```
global CS
  op CSenter(id: int) {call}    # must be called
  op CSexit()    # may be invoked by call or send
body CS
  process arbitrator
    do true ->
      in CSenter(id) by id ->
          write("user", id, "in its CS at", age())
      ni
      receive CSexit()
    od
  end
end

resource main()
  import CS
  var numusers, rounds: int
  getarg(1, numusers); getarg(2, rounds)

  process user(i := 1 to numusers)
    fa j := 1 to rounds ->
      call CSenter(i)          # enter critical section
      nap(int(random(100)))    # delay up to 100 msec
      send CSexit()            # exit critical section
      nap(int(random(1000)))   # delay up to 1 second
    af
  end
end
```

Figure 8.20 An SR program to simulate critical sections.