twit88.com

New SMS Library at http://www.twit88.com/

Java – Signal Handling

By admin on Feb 6, 2008 in Java, Programming

In my earlier article, I talked about how to do a graceful shutdown your of Java application when Ctr-C, or the termination signal is intercepted. Now I am going to roughly show you how to do it using Java signal handling.

JVM signal handling is slightly different across different platforms, e.g. across Linux, HP-UX and Windows. So the signal handling program you write in Windows may not work in Linux or HP-UX. Unlike the previous article which is using shutdown hook, which is platform independent, if you want to use signal handling in Java, you have to know which signal(s) you want to handle.

E.g. for Unix/Linux based platform the signals are SEGV, ILL, FPE, BUS, SYS, CPU, FSZ, ABRT, INT, TERM, HUP, USR1, QUIT, BREAK, TRAP, PIPE.

For Windows based platform, the signals are SEGV, ILL, FPE, ABRT, INT, TERM, BREAK.

However, to really determine which signal can be handled in your target platform, you have to test it yourself.

Here is a simple program that you can use to test out signal handling in Java.

```
import sun.misc.Signal;
import sun.misc.SignalHandler;

public class SignalHandlerExample implements SignalHandler {

    private SignalHandler oldHandler;

    public static SignalHandler install(String signalName) {
        Signal diagSignal = new Signal(signalName);
        SignalHandlerExample instance = new SignalHandlerExample();
        instance.oldHandler = Signal.handle(diagSignal, instance);
        return instance;
    }

    public void handle(Signal signal) {
        System.out.println("Signal handler called for signal "
              + signal);
        try {

            signalAction(signal);

            // Chain back to previous handler, if one exists
            if (oldHandler != SIG_DFL && oldHandler != SIG_IGN) {
                oldHandler.handle(signal);
            }
        } catch (Exception e) {
            System.out.println("handle|Signal handler
                 failed, reason " + e.getMessage());
            e.printStackTrace();
        }
    }

    public void signalAction(Signal signal) {
        System.out.println("Handling " + signal.getName());
        System.out.println("Just sleep for 5 seconds.");
        try {
            Thread.sleep(5000);
        } catch (InterruptedException e) {
            System.out.println("Interrupted: "
              + e.getMessage());
        }
    }

    public static void main(String[] args) {
        SignalHandlerExample.install("TERM");
        SignalHandlerExample.install("INT");
        SignalHandlerExample.install("ABRT");

        System.out.println("Signal handling example.");
        try {
            Thread.sleep(50000);
        } catch (InterruptedException e) {
            System.out.println("Interrupted: " + e.getMessage());
        }

    }
}
```

This program handles the TERM, INT, and ABRT signals. When it starts up, it will just sleep. During this time, if you press Ctr-C, the INT signal is triggered and **signalAction** is called.

Here is the sample output.

```
C:\>java -cp . SignalHandlerExample
Signal handling example.
Signal handler called for signal SIGINT
Handling INT
Just sleep for 5 seconds.
```