

OpenMP

(ne zamenjajte z OpenMPI!)

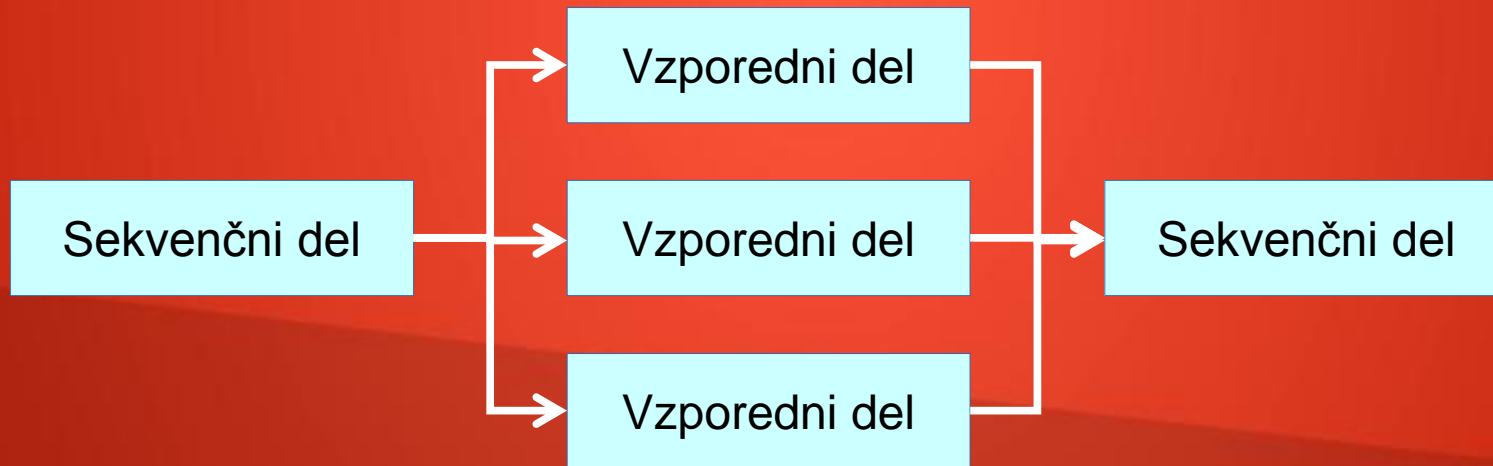
Vaja1

- `#include <omp.h>`
- `#pragma omp ...`
- Prevajanje:
 - `gcc -fopenmp vaja1.c -o vaja1`
- Poganjanje:
 - `./vaja1`

Vaja2

- Spominja kaj na MPI?

```
#pragma omp ...  
{  
    }  
}
```



Vaja3

- `#pragma omp ... private (id)`
- V nasprotju z MPI si pri OpenMP
 - Vse niti delijo spremenljivke
 - Ne rabijo eksplicitne komunikacije

Vaja4

- Implicitna komunikacija
- Je to sploh komunikacija
Ali raje skupinsko delo

Vaja5

- Parallel for
- Razdelitev dela
 - `schedule(static)`
 - `schedule(static, k)`
 - `schedule(dynamic)`
 - `schedule(guided)`

Vaja6

- Še vedno schedule



Vaja7

- reduction
- Globalnih (shared) spremenljivk se ne sme nezaščitenno spreminjati
- Omejitve na for zanki:
 - Kaj če pogoj spremenimo v $(\text{count} < 4)$?
 - Ali pa v $(i*i < 60*60)$?

Vaja8

- sections

vaja9

- Separacija omp parallel in ostalih direktiv

vaja10

- Gnezdenje paralelnih direktiv

Vaja11

- Statično in dinamično razporejanje delov zanke – merjenje časov

Vaja12

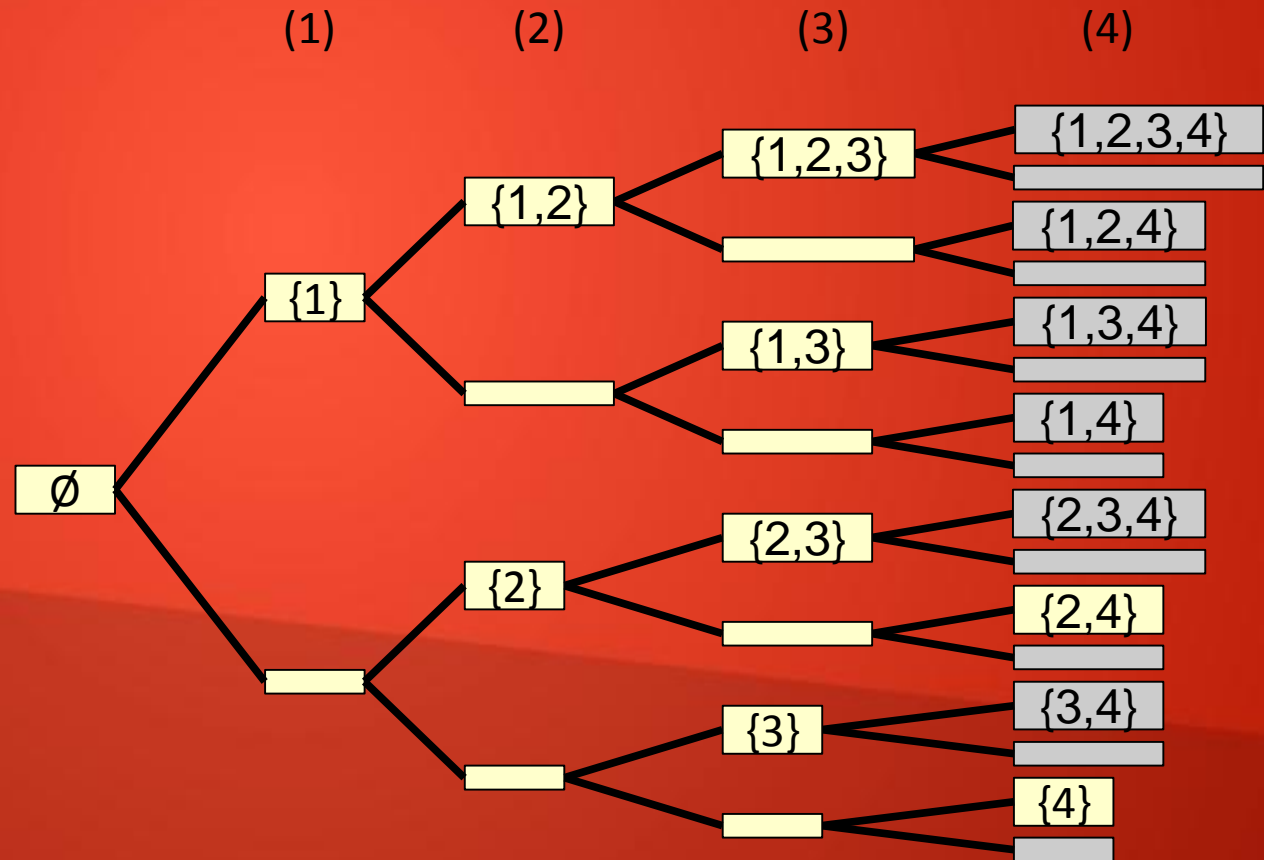
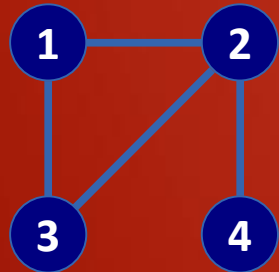
- Statično in dinamično razporejanje delov zanke – merjenje časov

Linki

- <http://openmp.org/mp-documents/OpenMP3.1-CCard.pdf>
- <http://www.mpi-forum.org/docs/docs.html>
- <http://www.mpich.org/static/docs/v3.1/www3/>
- <http://web.eecs.utk.edu/~dongarra/WEB-PAGES/SPRING-2006/mpi-quick-ref.pdf>

Branch and bound algoritem

- Rekurzijo se paralelizira z direktivo task



Branch and bound algoritem

- Rekurzijo se paralelizira z direktivo task

