

OpenMP Parallel Prefix Sum (1A)

•
•

Copyright (c) 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.

OpenMP Parallel Prefix Sum (1)

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "omp.h"
#define N 11 //33554432 // 2 ^ 25
```

<https://stackoverflow.com/questions/29200949/parallel-computing-for-prefix-sum-using-openmp-c-code>

OpenMP Parallel Prefix Sum (2)

```
void computeparallelprefix(int *iplist, int *_pprefixsum, unsigned long size)
{
    int nthr, *z, *x = _pprefixsum;
    #pragma omp parallel
    {
        int i;
        #pragma omp single
        {
            nthr = omp_get_num_threads();
            z = malloc(sizeof(int)*nthr+1);
            z[0] = 0;
        }
        int tid = omp_get_thread_num();
        int sum = 0;
```

<https://stackoverflow.com/questions/29200949/parallel-computing-for-prefix-sum-using-openmp-c-code>

OpenMP Parallel Prefix Sum (3)

```
#pragma omp for schedule(static)
```

```
for(i=0; i<size; i++) {
```

```
    sum += iplist[i];
```

```
    x[i] = sum;
```

```
}
```

```
z[tid+1] = sum;
```

```
#pragma omp barrier
```

```
int offset = 0;
```

```
for(i=0; i<(tid+1); i++) {
```

```
    offset += z[i];
```

```
}
```

<https://stackoverflow.com/questions/29200949/parallel-computing-for-prefix-sum-using-openmp-c-code>

OpenMP Parallel Prefix Sum (4)

```
#pragma omp for schedule(static)
for(i=0; i<size; i++) {
    x[i] += offset;
}
}
free(z);
}
```

<https://stackoverflow.com/questions/29200949/parallel-computing-for-prefix-sum-using-openmp-c-code>

OpenMP Parallel Prefix Sum (6)

```
int main(void ) {  
  
    int *iplist, *pprefixsum ;  
  
    iplist = (int*) malloc(sizeof(int) * N);  
    pprefixsum = (int*) malloc(sizeof(int) * N);  
  
    for(int i=0; i<N; i++) iplist[i] = i+1;  
    for(int i=0; i<N; i++) printf("%d ", iplist[i]); printf("\n");  
  
    computeparallelprefix(iplist, pprefixsum, N);  
    for(int i=0; i<N; i++) printf("%d ", pprefixsum[i]); printf("\n");  
    for(int i=0; i<N; i++) printf("%d ", (i+1)*(i+2)/2); printf("\n");  
    return 0;  
}
```

<https://stackoverflow.com/questions/29200949/parallel-computing-for-prefix-sum-using-openmp-c-code>

References

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