

## C++ PARALLEL PROGRAMMING

Function pointers  
`ret (*id) (p1, p2, ...)`

Thread `t1(f1, p1, p2)`

`threads`

`sync`

Thread `t1( [= ] { print(i); } )`

Lambda expression  
`[ c ] (p) -> ret {body}`

`auto lambda = [ c ] (p) -> ret {body}`

`// some other code`

`ret a = lambda (p)`

Atomic operations

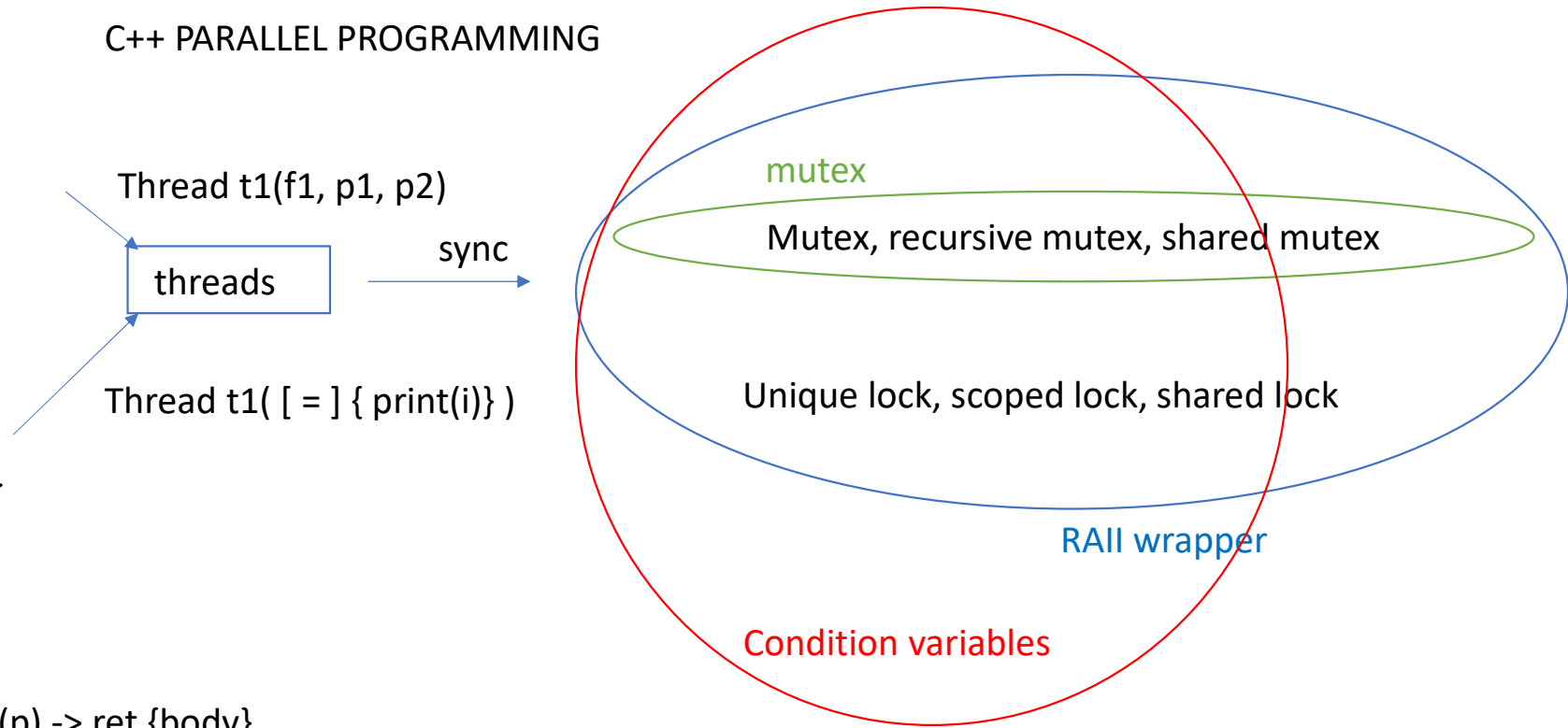
mutex

Mutex, recursive mutex, shared mutex

Unique lock, scoped lock, shared lock

RAII wrapper

Condition variables



## C++ TASK-BASED PARALLEL PROGRAMMING

