# Synchronization Constructs:

- Atomic Operations (fine-grain synchronization):

data-race.c 〉 No Selection

```c
#include <stdio.h>
#include <math.h>
#include <omp.h>

int main(){
    double result={0};

    #pragma omp parallel for shared(result)
    for(int i=0; i<1000000;i++) {
        result+=sin(i);
    }
    printf("%f",result);
}
```

```c
#include <stdio.h>
#include <math.h>
#include <omp.h>

int main(){
    double result={0};

    #pragma omp parallel for shared(result)
    for(int i=0; i<1000000;i++) {
        #pragma omp atomic
        result+=sin(i);
    }
    printf("%f",result);
}
```

```
    vmovsd  (%r12), %xmm1

L4:
    vxorpd  %xmm0, %xmm0, %xmm0
    vcvtsi2sd    %ebx, %xmm0, %xmm0
    vmovsd  %xmm1, 8(%rsp)
    addl    $1, %ebx
    call    _sin      ; return value in %xmm0
    vmovsd  8(%rsp), %xmm1  ; result in %xmm1
    cmpl    %ebx, %ebp
    vaddsd  %xmm0, %xmm1, %xmm1
    jne     L4

    vmovsd  %xmm1, (%r12)
```

```
L12:
    addl    $1, %ebx
    cmpl    %ebx, %r12d
    je      L9
L5:
    vxorpd  %xmm0, %xmm0, %xmm0
    vcvtsi2sd    %ebx, %xmm0, %xmm0
    call    _sin
    movq    0(%rbp), %rcx
    movq    (%rcx), %rdx
L4:
    vmovq   %rdx, %xmm2
    movq    %rdx, %rax
    vaddsd  %xmm2, %xmm0, %xmm1
    vmovq   %xmm1, %rsi
    lock cmpxchgq   %rsi, (%rcx)
    cmpq    %rax, %rdx
    je      L12
    movq    %rax, %rdx
    jmp     L4
```

Repeat until successful update