# The Structure of a Thread Dump

**Uriah Levy**
SOFTWARE ENGINEER
@iamuriahl www.medium.com/@iamuriahl

# Overview

**The structure of a thread dump**

**Thread states**

**Native methods**

# The Structure of a Thread Dump

# What's Inside a Thread Dump?

JVM threads

Application threads

Heap report

# Full Thread Dump

**JVM Threads**

```
"Signal Dispatcher" #4 daemon prio=9 os_prio=31 tid=0x00007f8beb00e800 nid=0x4903 waiting on
condition [0x0000000000000000]
   java.lang.Thread.State: RUNNABLE


"C2 CompilerThread0" #5 daemon prio=9 os_prio=31 tid=0x00007f8bea007800 nid=0x4b03 waiting on
condition [0x0000000000000000]
   java.lang.Thread.State: RUNNABLE
```

**Application Threads**

```
"main" #1 prio=5 os_prio=31 tid=0x00007f8bea802000 nid=0x1c03 waiting on condition
[0x000070000d5fa000]
   java.lang.Thread.State: TIMED_WAITING (sleeping)
      at java.lang.Thread.sleep(Native Method)
      at org.uriahl.ajtd.sleepyhelloworld.SleepyHelloWorld.sleep(SleepyHelloWorld.java:15)
      at org.uriahl.ajtd.sleepyhelloworld.SleepyHelloWorld.main(SleepyHelloWorld.java:10)
```

**Heap report**

```
Heap
 PSYoungGen      total 76288K, used 6553K [0x000000076ab00000, 0x0000000770000000,
0x00000007c0000000)
  eden space 65536K, 10% used [0x000000076ab00000,0x000000076b1667c0,0x000000076eb00000)
  from space 10752K, 0% used [0x000000076f580000,0x000000076f580000,0x0000000770000000)
  to   space 10752K, 0% used [0x000000076eb00000,0x000000076eb00000,0x000000076f580000)
 ParOldGen       total 175104K, used 0K [0x00000006c0000000, 0x00000006cab00000,
0x000000076ab00000)
  object space 175104K, 0% used [0x00000006c0000000,0x00000006c0000000,0x00000006cab00000)
 Metaspace       used 3120K, capacity 4494K, committed 4864K, reserved 1056768K
  class space    used 344K, capacity 386K, committed 512K, reserved 1048576K
```

# Full Thread Dump

```
"Signal Dispatcher" #4 daemon prio=9 os_prio=31 tid=0x00007f8beb00e800 nid=0x4903 waiting on condition
[0x0000000000000000]
   java.lang.Thread.State: RUNNABLE

"C2 CompilerThread0" #5 daemon prio=9 os_prio=31 tid=0x00007f8bea007800 nid=0x4b03 waiting on condition
[0x0000000000000000]
   java.lang.Thread.State: RUNNABLE
```

**JVM Threads**

Some of these

```
"main" #1 prio=5 os_prio=31 tid=0x00007f8bea802000 nid=0x1c03 waiting on condition [0x000070000d5fa000]
   java.lang.Thread.State: TIMED_WAITING (sleeping)
      at java.lang.Thread.sleep(Native Method)
      at org.uriahl.ajtd.sleepyhelloworld.SleepyHelloWorld.sleep(SleepyHelloWorld.java:15)
      at org.uriahl.ajtd.sleepyhelloworld.SleepyHelloWorld.main(SleepyHelloWorld.java:10)
```

**Application Threads**

Many of these

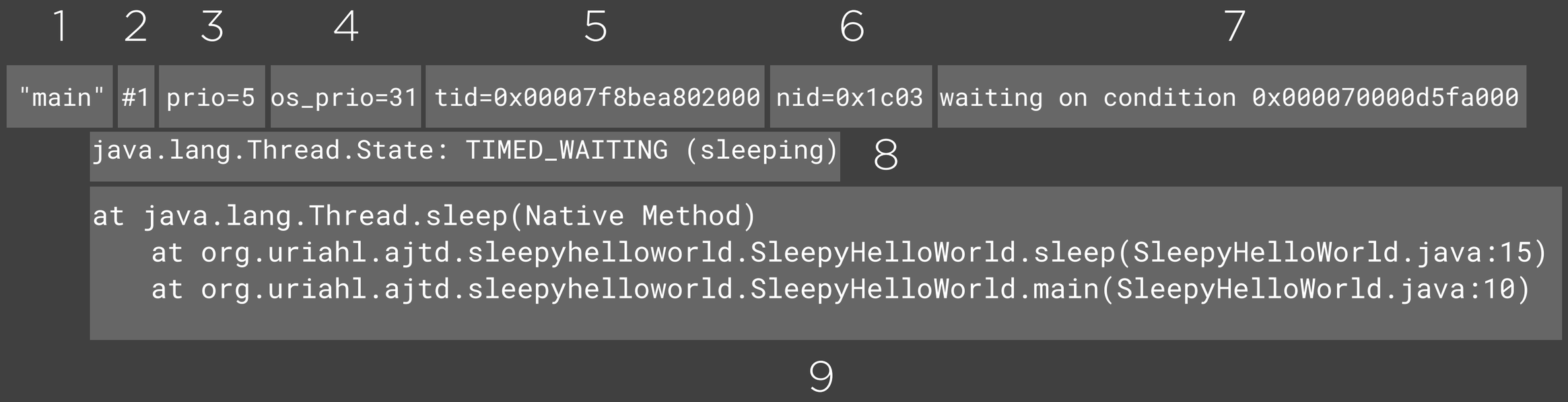
```
Heap
 PSYoungGen      total 76288K, used 6553K [0x000000076ab00000, 0x0000000770000000, 0x00000007c0000000)
  eden space 65536K, 10% used [0x000000076ab00000,0x000000076b1667c0,0x000000076eb00000)
  from space 10752K, 0% used [0x000000076f580000,0x000000076f580000,0x0000000770000000)
  to   space 10752K, 0% used [0x000000076eb00000,0x000000076eb00000,0x000000076f580000)
 ParOldGen       total 175104K, used 0K [0x00000006c0000000, 0x00000006cab00000, 0x000000076ab00000)
  object space 175104K, 0% used [0x00000006c0000000,0x00000006c0000000,0x00000006cab00000)
 Metaspace       used 3120K, capacity 4494K, committed 4864K, reserved 1056768K
  class space    used 344K, capacity 386K, committed 512K, reserved 1048576K
```

**Heap report**

One of these

# Inspecting Individual Threads

```
"main" #1 prio=5 os_prio=31 tid=0x00007f8bea802000 nid=0x1c03 waiting on condition 0x000070000d5fa000
    java.lang.Thread.State: TIMED_WAITING (sleeping)
    at java.lang.Thread.sleep(Native Method)
        at org.uriahl.ajtd.sleepyhelloworld.SleepyHelloWorld.sleep(SleepyHelloWorld.java:15)
        at org.uriahl.ajtd.sleepyhelloworld.SleepyHelloWorld.main(SleepyHelloWorld.java:10)
```


1
2
3
4
5
6
7
"main"
#1
prio=5
os_prio=31
tid=0x00007f8bea802000
nid=0x1c03
waiting on condition 0x000070000d5fa000
java.lang.Thread.State: TIMED_WAITING (sleeping)
8
at java.lang.Thread.sleep(Native Method)
at org.uriahl.ajtd.sleepyhelloworld.SleepyHelloWorld.sleep(SleepyHelloWorld.java:15)
at org.uriahl.ajtd.sleepyhelloworld.SleepyHelloWorld.main(SleepyHelloWorld.java:10)
9

```
"main" #1 prio=5 os_prio=31 tid=0x00007f8bea802000 nid=0x1c03 waiting on condition 0x000070000d5fa000
   java.lang.Thread.State: TIMED_WAITING (sleeping)
   at java.lang.Thread.sleep(Native Method)
      at org.uriahl.ajtd.sleepyhelloworld.SleepyHelloWorld.sleep(SleepyHelloWorld.java:15)
      at org.uriahl.ajtd.sleepyhelloworld.SleepyHelloWorld.main(SleepyHelloWorld.java:10)
```

**Name** Number ID JVM Priority OS Priority Thread Address Native ID Condition

Thread State Thread Call Stack

*The format could vary slightly between platforms

# Thread States

java.lang.Thread.State

# The Six States

RUNNABLE

WAITING

TIMED_WAITING

BLOCKED

NEW

TERMINATED

# RUNNABLE

The thread is actively invoking code [1]

[1] Watch out for Native Methods

# WAITING

The thread will wait indefinitely unless some other thread wakes it up

Object.wait

LockSupport.park

# TIMED_WAITING

Similar to WAITING, but timed

`Thread.sleep`

`LockSupport.parkNanos`

# BLOCKED

The thread is waiting for a monitor lock to become available

java.lang.Thread.State: WAITING / TIMED_WAITING

```
1 public class WaitingStatesDemo {
2    psvm(..) {
3        SomeThread someThread = new ThreadA();
4        someThread.start();
5
6        try {
7            synchronized (someThread) {
8                someThread.wait();
9            }
10        } catch (InterruptedException e) {
11            e.printStackTrace();
12        }
13    }
14 }
```

- Create & start a thread
- Enter the monitor of someThread

```
1 public class WaitingStatesDemo {
2    psvm(..) {
3        SomeThread someThread = new ThreadA();
4        someThread.start();
5
6        try {
7            synchronized (someThread) {
8                someThread.wait();
9            }
10        } catch (InterruptedException e) {
11            e.printStackTrace();
12        }
13    }
14 }
```

◀ Wait for the thread to notify us

```
1 class ThreadA extends Thread {
2    @Override
3    public void run() {
4       synchronized (this) {
5            try {
6                sleep(30000);
7            } catch (InterruptedException e) {
8                e.printStackTrace();
9            }
10        notify();
11     }
12    }
13 }
```

◀Sleep for 30s; some TIMED_WAITING here

```
1 class ThreadA extends Thread {
2    @Override
3    public void run() {
4       synchronized (this) {
5            try {
6                sleep(30000);
7            } catch (InterruptedException e) {
8                e.printStackTrace();
9            }
10       notify();
11     }
12    }
13 }
```

## ◀ Notify waiting threads

*This will wake up the main thread

# “main” Thread Stack

```
1 “main" #1 prio=5 os_prio=31 tid=0x00007faf4d800800 nid=0x1c03 in Object.wait()
2 [0x00007000091db000]
3    java.lang.Thread.State: WAITING (on object monitor)
4  at java.lang.Object.wait(Native Method)
5  - waiting on <0x000000076ac2fb68> (a org.uriahl.ajtd.threadstates.ThreadA)
6  at java.lang.Object.wait(Object.java:502)
7  at org.uriahl.ajtd.threadstates.WaitingStates.main(WaitingStates.java:14)
8  - locked <0x000000076ac2fb68> (a org.uriahl.ajtd.threadstates.ThreadA)
```

# “Thread-0” Thread Stack

```
1 "Thread-0" #11 prio=5 os_prio=31 tid=0x00007ff64200b000 nid=0x5903 waiting on
2 condition [0x0000700003117000]
3    java.lang.Thread.State: TIMED_WAITING (sleeping)
4  at java.lang.Thread.sleep(Native Method)
5  at org.uriahl.ajtd.threadstates.ThreadA.run(WaitingStates.java:28)
6  - locked <0x000000076ac2fb68> (a org.uriahl.ajtd.threadstates.ThreadA)
```

java.lang.Thread.State: BLOCKED

```
1 public class BlockedState {
2    psvm(..) {
3        Foo object = new Foo();
4        // Create 2 threads
5        ThreadB firstThread = new ThreadB(object);
6        ThreadB secondThread = new ThreadB(object);
7        // Start them
8        firstThread.start();
9        secondThread.start();
10    }
11 }
```

## ◀ Create two threads

*Use the same Foo instance with both threads

```java
class Foo {
    synchronized void someSynchronizedMethod() {
        try {
            Thread.sleep(30000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
```

## ◀Sleep 30s

*Sleeping inside this synchronized method means other threads will have to wait at least 30 seconds to lock this method

```
1 class ThreadB extends Thread {
2    private Foo object;
3
4    ThreadB(Foo object) {
5        this.object = object;
6    }
7
8    @Override
9    public void run() {
10        object.someSynchronizedMethod();
11    }
12 }
```

◀ Call Foo’s synchronized method

# Thread-0’s stack

```
1 “Thread-0" #11 prio=5 os_prio=31 tid=0x00007feee99b3000 nid=0x5903 waiting on
2 condition [0x0000700005341000]
3   java.lang.Thread.State: TIMED_WAITING (sleeping)
4  at java.lang.Thread.sleep(Native Method)
5  at org.uriahl.ajtd.threadstates.Foo.someSynchronizedMethod(Foo.java:6)
6  - locked <0x000000076ac2fb20> (a org.uriahl.ajtd.threadstates.Foo)
7  at org.uriahl.ajtd.threadstates.ThreadB.run(BlockedState.java:30)
```

# Thread-1's stack

```
1 "Thread-1" #12 prio=5 os_prio=31 tid=0x00007feee99b3800 nid=0x5b03 waiting for
2 monitor entry [0x0000700005444000]
3   java.lang.Thread.State: BLOCKED (on object monitor)
4  at org.uriahl.ajtd.threadstates.Foo.someSynchronizedMethod(Foo.java:6)
5  - waiting to lock <0x000000076ac2fb20> (a org.uriahl.ajtd.threadstates.Foo)
6  at org.uriahl.ajtd.threadstates.ThreadB.run(BlockedState.java:30)
```

# States Demoed

WAITING

TIMED_WAITING

BLOCKED

# Native Methods

# Platform Dependency

The Java Language:

NOT platform dependent

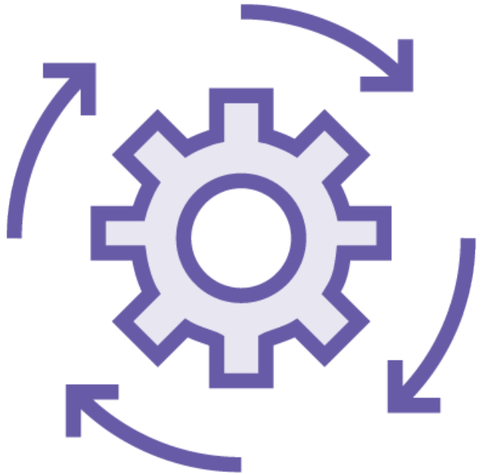
The JVM:

IS platform dependent

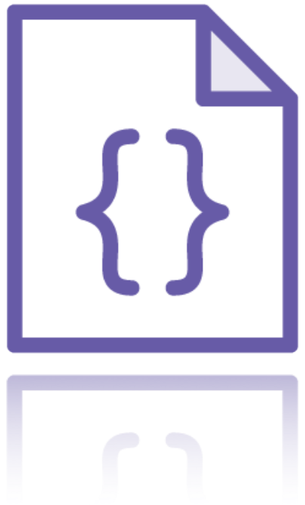
# The JNI

**Java Native Interface:**

**https://en.wikipedia.org/wiki/Java_Native_Interface**



SocketInputStream.c



```
java.lang.Thread.State: RUNNABLE

at java.net.SocketInputStream.socketRead0(Native Method)

at java.net.SocketInputStream.socketRead(SocketInputStream.java:116)

at java.net.SocketInputStream.read(SocketInputStream.java:170)

. . .
```