mmlsim <- function() {
  initglbls()
  # create simulation
  newsim()
  # get things going, generating and scheduling first arrival event
  arvtime <- rexp(1, rate=arvrate)
  schedevnt(arvtime, arvtype, arvtime)
  mainloop(10000.0)
  return(totwait/njobsdone)
}

# application: M/M/1 queue, arrival rate 0.5, service rate 1.0; we must
# set the globals and the function reactevnt() for this application

# initializes the global variables
initglbls <- function() {
  # globals
  rates
  arvrate <- 0.5 # arrival rate
  srvrate <- 1.0 # service rate
  # event types
  arvtype <- 1 # arrival type
  srvdotype <- 2 # service done type
  # server queue, consisting of arrival times of queued jobs
  srvq <- vector(length=0)
  # statistics
  njobsdone <- 0 # jobs done so far
  totwait <- 0.0 # total wait time so far
}

# application-specific event processing function required by mainloop()

# in the general DES library above
reactevnt <- function(head) {
  if (head == arvtype) {
    if (length(srvq) == 0) {
      # if server free, start service, else add to queue
      srvq <- c(head)
    } else {
      # generate next arrival
      arvtime <- sim$curntime + rexp(1, arvrate)
      schedevnt(arvtime, arvtype, arvtime)
      return(head)
    }
  }
  else {
    # process job that just finished
    # do accounting
    njobsdone <- njobsdone + 1
    totwait <- totwait + sim$curntime - head
    # remove from queue
    srvq <- c(srvq[-1])
    # more still in the queue?
    if (length(srvq) > 0) {
      # schedule new service
      srvdonetim <- sim$curntime + rexp(1, srvrate)
      schedevnt(srvdonetim, srvdotype, srvq[1])
    }
  }
}

mmldes.R  Fri Nov 12 08:41:12 2010  1
if (y < x[hi]) return(hi)
return(hi+1)