Can execute concurrently (or in parallel)?

Kernel functions / interrupt?
A parallel execution is control...
Atomicity ...

Note: Because updated z is written

\[ z = \sum (x y z) \]

while x ... is written

... & then
Hardware makes atomicity promises.

Every load and store is atomic
\[ z = x + y \]

Against \( z \) with respect to \( x, y \),
T = \frac{1}{2}

\text{locked}

\text{Leod} \iff \text{unlocked}

L = 0

\text{dynamic}

\text{static}

T = 1
use test set

must have

(test set) to do, but...

use information disorder

un processed
int test_and_set (int *lock_pointer);

.globl test_and_set

ret # return, old value in %eax

movl $1,%eax # get new_value into %eax

movl 4(%esp),%edx  # get lock_pointer into %edx

lock                            # next instruction is locked

xchgl %eax,(%edx)   # swap %eax with what is stored in (%edx)

# ... and don't let any other cpu touch that
# ... memory location while you're swapping

ret                               # return, old value in %eax
\( \text{Lx} \rightarrow 0 \rightarrow \text{locked} \rightarrow 1 \rightarrow \text{unlocked} \)

while (test & set (Lx)),

\[ \text{update} \rightarrow \text{atomic} \rightarrow \text{w.r.t. Lx} \rightarrow \text{create a lock} \rightarrow \text{lock} \]

\[ \text{Lx} = \text{0} \rightarrow \text{Lx} = \text{1} \rightarrow \text{Lx} = \text{Lx} \]

\[ \text{x} = \text{x} \rightarrow \text{x} = \text{x} \]
For a long time, Bob saw many ships.

Code used by user leave...
\[ (S_1 \cup S_2) \setminus \text{init} \]

Expression: \( \text{init} (S, \text{val}) \)

A triple representing a variable of a statement in \( S \)

Remark:
Cannot be checked.

Of course, the expression cannot be a value, as a glucose
(Dispose of waste in a safe and appropriate manner)
Let $V(s) \rightarrow$ increment value

if \( \text{val} \leq 0 \) then

Semaphore

get 1 process/thread

Value

from sem Q and

wake it up
$l_\text{even odd} = 3 \parallel \text{inlet}$
Check if block and wallet paying
(common paid)

Press
producer consumer
(bounded buffer)
reader writer
dining philosopher