type task = (double left, right, fleft, fright, lrarea);
queue bag(task);  # the bag of tasks
int size;  # number of tasks in bag
int idle = 0;  # number of idle workers
double total = 0.0;  # the total area

compute approximate area from \( a \) to \( b \);
insert task \((a, b, f(a), f(b), area)\) in the bag;
count = 1;

process Worker\([w = 1 \text{ to } PR]\) {
  double left, right, fleft, fright, lrarea;
double mid, fmid, larea, rarea;
while (true) {
  # check for termination
  \( \text{idle++}; \)
  if (idle == n \&\& size == 0) break;
  # get a task from the bag
  \( \text{await (size > 0)} \)
  remove a task from the bag;
  size--; idle--;
  mid = (left+right) / 2;
  fmid = f(mid);
  larea = (fleft+fmid) * (mid-left) / 2;
  rarea = (fmid+fright) * (right-mid) / 2;
  if (abs((larea+rarea) - lrarea) > EPSILON) {
    \( \text{put (left, mid, fleft, fmid, larea) in the bag; } \)
    \( \text{put (mid, right, fmid, fright, rarea) in the bag; } \)
    \( \text{size = size + 2; } \)
  } else
    \( \text{total = total + lrarea; } \)
}
if (w == 1)  # worker 1 prints the result
  printf("the total is %.1f
", total);
}

Figure 3.21  Adaptive quadrature using a bag of tasks.

Copyright © 2000 by Addison Wesley Longman, Inc.