

OpenMP 演習課題

問題 1

以下のサンプルプログラムをスレッド数を2、4、8 と変えて実行し、その動作を確認せよ。

[1] サンプルプログラム 1

Fortran版

```
program mythred
  integer :: OMP_GET_THREAD_NUM
  !$OMP PARALLEL
    write(*,*) "My thread id is ",OMP_GET_THREAD_NUM(), " Hello"
  !$OMP END PARALLEL
end program mythred
```

C版

```
#include <stdio.h>
#include <omp.h>
int main( void )
{
  int myid;
  #pragma omp parallel private(myid)
    printf("My thread id is %d Hallo¥n", myid = omp_get_thread_num());
}
```

[2] サンプルプログラム 2

Fortran版

```
program schedule
  real :: a(10000)
  integer :: omp_get_thread_num, myid
  integer :: i
  !$OMP PARALLEL PRIVATE(myid)
    myid=omp_get_thread_num()
  !$OMP DO SCHEDULE(STATIC,2)
    do i=1,10
      a(i)=i
      if (myid.eq.0) then
```

```

        write(*,*) i
    endif
enddo
!$OMP END PARALLEL
end program schedule

```

C版

```

#include <stdio.h>
#include <omp.h>
int main( void )
{
    int i, myid;
    float a[10000];

    #pragma omp parallel private(myid)
    {
        myid=omp_get_thread_num();
        #pragma omp for schedule(static,2)
        for(i=1;i<11;i++){
            a[i]=i;
            if(myid == 0){
                printf("%d¥n", i);
            }
        }
    }
}

```

問題2

以下のプログラムをOpenMP により並列化せよ。但し、プログラムは一部のみであるので、適当に配列の初期化、宣言等の必要部分を追加すること。

Fortran版

[1]

```

do i=1,n

```

```

    b(i)=f(i)
    a(i)=b(i-1)+c(i)
enddo

```

[2] 行列ベクトル積

```

do i=1,n
    do j=1,n
        x(i)=x(i)+a(i,j)*b(j)
    enddo
enddo

```

[3] 以下のループをスレッド数2 で並列化し、制御変数*i* が奇数の場合にはスレッド0、制御変数*i* が偶数の場合にはスレッド1 が処理するようにせよ。

```

do i=1,n
    a(i)=b(i)
enddo

```

[4] 以下のループをスレッド数2 で並列化せよ。

```

do i=2,n
    a(i)=a(i-2)+b(i)
enddo

```

[5] ベクトルの内積

```

do i=1,n
    v=v+a(i)*b(i)
enddo

```

[6] Stencil の計算

```

do j=2,n
    do i=1,n
        x(i,j)=x(i,j)+x(i,j-1)
    enddo
enddo

```

C版

[1]

```

for (i=1; i<n; i++){
    b[i]=f[i];
}

```

```

    a[i]=b[i-1]+c[i];
}

```

[2] 行列ベクトル積

```

for (j=0; j<n; j++){
    for (i=0; i<n; i++){
        x[i]=x[i]+a[j][i]*b[j];
    }
}

```

[3] 以下のループをスレッド数2 で並列化し、制御変数*i* が奇数の場合にはスレッド0、制御変数*i* が偶数の場合にはスレッド1 が処理するようにせよ。

```

for (i=0;i<n;i++){
    a[i]=b[i];
}

```

[4] 以下のループをスレッド数2 で並列化せよ。

```

for (i=2;i<n;i++){
    a[i]=a[i-2]+b[i];
}

```

[5] ベクトルの内積

```

for (i=0;i<n;i++){
    v=v+a[i]*b[i];
}

```

[6] Stencil の計算

```

for (j=1;j<n;j++){
    for (i=0;i<n;i++){
        x[j][i]=x[j][i]+x[j-1][i];
    }
}

```

問題3

以下のFORTRAN あるいはC のプログラムをOpenMP により並列化せよ

・FORTRANによるプログラム

```

program calculate_pi
    integer :: i, n

```

```

      real(kind=8) :: w, gsum, pi, v
!
      n=200000000
      w = 1.0d0 / n
      gsum = 0.0d0
      do i = 1, n
         v = (i - 0.5d0 ) * w
         v = 4.0d0 / (1.0d0 + v * v)
         gsum = gsum + v
      end do
      pi = gsum * w
      print *, 'PI is ', pi
end program calculate_pi

```

・C によるプログラム

```

#include <stdio.h>
#include <string.h>
#include <omp.h>
int main( void )
{
    int i,n;
    double w,gsum,pi,v;
    n = 200000000;
    w = (double)1.0 / n;
    gsum = (double)0.0;
    for( i=0; i<n; i++ ) {
        v = ( i + (double)0.5 ) * w;
        v = (double)4.0 / ( (double)1.0 + v * v );
        gsum = gsum + v;
    }
    pi = gsum * w;
    printf( "%s%f\n", "PI is ", pi );
    return 0;
}

```


OpenMP 演習課題 (自由課題)

1. レポート課題の拡散方程式の差分解析において、解析モデルの中央に穴があいていた場合には各スレッドのLoad balance をとるにはどうしたらよいか。
2. 2次元ポアソン方程式の境界値問題を5点差分公式により解く場合に生じる連立一次方程式をガウスザイデル法で解くとする。この場合、ガウスザイデル法を並列化することは可能か？

※ポアソン方程式は以下の通り。

$$\nabla^2 \phi = -\frac{\rho}{\epsilon_0}$$
$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = -\frac{\rho}{\epsilon_0}$$

※ガウスザイデル法は連立1次方程式の反復解法の一つ。