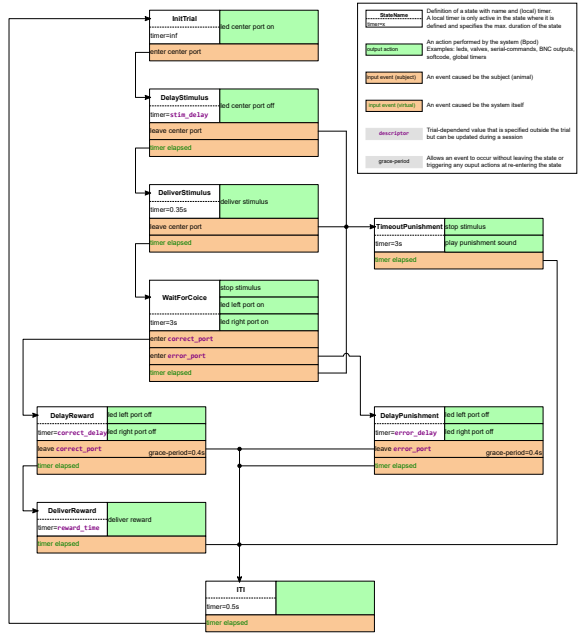
The Sparse Synchronous Model

Stephen A. Edwards and John Hui

FDL, September 15, 2020

```
sma = NewStateMatrix();
sma = AddState(sma,'Name', 'ITI',...
    'Timer',S.ITI,...
    'StateChangeConditions', {'Tup', 'PreState'},...
    'OutputActions',{});
%Pre task states
sma = AddState(sma, 'Name','PreState',...
    'Timer',S.GUI.PreCue,...
    'StateChangeConditions',{'Tup','CueDelivery'},...
    'OutputActions',{'BNCState',1});
%Cue
sma=AddState(sma,'Name', 'CueDelivery',...
    'Timer',S.GUI.CueDuration,...
    'StateChangeConditions',{'Tup', 'Delay'},...
    'OutputActions', {'SoftCode',S.Cue});
%Delay
sma=AddState(sma,'Name', 'Delay',...
    'Timer',S.Delay,...
    'StateChangeConditions',{'Tup', 'ExtraCueDelivery'},...
    'OutputActions',{});
%Extra Cue for L3-SecondaryCue
sma=AddState(sma,'Name', 'ExtraCueDelivery',...
    'Timer',S.ExtraCueDuration,...
    'StateChangeConditions', {'Tup', 'ExtraDelay'},...
    'OutputActions', {'SoftCode',S.ExtraCue});
%Extra Delay for L3-SecondaryCue
sma=AddState(sma,'Name', 'ExtraDelay',...
    'Timer',S.ExtraDelay,...
    'StateChangeConditions',{'Tup', 'Outcome'},...
    'OutputActions',{});
%Reward
sma=AddState(sma,'Name', 'Outcome',...
    'Timer',S.Outcome,...
    'StateChangeConditions', {'Tup', 'PostOutcome'},...
    'OutputActions', {'ValveState', S.Valve});
```



gcd(

Named routines, no return values

gcd(a, b,

Named routines, no return values

Pass-by-value (integer) arguments

gcd(a, b, &r)

Named routines, no return values

Pass-by-value (integer) arguments

Pass-by-reference arguments

```
gcd(a, b, &r)
  while a != b
```

Named routines, no return values

Pass-by-value (integer) arguments

Pass-by-reference arguments

Imperative *while* loops

```
gcd(a, b, &r)
  while a != b
    if  a < b then
```

Named routines, no return values

Pass-by-value (integer) arguments

Pass-by-reference arguments

Imperative *while* loops

Conditionals

```
gcd(a, b, &r)
  while a != b
    if  a < b then
      b = b − a
```

Named routines, no return values

Pass-by-value (integer) arguments

Pass-by-reference arguments

Imperative *while* loops

Conditionals

Imperative assignment

```
gcd(a, b, &r)
  while a != b
    if  a < b then
      b = b − a
    else
      a = a − b
  r = a
```

Named routines, no return values

Pass-by-value (integer) arguments

Pass-by-reference arguments

Imperative *while* loops

Conditionals

Imperative assignment

Assignment to a reference returns a value

```
fib(n, &r)
  var r1 = 0
```

Local variables

```
fib(n, &r)
  var r1 = 0
  if n < 2 then
    r = 1
  else
    fork fib(n − 1, r1)
    fork fib(n − 2, r)
    r = r1 + r
```

Local variables

Recursive routine calls

Concurrent recursive calls

```
main()
  var a = 1
  fork foo(a) bar(a)
```

```
foo(&a)
  a = a + 2

bar(&a)
  a = a * 4

main()
  var a = 1
  fork foo(a) bar(a)
```

Concurrent recursive calls

Concurrently running routines may interfere

```
foo(&a)
  a = a + 2

bar(&a)
  a = a * 4

main()
  var a = 1
  fork foo(a) bar(a)
```

Concurrent recursive calls

Concurrently running routines may interfere

Deterministic: execution order prescribed by call order

```
foo(&a)
  a = a + 2

bar(&a)
  a = a * 4

main()
  var a = 1
  fork foo(a) bar(a)
  // a = 12 = (1 + 2) * 4 here
```

Concurrent recursive calls

Concurrently running routines may interfere

Deterministic: execution order prescribed by call order

```
foo(&a)
  a = a + 2

bar(&a)
  a = a ∗ 4

main()
  var a = 1
  fork foo(a) bar(a)
  // a = 12 = (1 + 2) * 4 here
  fork bar(a) foo(a)
```

Concurrent recursive calls

Concurrently running routines may interfere

Deterministic: execution order prescribed by call order

```
foo(&a)
  a = a + 2

bar(&a)
  a = a * 4

main()
  var a = 1
  fork foo(a) bar(a)
  // a = 12 = (1 + 2) * 4 here
  fork bar(a) foo(a)
  // a = 50 = (12 * 4) + 2 here
```

Concurrent recursive calls

Concurrently running routines may interfere

Deterministic: execution order prescribed by call order

```
foo(&a)
  a = a + 2

bar(&a)
  a = a * 4

main()
  var a = 1
  fork foo(a) bar(a)
  // a = 12 = (1 + 2) * 4 here
  fork bar(a) foo(a)
  // a = 50 = (12 * 4) + 2 here
```

Concurrent recursive calls

Concurrently running routines may interfere

Deterministic: execution order prescribed by call order

No true parallelism, for now

Delayed assignment:
future update scheduled

```
blink(&led)
   while 1
      after 50 ms led = 1
```

```
blink(&led)
   while 1
      after 50 ms led = 1
      wait led
```

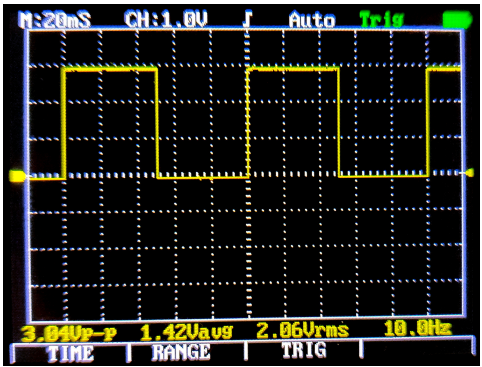
Delayed assignment:
future update scheduled

Blocking wait-for-write

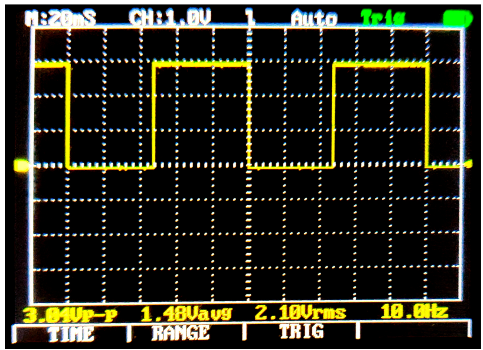
```
blink(&led)
   while 1
      after 50 ms led = 1
      wait led
      after 50 ms led = 0
      wait led
```

Delayed assignment:
future update scheduled

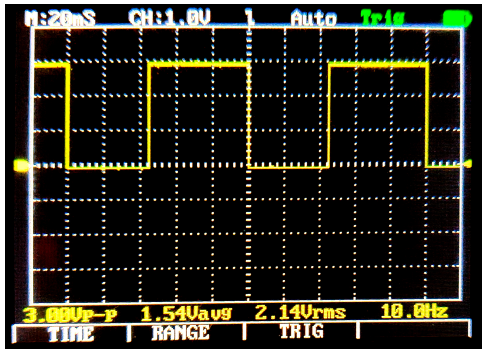
Blocking wait-for-write



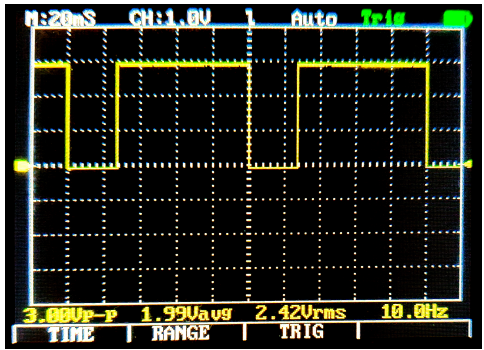
```
blink(&led)
  while 1

    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led
```

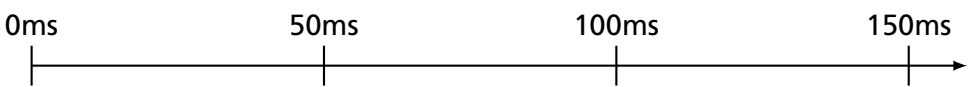


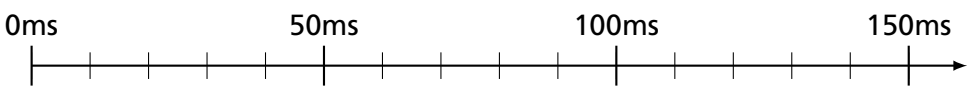
```
blink(&led)
  while 1
    fib(19, r)
    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led
```

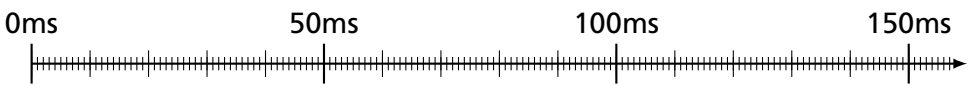


```
blink(&led)
  while 1
    fib(23, r)
    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led
```

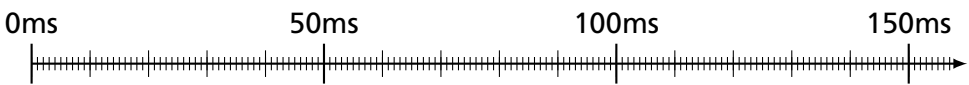



0ms
50ms
100ms
150ms

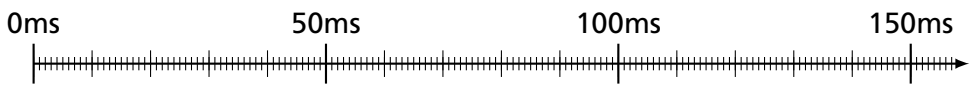

0ms
50ms
100ms
150ms


0ms
50ms
100ms
150ms

```
blink(&led)
   while 1
      after 50 ms led = 1
      wait led
      after 50 ms led = 0
      wait led
```

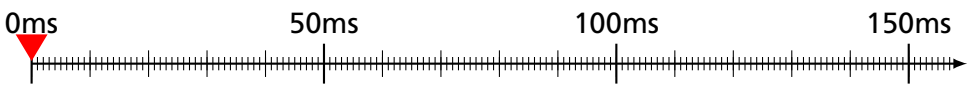


```
blink(&led)
   while 1
      after 50 ms led = 1
      wait led
      after 50 ms led = 0
      wait led
```



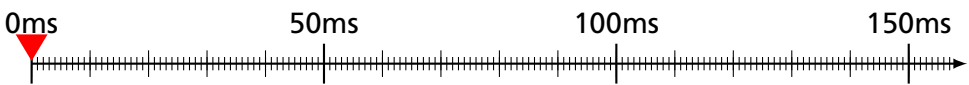
led 0

```
blink(&led)
   while 1
      after 50 ms led = 1
      wait led
      after 50 ms led = 0
      wait led
```



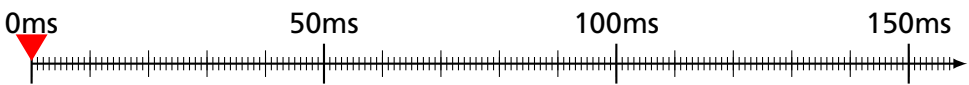
led 0

```
blink(&led)
    while 1
        after 50 ms led = 1
        wait led
        after 50 ms led = 0
        wait led
```



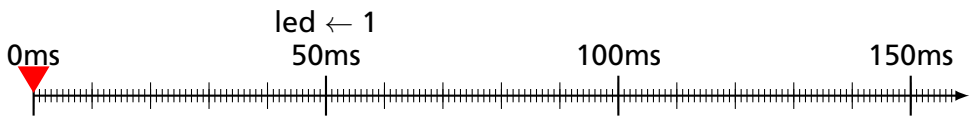
led 0

```
blink(&led)
    while 1
        after 50 ms led = 1
        wait led
        after 50 ms led = 0
        wait led
```



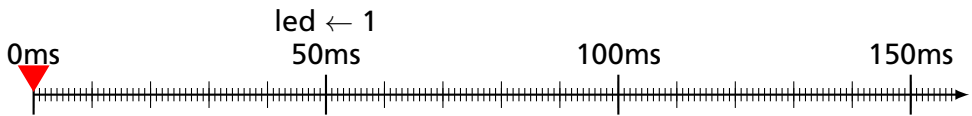
led 0

```
blink(&led)
   while 1
      after 50 ms led = 1
      wait led
      after 50 ms led = 0
      wait led
```



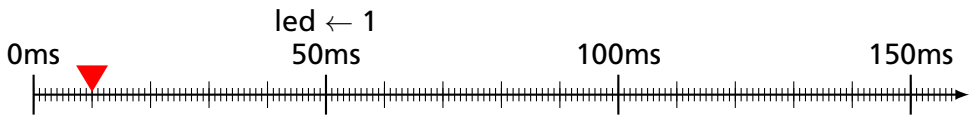
led 0

```
blink(&led)
  while 1
    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led
```



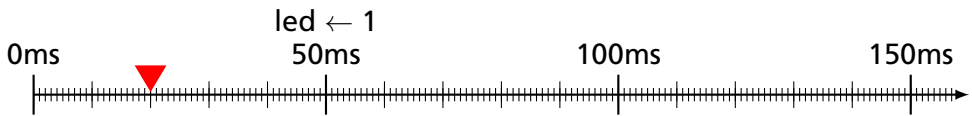
led 0

```
blink(&led)
   while 1
      after 50 ms led = 1
      wait led
      after 50 ms led = 0
      wait led
```



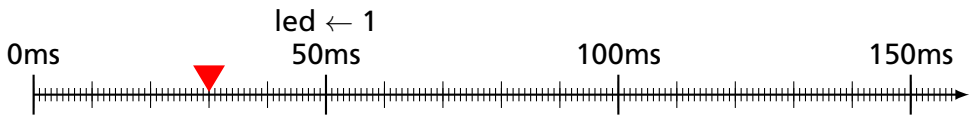
led

```
blink(&led)
   while 1
      after 50 ms led = 1
      wait led
      after 50 ms led = 0
      wait led
```



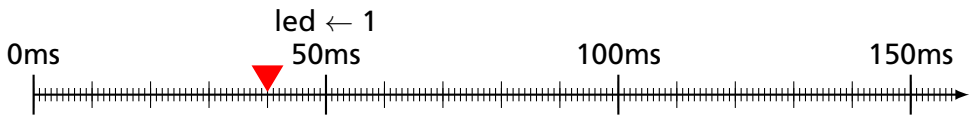
led

```
blink(&led)
   while 1
      after 50 ms led = 1
      wait led
      after 50 ms led = 0
      wait led
```



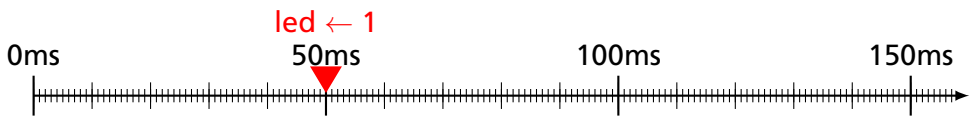
led

```
blink(&led)
   while 1
      after 50 ms led = 1
      wait led
      after 50 ms led = 0
      wait led
```

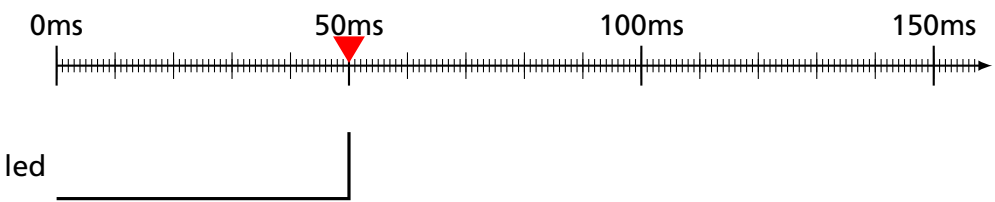


led

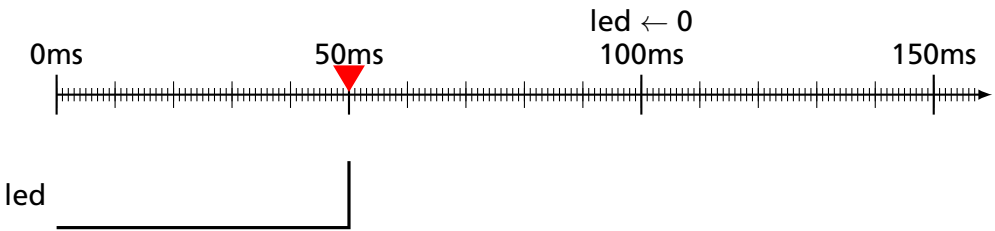
```
blink(&led)
   while 1
      after 50 ms led = 1
      wait led
      after 50 ms led = 0
      wait led
```



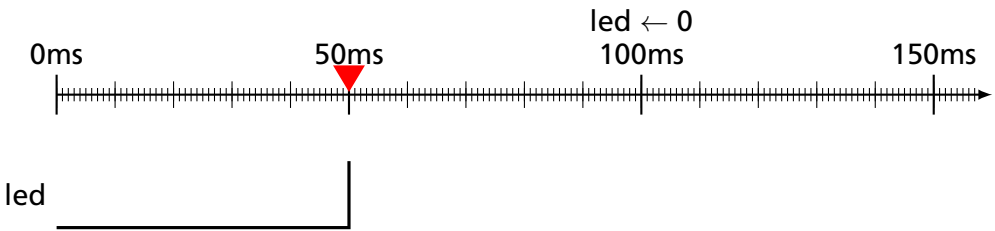
```
blink(&led)
   while 1
      after 50 ms led = 1
      wait led
      after 50 ms led = 0
      wait led
```



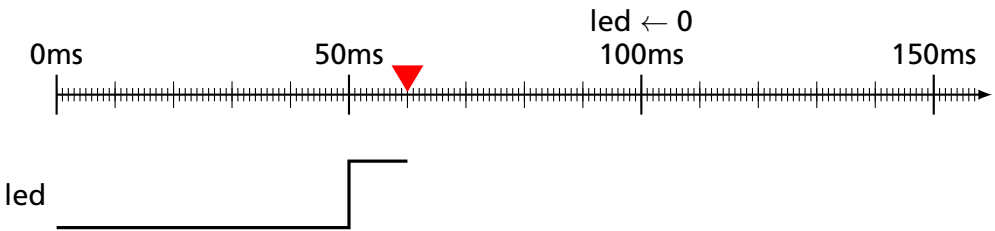
```
blink(&led)
   while 1
      after 50 ms led = 1
      wait led
      after 50 ms led = 0
      wait led
```



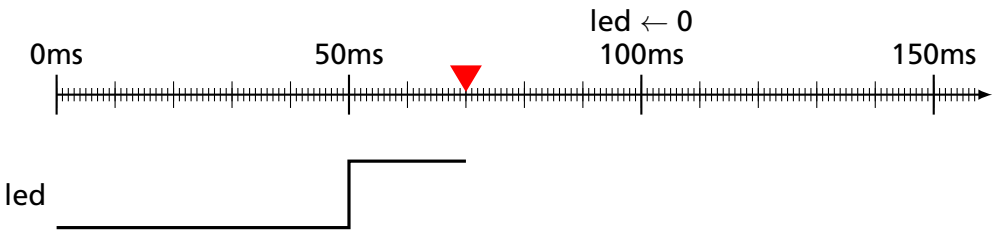
```
blink(&led)
   while 1
      after 50 ms led = 1
      wait led
      after 50 ms led = 0
      wait led
```



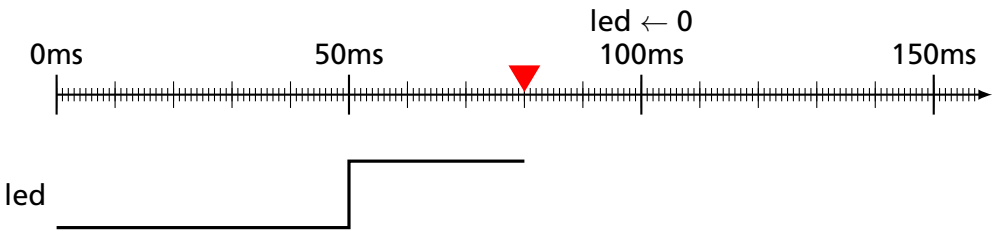
```
blink(&led)
   while 1
      after 50 ms led = 1
      wait led
      after 50 ms led = 0
      wait led
```



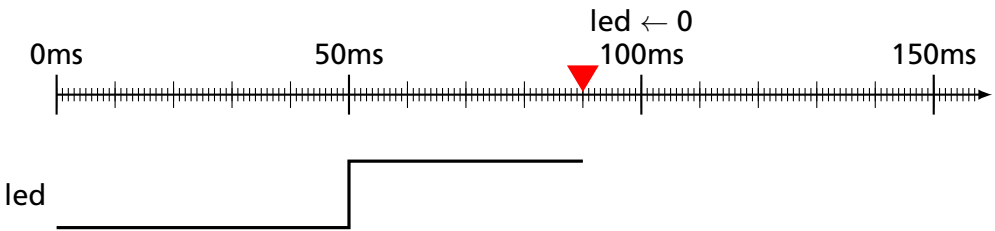
```
blink(&led)
   while 1
      after 50 ms led = 1
      wait led
      after 50 ms led = 0
      wait led
```



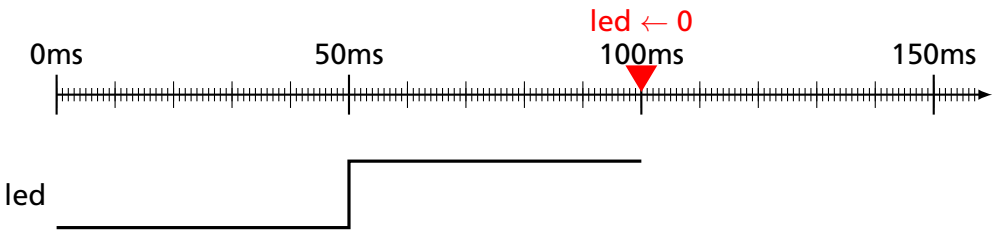
```
blink(&led)
   while 1
      after 50 ms led = 1
      wait led
      after 50 ms led = 0
      wait led
```



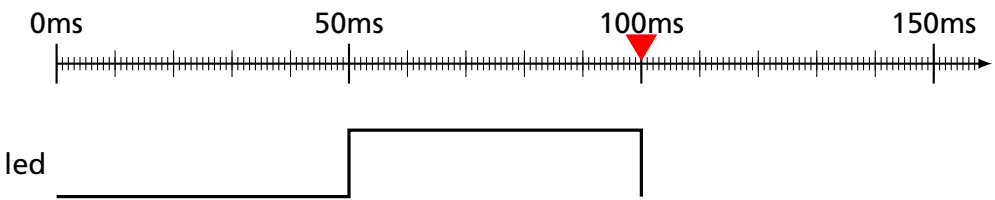
```
blink(&led)
   while 1
      after 50 ms led = 1
      wait led
      after 50 ms led = 0
      wait led
```



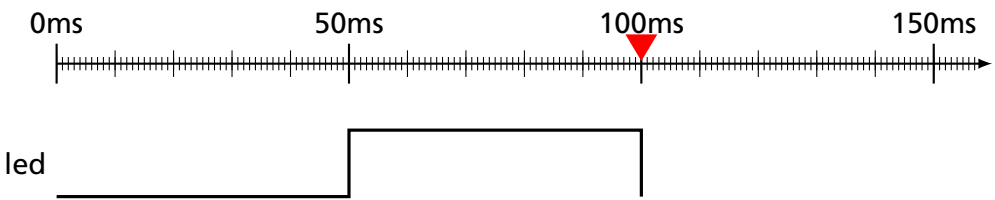
```
blink(&led)
   while 1
      after 50 ms led = 1
      wait led
      after 50 ms led = 0
      wait led
```



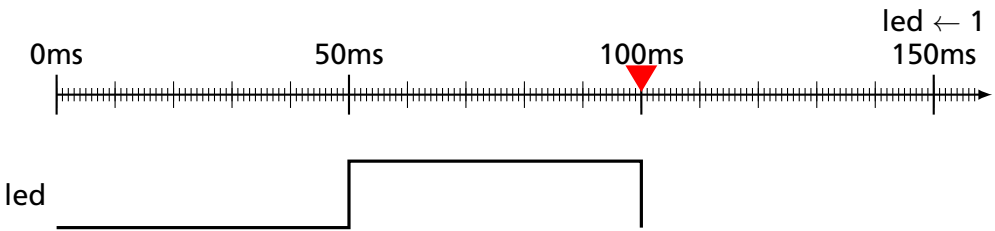
```
blink(&led)
   while 1
      after 50 ms led = 1
      wait led
      after 50 ms led = 0
      wait led
```



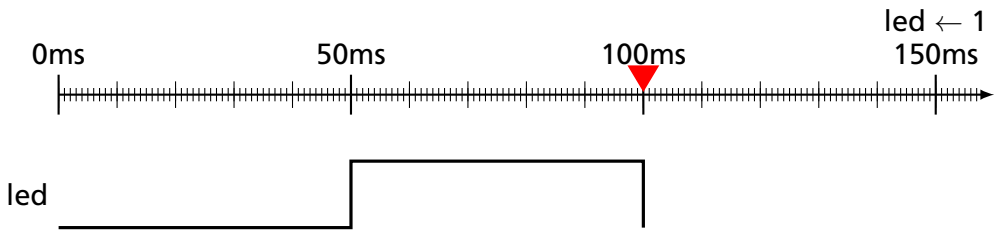
```
blink(&led)
  while 1
    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led
```



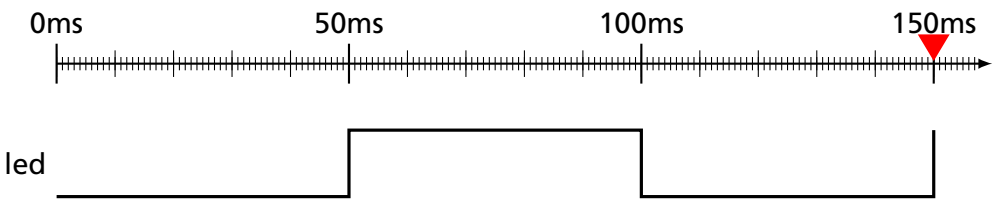
```
blink(&led)
  while 1
    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led
```



```
blink(&led)
  while 1
    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led
```



```
blink(&led)
   while 1
      after 50 ms led = 1
      wait led
      after 50 ms led = 0
      wait led
```



main(&led)

```
main(&led)
   fork blink(led,        ) blink(led,        ) blink(led,        )
```

```
main(&led)
   fork blink(led, 50ms) blink(led,          ) blink(led,          )
```

```
main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led,          )
```

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)

blink(&led, period)

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)

```
blink(&led, period)
   var e = 0
```

```
main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

```
blink(&led, period)
   var e = 0
   while 1
```

```
main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

```
blink(&led, period)
   var e = 0
   while 1
      toggle(led)


main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)



main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0


main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

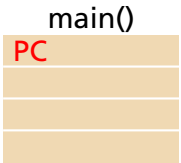
now = 0 ms

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

main()

now = 0 ms

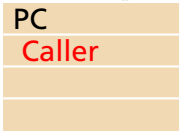
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



now = 0 ms

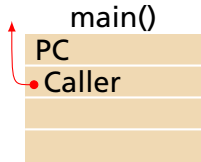
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

main()



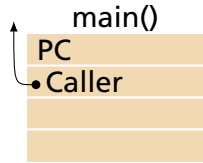
now = 0 ms

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



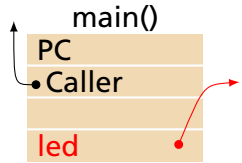
now = 0 ms

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



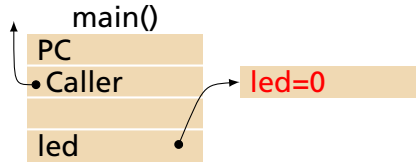
now = 0 ms

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



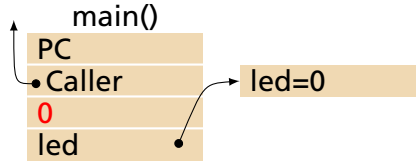
now = 0 ms

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



now = 0 ms

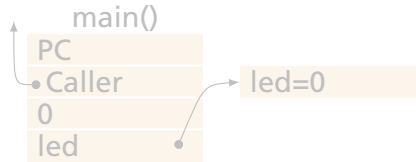
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



now = 0 ms

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

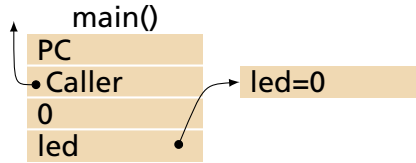
Ready:



now = 0 ms

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

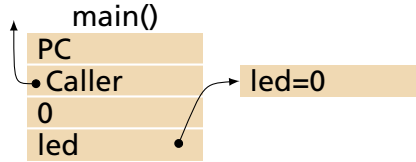
Ready: 0



now = 0 ms

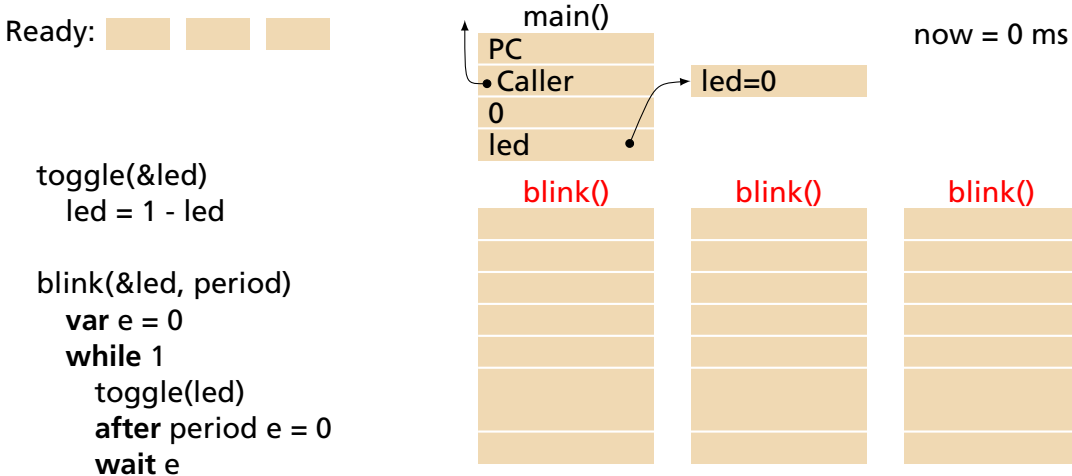
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

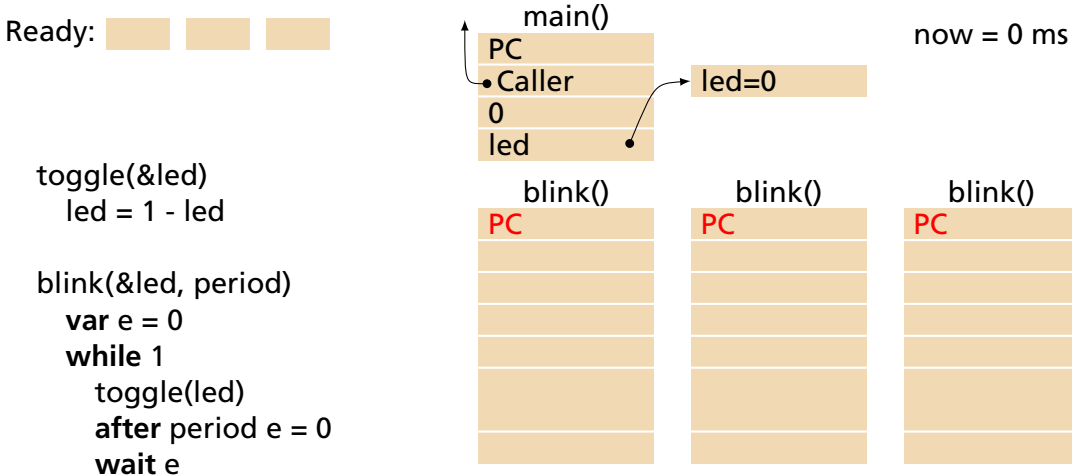
Ready:



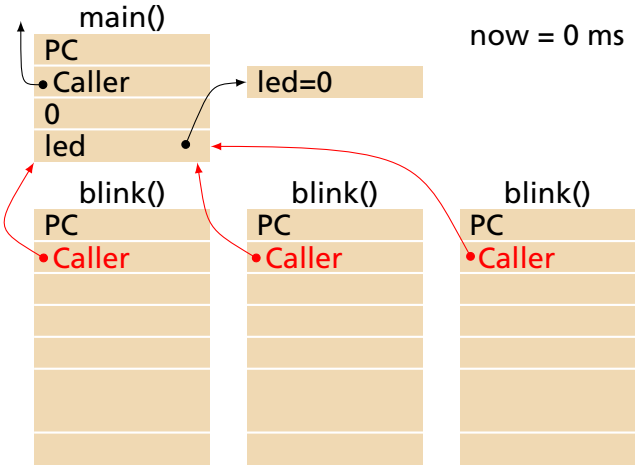
now = 0 ms

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```


Ready:
main()
PC
Caller
0
led
led=0
now = 0 ms
toggle(&led)
led = 1 - led
blink(&led, period)
var e = 0
while 1
toggle(led)
after period e = 0
wait e
blink()
blink()
blink()
main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)

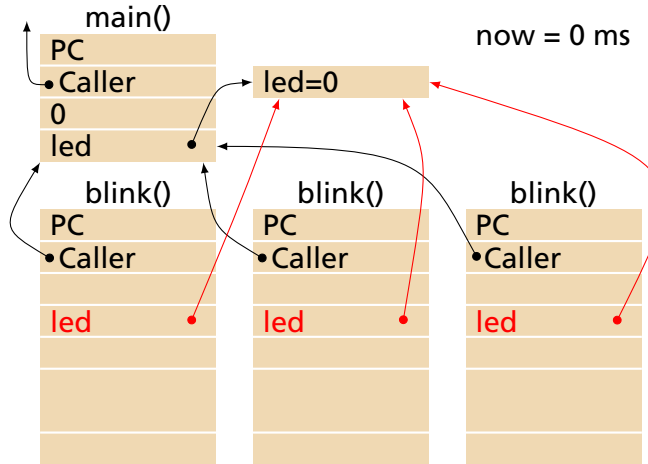

Ready:
main()
PC
Caller
0
led
led=0
now = 0 ms
toggle(&led)
led = 1 - led
blink(&led, period)
var e = 0
while 1
toggle(led)
after period e = 0
wait e
blink()
PC
blink()
PC
blink()
PC
main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)

Ready:



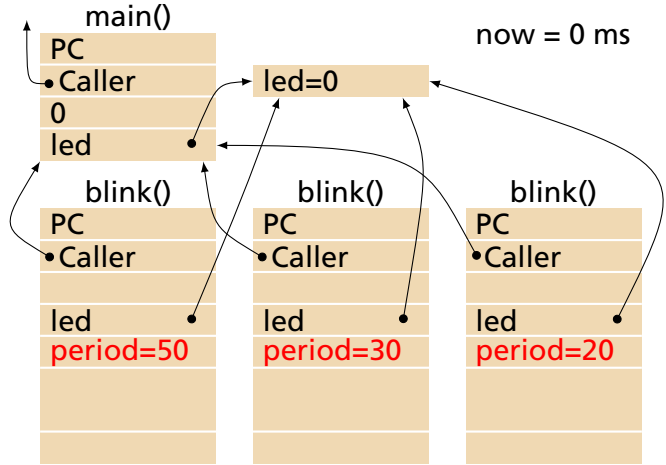

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

Ready:




```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

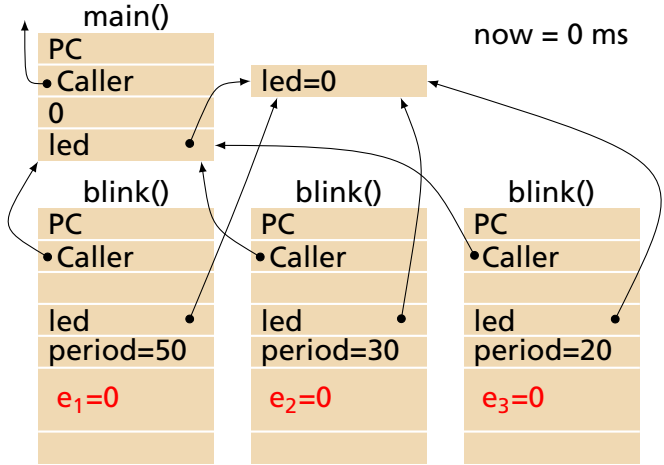
Ready:




```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

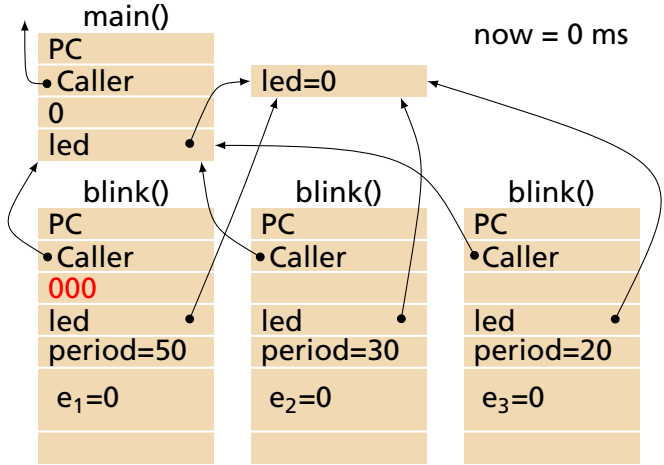
Ready:

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



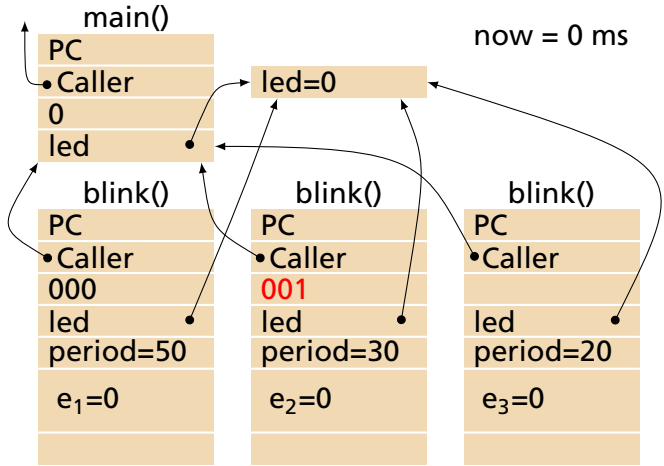
Ready:

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



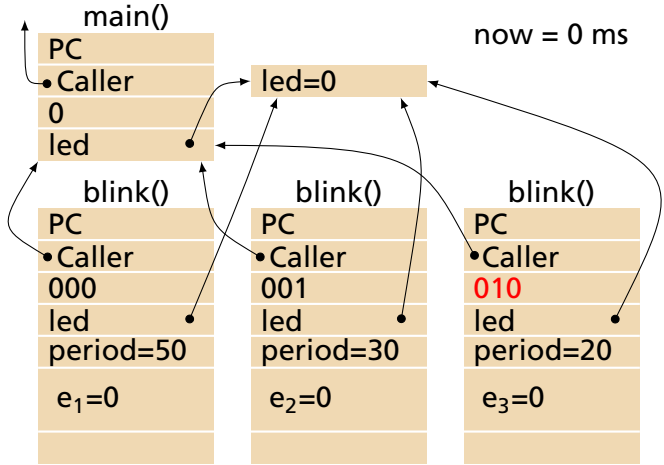
Ready:

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



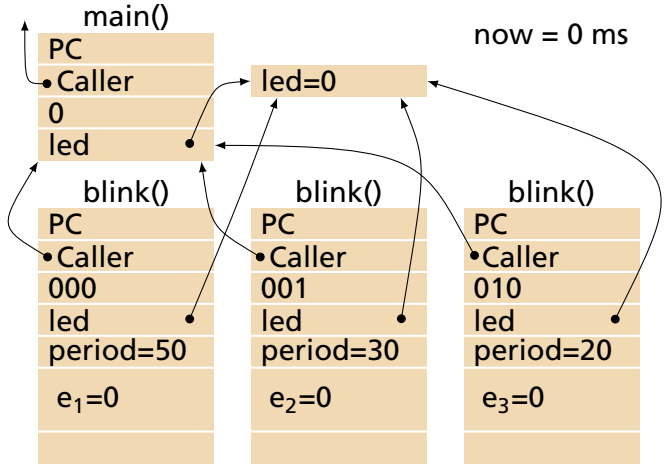
Ready:

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



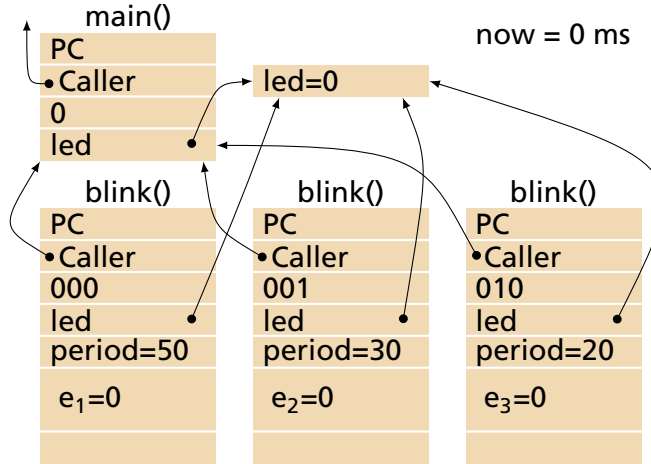
Ready: 000 001 010

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

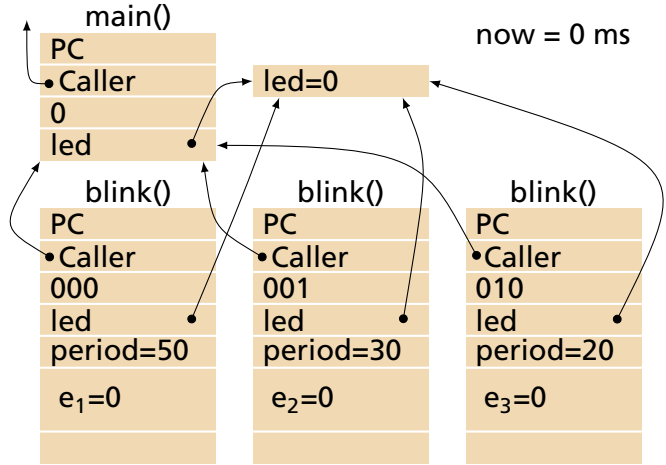


Ready: 001 010

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

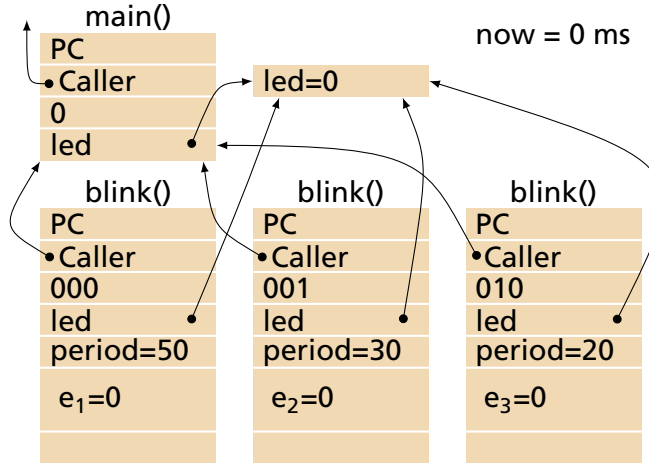


Ready: 001 010




```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

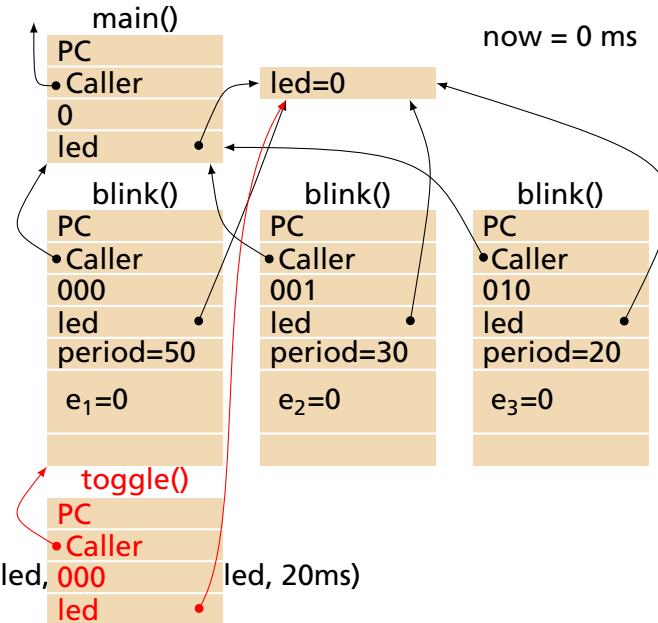
Ready: 001 010




```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

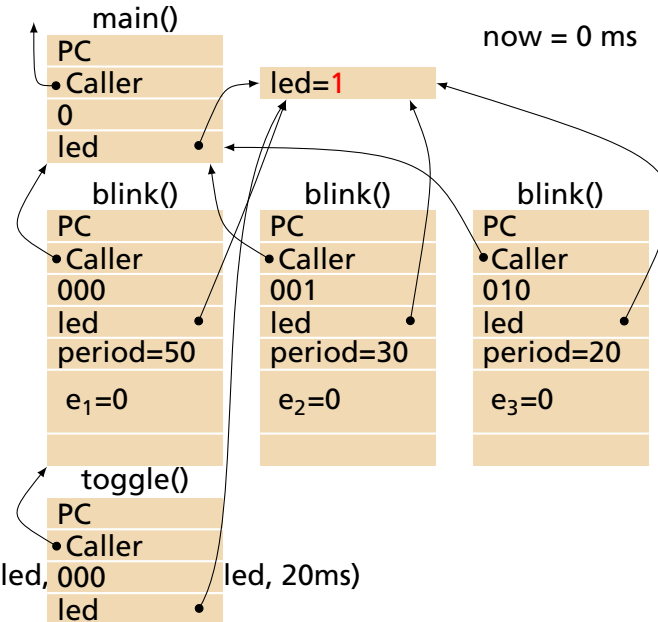
Ready: 001 010

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



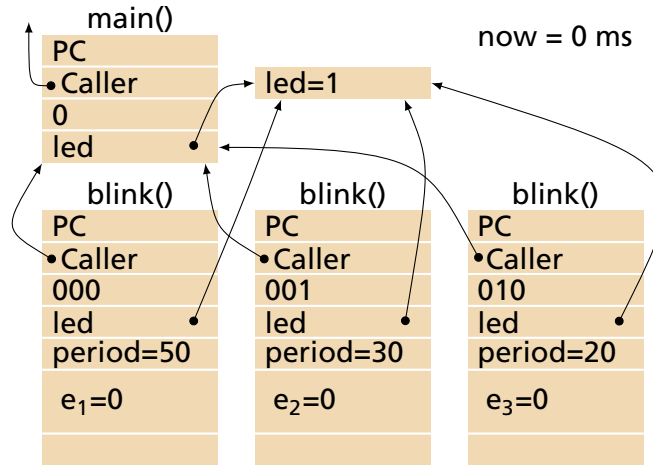
Ready: 001 010

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



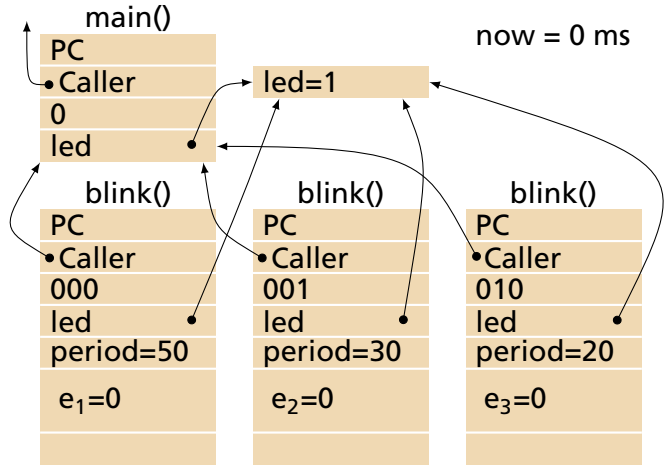
Ready: 001 010

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led,          led, 20ms)
```

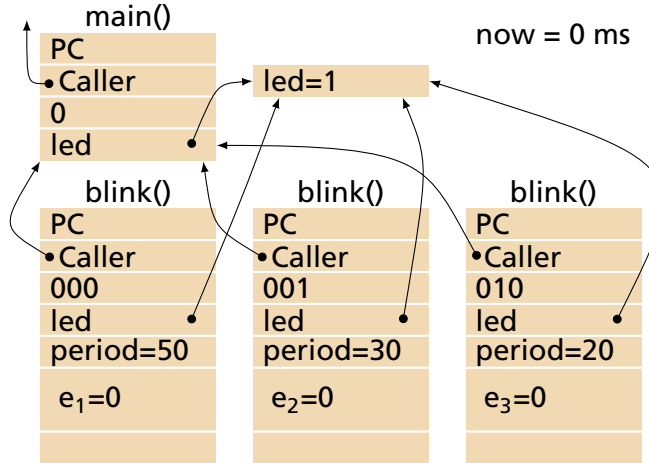


Ready: 001 010

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



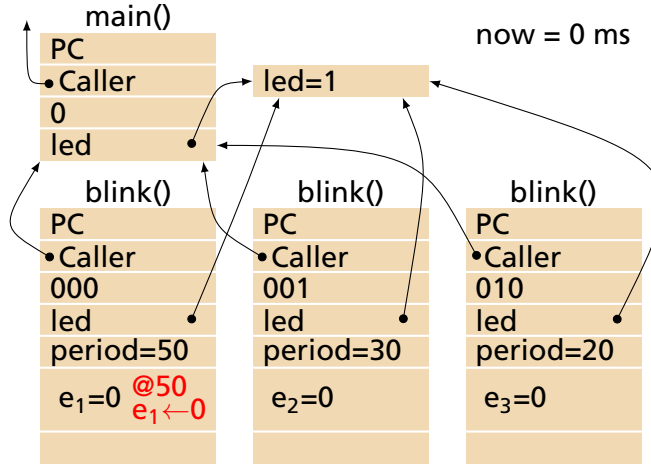
Ready: 001 010




```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

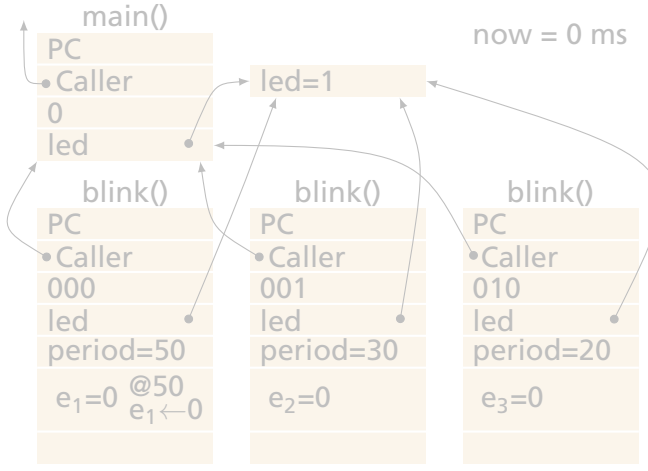
Ready: 001 010

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

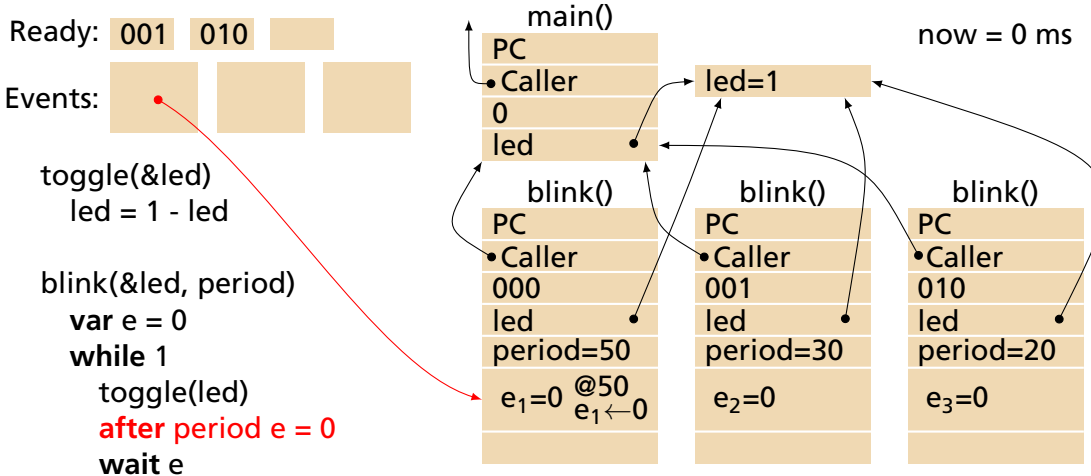


Ready: 001 010

Events:



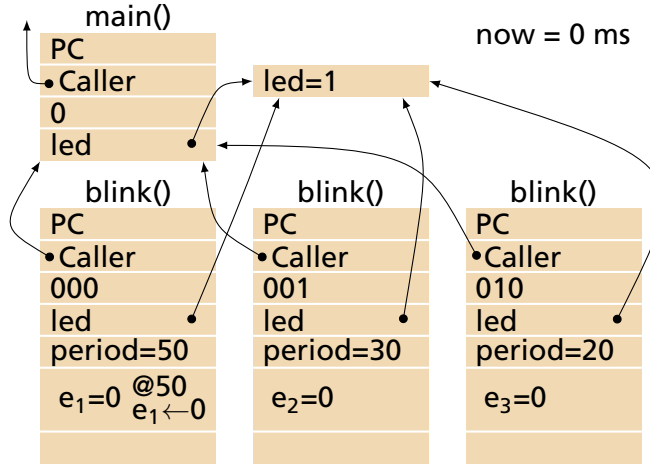

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```


Ready: 001 010
Events:
toggle(&led)
led = 1 - led
blink(&led, period)
var e = 0
while 1
toggle(led)
after period e = 0
wait e
main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
main()
PC
Caller
0
led
led=1
now = 0 ms
blink()
PC
Caller
000
led
period=50
e1=0 @50 e1←0
blink()
PC
Caller
001
led
period=30
e2=0
blink()
PC
Caller
010
led
period=20
e3=0

Ready: 001 010

Events: @50 $e_1 \leftarrow 0$

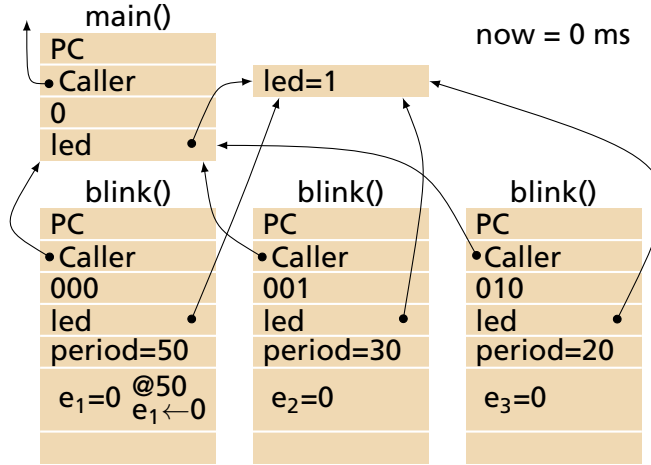
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 001 010

Events: @50 $e_1 \leftarrow 0$

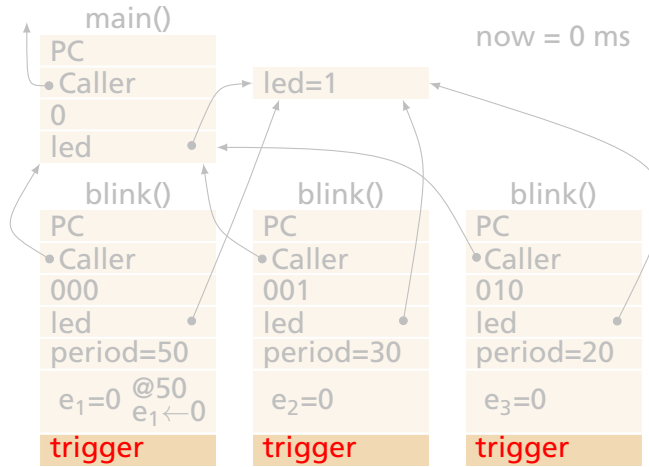
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 001 010

Events: @50 $e_1 \leftarrow 0$

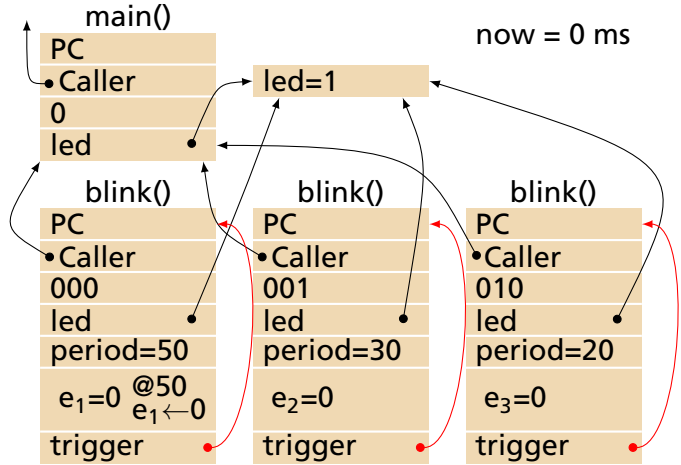
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 001 010

Events: @50 $e_1 \leftarrow 0$

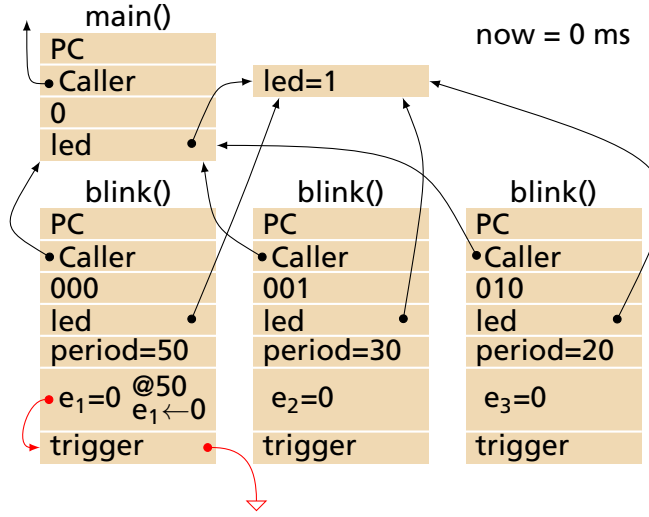
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 001 010

Events: @50 $e_1 \leftarrow 0$

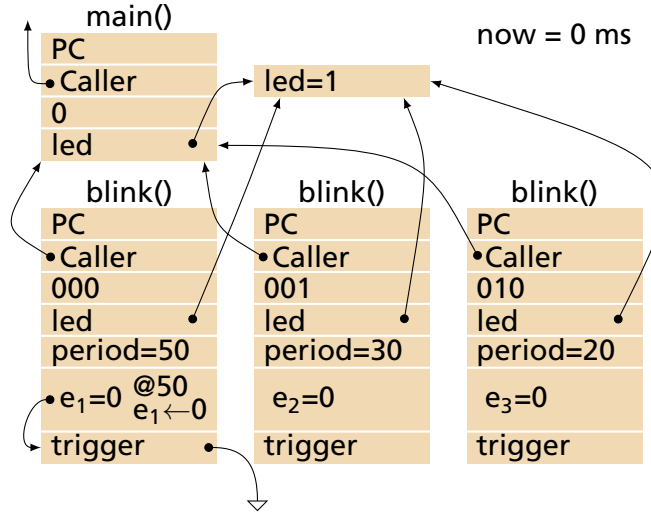
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 001 010

Events: @50 $e_1 \leftarrow 0$

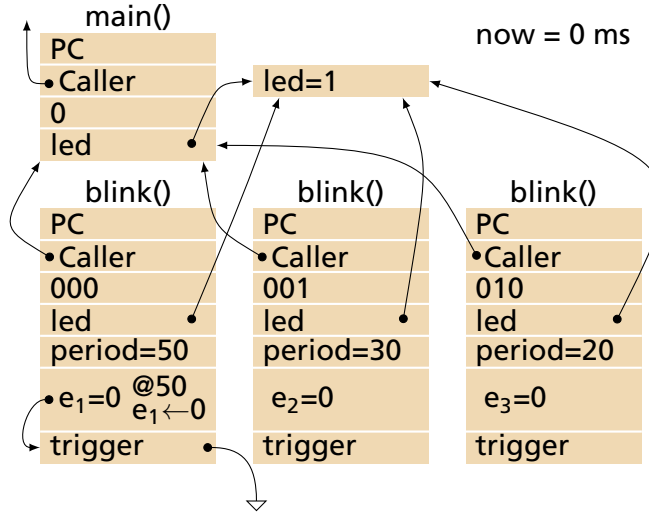
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 010

Events: @50 $e_1 \leftarrow 0$

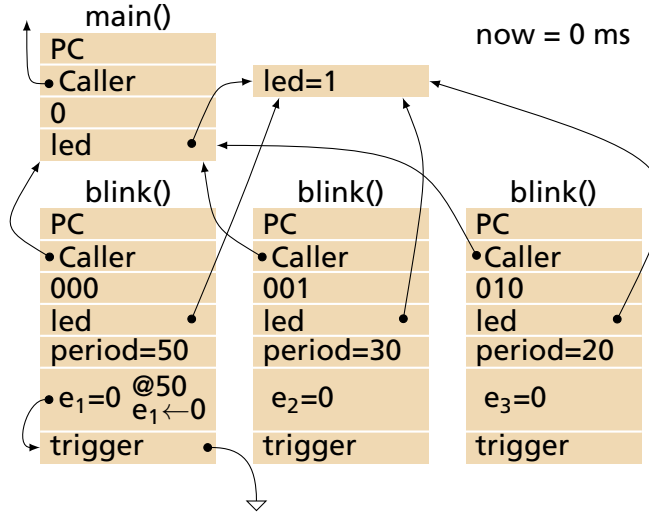
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 010

Events: @50 $e_1 \leftarrow 0$

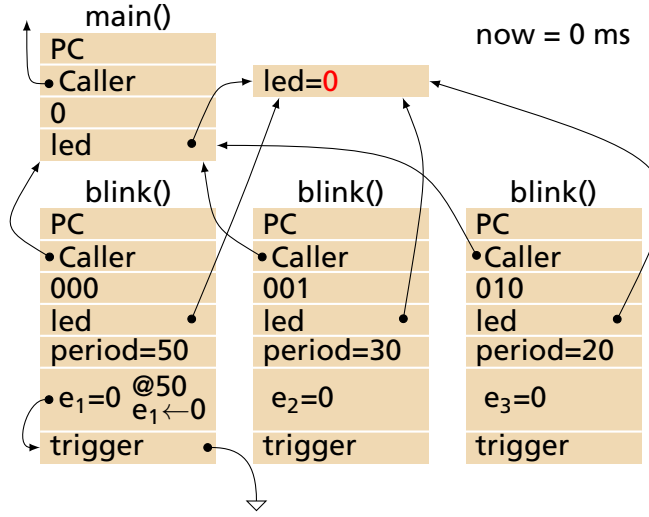
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 010

Events: @50 $e_1 \leftarrow 0$

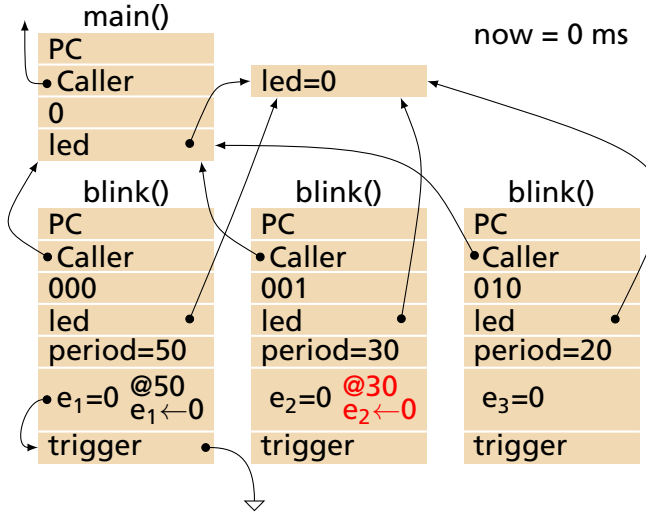
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 010

Events: @50 $e_1 \leftarrow 0$

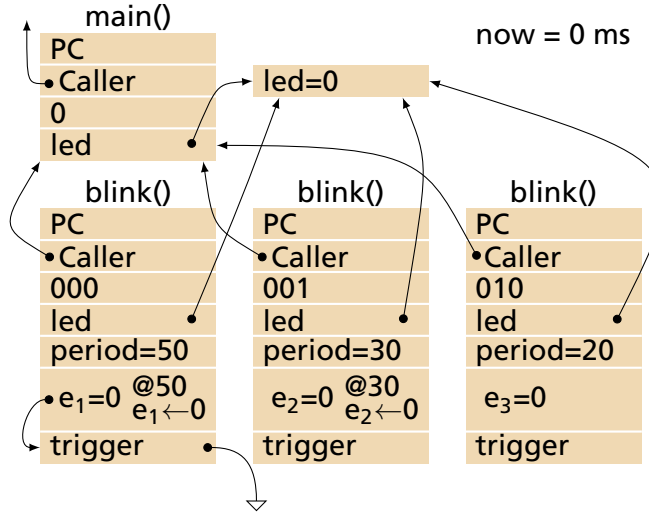
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 010

Events: @30 $e_2 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

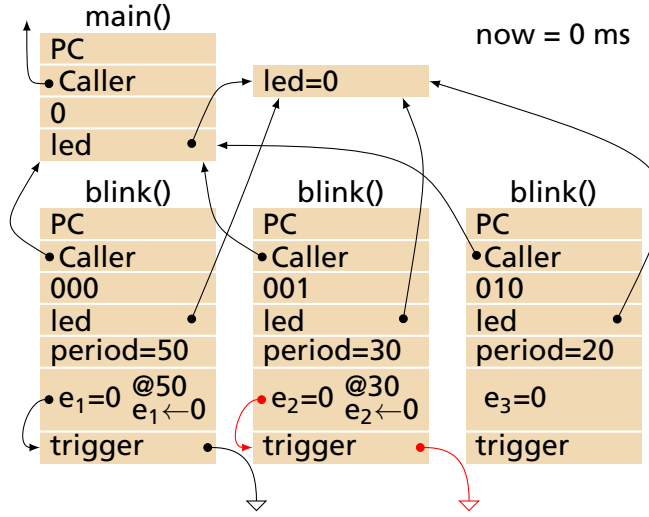
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 010

Events: @30 $e_2 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

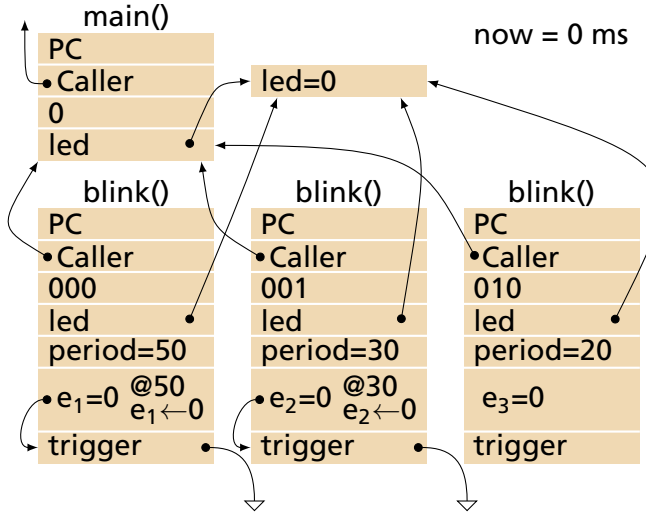
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 010

Events: @30 $e_2 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

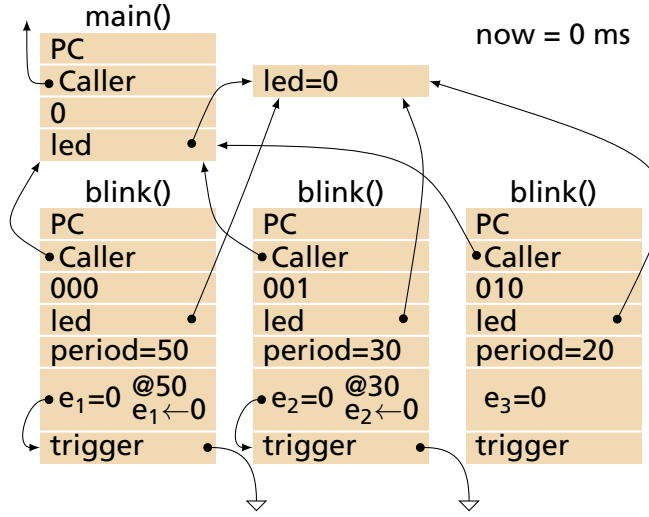
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 010

Events: @30 $e_2 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

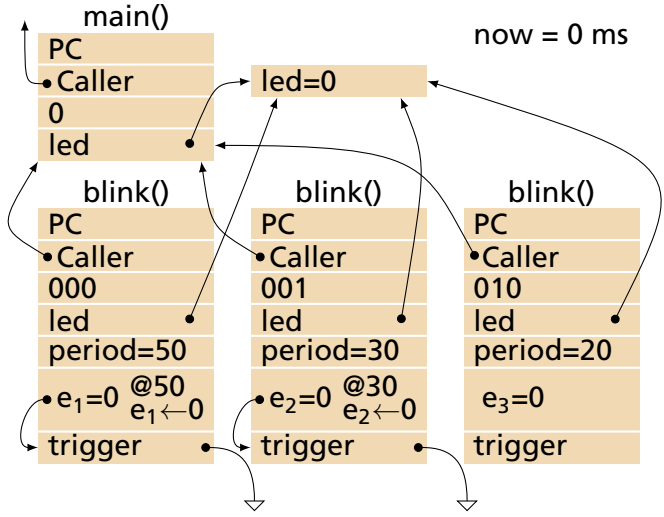
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

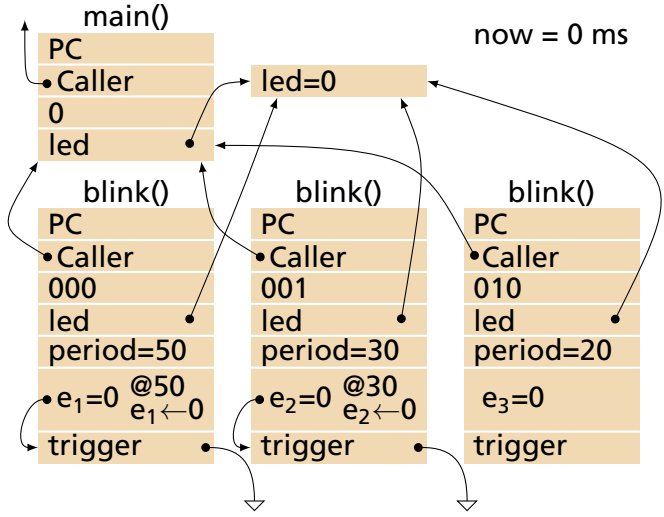
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

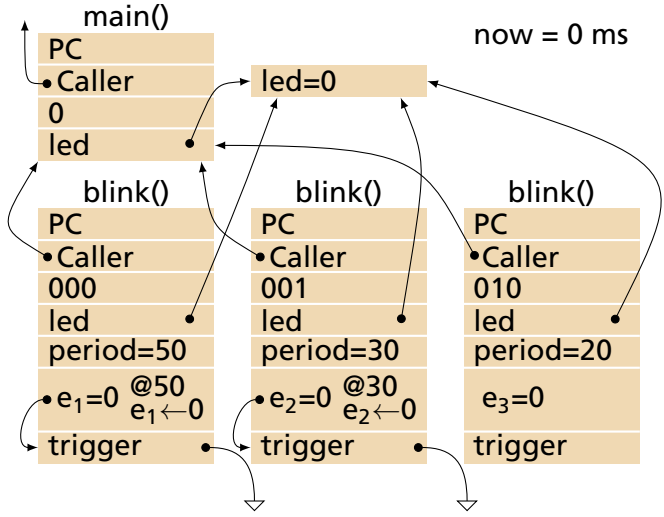
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

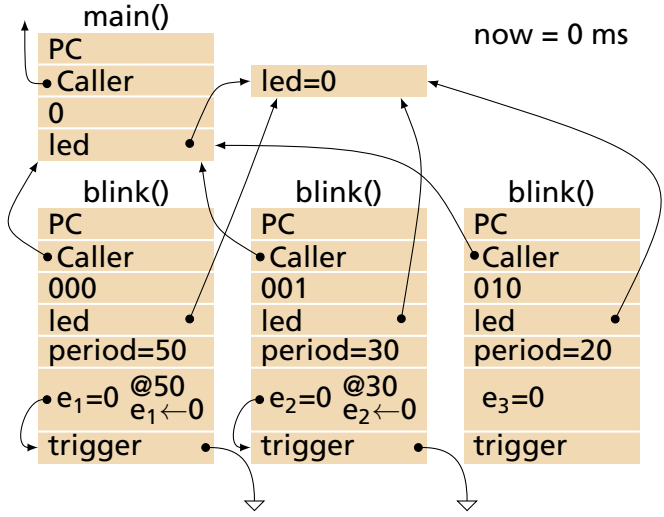
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ @50 $e_1 \leftarrow 0$

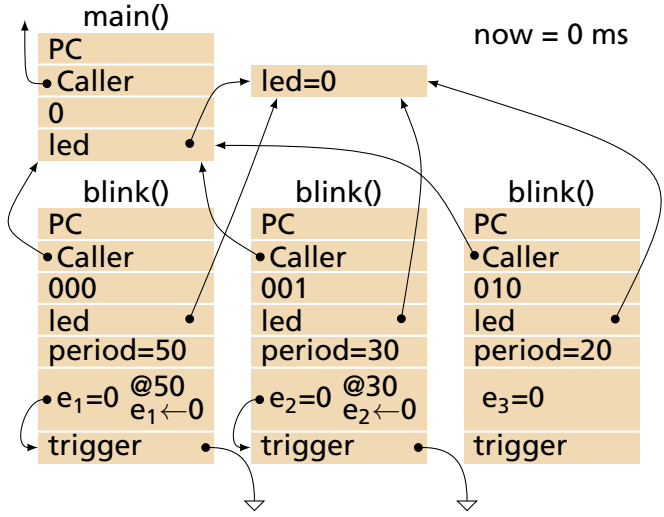
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

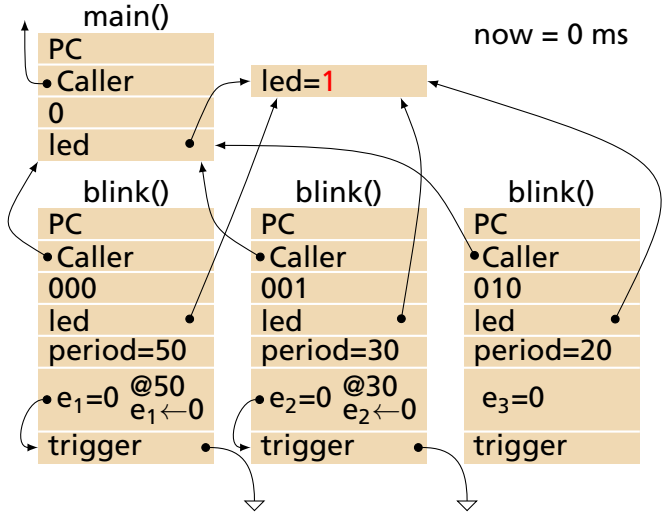
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

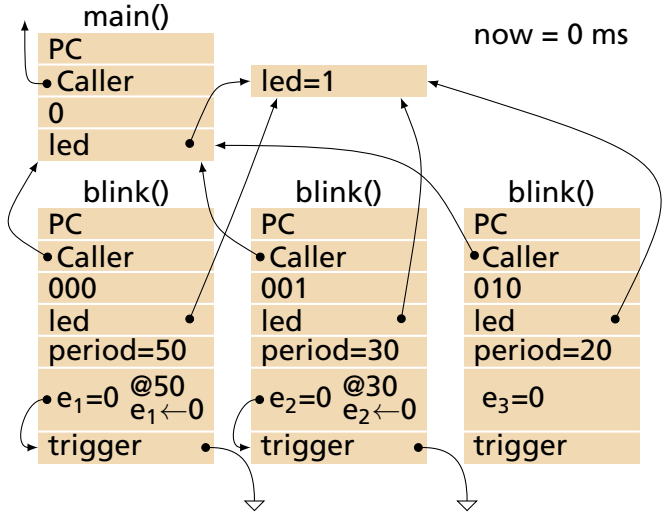
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ @50 $e_1 \leftarrow 0$

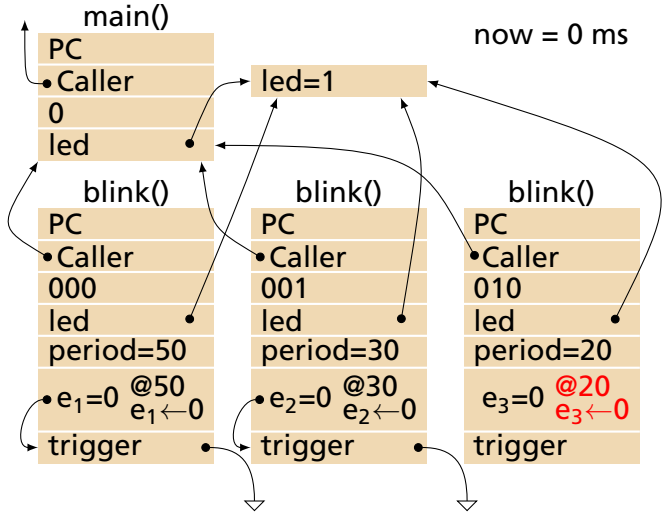
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

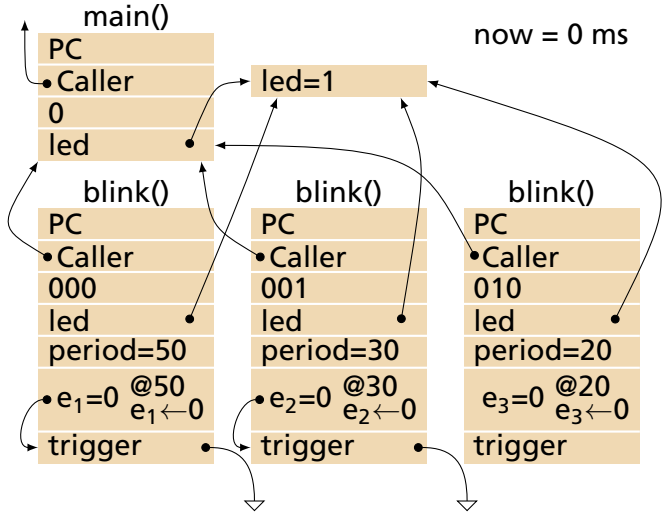
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @20 $e_3 \leftarrow 0$ | @30 $e_2 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

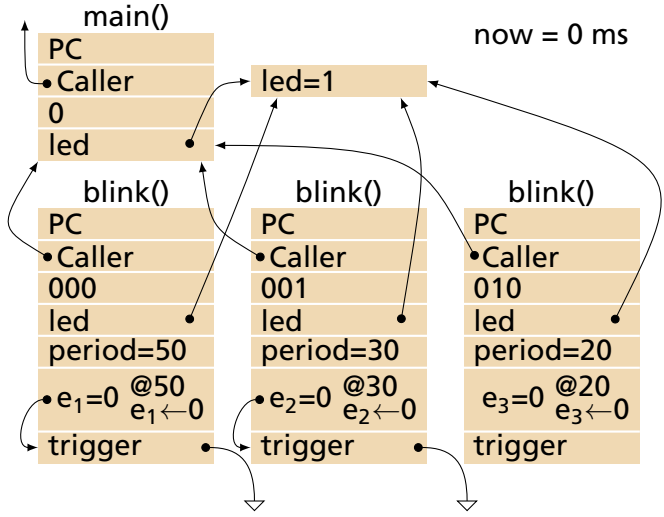
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @20 $e_3 \leftarrow 0$ | @30 $e_2 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

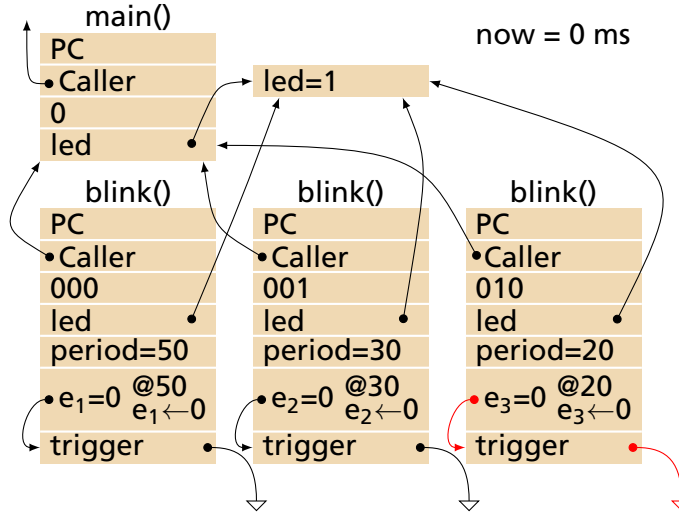
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @20 $e_3 \leftarrow 0$ | @30 $e_2 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

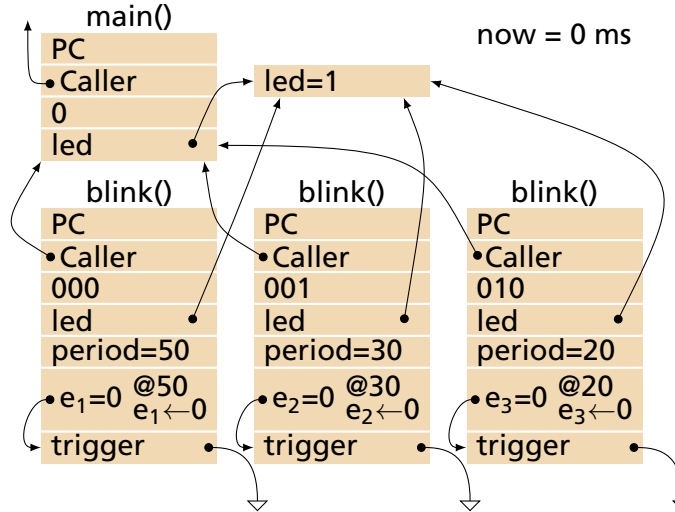
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @20 $e_3 \leftarrow 0$ | @30 $e_2 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

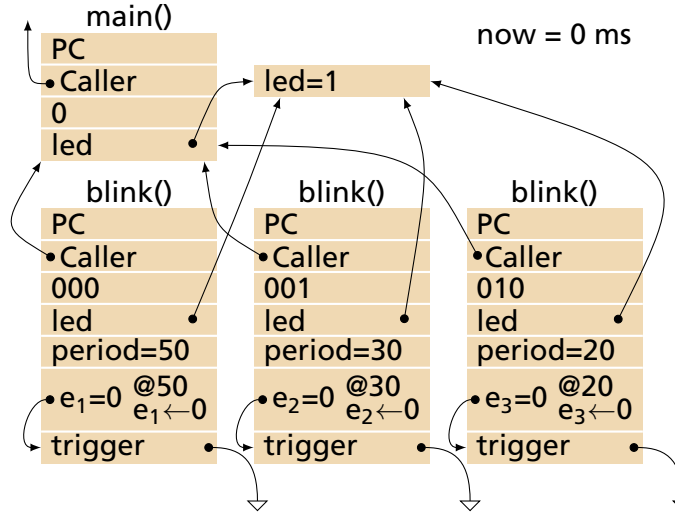
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @20 $e_3 \leftarrow 0$ | @30 $e_2 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

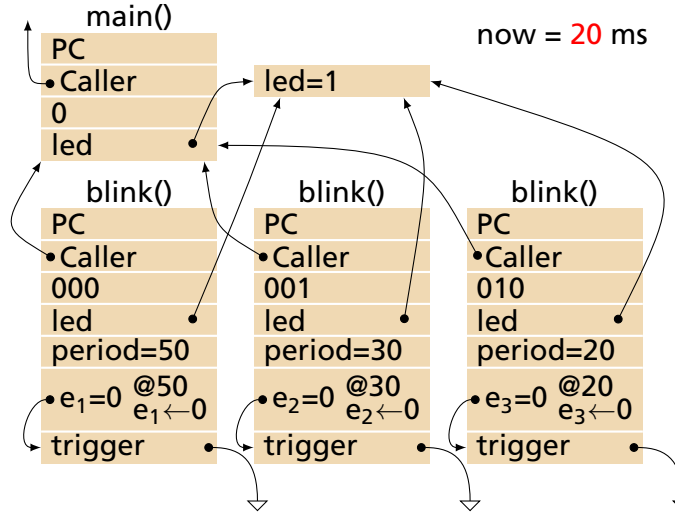
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @20 $e_3 \leftarrow 0$ | @30 $e_2 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

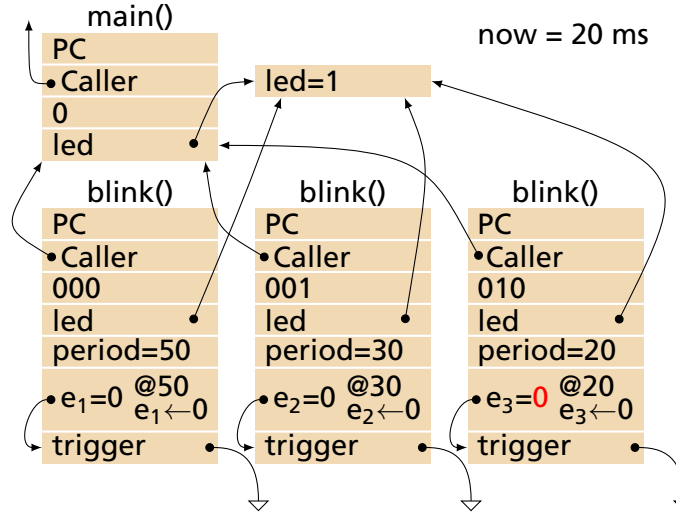
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @20 $e_3 \leftarrow 0$ | @30 $e_2 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

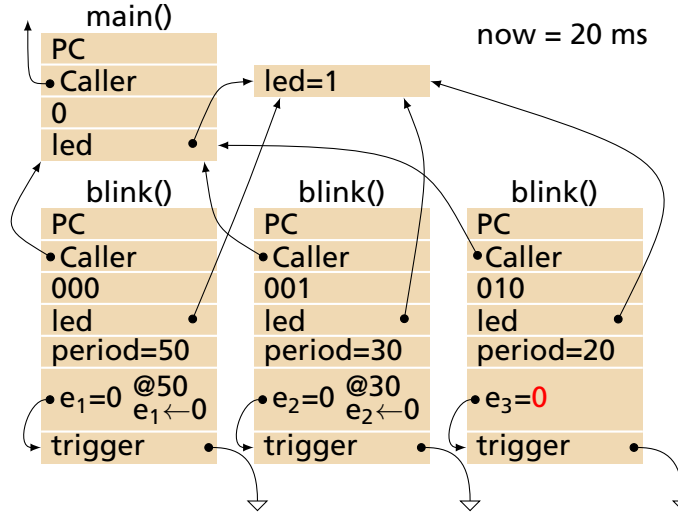
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

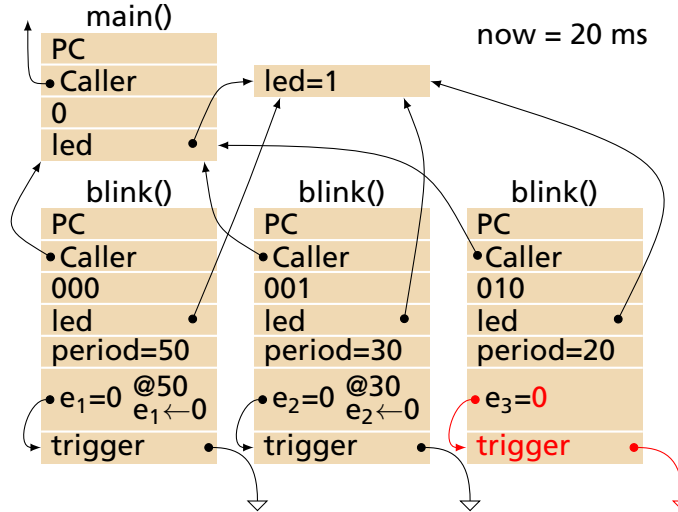
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

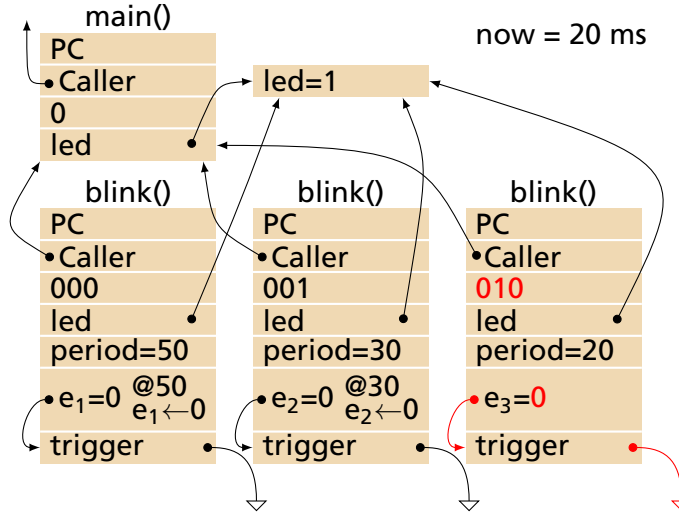
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ @50 $e_1 \leftarrow 0$

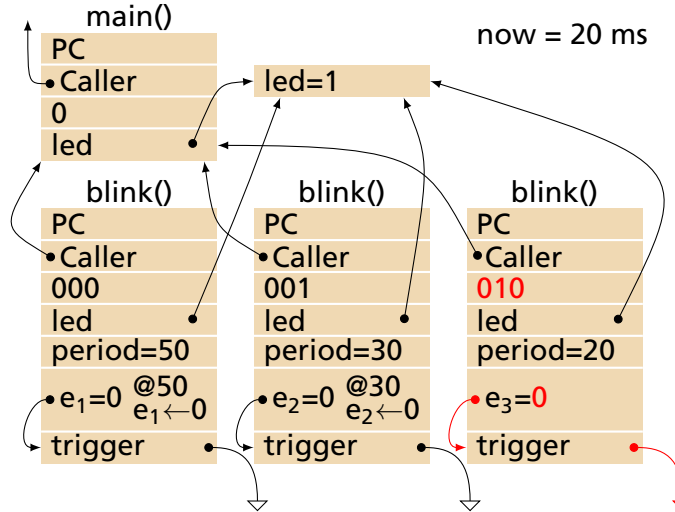
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 010

Events: @30 $e_2 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

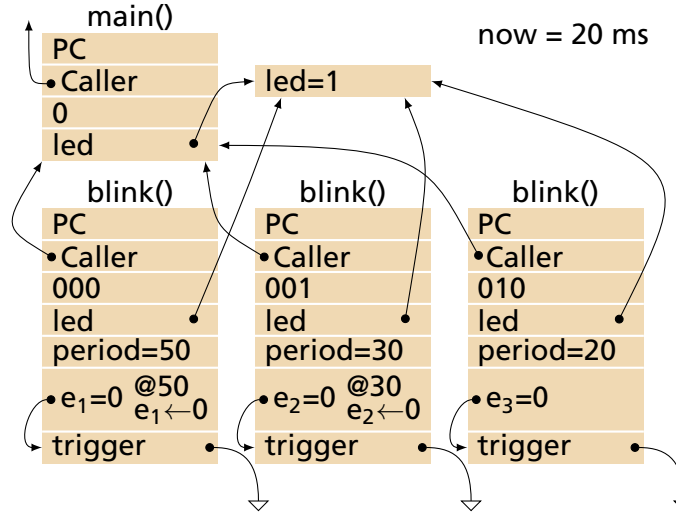
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 010

Events: @30 $e_2 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

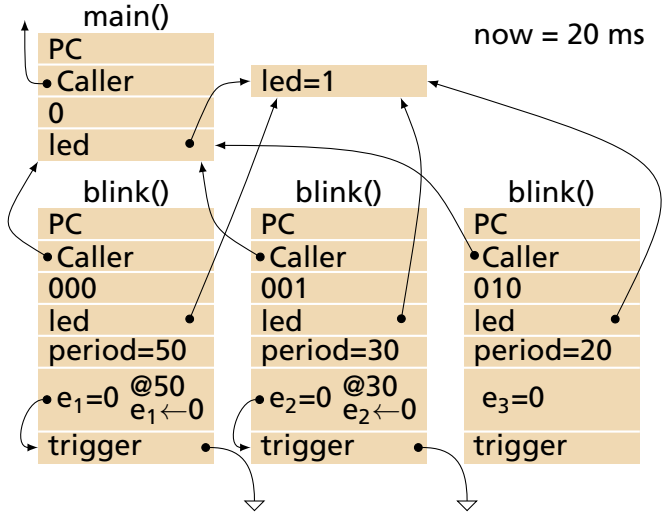
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ @50 $e_1 \leftarrow 0$

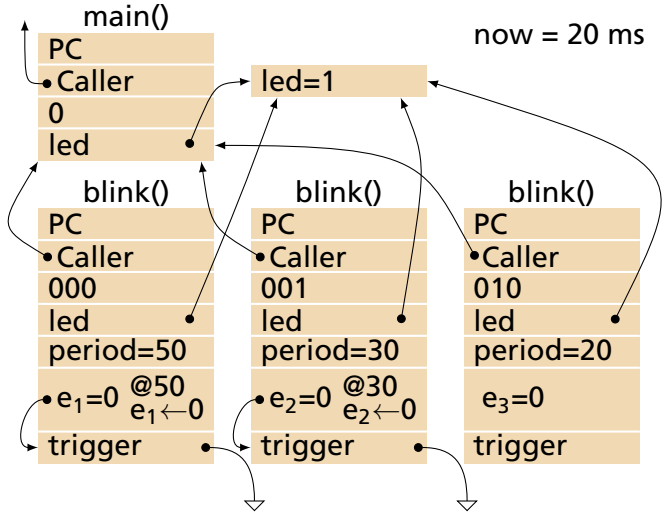
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ @50 $e_1 \leftarrow 0$

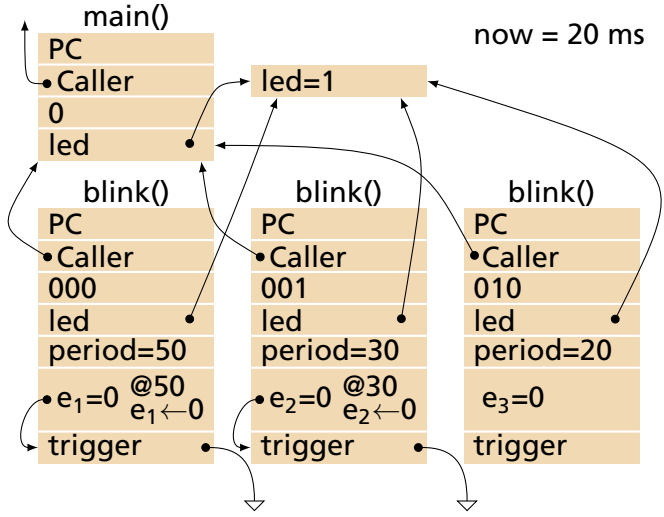
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

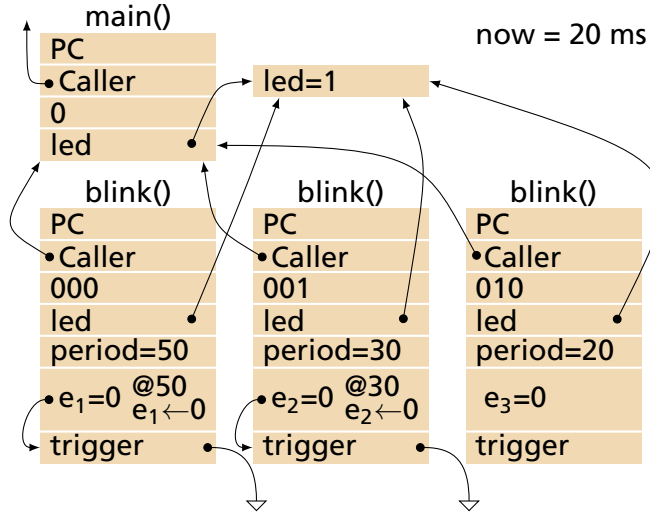
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ @50 $e_1 \leftarrow 0$

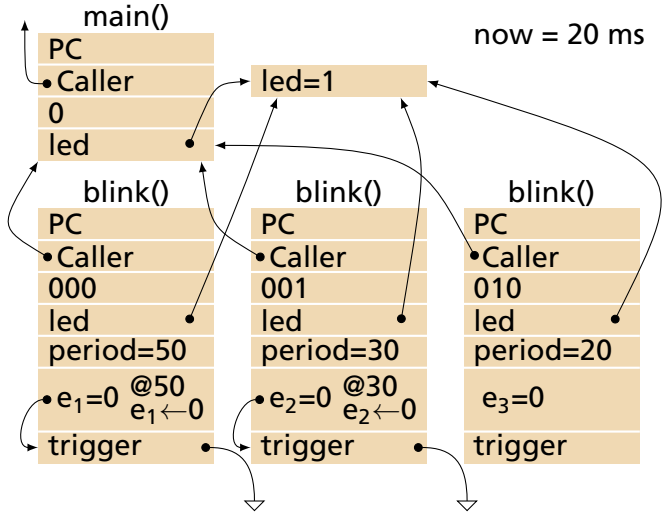
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ @50 $e_1 \leftarrow 0$

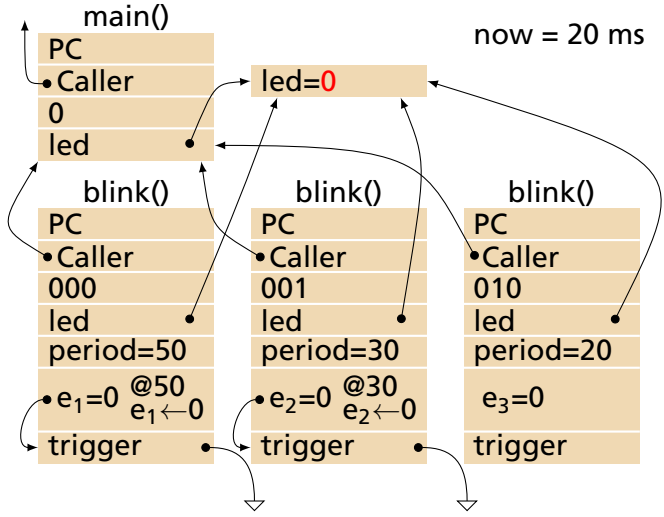
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ @50 $e_1 \leftarrow 0$

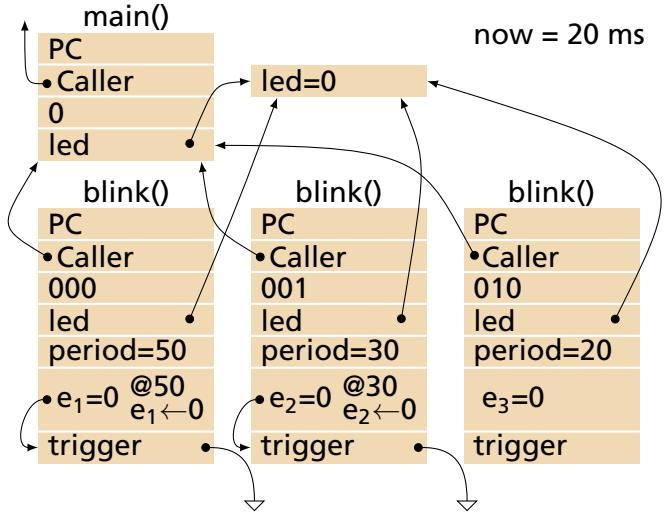
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ @50 $e_1 \leftarrow 0$

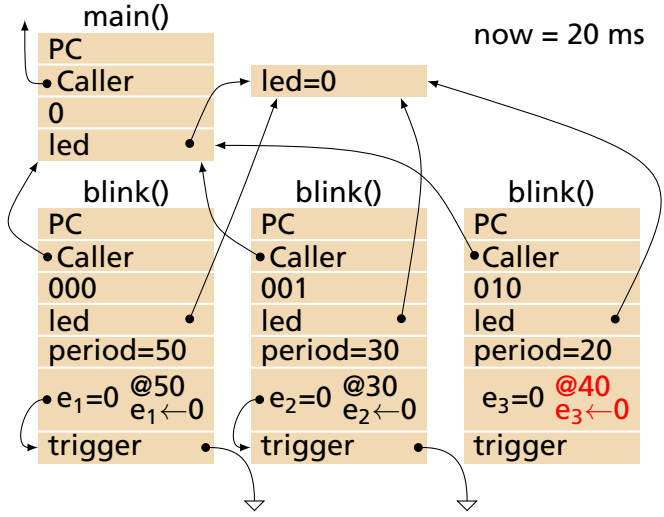
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

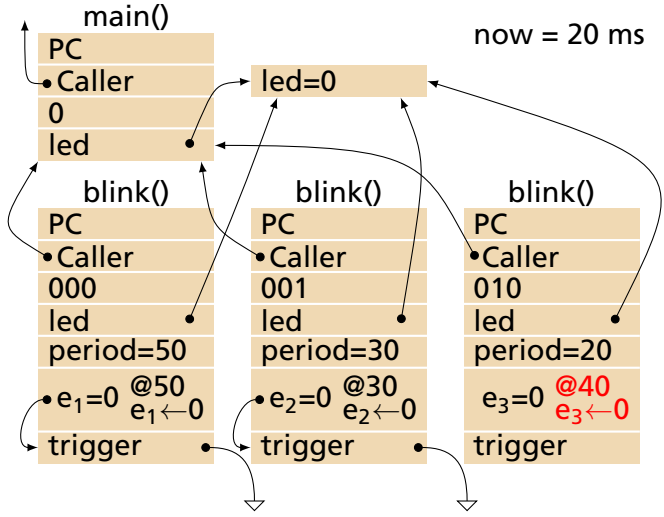
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ | @40 $e_3 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

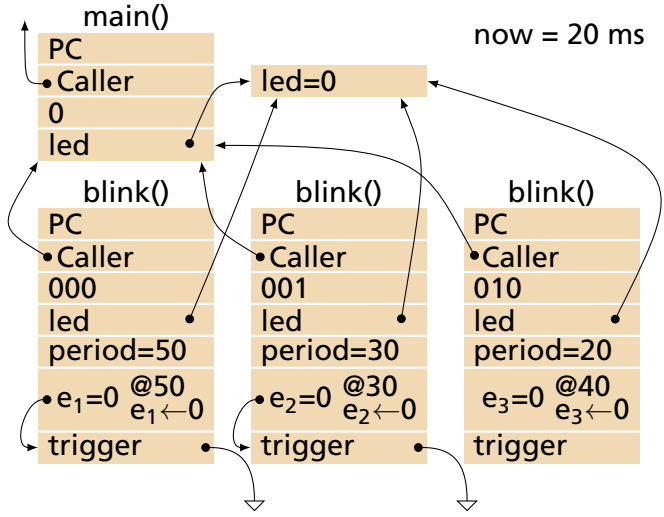
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ | @40 $e_3 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

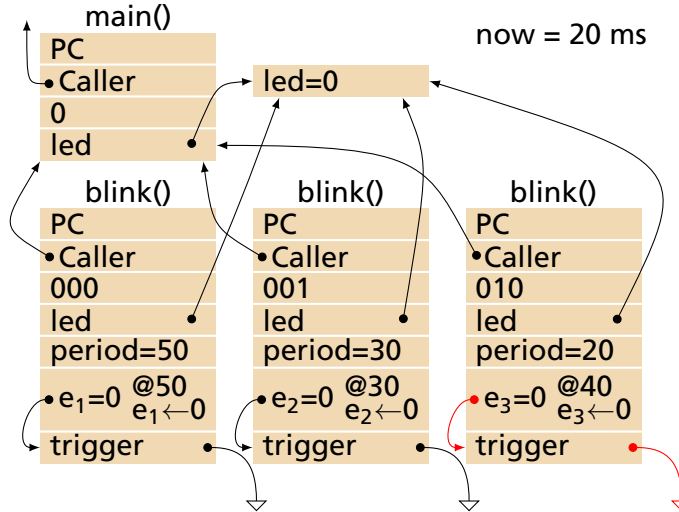
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ | @40 $e_3 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

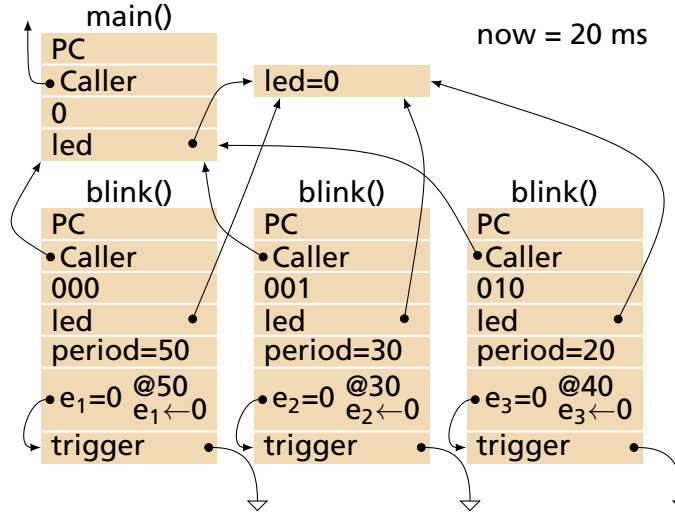
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ | @40 $e_3 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

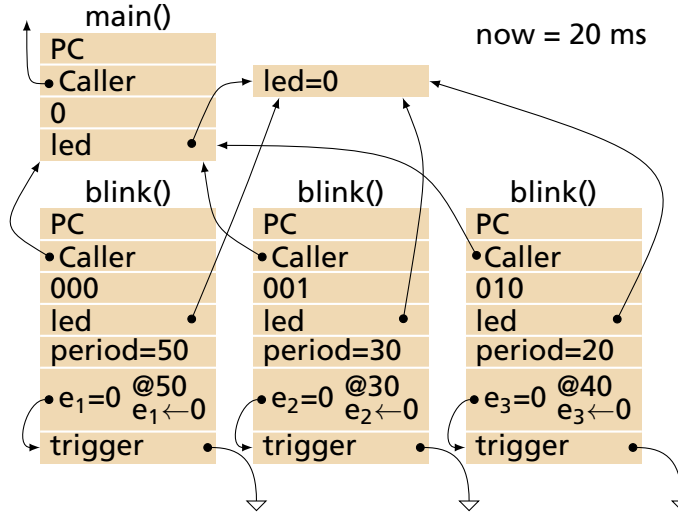
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ | @40 $e_3 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

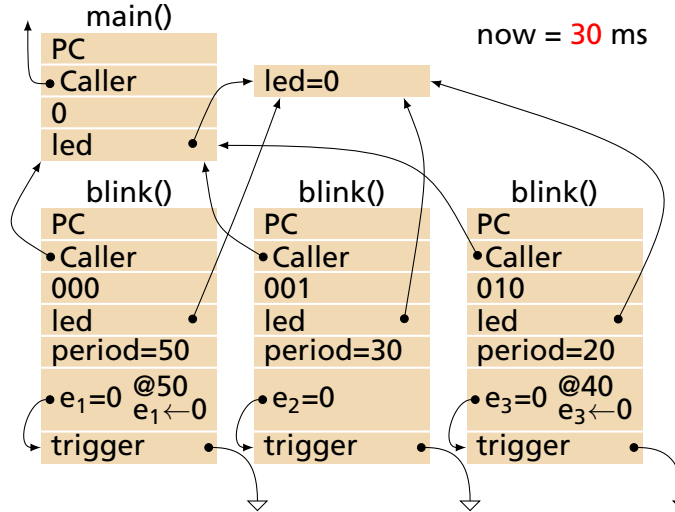
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @30 $e_2 \leftarrow 0$ | @40 $e_3 \leftarrow 0$ | @50 $e_1 \leftarrow 0$

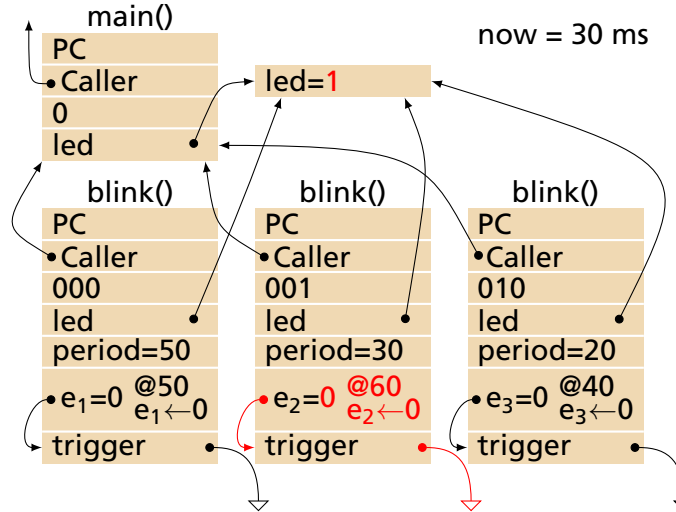
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

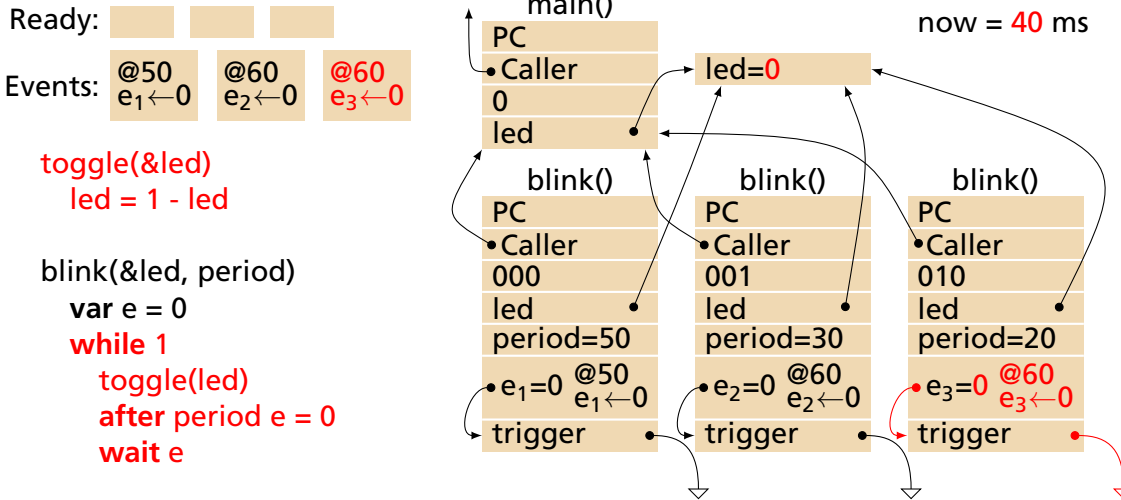


Ready:

Events: @40 $e_3 \leftarrow 0$ | @50 $e_1 \leftarrow 0$ | @60 $e_2 \leftarrow 0$

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

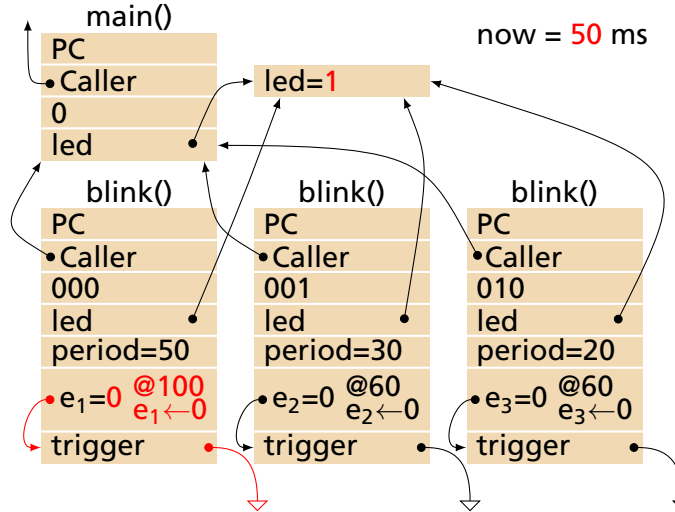



Ready:
Events:
@50 e1←0
@60 e2←0
@60 e3←0
toggle(&led)
led = 1 - led
blink(&led, period)
var e = 0
while 1
toggle(led)
after period e = 0
wait e
main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
main()
PC
Caller
0
led
led=0
now = 40 ms
blink()
PC
Caller
000
led
period=50
e1=0 @50 e1←0
trigger
blink()
PC
Caller
001
led
period=30
e2=0 @60 e2←0
trigger
blink()
PC
Caller
010
led
period=20
e3=0 @60 e3←0
trigger

Ready:

Events: @60 $e_2 \leftarrow 0$ | @60 $e_3 \leftarrow 0$ | @100 $e_1 \leftarrow 0$

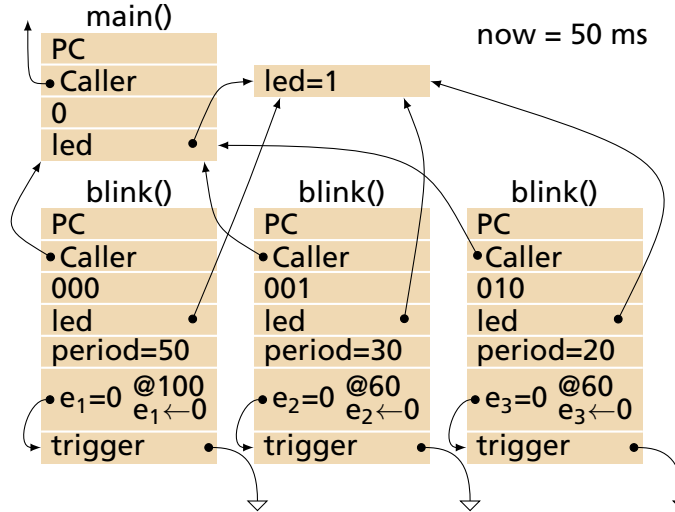
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @60 $e_2 \leftarrow 0$ | @60 $e_3 \leftarrow 0$ | @100 $e_1 \leftarrow 0$

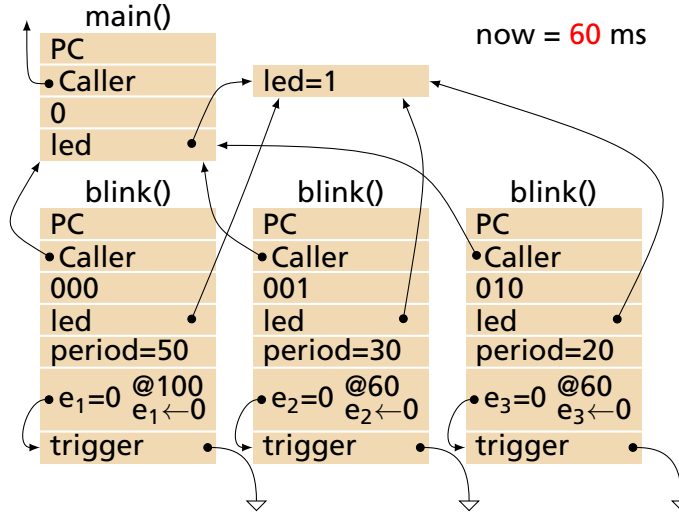
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @60 $e_2 \leftarrow 0$ | @60 $e_3 \leftarrow 0$ | @100 $e_1 \leftarrow 0$

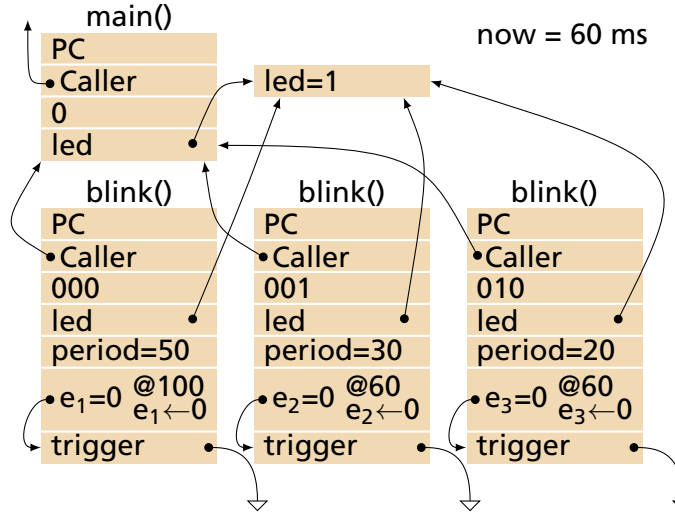
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @60 $e_2 \leftarrow 0$ | @60 $e_3 \leftarrow 0$ | @100 $e_1 \leftarrow 0$

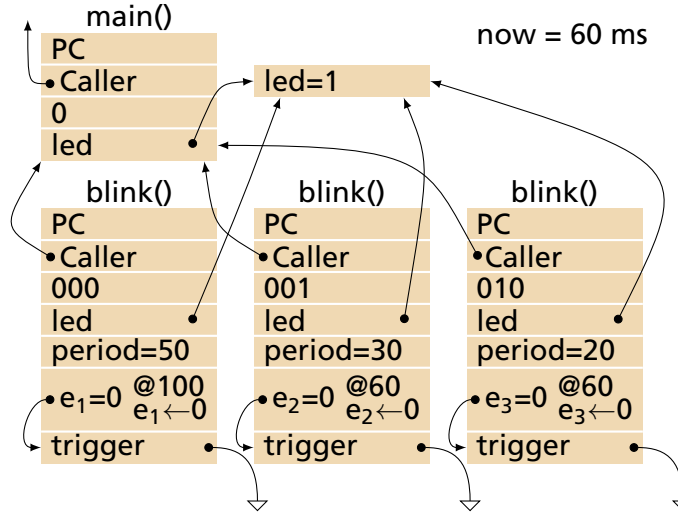
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @60 $e_3 \leftarrow 0$ | @100 $e_1 \leftarrow 0$

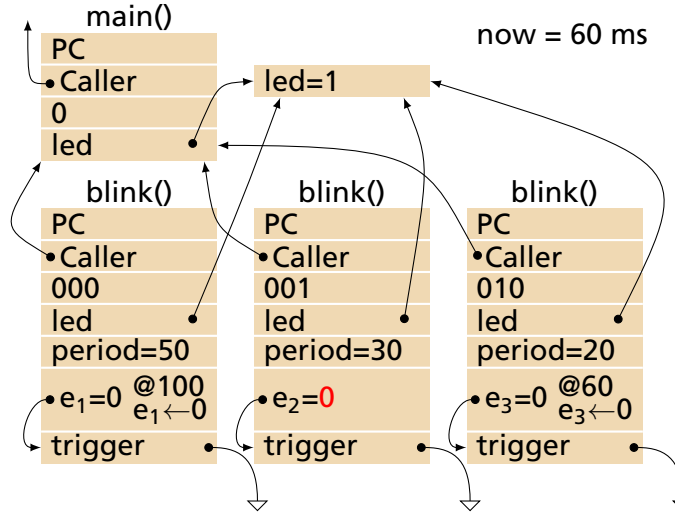
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @60 $e_3 \leftarrow 0$ | @100 $e_1 \leftarrow 0$

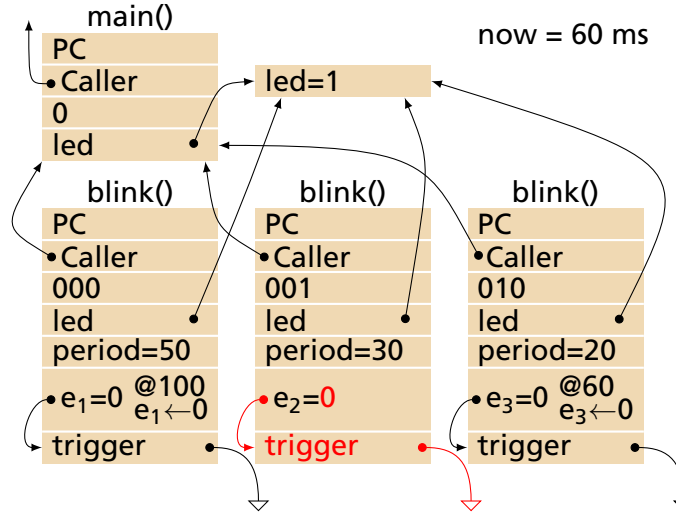
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @60 $e_3 \leftarrow 0$ | @100 $e_1 \leftarrow 0$

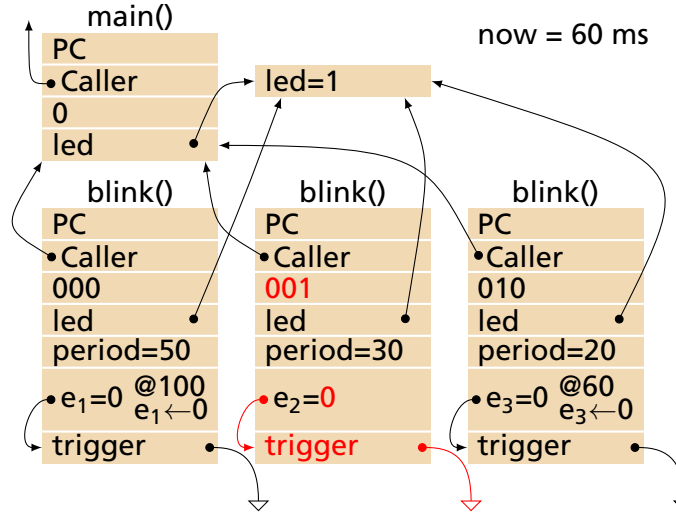
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @60 $e_3 \leftarrow 0$ | @100 $e_1 \leftarrow 0$

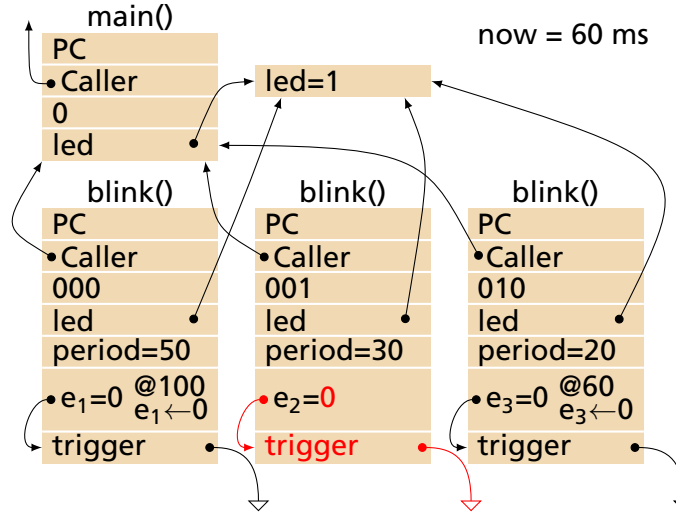
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 001

Events: @60 $e_3 \leftarrow 0$ | @100 $e_1 \leftarrow 0$

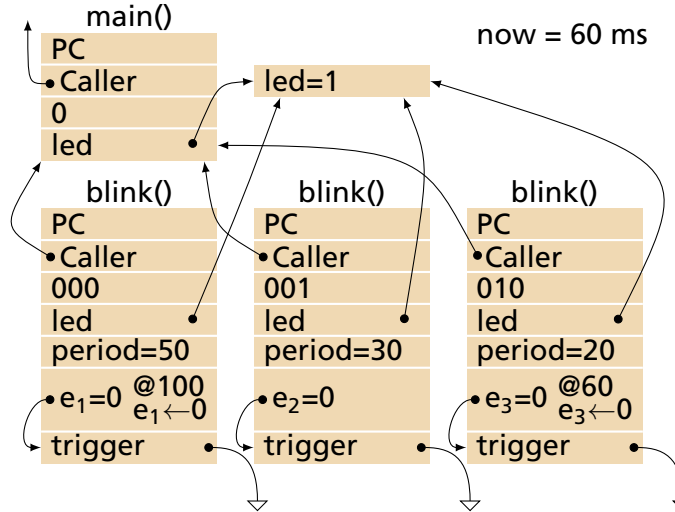
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 001

Events: @100 $e_1 \leftarrow 0$

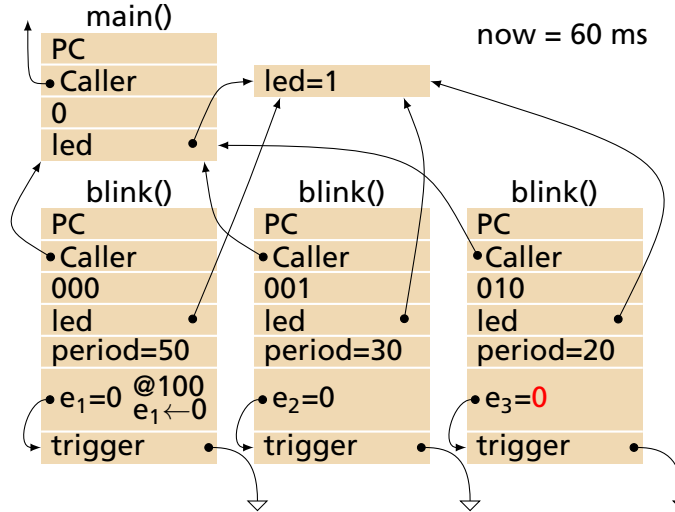
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 001

Events: @100 $e_1 \leftarrow 0$

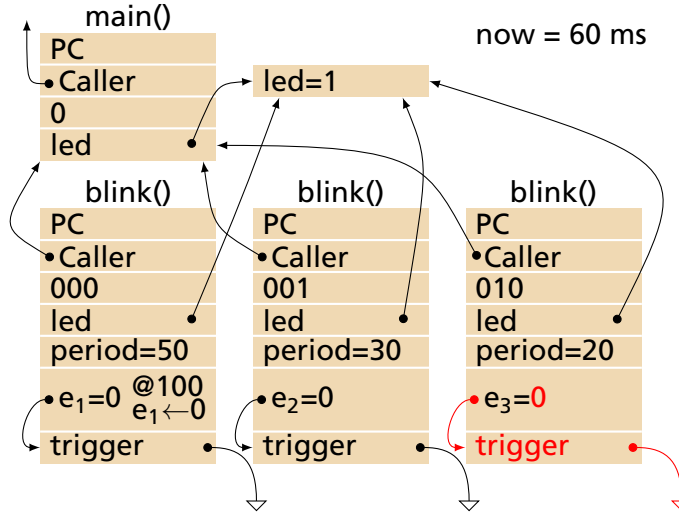
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 001

Events: @100 $e_1 \leftarrow 0$

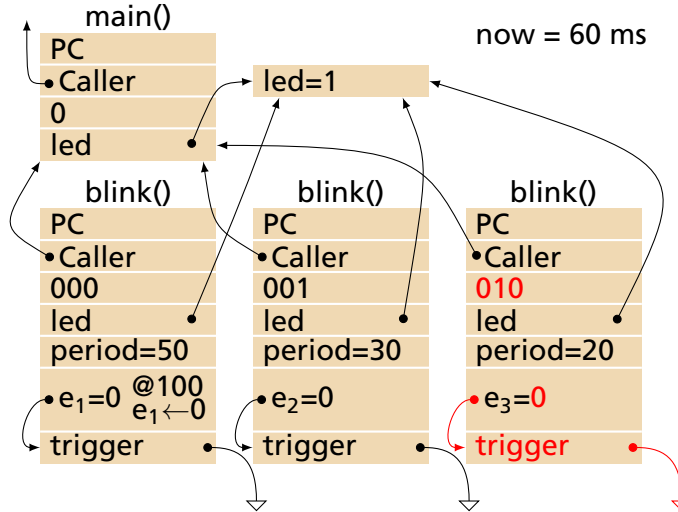
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 001

Events: @100 $e_1 \leftarrow 0$

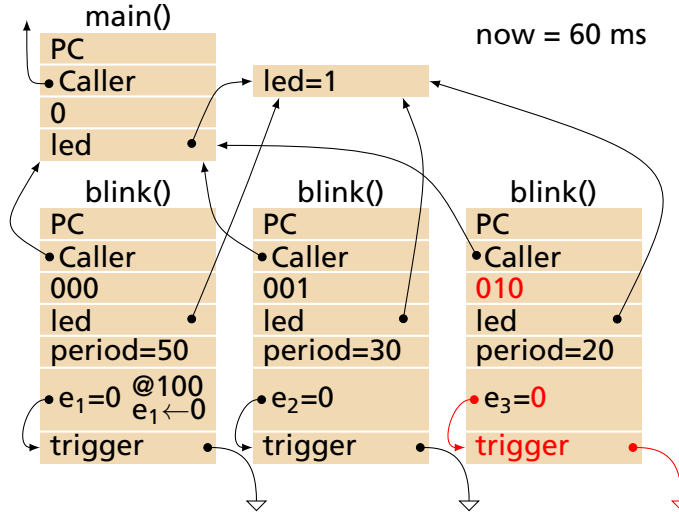
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 001 010

Events: @100 $e_1 \leftarrow 0$

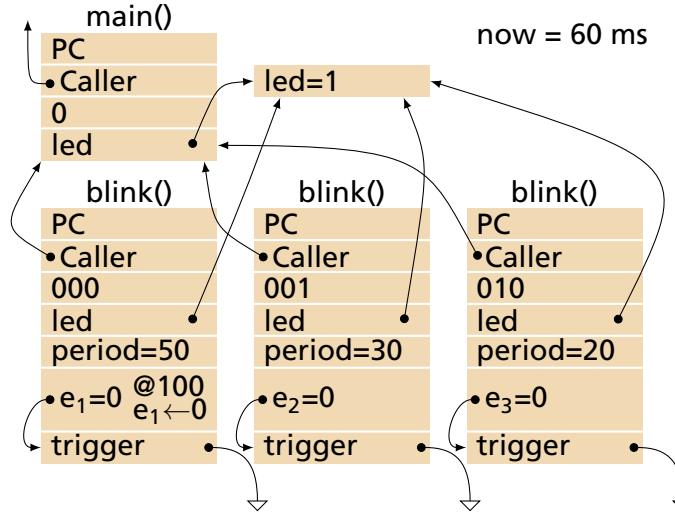
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 001 010

Events: @100 $e_1 \leftarrow 0$

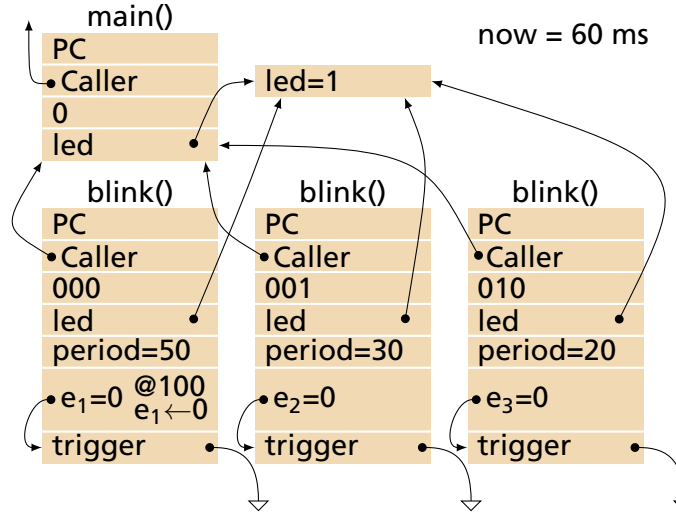
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 001 010

Events: @100 $e_1 \leftarrow 0$

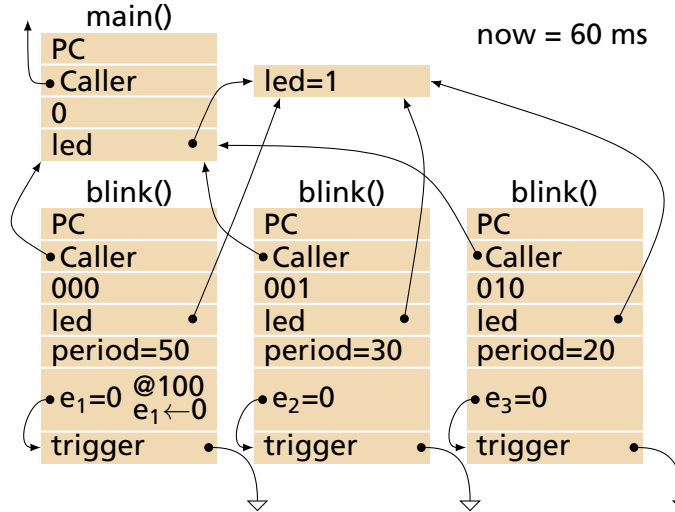
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 010

Events: @100 $e_1 \leftarrow 0$

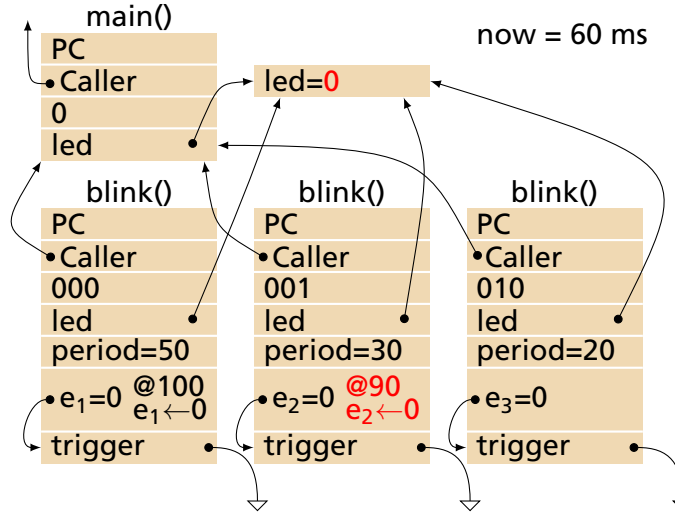
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 010

Events: @90 $e_2 \leftarrow 0$ | @100 $e_1 \leftarrow 0$

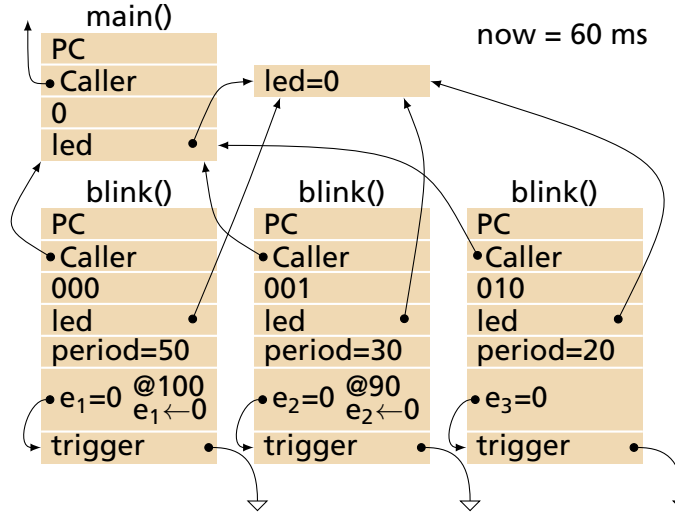
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready: 010

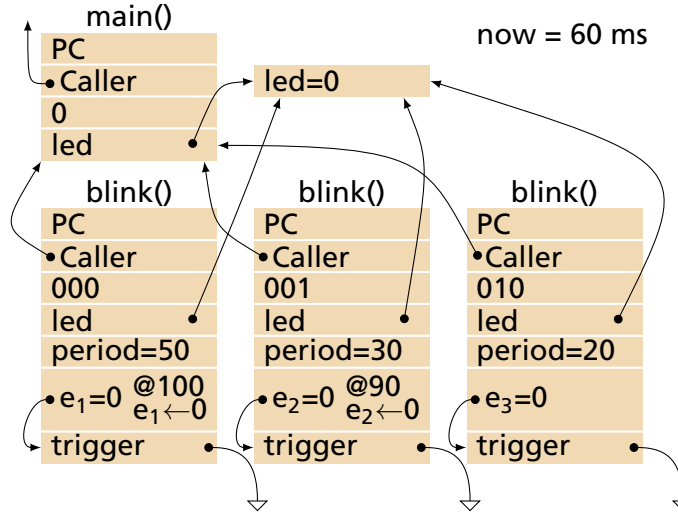
Events: @90 $e_2 \leftarrow 0$ | @100 $e_1 \leftarrow 0$

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @90 $e_2 \leftarrow 0$ | @100 $e_1 \leftarrow 0$

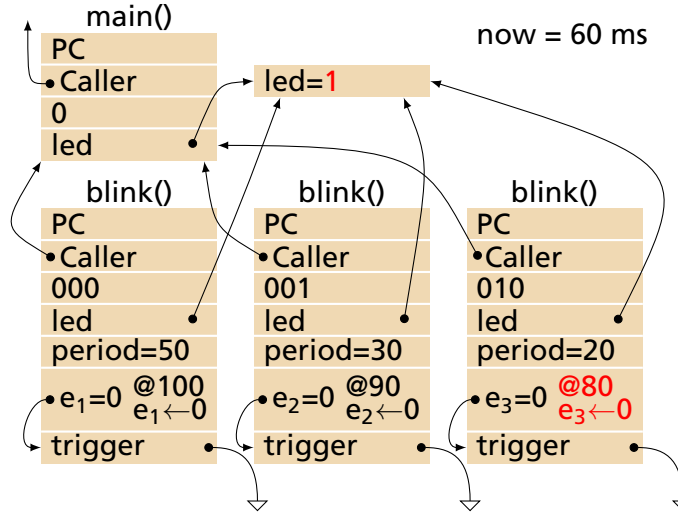



```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

Ready:

Events: @80 $e_3 \leftarrow 0$ | @90 $e_2 \leftarrow 0$ | @100 $e_1 \leftarrow 0$

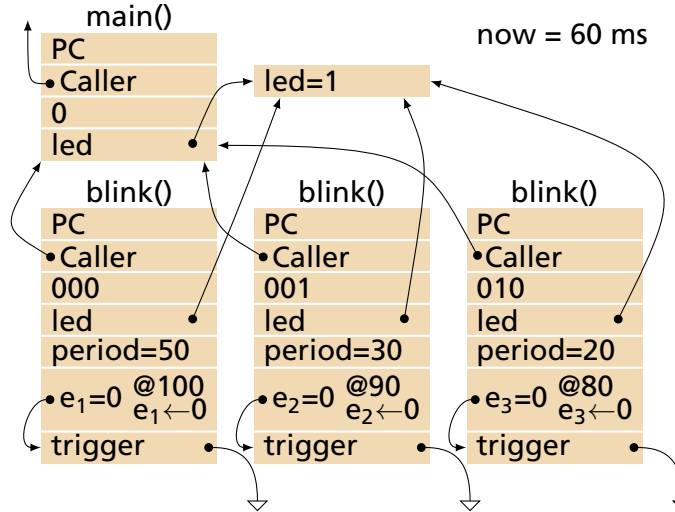
```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



Ready:

Events: @80 $e_3 \leftarrow 0$ | @90 $e_2 \leftarrow 0$ | @100 $e_1 \leftarrow 0$

```
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```



SSM vs. Ptides

[Zhao, Liu, and Lee, RTAS 2007]

SSM vs. Ptides

[Zhao, Liu, and Lee, RTAS 2007]

	SSM	**Ptides**
Between instants	Discrete-event	Discrete-Event

SSM vs. Ptides

[Zhao, Liu, and Lee, RTAS 2007]

	SSM	**Ptides**
Between instants	Discrete-event	Discrete-Event
Within instants	Totally-ordered	Discrete-Event

SSM vs. Ptides

[Zhao, Liu, and Lee, RTAS 2007]

	SSM	**Ptides**
Between instants	Discrete-event	Discrete-Event
Within instants	Totally-ordered	Discrete-Event
Topology	Dynamic, recursive	Static

SSM vs. Ptides

[Zhao, Liu, and Lee, RTAS 2007]

	SSM	Ptides
Between instants	Discrete-event	Discrete-Event
Within instants	Totally-ordered	Discrete-Event
Topology	Dynamic, recursive	Static
Implementation	Single-threaded	Distributed

[Zou Ph.D 2011]

SSM vs. Esterel

[Berry and Gonthier, SCP 1992]

SSM vs. Esterel

[Berry and Gonthier, SCP 1992]

	SSM	**Esterel**
Deterministic	Yes	Yes

SSM vs. Esterel

[Berry and Gonthier, SCP 1992]

	SSM	**Esterel**
Deterministic	Yes	Yes
Time	Sparse	Dense

SSM vs. Esterel

[Berry and Gonthier, SCP 1992]

	SSM	**Esterel**
Deterministic	Yes	Yes
Time	Sparse	Dense
Within instants	Totally-ordered	Constructive

SSM vs. Esterel

[Berry and Gonthier, SCP 1992]

	SSM	**Esterel**
Deterministic	Yes	Yes
Time	Sparse	Dense
Within instants	Totally-ordered	Constructive
Compilation	Separate	Whole-program

SSM vs. Esterel

[Berry and Gonthier, SCP 1992]

	SSM	**Esterel**
Deterministic	Yes	Yes
Time	Sparse	Dense
Within instants	Totally-ordered	Constructive
Compilation	Separate	Whole-program
Runtime	Dynamic Event Queues	Statically Scheduled

SSM vs. Esterel

[Berry and Gonthier, SCP 1992]

	SSM	**Esterel**
Deterministic	Yes	Yes
Time	Sparse	Dense
Within instants	Totally-ordered	Constructive
Compilation	Separate	Whole-program
Runtime	Dynamic Event Queues	Statically Scheduled
Topology	Dynamic, recursive	Static

https://github.com/sedwards-lab/ssm