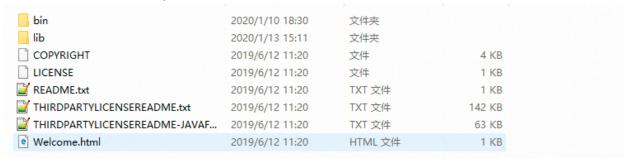
说一下前提:最近写了一个excel导出的程序,使用exe4j将jar转化成exe文件,在运行时需要提供java运行环境,jre1.8有130M,程序只有3M,这肯定忍不了了。jre中的很多东西根部没有用到,所以需要精简一下。

结果: 130M的jre优化到25M, 是不是很可怕。

说一下思路,以jre结构为例:

- 1.优化bin下的.dll文件,去除没有用到的。
- 2.优化lib下使用的.jar文件,去除没有用到的。
- 3.精简lib下rt.jar文件,过滤,重组有用到内容



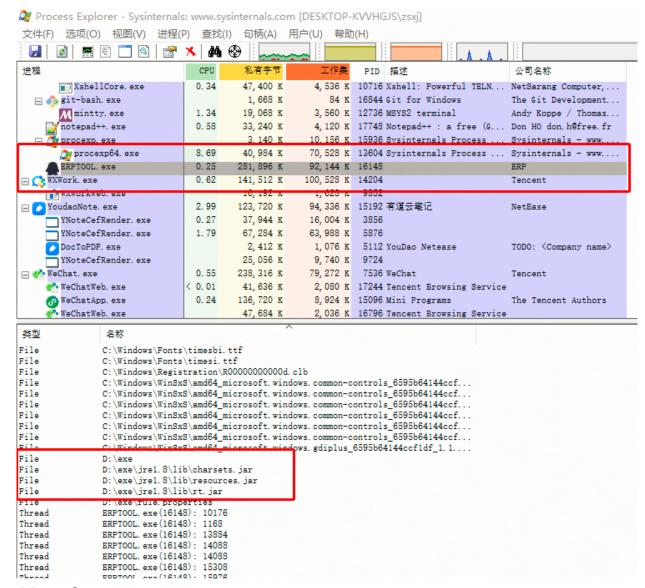
## 第一步

下载process explorer,在进程列表中选中执行的程序,按下ctrl+d,就可以在下方面板中查看到程序运行时所使用的.dll文件,在jre\bin\下移除掉其他无用的.dll文件。(路径排序,找文件更方便)



# 第二步

按下ctrl+h,就可以在下方面板中查看到程序运行时所使用的jar,在jre\lib\下移除掉 其他无用的jar。



# 第三步

## 精简rt.jar比较复杂:

首先,创建批处理(.bat),或者脚本(.sh),把下面的代码粘上去,启动一下jar,上去把所有功能都执行一遍,程序所有需要rt.jar内的文件,都会打印在classes.txt 文件中

```
1 @echo off
2 set path=%cd%\jre1.8\bin
3 java -jar -verbose:class ERPTOOL.jar >> classes.txt
4 pause
```

### 然后,再把下面的程序复制执行一下RTCutter.getJar()方法,需要配置

```
1 RTCutter obj = new RTCutter(
2 "D:/exe/rt",//jre内原始rt解压路径
3 "D:/exe/rt2", //精简后的rt输出路径
4 jarArr);
5 obj.cutRT("D:/exe/classes.txt");//上面脚本输出文件的路径
```

### RTCutter类

```
import java.io.BufferedReader;
2 import java.io.File;
3 import java.io.FileInputStream;
4 import java.io.FileOutputStream;
5 import java.io.FileReader;
6 import java.io.IOException;
7 import java.util.ArrayList;
8 import java.util.List;
                   ******************
***
   * @Title: RTCutter
10
   * @Package com.wdt.main.common.utils
11
   * @Description: 精简JRE
12
   * @author shimingda
13
  * @date 2020/1/13
14
   * @version V1.0
15
   **************************
****/
17 public class RTCutter
   private String source = null; // 类源目录
19
   private String dest = null; // 类拷贝目的目录
20
   String[] jarArr = new String[] {"rt" };
21
22
    public static void getJar(){
23
   String[] jarArr = new String[] {"rt" };
24
   RTCutter obj = new RTCutter(
25
    "D:/exe/rt",
26
    "D:/exe/rt2", jarArr);
27
   obj.cutRT("D:/exe/classes.txt");
28
29
    }
    /***
30
31
   * @param source
32
   * 类源目录
33
   * @param dest
34
   * 类拷贝目的目录
35
36
   * @param jarArr
   * 需要的提取的jar文件
```

```
38
    */
    public RTCutter(String source, String dest, String[] jarArr) {
39
   this.source = source;
40
   this.dest = dest;
41
    this.jarArr = jarArr;
42
43
44
   /***
45
   * 根据classes.txt文件精简rt目录
46
    * @param logName
47
    * 提取class明细
48
    */
49
    public void cutRT(String logName) {
50
    List<String> cList = getClassesListFromFile(logName);
51
    int count = 0; // 用于记录成功拷贝的类数
52
53
    try {
   for(String klass : cList) {
54
    if (copyClass(klass)) {
   count++;
56
    } else {
57
    System.out.println("ERROR " + count + ": " + klass);
58
    }
59
   }
60
   } catch (IOException e) {
61
    System.out.println("ERROR: " + e);
62
63
    }
64
    System.out.println("count: " + count);
    }
65
66
    /***
67
    * 从原jar路径提取相应的类到目标路径,如将java/lang/CharSequence类从rt目录提
取到rt1目录
69
   * @param string
   * 提取类的全路径
71
    * @return
72
    * @throws IOException
73
74
75
   public boolean copyClass(String string) throws IOException {
    if (string.lastIndexOf("/") == -1) {
```

```
77
    return false;
    }
78
    String classDir = string.substring(0, string.lastIndexOf("/"));
79
    String className = string.substring(string.lastIndexOf("/") + 1,
80
    string.length())
81
    + ".class";
82
83
    boolean result = false;
84
85
    for (String jar : jarArr) {
86
    File srcFile = new File(source + "/" + jar + "/" + classDir + "/"
87
88
    + className);
    if (!srcFile.exists()) {
89
    continue;
90
    }
91
92
    byte buf[] = new byte[256];
93
94
    FileInputStream fin = new FileInputStream(srcFile);
95
    /* 目标目录不存在,创建 */
96
    File destDir = new File(dest + "/" + jar + "1/" + classDir);
    if (!destDir.exists()) {
98
    destDir.mkdirs();
99
100
101
    File destFile = new File(destDir + "/" + className);
102
    FileOutputStream fout = new FileOutputStream(destFile);
    int len = 0;
103
    while ((len = fin.read(buf)) != -1) {
104
105
    fout.write(buf, 0, len);
106
    fout.flush();
107
    result = true;
108
    fin.close();
109
    fout.close();
110
111
    break;
112
    }
    return result;
113
     }
114
115
116 /**
```

```
* 将原始的classes.txt转换为类名列表
    * @param source 由-verbose:classes选项生成的classes.txt的路径
118
119
    private List<String> getClassesListFromFile(String source) {
120
    BufferedReader reader = null;
121
    ArrayList<String> classList = new ArrayList<>();
123
    try {
    reader = new BufferedReader(new FileReader(source));
124
    String line = null;
    while((line = reader.readLine()) != null) {
126
    if (line.startsWith("[")) {
127
    if (line.startsWith("[Opened")) {
128
    continue;
129
130
    line = line.replace("[Loaded ", "");
131
    int index = line.indexOf("from");
132
    if (index >= 0) {
133
    line = line.substring(0, index);
134
135
    line = line.trim().replace(".", "/");
136
    classList.add(line);
137
138
139
    } catch (Exception e) {
140
    e.printStackTrace();
141
    } finally {
142
    try {
143
    reader.close();
144
    } catch (Exception e) {
145
146
147
148
    return classList;
149
150
151
152
```

最后,在精简后的rt2文件夹上,执行一下代码,就会打包成一个精简后的rt.jar。把jre/lib下原始的rt替换一下就可以。

```
1 jar cf rt.jar com java javax org sun jdk
```

注:可能会缺少文件,根据路径复制到精简后的文件下,对应路径重新打包就可以exe文件需要复制rt下java/nio, java/util和sun/security/util, sun/util所有内容