03、死循环

死循环

1、代码

```
package com.vrv.tune;

public class FindDeadWhile {
public static void main(String[] args) throws Inte
rruptedException {
while (true) {
System.out.println(111);
}
}
}
```

2、打包运行通过jps找到pid

java -jar jstackDeadWhile.jar

jps

找到pid

19940

3、获取线程号

ps p 19940 -L -o pcpu,pid,**tid**,time,tname,cmd

4、将获取的线程号（十进制数）转换成十六进制,此处为0x**4de5**

 printf "%x\n" 19941

0x4de5

5、jstack找到死循环

jstack -l 19940

```
    Locked ownable synchronizers:
        - None

"main" prio=10 tid=0x00007f5c70008800 nid=0x4de5 runnable [0x00007f5c7574f000]
    java.lang.Thread.State: RUNNABLE
        at java.io.FileOutputStream.writeBytes(Native Method)
        at java.io.FileOutputStream.write(FileOutputStream.java:345)
        at java.io.BufferedOutputStream.flushBuffer(BufferedOutputStream.java:82)
        at java.io.BufferedOutputStream.flush(BufferedOutputStream.java:140)
        - locked <0x000000061fe0d160> (a java.io.BufferedOutputStream)
        at java.io.PrintStream.write(PrintStream.java:482)
        - locked <0x000000061fe0c6a8> (a java.io.PrintStream)
        at sun.nio.cs.StreamEncoder.writeBytes(StreamEncoder.java:221)
        at sun.nio.cs.StreamEncoder.implFlushBuffer(StreamEncoder.java:291)
        at sun.nio.cs.StreamEncoder.flushBuffer(StreamEncoder.java:104)
        - locked <0x000000061fe0c638> (a java.io.OutputStreamWriter)
        at java.io.OutputStreamWriter.flushBuffer(OutputStreamWriter.java:185)
        at java.io.PrintStream.newLine(PrintStream.java:546)
        - eliminated <0x000000061fe0c6a8> (a java.io.PrintStream)
        at java.io.PrintStream.println(PrintStream.java:737)
        - locked <0x000000061fe0c6a8> (a java.io.PrintStream)
        at com.vrv.tune.FindDeadWhile.main(FindDeadWhile.java:6)

    Locked ownable synchronizers:
        - None
```