

打印线程堆栈的特殊情况

(本文档仅供参考)

✔

使用背景

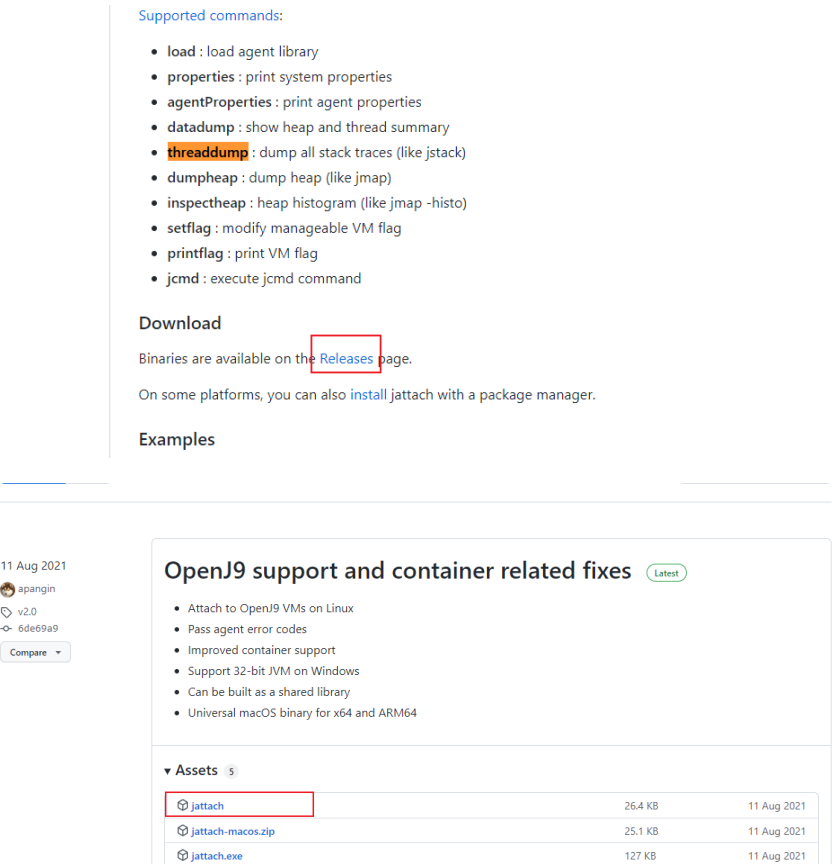
如何在没有jstack和jmap的情况下打印线程和堆？

操作步骤:

一、下载jattach文件

参考资料: <https://github.com/apangin/jattach>

访问以上网址, 依据实际的操作系统下载对应的jattach文件, 比如这里我们下载了Linux的: [jattach](#)



二、上传jattach文件到服务器

将jattach文件上传到服务器任意位置, 并赋予可执行权限`chmod 777 ./jattach`



三、打印线程

`./jattach <pid> threaddump`

```
[root@localhost bin]# ./jattach 19212 threaddump
Connected to remote JVM
JVM response code = 0
2022-07-05 19:01:04
Full thread dump Java HotSpot(TM) 64-Bit Server VM (25.181-b13 mixed mode):

"pool-5-thread-31789" #54426 prio=5 os_prio=0 tid=0x00007f4dc0256000 nid=0x547f waiting on condition [0x00007f4d74db1000]
  java.lang.Thread.State: TIMED_WAITING (parking)
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for <0x00000000c1f942fb> (a java.util.concurrent.SynchronousQueue$TransferStack)
    at java.util.concurrent.locks.LockSupport.parkNanos(LockSupport.java:215)
    at java.util.concurrent.SynchronousQueue$TransferStack.awaitFulfill(SynchronousQueue.java:460)
    at java.util.concurrent.SynchronousQueue$TransferStack.transfer(SynchronousQueue.java:362)
    at java.util.concurrent.SynchronousQueue.poll(SynchronousQueue.java:941)
    at java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1073)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)

"pool-5-thread-31788" #54425 prio=5 os_prio=0 tid=0x00007f4dc024f800 nid=0x547e waiting on condition [0x00007f4d565a4000]
  java.lang.Thread.State: TIMED_WAITING (parking)
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for <0x00000000c1f942fb> (a java.util.concurrent.SynchronousQueue$TransferStack)
    at java.util.concurrent.locks.LockSupport.parkNanos(LockSupport.java:215)
    at java.util.concurrent.SynchronousQueue$TransferStack.awaitFulfill(SynchronousQueue.java:460)
    at java.util.concurrent.SynchronousQueue$TransferStack.transfer(SynchronousQueue.java:362)
    at java.util.concurrent.SynchronousQueue.poll(SynchronousQueue.java:941)
    at java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1073)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)

"smartbi_valid_thread_pool-1888" #54423 prio=5 os_prio=0 tid=0x00007f4dd8206000 nid=0x4b61 waiting on condition [0x00007f4d2b9fb000]
  java.lang.Thread.State: TIMED_WAITING (parking)
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for <0x00000000c1f942fb> (a java.util.concurrent.SynchronousQueue$TransferStack)
```

若需要生成文件到指定路径，可输入./jattach <pid> threaddump >> /路径/文件名.txt

四、打印堆栈

./jattach <pid> dumpheap ./test.hprof

```
[root@localhost bin]# ./jattach 19212 dumpheap ./test.hprof
Connected to remote JVM
JVM response code = 0
Heap dump file created

[root@localhost bin]# ll | grep test
-rwxrwxrwx 1 root root 2040 4月 11 2019 configtest.bat
-rwxrwxrwx 1 root root 1922 4月 11 2019 configtest.sh
-rw----- 1 root root 906101665 7月 5 19:04 test.hprof
[root@localhost bin]#
```