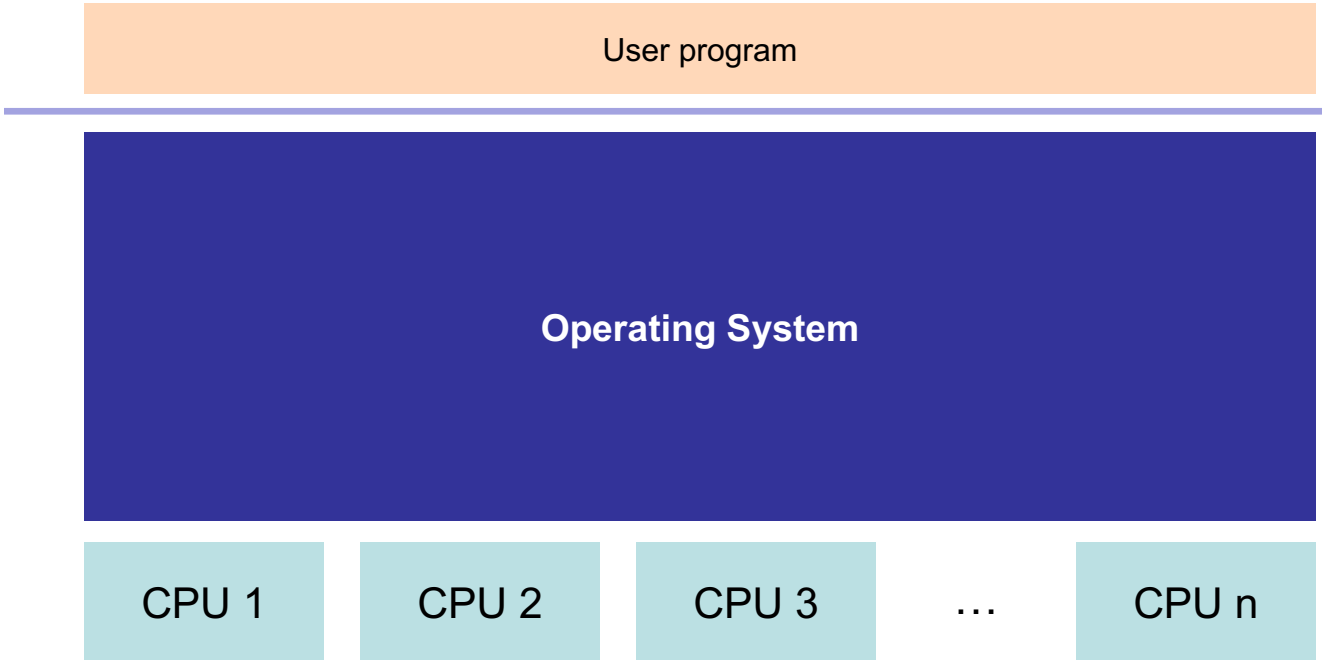
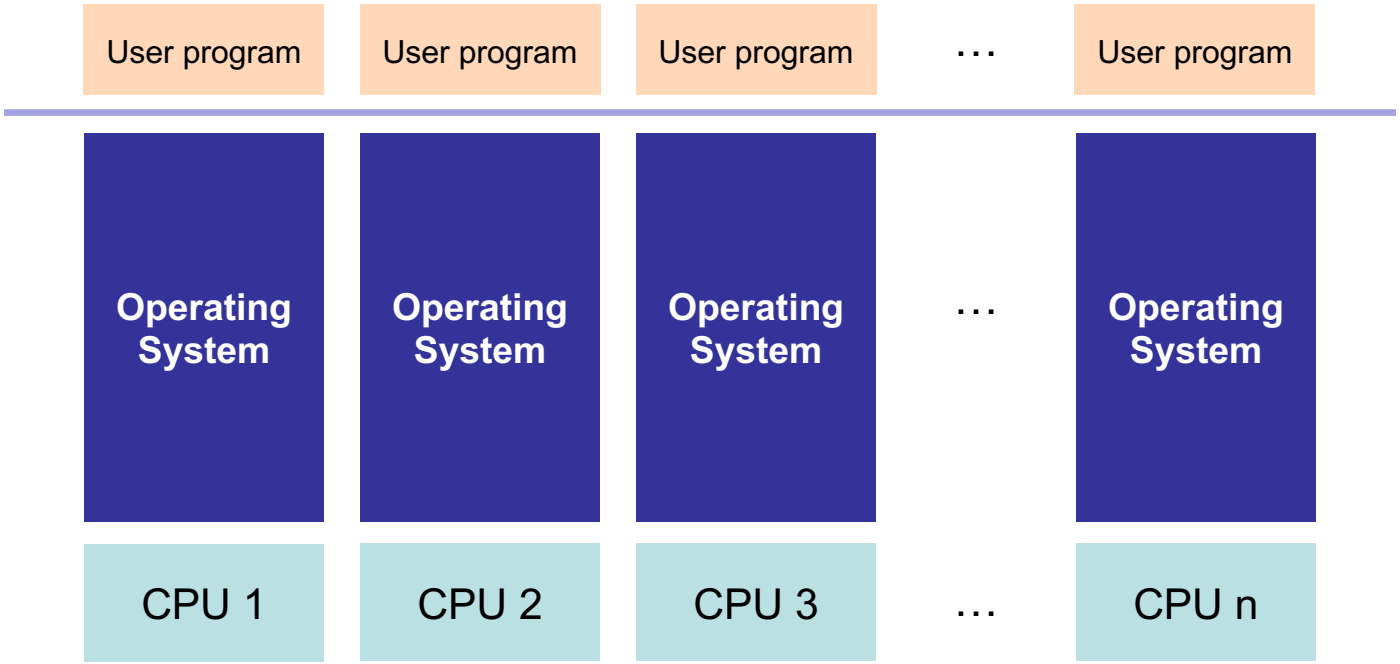
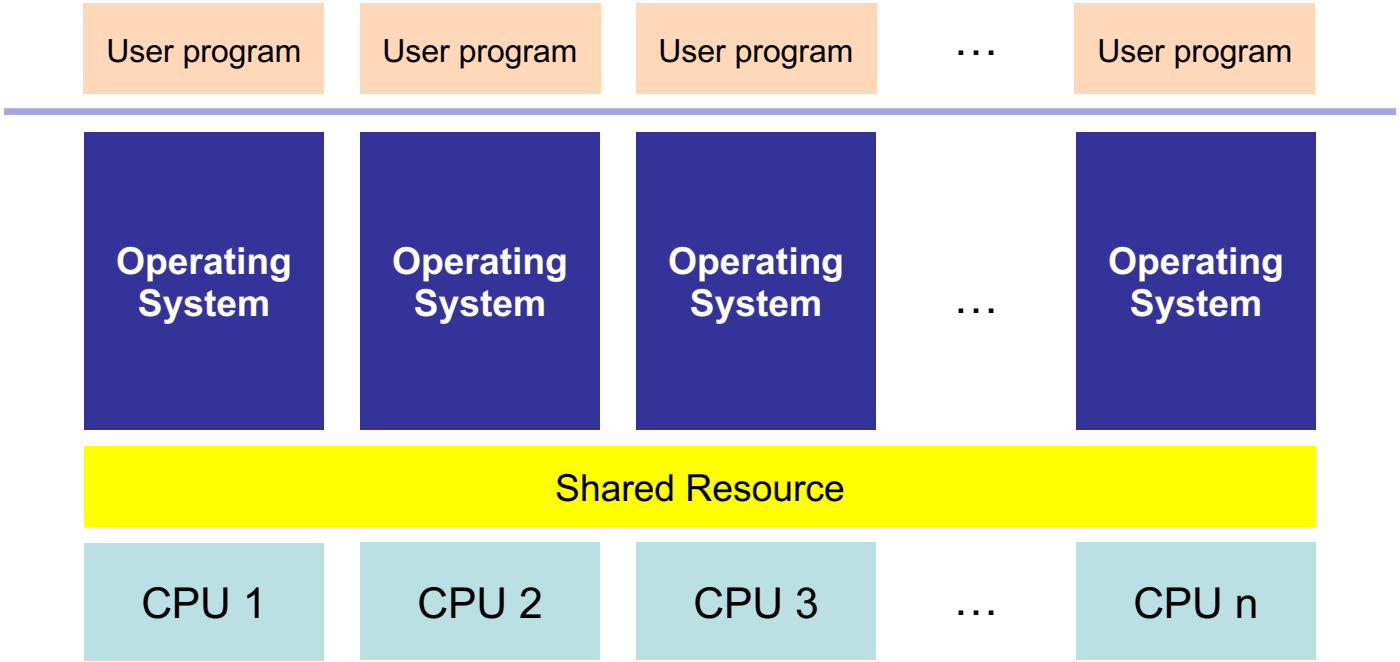
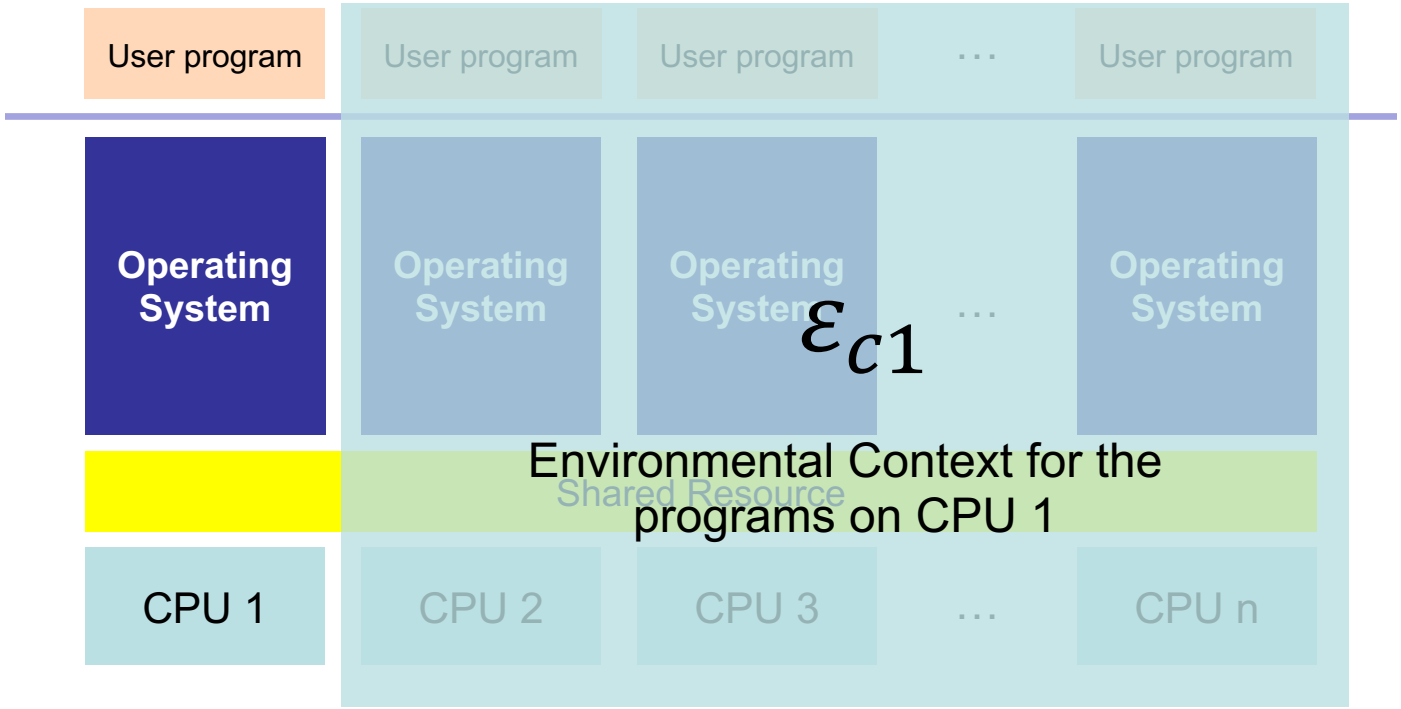
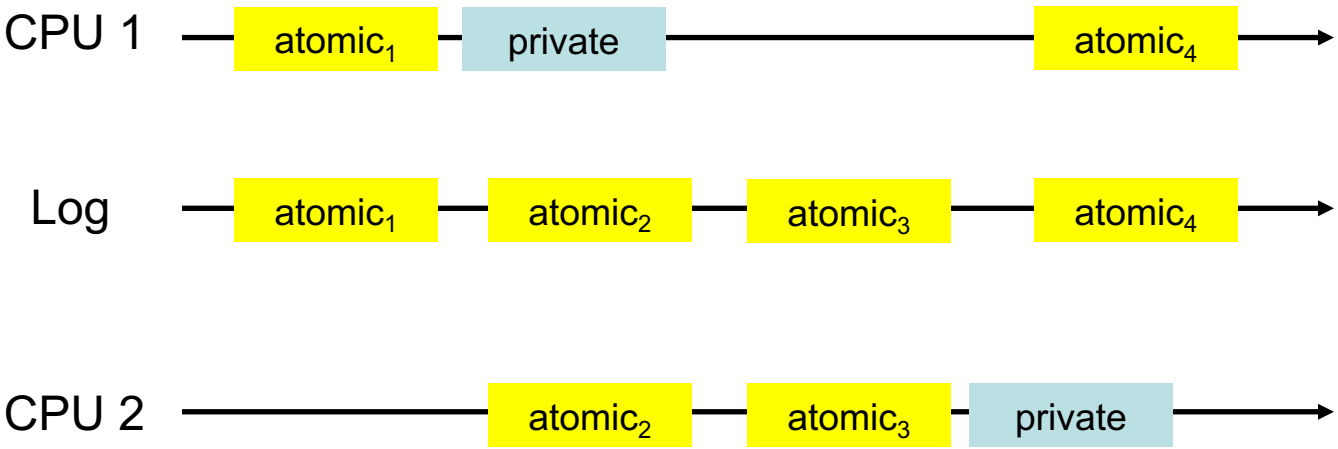
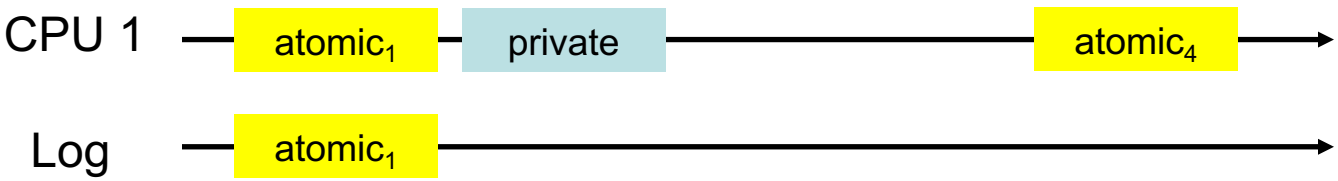

User program
Operating System
CPU 1
CPU 2
CPU 3
…
CPU n


User program
User program
User program
...
User program
Operating System
Operating System
Operating System
...
Operating System
CPU 1
CPU 2
CPU 3
...
CPU n

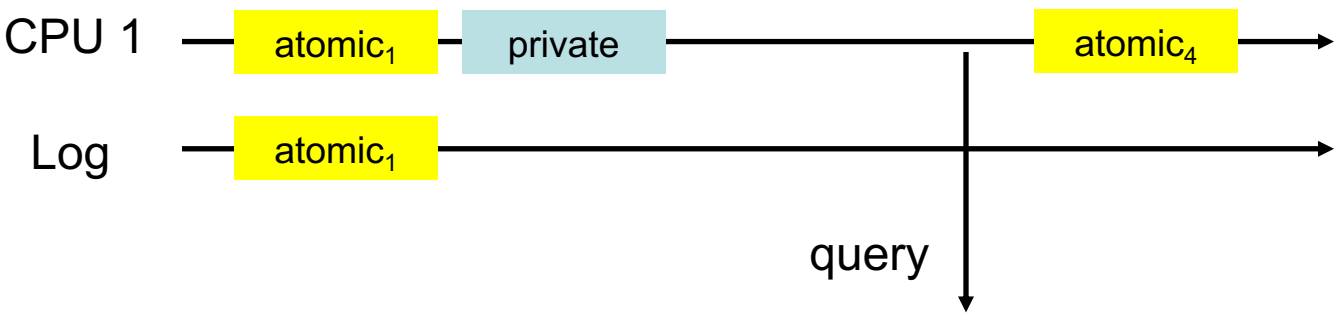

User program
User program
User program
…
User program
Operating System
Operating System
Operating System
…
Operating System
Shared Resource
CPU 1
CPU 2
CPU 3
…
CPU n

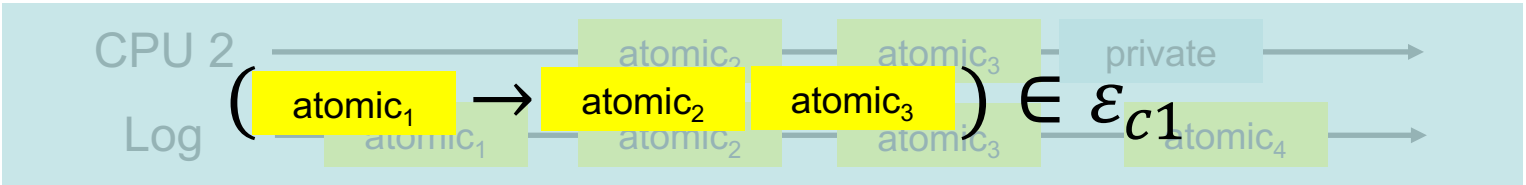

User program
User program
User program
…
User program
Operating System
Operating System
Operating System
…
Operating System
ε_{c1}
Shared Resource
Environmental Context for the programs on CPU 1
CPU 1
CPU 2
CPU 3
…
CPU n

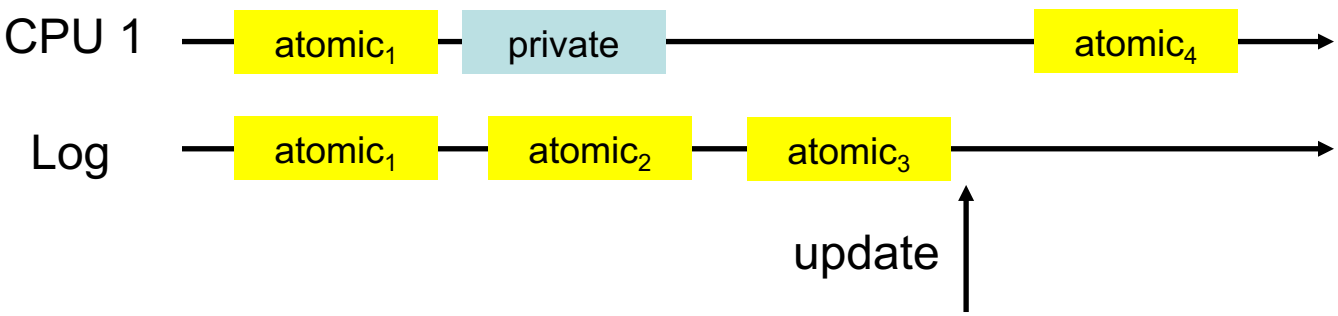

CPU 1
atomic1
private
atomic4
Log
atomic1
atomic2
atomic3
atomic4
CPU 2
atomic2
atomic3
private

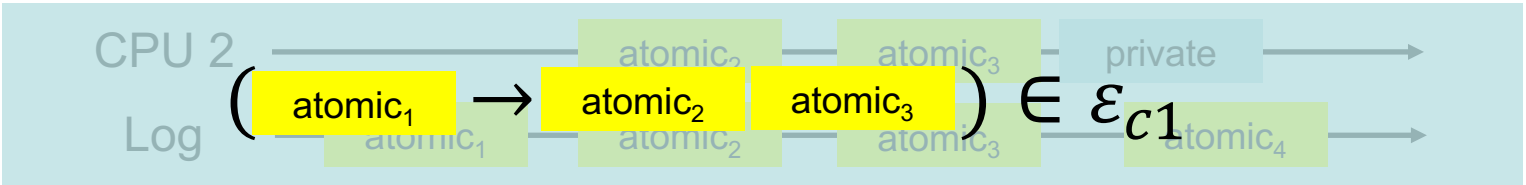

CPU 1
atomic1
private
atomic4
Log
atomic1

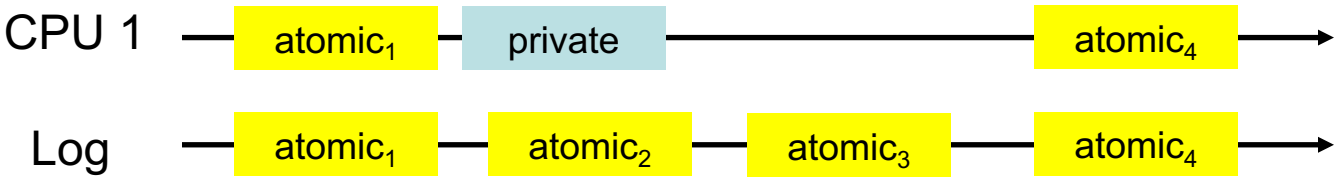

CPU 2
atomic2
atomic3
private
Log
atomic1
atomic2
atomic3
atomic4

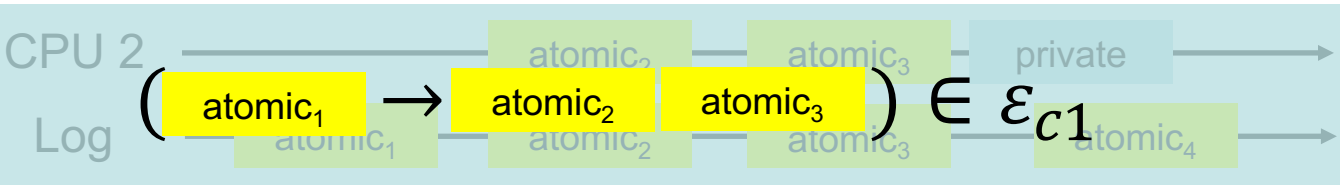

CPU 1
atomic1
private
atomic4
Log
atomic1
query

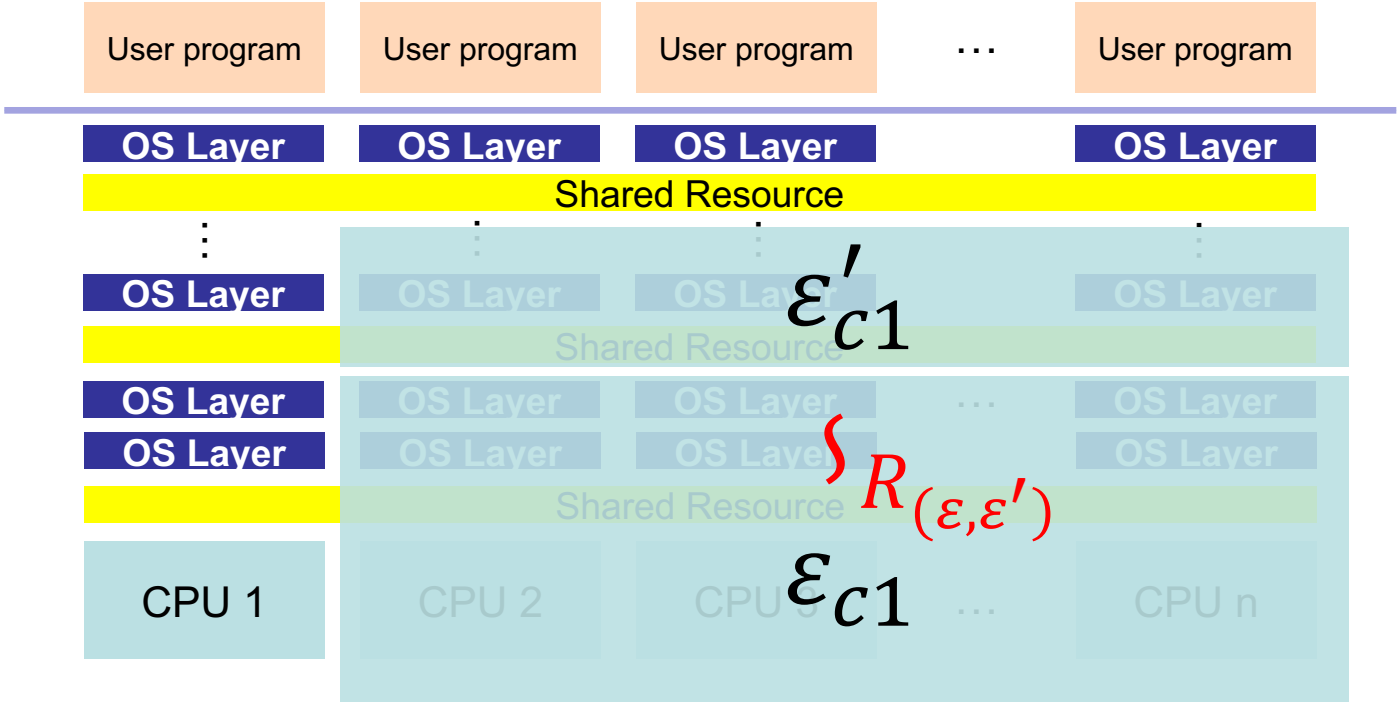

CPU 2
Log
atomic1
atomic2
atomic3
private
atomic4
(atomic1 → atomic2 atomic3) ∈ εc1

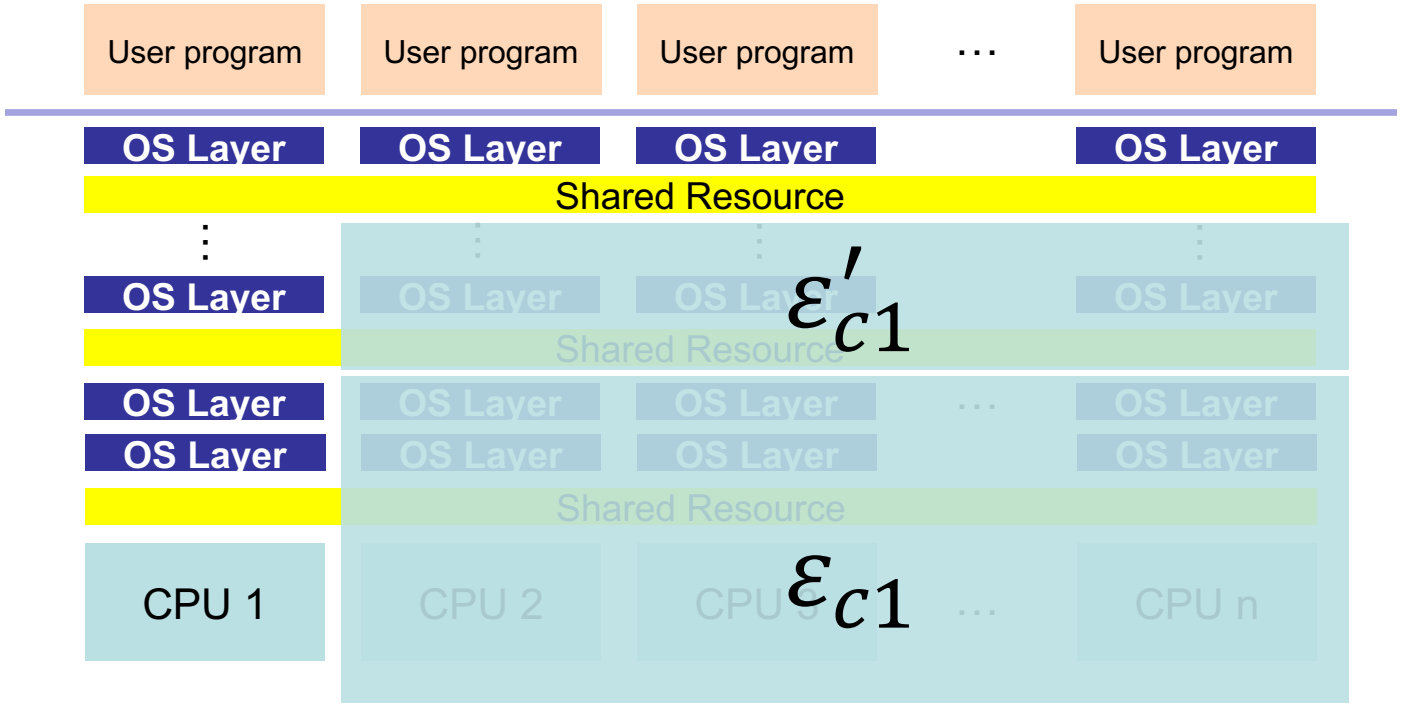

CPU 1
atomic1
private
atomic4
Log
atomic1
atomic2
atomic3
update


CPU 2
Log
atomic1
atomic2
atomic3
private
atomic4
(atomic1 → atomic2 atomic3) ∈ ε_{c1}


CPU 1
atomic1
private
atomic4
Log
atomic1
atomic2
atomic3
atomic4


CPU 2
Log
atomic1
atomic2
atomic3
private
atomic4
(atomic1 → atomic2 atomic3) ∈ ε_{c1}

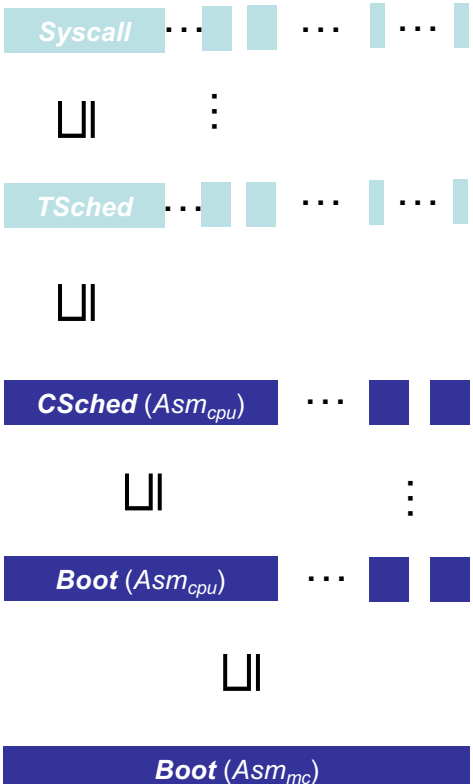

User program
User program
User program
…
User program
OS Layer
OS Layer
OS Layer
OS Layer
Shared Resource
OS Layer
OS Layer
OS Layer
OS Layer
Shared Resource
ε'_{c1}
OS Layer
OS Layer
OS Layer
OS Layer
OS Layer
OS Layer
OS Layer
OS Layer
…
OS Layer
Shared Resource
$\varsigma_{R_{(\varepsilon,\varepsilon')}}$
CPU 1
CPU 2
CPU n
ε_{c1}


User program
User program
User program
···
User program
OS Layer
OS Layer
OS Layer
OS Layer
Shared Resource
OS Layer
OS Layer
OS Layer
OS Layer
OS Layer
OS Layer
Shared Resource
CPU 1
CPU 2
CPU n
ε'_{c1}
ε_{c1}

$$Asm_{thrd}(Syscall[tid, \varepsilon'_{cpu}, \varepsilon'_{thrd}]) \vdash [\![\mathbf{Ctxt}]\!]$$

$$\sqcup\!|$$

$$Asm_{mc}(Boot) \vdash [\![\mathbf{CertiKOS} \oplus \mathbf{Ctxt}]\!]$$



$$Asm_{thrd}(Syscall[tid, \varepsilon'_{cpu}, \varepsilon'_{thrd}]) \vdash [\![\mathbf{Ctxt}]\!]$$

$$\sqcup\!|$$

$$Asm_{thrd}(TSched[tid, \varepsilon'_{cpu}, \varepsilon_{thrd}]) \vdash [\![\mathbf{CertiKOS_{td}} \oplus \mathbf{Ctxt}]\!]$$

$$\sqcup\!|$$

$$Asm_{cpu}(CSched[cid, \varepsilon'_{cpu}]) \vdash [\![\mathbf{CertiKOS_{td}} \oplus \mathbf{Ctxt}]\!]$$

$$\sqcup\!|$$

$$Asm_{cpu}(Boot[cid, \varepsilon_{cpu}]) \vdash [\![\mathbf{CertiKOS} \oplus \mathbf{Ctxt}]\!]$$

$$\sqcup\!|$$

$$Asm_{mc}(Boot) \vdash [\![\mathbf{CertiKOS} \oplus \mathbf{Ctxt}]\!]$$

(where $\mathbf{CertiKOS} \coloneqq \mathbf{CertiKOS_{cpu}} \oplus \mathbf{CertiKOS_{td}}$)

$$Asm_{cpu}(Boot[cid, \varepsilon_{cpu}]) \vdash [\![\textbf{CertiKOS} \oplus \textbf{Ctxt}]\!]$$

⨆

$$Asm_{mc}(Boot) \vdash [\![\textbf{CertiKOS} \oplus \textbf{Ctxt}]\!]$$

- Environment
  - Fixed number of CPUs
  - Fixed initial state for all CPUs
  - Fairness assumptions
- Things to solve
  - Hide non-determinism ①
  - Build environmental context for each CPU ②
  - Prove compositionality of multiple per-CPU machines ③
  - Provide simple environmental context for per-CPU machines ④

Link with Asm_{cpu} ④

④

Optimize
environmental context

Introduce per-CPU machine ②

Introduce partial machine and prove linking theorem ②③

Introduce hardware scheduler ①

$$Asm_{cpu}(Boot[cid, \varepsilon_{cpu}]) \vdash [\![\textbf{CertiKOS} \oplus \textbf{Ctxt}]\!]$$

$$\sqsupseteq$$

$$Asm_{sep}(Boot[cid, \varepsilon_{sep}]) \vdash [\![\textbf{CertiKOS} \oplus \textbf{Ctxt}]\!]$$

$$\sqsupseteq$$

$$Asm_{reorder}(Boot[cid, \varepsilon'_{reorder}]) \vdash [\![\textbf{CertiKOS} \oplus \textbf{Ctxt}]\!]$$

$$\sqsupseteq$$

$$Asm_{reorder}(Boot[cid, \varepsilon_{reorder}]) \vdash [\![\textbf{CertiKOS} \oplus \textbf{Ctxt}]\!]$$

$$\sqsupseteq$$

$$Asm_{split}(Boot[cid, \varepsilon]) \vdash [\![\textbf{CertiKOS} \oplus \textbf{Ctxt}]\!]$$

$$\sqsupseteq$$

$$Asm_{big2}(Boot[cid, \varepsilon]) \vdash [\![\textbf{CertiKOS} \oplus \textbf{Ctxt}]\!]$$

$$\sqsupseteq$$

$$Asm_{big}(Boot[cid, \varepsilon]) \vdash [\![\textbf{CertiKOS} \oplus \textbf{Ctxt}]\!]$$

$$\sqsupseteq$$

$$Asm_{single}(Boot[cid, \varepsilon]) \vdash [\![\textbf{CertiKOS} \oplus \textbf{Ctxt}]\!]$$

$$\sqsupseteq$$

$$Asm_{env}(Boot[cid, \varepsilon]) \vdash [\![\textbf{CertiKOS} \oplus \textbf{Ctxt}]\!]$$

$$\sqsupseteq$$

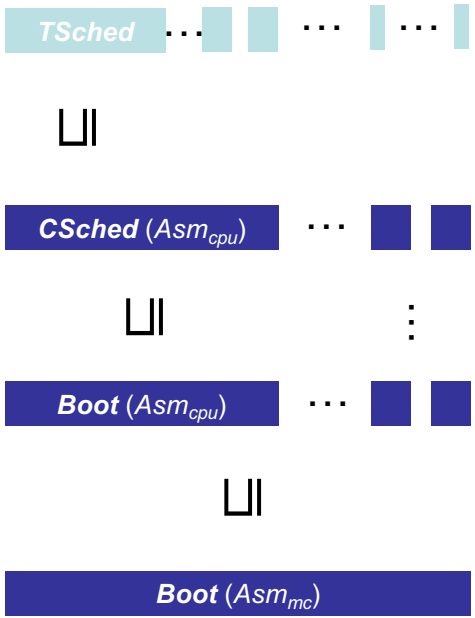
$$Asm_{env}(\|_{i \in CoreSet}\ Boot[CoreSet, \varepsilon_{CoreSet}]) \vdash [\![\textbf{CertiKOS} \oplus \textbf{Ctxt}]\!]$$

$$\sqsupseteq$$

$$Asm_{oracle}(Boot[\varepsilon_{CoreSet}]) \vdash [\![\textbf{CertiKOS} \oplus \textbf{Ctxt}]\!]$$

$$\sqsupseteq$$

$$Asm_{mc}(Boot) \vdash [\![\textbf{CertiKOS} \oplus \textbf{Ctxt}]\!]$$



$$Asm_{thrd}(TSched[tid, \varepsilon'_{cpu}, \varepsilon_{thrd}]) \vdash [\![\mathbf{CertiKOS_{td}} \oplus \mathbf{Ctxt}]\!]$$

$$\sqcup\!|$$

$$Asm_{cpu}(CSched[cid, \varepsilon'_{cpu}]) \vdash [\![\mathbf{CertiKOS_{td}} \oplus \mathbf{Ctxt}]\!]$$

$$Asm_{cpu}(Boot[cid, \varepsilon_{ccpu}]) \vdash [\![\mathbf{CertiKOS} \oplus \mathbf{Ctxt}]\!]$$

$$\sqcup\!|$$

$$Asm_{mc}(Boot) \vdash [\![\mathbf{CertiKOS} \oplus \mathbf{Ctxt}]\!]$$

(where $\mathbf{CertiKOS} \coloneqq \mathbf{CertiKOS_{cpu}} \oplus \mathbf{CertiKOS_{td}}$)

$$TSched[tid, \varepsilon'_{cpu}, \varepsilon_{thrd}](yield) \; - \; (lst, log) \rightarrow (lst, log')$$

$$Asm_{thrd}(TSched[tid, \varepsilon'_{cpu}, \varepsilon_{thrd}]) \vdash [\![\mathbf{CertiKOS_{td}} \oplus \mathbf{Ctxt}]\!]$$

$$Asm_{cpu}(CSched[cid, \varepsilon'_{cpu}]) \vdash [\![\mathbf{CertiKOS_{td}} \oplus \mathbf{Ctxt}]\!]$$

$CSched[cid, \varepsilon'_{ci}]$ contains software scheduler primitives

- spawn / yield / sleep / wakeup

$$CSched[cid, \varepsilon'_{ci}](yield) \; - \; (lst, log) \rightarrow (lst/[tid = \cdots, \rho = \cdots, \cdots], log')$$

- Environment
  - Arbitrary active or available thread set on the CPU
  - Dynamic initial states
- Things to solve
  - Hide context switching between threads ①
  - Build environmental context for each thread ②
  - Assign proper initial states for each thread ③
  - Prove compositionality of multiple per-thread machines ④
  - Use the same compiler for per-CPU/thread machines ⑤

Link per-CPU machine compiler with per-thread machine (5)

Introduce per-thread machine (1)(2)(3)

Introduce multithreaded machine and prove linking theorem (1)(2)(3)(4)

$$Asm_{thrd}(TSched[tid, \varepsilon'_{cpu}, \varepsilon_{thrd}]) \vdash [\![\mathbf{CertiKOS_{td}} \oplus \mathbf{Ctxt}]\!]$$

$$\sqcup\!|$$

$$IAsm_{thrd}(TLink[tid, \varepsilon'_{cpu}, \varepsilon_T^{zip}]) \vdash [\![\mathbf{CertiKOS_{td}} \oplus \mathbf{Ctxt}]\!]$$

$$\sqcup\!|$$

$$IAsm_{mt}(TLink[tid, \varepsilon'_{cpu}, \varepsilon_T]) \vdash [\![\mathbf{CertiKOS_{td}} \oplus \mathbf{Ctxt}]\!]$$

$$\sqcup\!|$$

$$IAsm_{mt}(\|_{ti \in TSet}\ TLink[cid, \varepsilon'_{cpu}]) \vdash [\![\mathbf{CertiKOS_{td}} \oplus \mathbf{Ctxt}]\!]$$

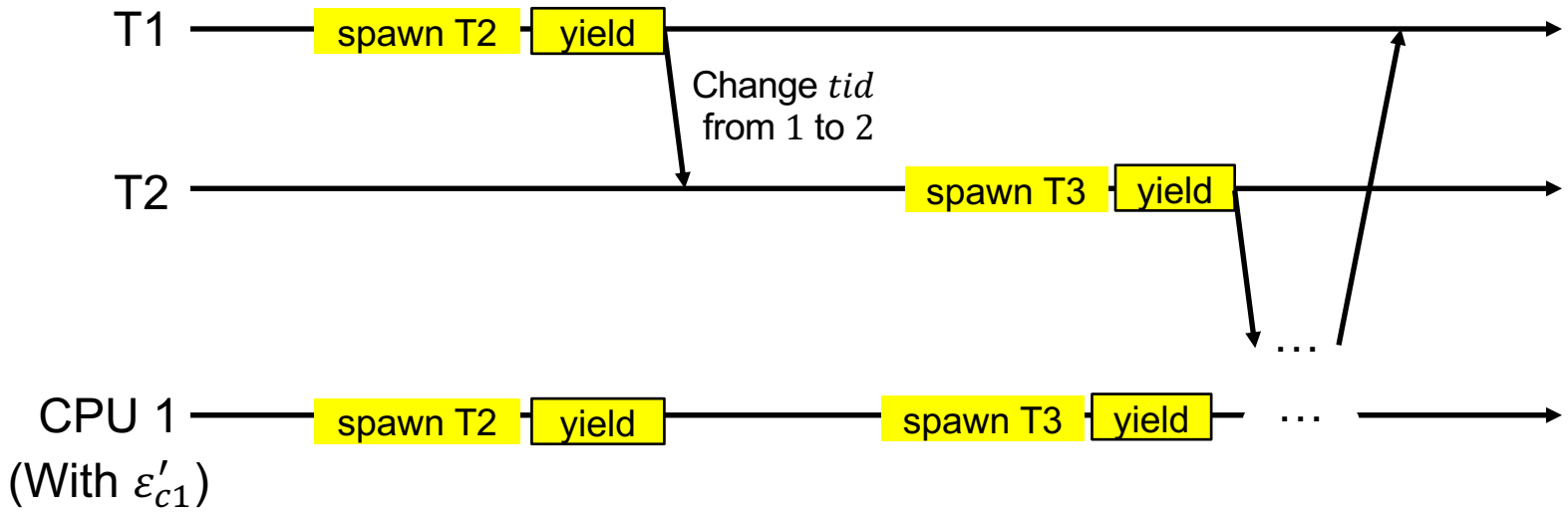
$$\sqcup\!|$$

$$Asm_{mt}(\|_{ti \in TSet}\ TLink[cid, \varepsilon'_{cpu}]) \vdash [\![\mathbf{CertiKOS_{td}} \oplus \mathbf{Ctxt}]\!]$$

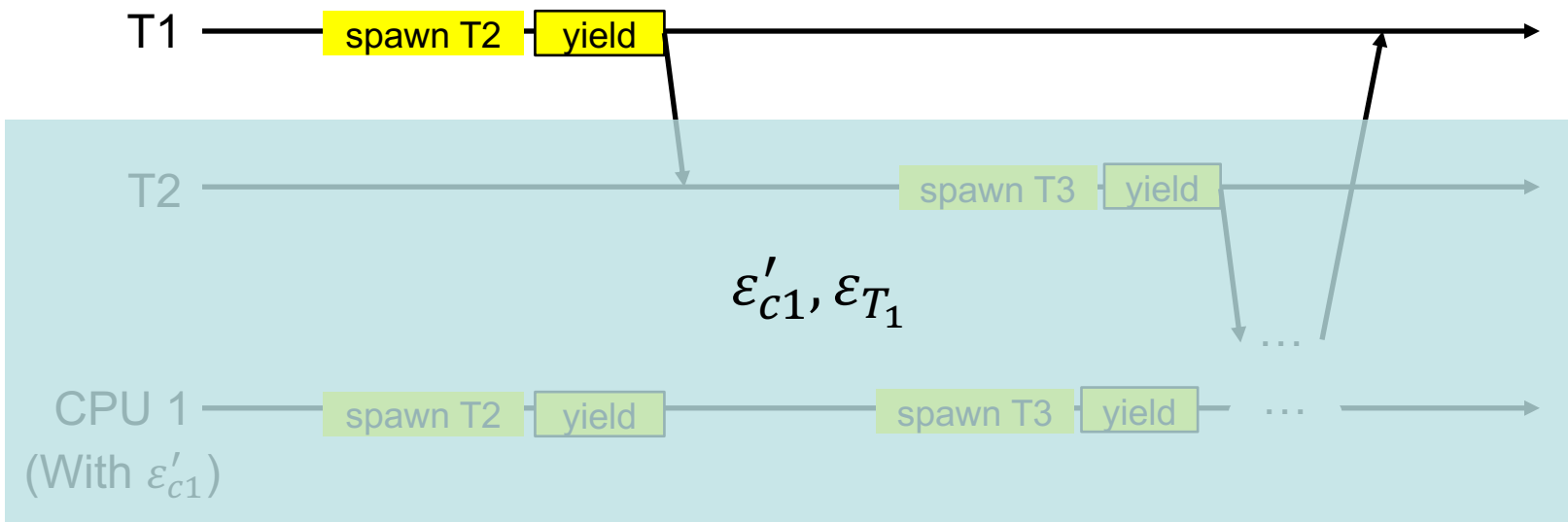
$$\sqcup\!|$$

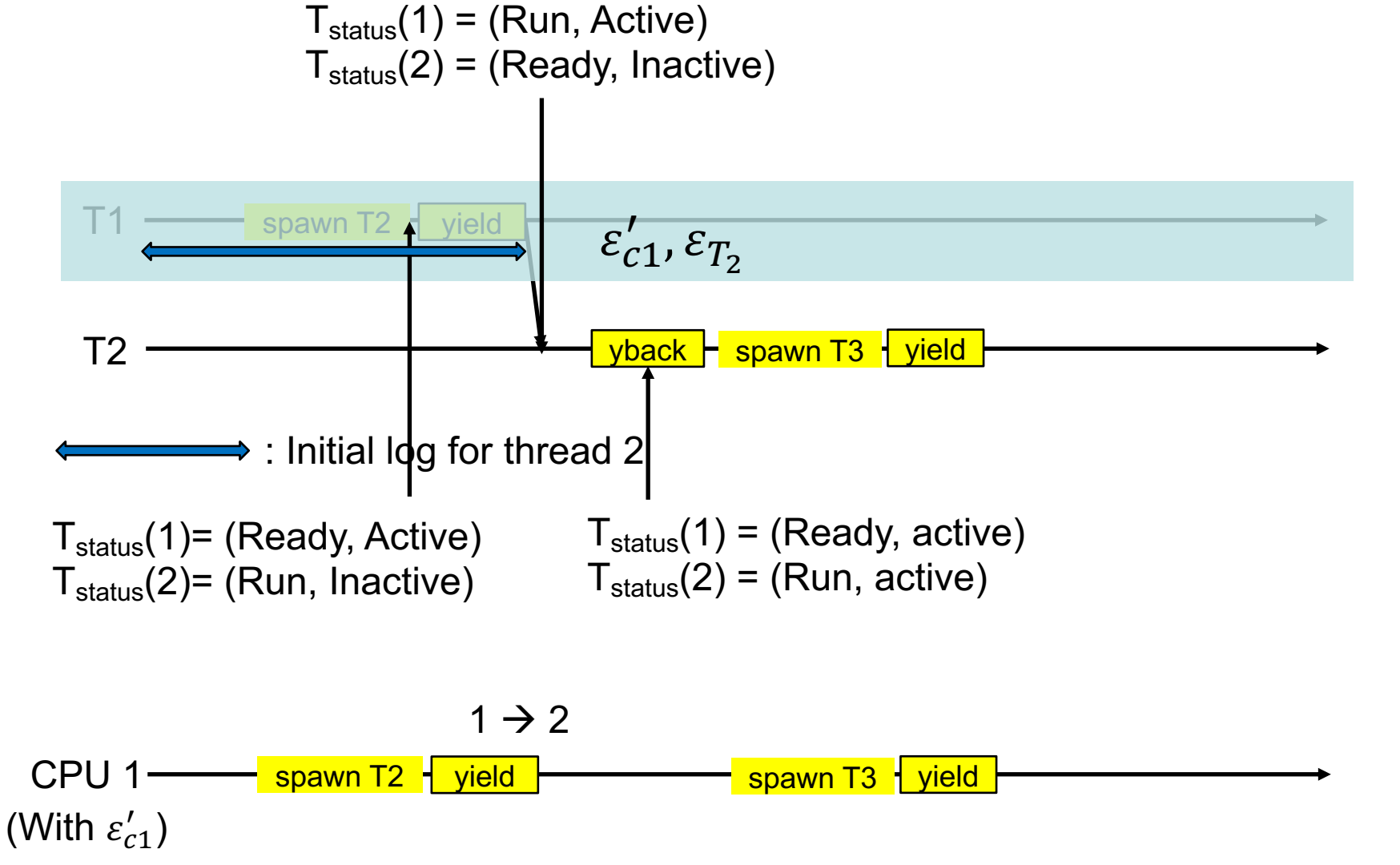
$$Asm_{cpu}(CSched[cid, \varepsilon'_{cpu}]) \vdash [\![\mathbf{CertiKOS_{td}} \oplus \mathbf{Ctxt}]\!]$$

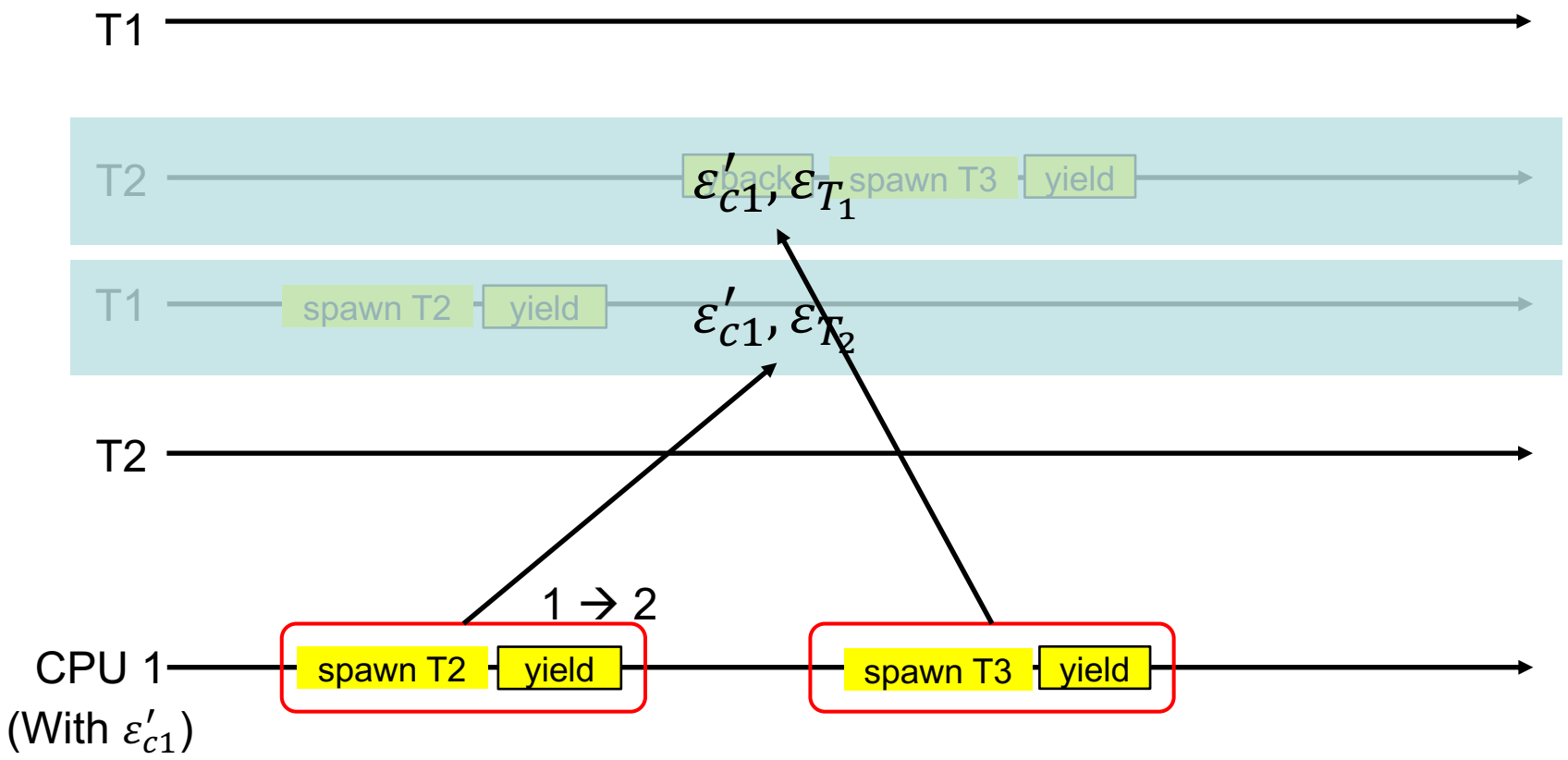
$st_{TSet} \coloneqq (tid, \{ti \mapsto lst_{ti}\}, log)\ (\forall ti, ti \in TSet)$



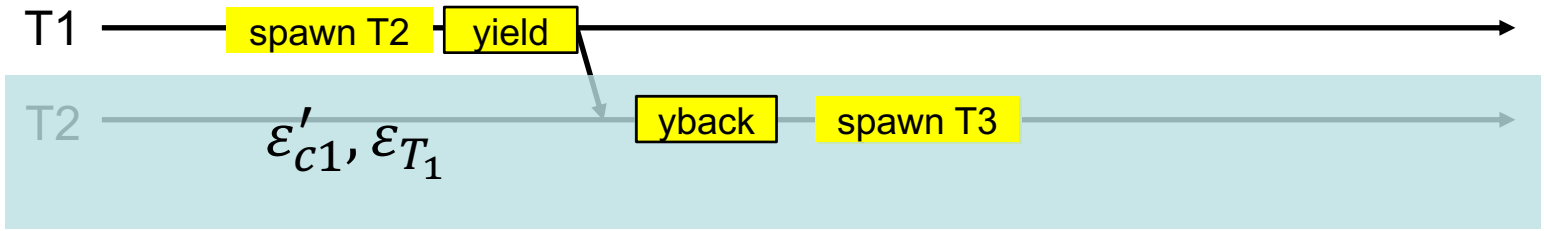
$st_{TSet} \coloneqq (tid, \{ti \mapsto lst_{ti}\}, log)\ (\forall ti, ti \in TSet)$




$T_{status}(1)$ = (Run, Active)
$T_{status}(2)$ = (Ready, Inactive)
T1
spawn T2
yield
$\varepsilon'_{c1}, \varepsilon_{T_2}$
T2
yback
spawn T3
yield
: Initial log for thread 2
$T_{status}(1)$= (Ready, Active)
$T_{status}(2)$= (Run, Inactive)
$T_{status}(1)$ = (Ready, active)
$T_{status}(2)$ = (Run, active)
1 → 2
CPU 1
(With ε'_{c1})
spawn T2
yield
spawn T3
yield


T1
T2
spawn T3
yield
$\varepsilon'_{c1}, \varepsilon_{T_1}$
T1
spawn T2
yield
$\varepsilon'_{c1}, \varepsilon'_{T_2}$
T2
1 → 2
CPU 1
(With ε'_{c1})
spawn T2
yield
spawn T3
yield

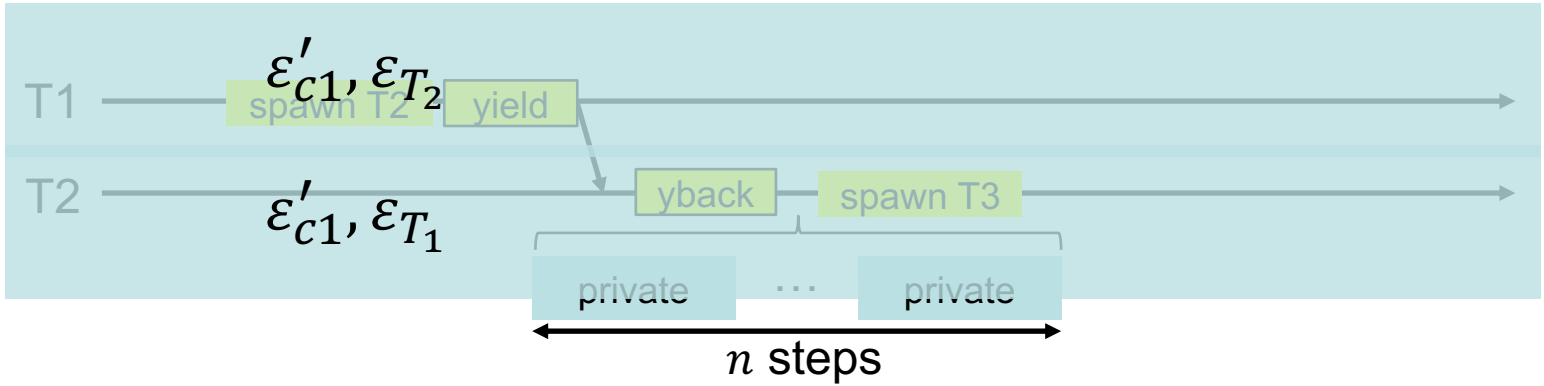
$IAsm_{mt}$ with T1



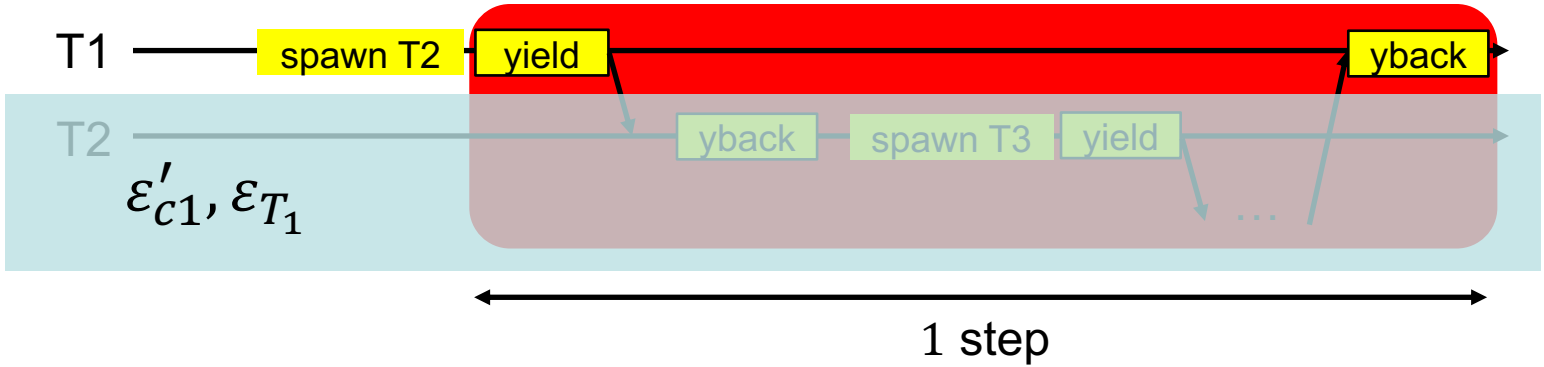
$n < progress$

Every thread will generate at least one event within $progress$ steps

$IAsm_{mt}$ with $TSet$



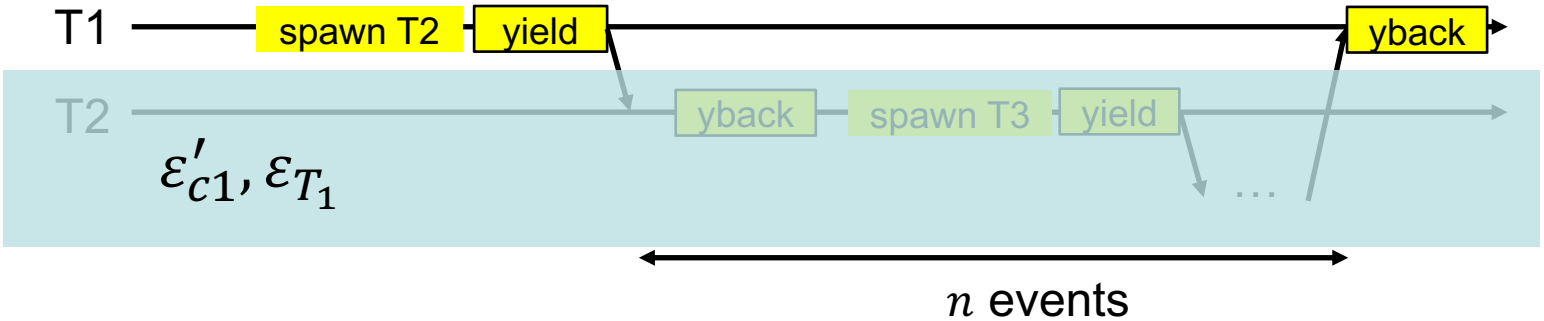
$IAsm_{thrd}$ with T1



$n \leq limit$

Every thread will be eventually scheduled within $limit \times progress$ steps

$IAsm_{mt}$ with T1



Initial state: Calculate initial log to find the proper initial state

Yield rule: $TSched\left[tid, \varepsilon'_{cpu}, \varepsilon_{thrd}\right](yield) \; - \; (lst, log) \rightarrow (lst, log')$

$Asm_{thrd}(TSched\left[tid, \varepsilon'_{cpu}, \varepsilon_{thrd}\right]) \vdash [\![\mathbf{CertiKOS_{td}} \oplus \mathbf{Ctxt}]\!]$

⨆

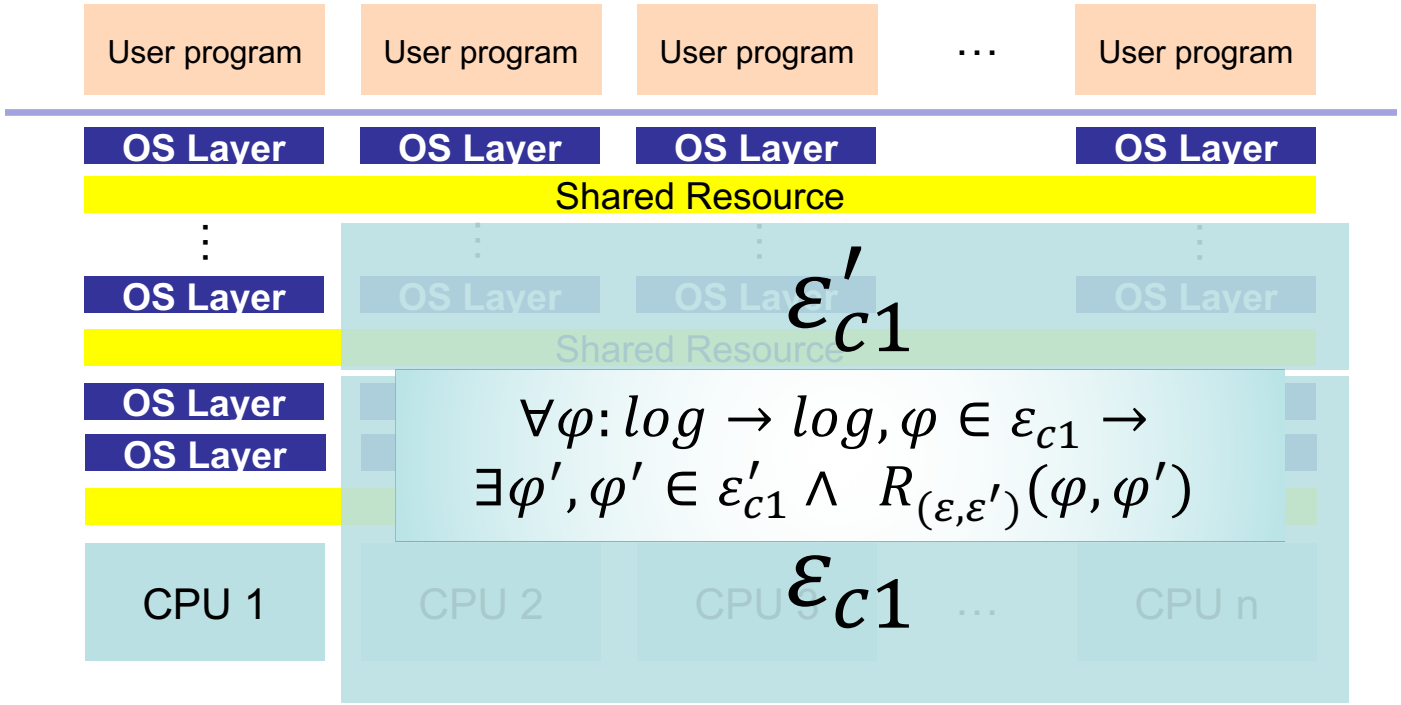
⨆

$Asm_{cpu}(CSched\left[tid, \varepsilon'_{cpu}\right]) \vdash [\![\mathbf{CertiKOS_{td}} \oplus \mathbf{Ctxt}]\!]$

Initial state: Fixed initial state

Yield rule: $CSched\left[cid, \varepsilon'_{cpu}\right](yield) \; - \; (lst, log) \rightarrow (lst/[tid = \cdots, \rho = \cdots, \cdots], log')$

Environmental Context Relation



Hide Nondeterminism

$$Asm_{oracle}(Boot[\varepsilon_{CoreSet}]) \vdash [\![\mathbf{CertiKOS} \oplus \mathbf{Ctxt}]\!]$$

$$\sqcup\!|$$

$$Asm_{mc}(Boot) \vdash [\![\mathbf{CertiKOS} \oplus \mathbf{Ctxt}]\!]$$