```
chan getTask(int worker), task[1:PR](int block1, block2);
chan bodies[1:PR](int worker; point pos[*], vel[*]);
chan forces[1:PR](point force[*]);

process Manager {
  declare and initialize local variables;
  for [time = start to finish by DT] {
    initialize the bag of tasks;
    for [i = 1 to numTasks+PR ] {
      receive getTask(worker);
      select next task; use (0, 0) to signal bag is empty;
      send task[worker](block1, block2);
    }
  }
}

process Worker[w = 1 to PR] {
  point p[1:n], v[1:n], f[1:n];  # position, velocity
  double m[1:n];        # force and mass for each body
  declare other local variables; initialize all local variables;
  for [time = start to finish by DT] {
    while (true) {
      send getTask(w); receive task[w](block1, block2);
      if (block1 == 0) break;  # bag is empty
      calculate forces between bodies in block1 and block2;
    }
    for [i = 1 to PR st i != w]     # exchange forces
      send forces[i](f[*]);
    for [i = 1 to PR st i != w] {
      receive forces[w](tf[*]);
      add values in tf to those in f;
    }
    update p and v for my block of bodies;
    for [i = 1 to PR st i != w]     # exchanges bodies
      send bodies[i](w, p[*], v[*]);
    for [i = 1 to PR st i != w] {
      receive bodies[w](worker, tp[*], tv[*]);
      move bodies of worker from tp and tv to p and v;
    }
    reinitialize f to zeros;
  }
}
```

Figure 11.12 Manager/workers program for the *n*-body problem.