

```
1  iter BlockCyclicDom.these(param tag: iterKind) var where tag == iterKind.leader {
2
3      //calculate blockcyclesize
4      var blockcyclesize = blocksize*numLocales;
5
6      //assign loop iterations to locales
7      coforall locDom in locDoms do on locDom {
8
9          //determine the index of the first element in the locDom
10         var start = locDom.myStarts.low;
11         var tasks = here.numCores;
12
13         //each core on a locale can handle its own chunk of work in parallel
14         coforall core in 0..tasks-1 do
15
16             //serialize the division of work in case there are
17             //more elements within a block than there are cores
18             for i in core..blocksize-1 by tasks {
19
20                 yield (start+i)..end by blockcyclesize;
21     } }
```