



链滴

# java 调用 shell 命令并获取执行结果

作者: [huihui](#)

原文链接: <https://ld246.com/article/1480413228735>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

使用到Process和Runtime两个类, 返回值通过Process类的getInputStream()方法获取

1. package ark;

2.

3. import java.io.BufferedReader;

4. import java.io.IOException;

5. import java.io.InputStreamReader;

6. import java.util.ArrayList;

7. import java.util.List;

8.

9. public class ReadCmdLine {

10.

```
public static void main(String args[]) {
```

11.

```
    Process process = null;
```

12.

```
    List processList = new ArrayList();
```

13.

```
    try {
```

14.

```
        process = Runtime.getRuntime().exec("ps -aux");
```

15.

```
        BufferedReader input = new BufferedReader(new InputStreamReader(process.getInputStream()));
```

16.

```
        String line = "";
```

17.

```
        while ((line = input.readLine()) != null) {
```

18.

```
        processList.add(line);
19.
    }
20.
    input.close();
21.
} catch (IOException e) {
22.
    e.printStackTrace();
23.
}
24.
25.
    for (String line : processList) {
26.
        System.out.println(line);
27.
    }
28.
}
29. }
```

调用shell脚本，判断是否正常执行，如果正常结束，Process的waitFor()方法返回0

```
1. public static void callShell(String shellString) {
2.
    try {
3.
```

```
Process process = Runtime.getRuntime().exec(shellString);
4.
    int exitValue = process.waitFor();
5.
    if (0 != exitValue) {
6.
        log.error("call shell failed. error code is :" + exitValue);
7.
    }
8.
} catch (Throwable e) {
9.
    log.error("call shell failed. " + e);
10.
}
11. }
```