



OpenMP

- Lab 3 -

# Sumar

- Constructii de partajare a taskurilor/threadurilor
- Constructii de sincronizare a taskurilor/threadurilor
  - Copii locale ale variabilelor globale
  - Contextul variabilelor si al datelor

# Constructii de partajare a taskurilor/threadurilor

## SECTIONS - [docs](#)

- 1 thread executa un “section”
- bariera implicita la terminarea “sections”

```
#pragma omp sections [clause ...]  
{  
    #pragma omp section  
        structured_block  
    #pragma omp section  
        structured_block  
}
```

# Constructii de partajare a taskurilor/threadurilor

## SINGLE – [docs](#)

- 1 singur thread executa o regiune single
- util pentru zone non-thread safe
- restul thread-urilor asteapta terminarea SINGLE
  - exceptie nowait

*#pragma omp single [clause ...]*

*structured\_block*

# Constructii de partajare a taskurilor/threadurilor

## DIRECTIVE COMBINATE

```
#pragma omp parallel for \  
    shared(a,b,c,chunk) private(i) \  
    schedule(static,chunk)  
        for (i=0; i < N; i++)  
            c[i] = a[i] + b[i];  
  
}
```

# Constructii de sincronizare a taskurilor/threadurilor

- MASTER – thread-ul 0 executa regiunea
- CRITICAL – executie serializata a thread-urilor
- BARRIER – asteapta toate thread-urile
- ATOMIC – actualizare atomica a unei zone de memorie

```
#pragma omp atomic
sum += a[i]
```
- ORDERED – asigura executia fireasca a chunk-urilor unui for

```
# pragma omp for ordered schedule(static, chunk)
chunk0, chunk1, chunk2...
```

# Copii locale ale variabilelor globale

THREADPRIVATE

*#pragma omp threadprivate (list)*

# Contextul variabilelor si al datelor

- PRIVATE – referinta globala -> referinta locala
- FIRSTPRIVATE – la fel ca private + initializare
- LASTPRIVATE – la fel ca private + update global
- SHARED – pastreaza referina globala
- DEFAULT – specifica comportamentul implicat
- REDUCTION – reduction(+:sum)
- COPYIN – copiaza valoarea globala local



# Task-uri

1. Scrieti un program paralel in care sa folositi SECTIONS. Afisati thread-id in cadrul programului. Ce observati?
2. Modificati programul ex\_sections.c pentru a beneficia de OpenMP sections.
3. Aplicati lucrurile invatate in acest laborator pentru a imbunatati performantele:
  1. Inmultirii unei matrice cu un vector - [docs](#)
  2. Inmultirii a doua matrice - [docs](#)
  3. Legat de lucrurile invatate in laboratorul precedent, variati tipurile de planificare in cadrul celor 2 probleme si observati modificarile performantelor.