**Linux下获取java堆栈的方法（无需完整JDK）**

1. 将附件工具包上传到服务器/opt/javatools
2. 解压该文件,并且修改目录属主和权限

unzip  java\_tools\_linux.zip

chmod -R 750 java\_tools\_linux

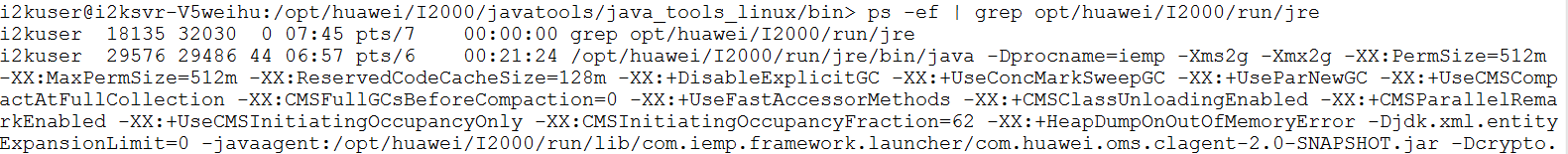
chown -R i2kuser: i2kgrp java\_tools\_linux

1. 环境变量，定义I2000使用的jre位置

su – i2kuser

  export I2K\_JAVA\_HOME=/opt/huawei/I2000/run/jre

1. 获取I2000服务的线程PID

ps -ef | grep opt/huawei/I2000/run/jre找到以下信息

29576 即为I2000的进程ID号

1. 获取堆栈

cd /opt/javatools/java\_tools\_linux/bin

./jstack.sh -l 29576 > stack.txt

(运行时替换为真实的PID，如本例就是29576)

1. 获取镜像

cd /opt/javatools/java\_tools\_linux/bin

./jmap.sh -dump:format=b,file=heap.bin 29576

(运行时替换为真实的PID，如本例就是29576)

1. 将/opt/javatools/java\_tools\_linux/bin/stack.txt即为当前进程的堆栈信息，/opt/javatools/java\_tools\_linux/bin/heap.bin即为当前进程的镜像
2. 当FullGC的次数到达一定上限时，自动导出内存镜像
3. cd /opt/javatools/java\_tools\_linux/bin

./dump.sh PID FGC\_limit dump\_count dump\_over\_sleep

PID为进程号

FGC\_limit为FGC超过多少开始导出镜像的阈值

dump\_count为导出镜像次数上限

dump\_over\_sleep为导出一次镜像后休息时间

实例：./dump.sh 128643 10 3 60

当FGC次数超过10次后，导出进程号为128643的内存镜像，导出完成后休息60s再进行下次检测，最多导出3次镜像后退出（平时检测周期为5s，可以自行调整脚本）

1. 常见问题

**8.1 问题**

**提示：well-known file is not secure**

**提示：Unable to open socket file: target process not responding or HotSpot VM not loadedThe -F option can be used when the target process is not responding**

**Unable to open socket file: target process not responding or HotSpot VM not loaded**

**8.2        修改方法**

**以上两个是常见的报错，是由下面的原因引起的：**

**1)        使用命令的用户和要获取信息的进程的用户不同，导致权限不够，请合理修改工具目录的属主和权限。然后操作。**

**2)        使用的JDK版本过低。请使用export变量修改当前shell的jre运行环境信息。然后操作。**

**说明：**

***jre版本检查办法，可以在当前命令行下执行java –version来识别。默认的JAVATOOLS服务器环境，JAVATOOLS程序运行使用的都是IMAP\_JAVA\_HOME下的jre环境。但是系统是寻找path的，在默认环境中，使用的是/usr/bin，实际是/etc/alternatives下的jre环境，版本是1.4，太低了。***

8.3 如果执行提示如下错误，原因一般为jre和tools.jar版本不配套或缺失类库

Exception in thread "main" java.lang.UnsatisfiedLinkError: no attach in java.library.path

at java.lang.ClassLoader.loadLibrary(Unknown Source)

at java.lang.Runtime.loadLibrary0(Unknown Source)

at java.lang.System.loadLibrary(Unknown Source)

at sun.tools.attach.LinuxVirtualMachine.<clinit>(LinuxVirtualMachine.java:336)

at sun.tools.attach.LinuxAttachProvider.attachVirtualMachine(LinuxAttachProvider.java:63)

at com.sun.tools.attach.VirtualMachine.attach(VirtualMachine.java:213)

at sun.tools.jstack.JStack.runThreadDump(JStack.java:159)

at sun.tools.jstack.JStack.main(JStack.java:112)