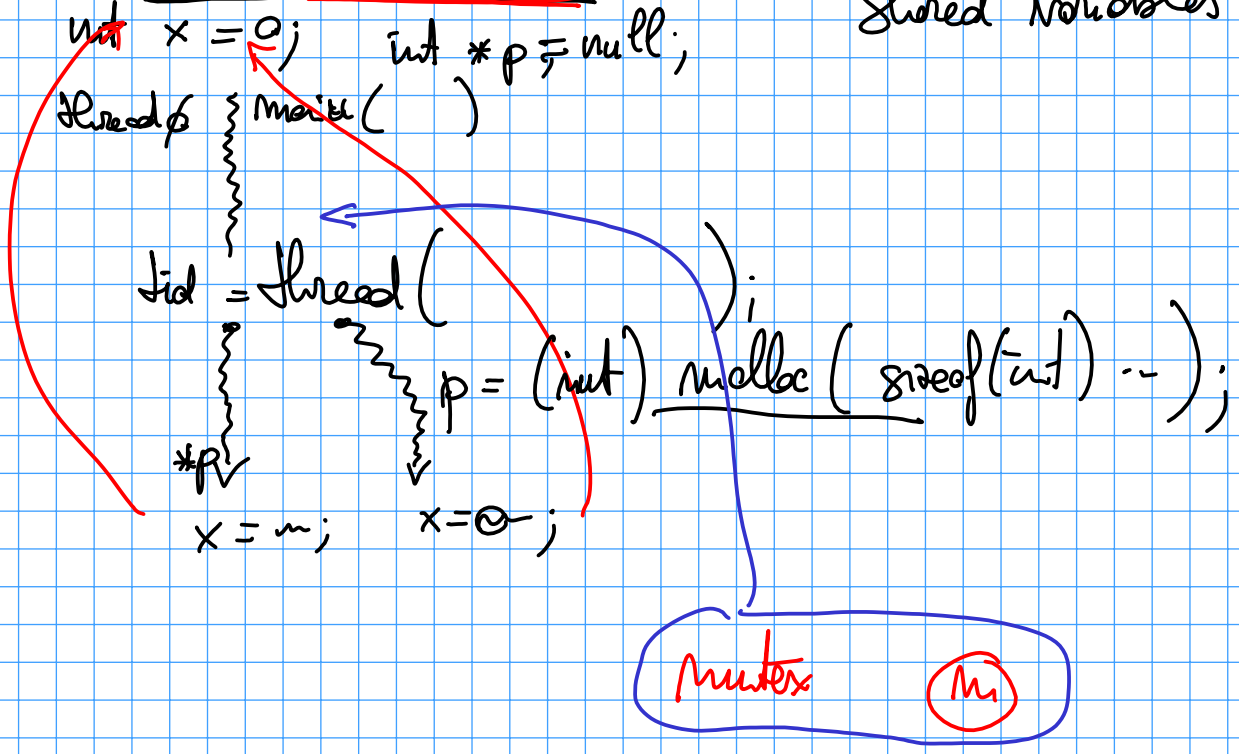
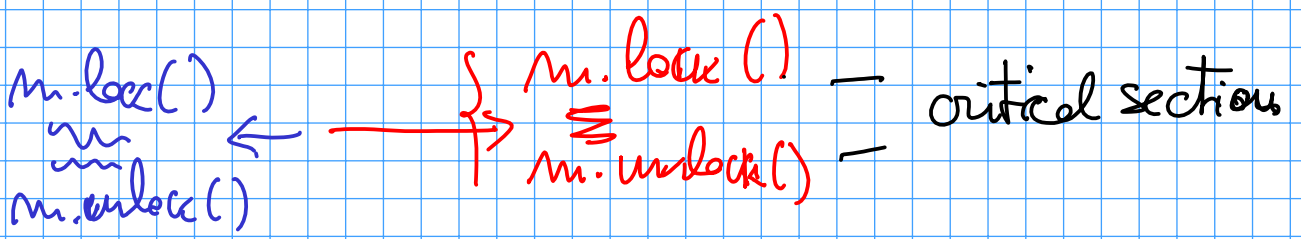
C++ 11 (14, 17) threads

↳ lock / mutexes / cond_var

shared variables

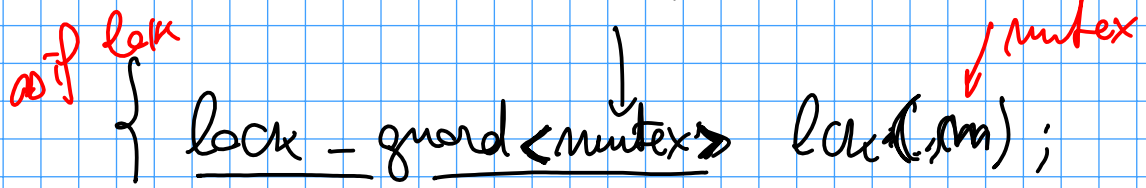


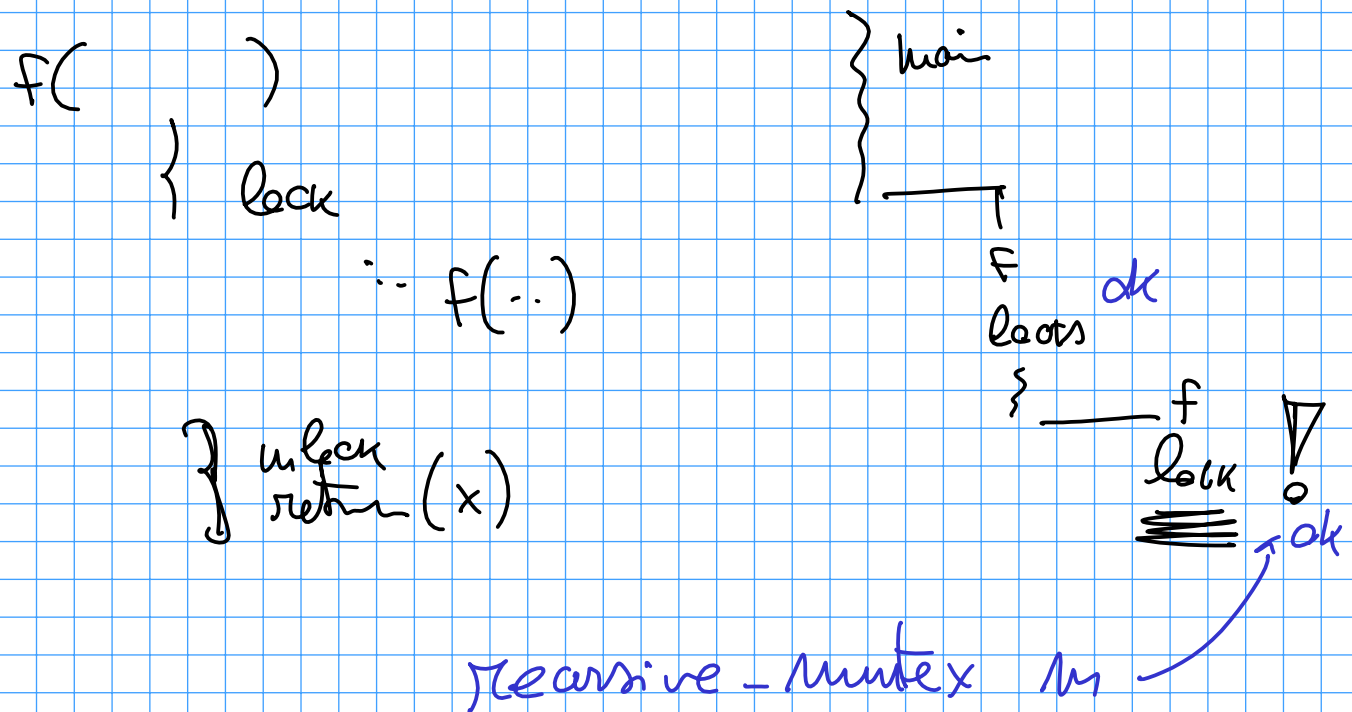


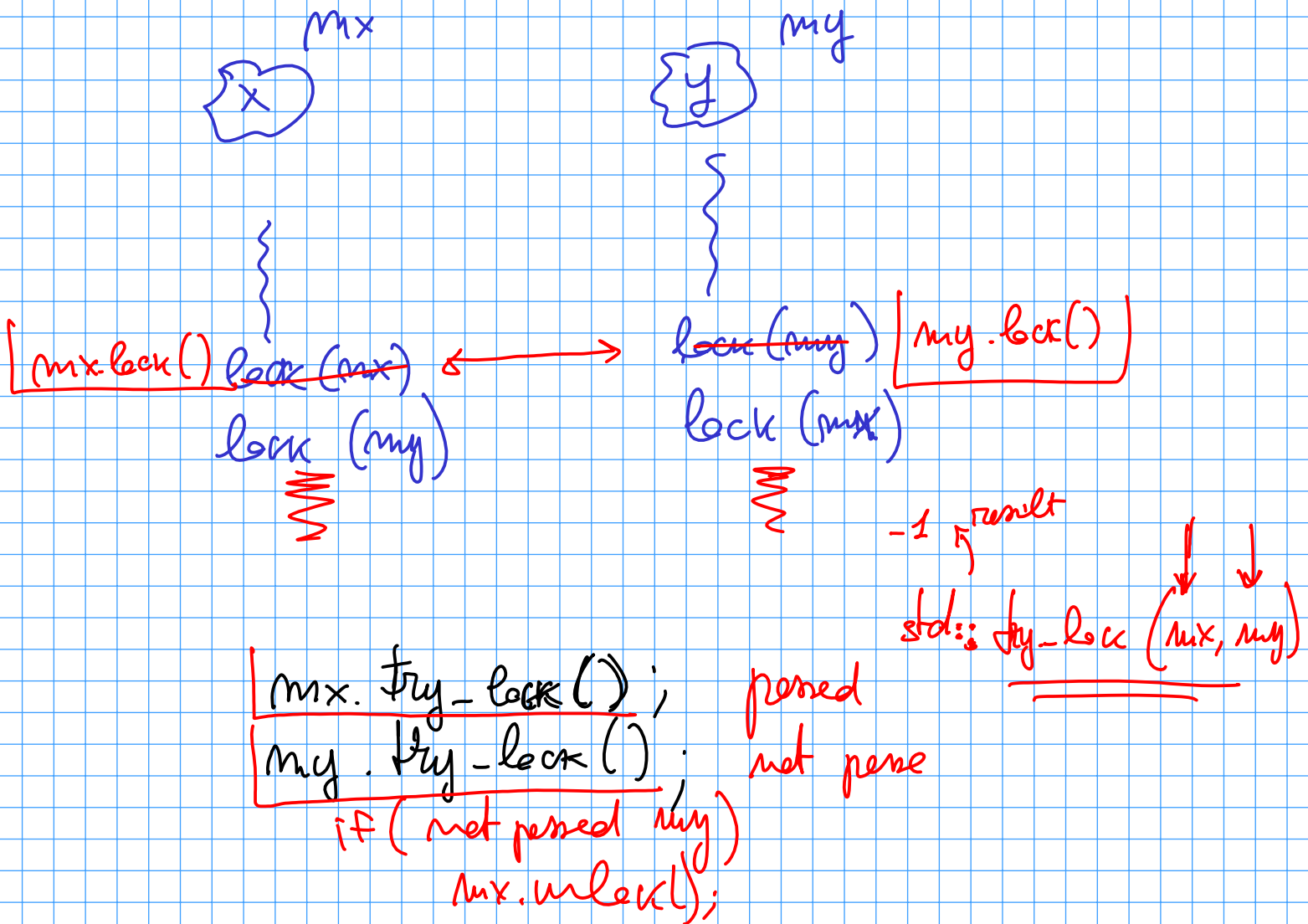
forget unlock

```
m.lock();
x++;
m.unlock();
```

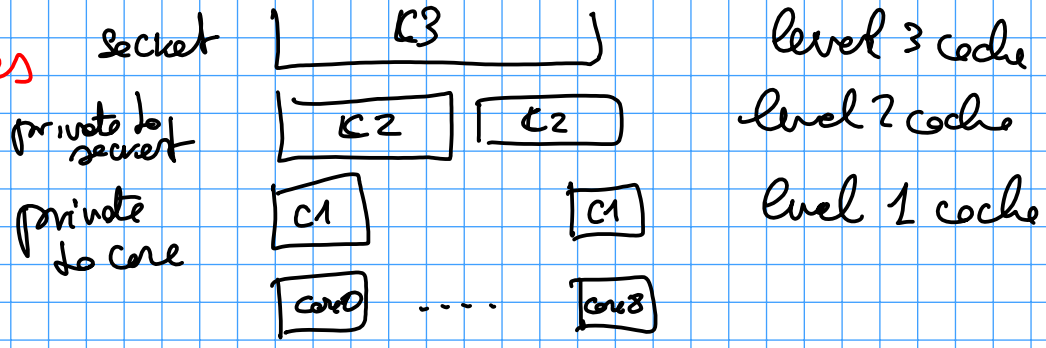
1k lines of C++ code




f( )
{ lock
... f(..)
} unlock
return (x)
main
f
locks
ok
f
lock
ok
recursive_mutex m

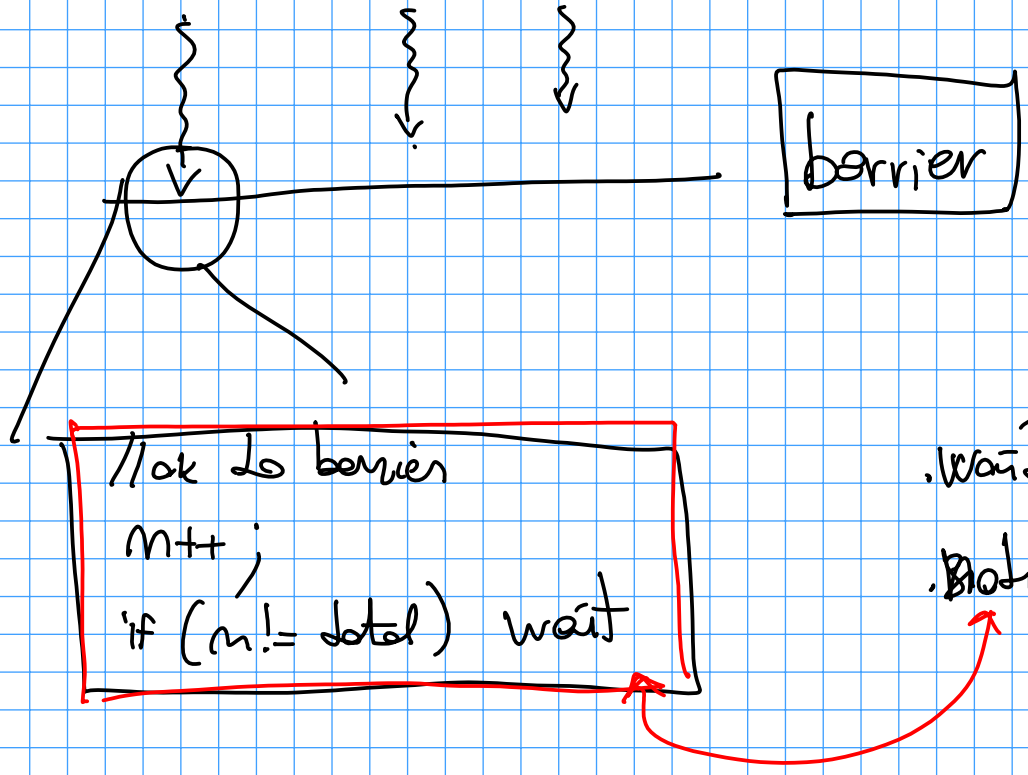

mx
x
my
y
mx.lock() lock(mx)
lock(my)
lock(my) my.lock()
lock(mx)
-1 result
std::try_lock(mx, my)
mx.try_lock();
my.try_lock();
passed
not pass
if(not passed my)
mx.unlock();

Atomic variables




```
std::atomic<int>  x;
```

```
x++;          x+= value;
```



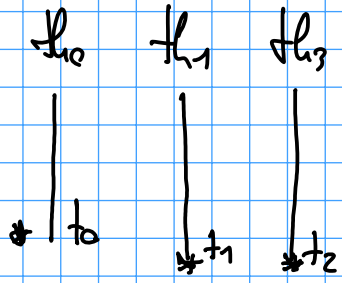
atomic variable

* //ok to barrier

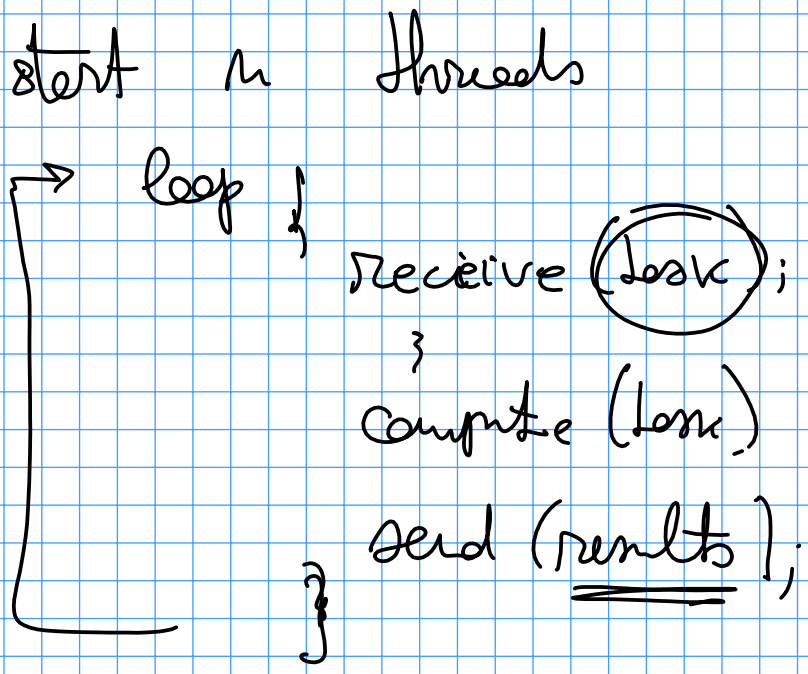
n++;

while(n != total); active wait

n++

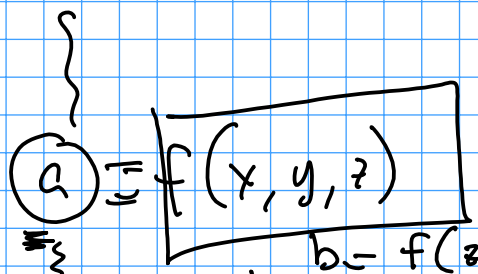


thread pool



asynchronous computations

std::async



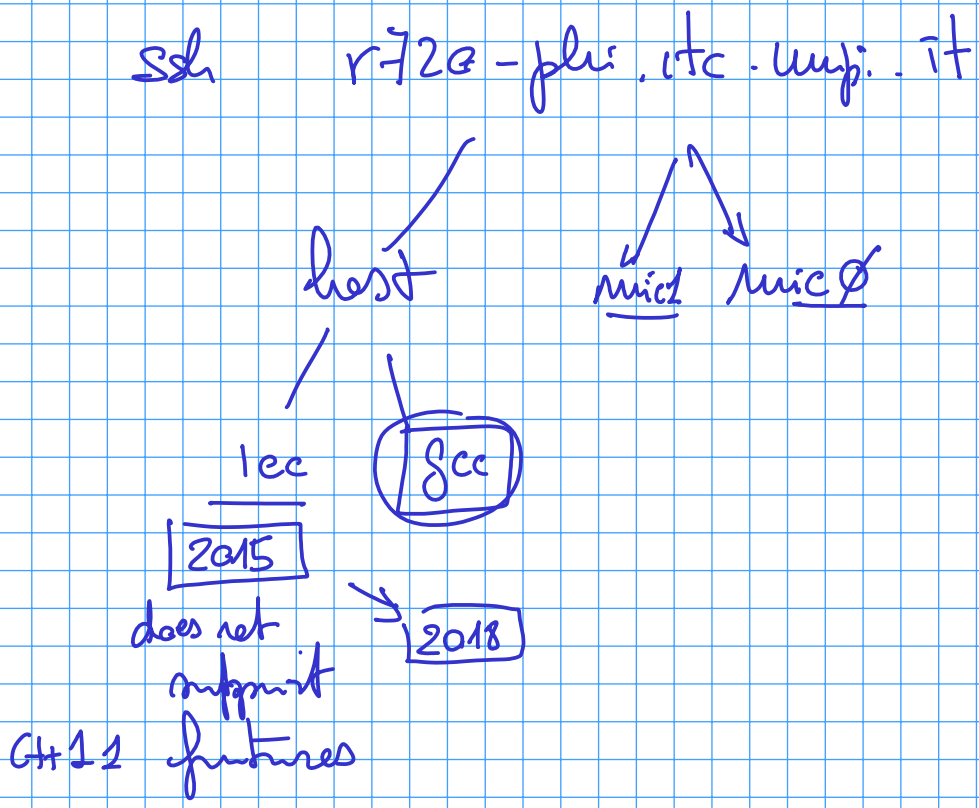

```
x = a + b;
```

```
std::future a = async(f, x, y, z);
b = async(f, z, t, w);
a.wait()
x = a.get() + b.get();
```

Cilk

```
a = spawn f(x, y, z);
b = spawn f(z, t, w);
sync
x = a + b;
```

- fcilkplus
  g++


ssh r720-plx.itc.unipi.it
host
mic1
mic0
icc
2015
does not support C++11 features
scc
2018