

Get 新技能——通过 Jdb 命令行调试 Java 代码

2019 年 1 月 24 日 星期四 晴

在能力圈努力工作，在兴趣圈尽情尝试

Word hard inside, Learn happy outside

一直习惯于通过 IDE 编程，也老是很想能通过文本编辑器编程。对代码助手功能已经不太感冒了，代码还是要一点一点自己敲的好，复制粘贴模式并不可取。无奈一直卡在调试这个节点上。

通过文本编辑器编写代码，要怎么方便和 IDE 一样进行调试呢？

所幸，今天终于学会了命令行基于 Jdb 进行 Java 代码的调试。

Jdb—Java Debug 工具，以下内容适用于 Windows 平台，Linux 平台仅供参考。

调试主要涉及指令——设置断点(Stop)，取消断点(Clear)，步进(F5/Step)，F6-Next，F8-Cont 继续执行,恢复-Resume

Jdb 有两种调试方式，这篇仅介绍其中一种调试方式，

JVM 要进入调试模式，需要在启动时设置 JVM 参数，其中一种调试设置方式是：

```
set JAVA_OPTS=%JAVA_OPTS%
```

```
-Xdebug
```

```
-Xrunjdwp:transport=dt_shmem,address=8787,server=y,suspend=y
```

transport 指定调试数据的传送方式，有 dt_socket 和 dt_shmem。

Windows 平台仅 dt_shmem 可用;address 指定调试端口

具体使用过程如下：

Step1:编写或使用简单的 Java 程序,下面以一个简单的控制台程序为例

```
public class Demo
{
    public static void main(String[] args)
    {
        int i = 0;

        String future = "happy always";

        System.out.println(i);

        System.out.println(future);
    }
}
```

Step2:javac 编译

Step3:正常情况下使用 java 执行 class 文件,但这里需要临时指定环境变量,因此执行命令如下：

```
java -Xdebug
-Xrunjdwp:transport=dt_shmem,address=8989,server=y,suspend=y
Demo
```

调试监听端口 8989

Step4：连接调试，在另一个控制台执行命令：

```
jdb -attach 8989
```

返回结果类似下面：

=====

设置未捕获的 java.lang.Throwable

设置延迟的未捕获的 java.lang.Throwable

正在初始化 jdb...

>

VM 已启动: 当前调用堆栈上没有帧

=====

表示已正常连接

可以使用 Stop 设置断点, Clear 取消断点, 命令如下:

Stop at Demo;5

Clear Demo;5

run 命令执行, Resume 命令恢复, F5-Step, F6-Next, F8-Cont