var)
    for (int i = 0; i < 4; i++)
    {
        printf("var's address in thread %d is %p, value is %d\n", omp_get_thread_num(), &var, var);
        var = i;
    }

    printf("After parallelism, var's address is %p, and value is %d\n", &var, var);
    return 0;
}
```

https://nanxiao.gitbooks.io/openmp-little-book/content/posts/collapse-clause.html?q=

Collapse example (3)

```
# gcc -fopenmp parallel.c
# ./a.out
Before parallelism, var's address is 0x7ffde0e4c124
var's address in thread 3 is 0x7f0fe9bf0e20, value is 0
var's address in thread 1 is 0x7f0feabf2e20, value is 0
var's address in thread 2 is 0x7f0fea3f1e20, value is 0
var's address in thread 0 is 0x7ffde0e4c0c0, value is 24
After parallelism, var's address is 0x7ffde0e4c124, and value is 100
```

https://nanxiao.gitbooks.io/openmp-little-book/content/posts/collapse-clause.html?q=

Collapse example (3)

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

int main(void)
{
    int var = 100;

    printf("Before parallelism, var's address is %p\n", &var);

    #pragma omp parallel for firstprivate(var)
    for (int i = 0; i < 4; i++)
    {
        printf("var's address in thread %d is %p, value is %d\n", omp_get_thread_num(), &var, var);
        var = i;
    }

    printf("After parallelism, var's address is %p, and value is %d\n", &var, var);
    return 0;
}
```

https://nanxiao.gitbooks.io/openmp-little-book/content/posts/collapse-clause.html?q=

Collapse example (3)

```
# gcc -fopenmp parallel.c
# ./a.out
Before parallelism, var's address is 0x7ffe3ae46e00
var's address in thread 0 is 0x7ffe3ae46da0, value is 100
var's address in thread 3 is 0x7ff6870ede20, value is 100
var's address in thread 2 is 0x7ff6878eee20, value is 100
var's address in thread 1 is 0x7ff6880efe20, value is 100
After parallelism, var's address is 0x7ffe3ae46e00, and value is 100
```

https://nanxiao.gitbooks.io/openmp-little-book/content/posts/collapse-clause.html?q=

Collapse example (3)

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

int main(void)
{
    int var = 100;

    printf("Before parallelism, var's address is %p\n", &var);

    #pragma omp parallel for lastprivate(var)
    for (int i = 0; i < 4; i++)
    {
        printf("var's address in thread %d is %p, value is %d\n", omp_get_thread_num(), &var, var);
        var = i;
    }

    printf("After parallelism, var's address is %p, and value is %d\n", &var, var);
    return 0;
}
```

https://nanxiao.gitbooks.io/openmp-little-book/content/posts/collapse-clause.html?q=

Collapse example (3)

```
# gcc -fopenmp parallel.c
# ./a.out
Before parallelism, var's address is 0x7ffd53ea3fb0
var's address in thread 0 is 0x7ffd53ea3f50, value is 24
var's address in thread 3 is 0x7fd70584fe20, value is 0
var's address in thread 2 is 0x7fd706050e20, value is 0
var's address in thread 1 is 0x7fd706851e20, value is 0
After parallelism, var's address is 0x7ffd53ea3fb0, and value is 3
# ./a.out
Before parallelism, var's address is 0x7fff72604380
var's address in thread 0 is 0x7fff72604320, value is 24
var's address in thread 2 is 0x7f6f3ac49e20, value is 0
var's address in thread 1 is 0x7f6f3b44ae20, value is 0
var's address in thread 3 is 0x7f6f3a448e20, value is 0
After parallelism, var's address is 0x7fff72604380, and value is 3
```

https://nanxiao.gitbooks.io/openmp-little-book/content/posts/collapse-clause.html?q=

Clauses (3)

copyprivate (list)

broadcasts the values of variables specified in list
from one thread of the team to other threads

This occurs after the execution of the structured block
associated with the omp single directive,
and before any of the threads leave
the barrier at the end of the construct.

For all other threads in the team,
each variable in the list becomes defined
with the value of the corresponding variable
in the thread that executed the structured block.

Data variables in list are separated by commas.

https://www.ibm.com/docs/en/xl-c-aix/13.1.2?topic=processing-pragma-omp-section-pragma-omp-sections

References

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