```
1   iter BlockCyclicDom.these(param tag: iterKind) var where tag == iterKind.leader {
3
4      //calculate blockcyclesize
5       var blockcyclesize = blocksize*numLocales;
6
7       //assign loop iterations to locales
8       coforall locDom in locDoms do on locDom {
9
10           //determine the index of the first element in the locDom
11          var start = locDom.myStarts.low;
12          var tasks = here.numCores;
13
14          //each core on a locale can handle its own chunk of work in parallel
15          coforall core in 0..tasks-1 do
16
17               //serialize the division of work in case there are
18               //more elements within a block than there are cores
19              for i in core..blocksize-1 by tasks {
20
21                   yield (start+i)..end by blockcyclesize;
22  }  }          }
```