```
double A[1:n,1:n], LU[1:n,1:n];  # assume A initialized
int ps[1:n];                     # pivot row indices

procedure barrier(int id) { ... }  # see Chapter 3

process Worker(w = 1 to PR) {
  double pivot, mult;
  declarations of other local variables, such as a copy of ps;
  for [i = w to n by PR]
    initialize ps and my stripes of LU;
  barrier(w);

  # perform Gaussian elimination with partial pivoting
  for [k = 1 to n-1] {     # iterate down main diagonal
    find maximum pivot element — see text;
    if necessary, swap pivot row and row k, then call barrier(w);
    pivot = LU[ps[k],k];     # get actual value of pivot
    for [i = k+1 to n st (i%PR == 0)] {  # for my stripe
      mult = LU[ps[i],k]/pivot;  # calculate multiplier
      LU[ps[i],k] = mult;        #   and save it
      for [j = k+1 to n]     # eliminate across columns
        LU[ps[i],j] = LU[ps[i],j] - mult*LU[ps[k],j];
    }
    barrier(w);
  }
}
```

Figure 11.18 Outline of shared variable program for LU decomposition.

Copyright © 2000 by Addison Wesley Longman, Inc.