

restricted keywords in C++?

Amdahl's Law
diminishing return of resources

30 sequential 1 processor → 100 unit time

70 parallel p processor → $30 + \frac{70}{p}$

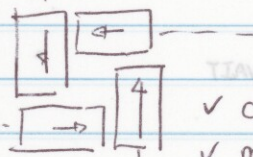
parallelism

- data → privatization
- instruction

DEADLOCK

11/12/2010

- mutual exclusion
- hold & wait
- no preemption [CANNOT VANISH]
- circular wait



- ✓ circular wait
- ✓ mutual exclusion
- ✓ no preemption
- ✓ hold & wait

LIVELOCK
doesn't do this

```
class Account {
    synchronized void f1(Account a) {
        a.f2();
    }
    synchronized void f2() {}
}
```

```
class MyThread extends Thread {
    Account a1;
    Account a2;
    public MyThread (Account x1, Account x2) {
        a1 = x1;
        a2 = x2;
    }
}
```

```
main() {
    Account a1 = new Account();
    Account a2 = new Account();
    MyThread t1 = new MyThread(a1, a2);
    MyThread t2 = new MyThread(a2, a1);
}
```

```
public void run() {
    a1.f1(a2);
}
```

```
t1.start();
t2.start();
<join>
```

No DEADLOCK IF t2 is removed (NO CIRCULAR WAIT)
MyThread(A1, A1) doesn't cause deadlock

SEE ↗