(1) 先用pgrep [str] 命令进行模糊匹配，找到匹配该特征串的进程ID；

(2) 其次根据进程ID显示指定的进程信息，ps --pid [pid]；

(3) 因为查找出来的进程ID需要被作为参数传递给ps命令，故使用xargs命令，通过管道符号连接；

(4) 最后显示进程详细信息，需要加上-u参数。

pgrep java | xargs ps -u --pid

1、windows和linux执行jar命令是一样的，java -jar xxx.jar

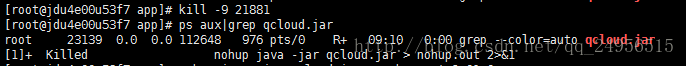
2、使用nohup命令将jar程序设置成后台运行，运行日志输出到nohup.out，关闭窗口无影响

nohup java -jar xxx.jar > nohup.out 2>&1 &  
IMG_256

3. 查看指定jar进程命令

ps aux|grep xxx.jar  
IMG_257

4. 杀死指定jar进程命令

kill -9 进程号  


jps   
列出本机所有的jvm实例

 jps是jdk提供的一个查看当前java进程的小工具， 可以看做是JavaVirtual Machine Process Status Tool的缩写。非常简单实用。

       命令格式：jps [options ] [ hostid ]

       [options]选项 ：  
-q：仅输出VM标识符，不包括classname,jar name,arguments in main method   
-m：输出main method的参数   
-l：输出完全的包名，应用主类名，jar的完全路径名   
-v：输出jvm参数   
-V：输出通过flag文件传递到JVM中的参数(.hotspotrc文件或-XX:Flags=所指定的文件   
-Joption：传递参数到vm,例如:-J-Xms512m

        [hostid]：

[protocol:][[//]hostname][:port][/servername]

        命令的输出格式 ：  
lvmid [ [ classname| JARfilename | "Unknown"] [ arg\* ] [ jvmarg\* ] ]

jstack用于打印出给定的java进程ID或core file或远程调试服务的Java堆栈信息，如果是在64位机器上，需要指定选项"-J-d64"，Windows的jstack使用方式只支持以下的这种方式：

jstack [-l] pid

如果打印出来还不够 还要输出到TXT文件，则使用如下命令

jstack -l 5500 >out.txt

启动脚本脚本

#!/bin/bash

workdir=$(cd $(dirname $0); pwd)

echo "当前路径是：${workdir}"

files=`ls ${workdir} | grep -i .jar | grep -v old.jar`

for file in ${files}

do

echo ${file}

PID=`ps ux | grep -i ${file}| grep -v "grep" | awk {'print $2'}`

echo "=================当前服务进程是：'$PID',准备关闭进程=========="

kill -9 $PID

echo "=================当前进程'$PID'已关闭，准备启动新服务========="

nohup java -jar $file >run.log 2>&1 &

echo "================='$file'服务启动完毕====================="

done