```
interrupts: off -> on

== Tick 10 ==
        interrupts: on -> off
Time: 10, interrupts off
Pending interrupts:
End of pending interrupts
        interrupts: off -> on
Entering SimpleTestForking thread child1 #0 with
func=0x804a6c8, arg=1, join=NO
        interrupts: on -> off
Putting thread child1 #0 on ready list.
        interrupts: off -> on

== Tick 20 ==
        interrupts: on -> off
Time: 20, interrupts off
Pending interrupts:
End of pending interrupts
        interrupts: off -> on
Forking thread child2 #0 with func=0x804a6c8, arg=2,
join=NO
        interrupts: on -> off
Putting thread child2 #0 on ready list.
        interrupts: off -> on

== Tick 30 ==
        interrupts: on -> off
Time: 30, interrupts off
Pending interrupts:
End of pending interrupts
        interrupts: off -> on
*** thread 0 looped 0 times
        interrupts: on -> off
Yielding thread main #0
Putting thread main #0 on ready list.
Switching from thread main #0 to thread child1 #0
        interrupts: off -> on

== Tick 40 ==
        interrupts: on -> off
Time: 40, interrupts off
Pending interrupts:
End of pending interrupts
        interrupts: off -> on
*** thread 1 looped 0 times
        interrupts: on -> off
```

```
Yielding thread child1 #0
Putting thread child1 #0 on ready list.
Switching from thread child1 #0 to thread child2 #0
        interrupts: off -> on

== Tick 50 ==
        interrupts: on -> off
Time: 50, interrupts off
Pending interrupts:
End of pending interrupts
        interrupts: off -> on
*** thread 2 looped 0 times
        interrupts: on -> off
Yielding thread child2 #0
Putting thread child2 #0 on ready list.
Switching from thread child2 #0 to thread main #0
Now in thread main #0
        interrupts: off -> on

== Tick 60 ==
        interrupts: on -> off
Time: 60, interrupts off
Pending interrupts:
End of pending interrupts
        interrupts: off -> on
*** thread 0 looped 1 times
        interrupts: on -> off
Yielding thread main #0
Putting thread main #0 on ready list.
Switching from thread main #0 to thread child1 #0
Now in thread child1 #0
        interrupts: off -> on

== Tick 70 ==
        interrupts: on -> off
Time: 70, interrupts off
Pending interrupts:
End of pending interrupts
        interrupts: off -> on
*** thread 1 looped 1 times
        interrupts: on -> off
Yielding thread child1 #0
Putting thread child1 #0 on ready list.
Switching from thread child1 #0 to thread child2 #0
Now in thread child2 #0
        interrupts: off -> on
```

```
== Tick 80 ==
        interrupts: on -> off
Time: 80, interrupts off
Pending interrupts:
End of pending interrupts
        interrupts: off -> on
*** thread 2 looped 1 times
        interrupts: on -> off
Yielding thread child2 #0
Putting thread child2 #0 on ready list.
Switching from thread child2 #0 to thread main #0
Now in thread main #0
        interrupts: off -> on

== Tick 90 ==
        interrupts: on -> off
Time: 90, interrupts off
Pending interrupts:
End of pending interrupts
        interrupts: off -> on
*** thread 0 looped 2 times
        interrupts: on -> off
Yielding thread main #0
Putting thread main #0 on ready list.
Switching from thread main #0 to thread child1 #0
Now in thread child1 #0
        interrupts: off -> on

== Tick 100 ==
        interrupts: on -> off
Time: 100, interrupts off
Pending interrupts:
End of pending interrupts
        interrupts: off -> on
*** thread 1 looped 2 times
        interrupts: on -> off
Yielding thread child1 #0
Putting thread child1 #0 on ready list.
Switching from thread child1 #0 to thread child2 #0
Now in thread child2 #0
        interrupts: off -> on

== Tick 110 ==
        interrupts: on -> off
Time: 110, interrupts off
Pending interrupts:
End of pending interrupts
```

```
        interrupts: off -> on
*** thread 2 looped 2 times
        interrupts: on -> off
Yielding thread child2 #0
Putting thread child2 #0 on ready list.
Switching from thread child2 #0 to thread main #0
Now in thread main #0
        interrupts: off -> on

== Tick 120 ==
        interrupts: on -> off
Time: 120, interrupts off
Pending interrupts:
End of pending interrupts
        interrupts: off -> on
        interrupts: on -> off
Finishing thread main #0
Sleeping thread main #0
Switching from thread main #0 to thread child1 #0
Now in thread child1 #0
Deleting thread main #0
        interrupts: off -> on

== Tick 130 ==
        interrupts: on -> off
Time: 130, interrupts off
Pending interrupts:
End of pending interrupts
        interrupts: off -> on
        interrupts: on -> off
Finishing thread child1 #0
Sleeping thread child1 #0
Switching from thread child1 #0 to thread child2 #0
Now in thread child2 #0
Deleting thread child1 #0
        interrupts: off -> on

== Tick 140 ==
        interrupts: on -> off
Time: 140, interrupts off
Pending interrupts:
End of pending interrupts
        interrupts: off -> on
        interrupts: on -> off
Finishing thread child2 #0
Sleeping thread child2 #0
Machine idling; checking for interrupts.
```

```
Time: 140, interrupts off
Pending interrupts:
End of pending interrupts
Machine idle.  No interrupts to do.
No threads ready or runnable, and no pending interrupts.
Assuming the program completed.
Machine halting!

Ticks: total 140, idle 0, system 140, user 0
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 0
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
```