Tugas Pemrograman Parallel

Oleh:

D121171519 - Glenn Claudio Ivan Petrus

Departemen Teknik Informatika

Fakultas Teknik

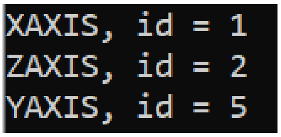
Universitas Hasanuddin

2020

The parallel sections Construct

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

void XAXIS();
void YAXIS();
void ZAXIS();

int main(){
    #pragma omp parallel sections
    {
        #pragma omp section
        {
            XAXIS();
        }

        #pragma omp section
        {
            YAXIS();
        }

        #pragma omp section
        {
            ZAXIS();
        }
    }
}

void XAXIS (){
    printf ("XAXIS, id = %d\n", omp_get_thread_num());
}

void YAXIS (){
    printf ("YAXIS, id = %d\n", omp_get_thread_num());
}

void ZAXIS (){
    printf ("ZAXIS, id = %d\n", omp_get_thread_num());
}
```

Output



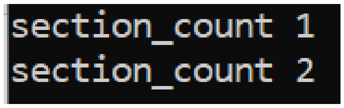
```
XAXIS, id = 1
ZAXIS, id = 2
YAXIS, id = 5
```

The firstprivate Clause and the sections Construct (1)

```
#include <omp.h>
#include <stdio.h>
#define NT 1

int main (){
    int section_count = 0;
    omp_set_dynamic(0);
    omp_set_num_threads(NT);

    #pragma omp parallel
    #pragma omp sections firstprivate(section_count)
    {
        #pragma omp section
        {
            section_count++;
            /* may print the number one or two */
            printf("section_count %d\n", section_count);
        }

        #pragma omp section
        {
            section_count++;
            /* may print the number one or two */
            printf("section_count %d\n", section_count);
        }

    }

    return 0;
}
```

Note: Using 1 thread (the same thread)

Output



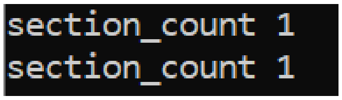
```
section_count 1
section_count 2
```

The firstprivate Clause and the sections Construct (2)

```
#include <omp.h>
#include <stdio.h>
#define NT 4

int main (){
     int section_count = 0;
     omp_set_dynamic(0);
     omp_set_num_threads(NT);

     #pragma omp parallel
     #pragma omp sections firstprivate(section_count)
     {
          #pragma omp section
          {
               section_count++;
               /* may print the number one or two */
               printf("section_count %d\n", section_count);
          }

          #pragma omp section
          {
               section_count++;
               /* may print the number one or two */
               printf("section_count %d\n", section_count);
          }

     }

     return 0;
}
```

Note: Using 4 threads (the different threads)

Output



```
section_count 1
section_count 1
```