

# Parallel Computing for Science & Engineering (PCSE 374C/394C )

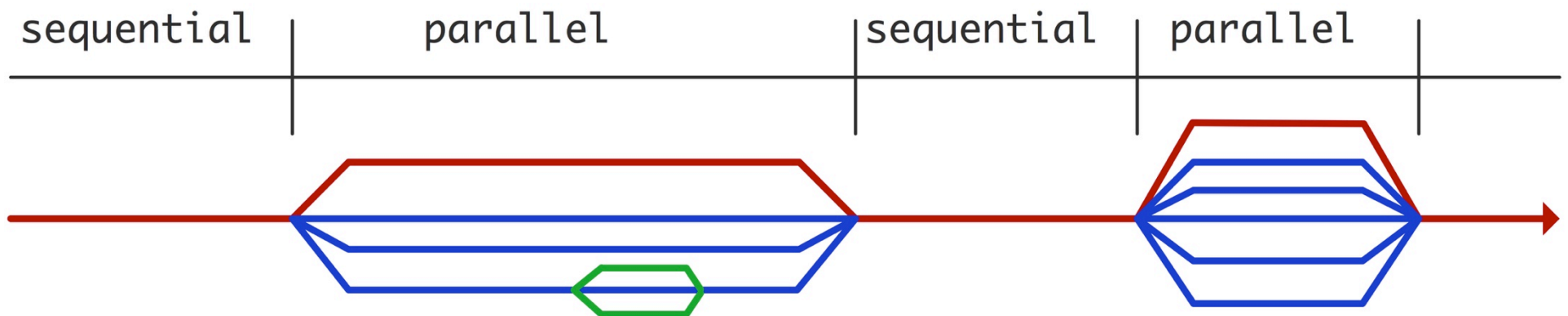
About the OpenMP  
process and data model

Instructors:

Victor Eijkhout, Cyrus Proctor

# Parallel regions

- OpenMP uses fork/join parallelism
- Fork: one thread splits itself into a team
- Join: the team dissolves and only the original thread remains.
- This can be done recursively.



# OpenMP parallel regions

- C: pragma followed by statement or block
- Fortran: delimiting 'sentinel' comments.

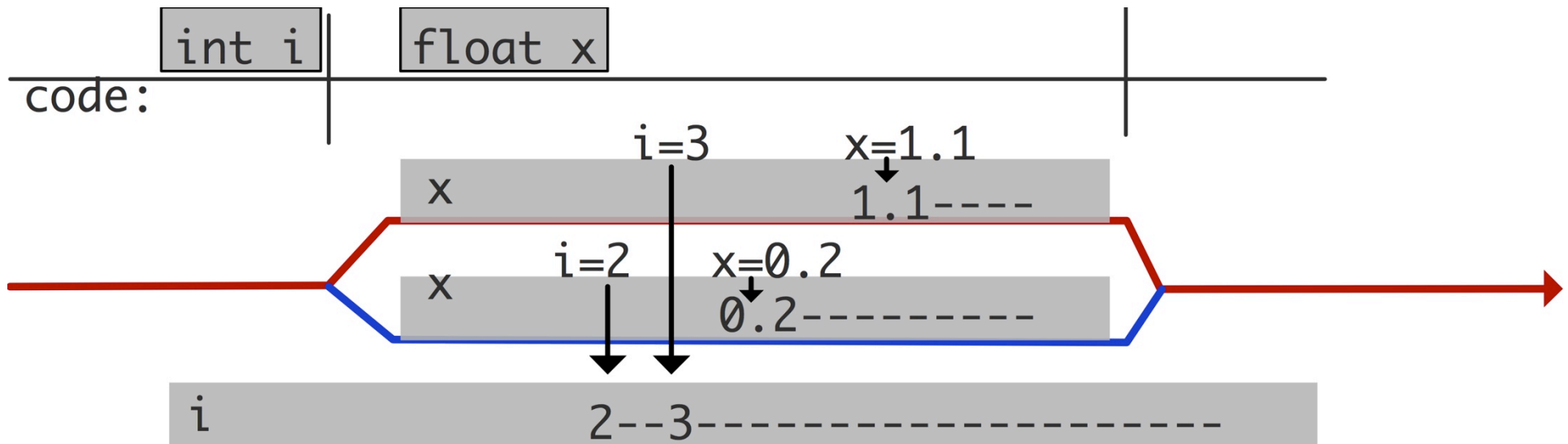
```
#pragma omp parallel
// single statement

#pragma omp parallel
{
    // block of statements
}

!$OMP parallel
    ! sequence of statements
!$OMP end parallel
```

# Private and shared data

- OpenMP threads are based on shared memory: when they are created they see everything from the initial thread
- Threads can also have private data



# Data in parallel regions

- C: local allocation is possible
- Fortran: make data private by using a clause
- (clauses can also be used in C)

```
int i;
#pragma omp parallel
{
    double x;
    i = ... // global assignment
    x = ... // local assignment
}

integer :: i
double :: x
!$OMP parallel private(x)
    i = ... ! global
    x = ... ! local
!$OMP end parallel
```