

```

Result: Read two sides of one track
1 ENTRANCE: call readloop();
2 Procedure readloop()
3 |   clear the times of failed to 0,  $si \leftarrow 0$ ;
4 |   call retry();
5 Procedure retry()
6 |   register parameter preparing;
7 |   read a sector;
8 |   if no carry then
9 |       |   call next();
10 |  else
11 |       |   add 1 to  $si$ ,  $si \leftarrow si + 1$ ;
12 |       |   compare  $si$  with 5;
13 |       |   if  $si \geq 5$  then
14 |           |   goto error, FINISHED;
15 |       |   else
16 |           |   reset registers and call retry() to read again;
17 |       |   end
18 |   end
19 Procedure next()
20 |   memory address moved back 0x200;
21 |   add 1 to  $cl$ , preparing for reading the next sector,  $cl \leftarrow cl + 1$ ;
22 |   if  $cl \leq 18$  then
23 |       |   call readloop() to read this sector;
24 |   else
25 |       |    $cl > 18$ , it means that one side of this track is read already;
26 |       |   add 1 to  $dh$ ,  $dh \leftarrow dh + 1$ , reverse the head pointer;
27 |       |   if  $dh < 2$  then
28 |           |   it means the 1 side has not read yet, call readloop();
29 |       |   else
30 |           |   both sides have finished reading, FINISHED;
31 |       |   end
32 |   end

```

Algorithm 1: read two sides of one track