

# Linux 上机作业 1 实验报告

## 1. 实验题目

从因特网上搜索 Web 页，用 wget 获取网页，处理网页 html 文本数据，从中提取出当前时间点北京各监测站的 PM2.5 浓度，输出如下 CSV 格式数据：

2021-03-09 13:00:00,海淀区万柳,73

2021-03-09 13:00:00,昌平镇,67

2021-03-09 13:00:00,奥体中心,66

2021-03-09 13:00:00,海淀区万柳,73














2021-03-09 13:00:00,昌平镇,73

2021-03-09 13:00:00,奥体中心,75

## 2. 实验步骤

a) 在如下网址中发现有我们想要的数据：

<http://www.86pm25.com/city/beijing.html>

各监测站点实时数据				
监测站点	AQI	污染等级	PM2.5浓度	PM10浓度
奥体中心	28	 优	19µg/m³	20µg/m³
昌平镇	45	 优	31µg/m³	35µg/m³
大兴旧宫	28	 优	19µg/m³	23µg/m³
定陵(对照点)	28	 优	8µg/m³	11µg/m³
东四	23	 优	16µg/m³	21µg/m³
房山燕山	57	 良	40µg/m³	42µg/m³
丰台小屯	28	 优	19µg/m³	21µg/m³
古城	33	 优	23µg/m³	27µg/m³
官园	26	 优	18µg/m³	21µg/m³
海淀万柳	32	 优	22µg/m³	27µg/m³
怀柔新城	35	 优	24µg/m³	29µg/m³
怀柔镇	33	 优	23µg/m³	30µg/m³
门头沟三家店	32	 优	22µg/m³	26µg/m³
密云新城	24	 优	14µg/m³	16µg/m³
密云镇	24	 优	11µg/m³	17µg/m³
农展馆	26	 优	18µg/m³	18µg/m³
平谷新城	26	 优	10µg/m³	13µg/m³
顺义新城	24	 优	16µg/m³	19µg/m³

b) 打开 putty，在命令行中使用 wget 命令获取这个页面：

```
b285@Ubuntu-bupt:~$ wget http://www.86pm25.com/city/beijing.html
--2022-03-19 13:40:51-- http://www.86pm25.com/city/beijing.html
Resolving www.86pm25.com (www.86pm25.com)... 120.27.42.216
Connecting to www.86pm25.com (www.86pm25.com)|120.27.42.216|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 21817 (21K) [text/html]
Saving to: 'beijing.html.1'

beijing.html.1      100%[=====] 21.31K  --.-KB/s   in 0.01s

2022-03-19 13:40:52 (1.52 MB/s) - 'beijing.html.1' saved [21817/21817]
```

- c) 由于之前已经获取了这个网页，所以这里将文件命名为“beijing.html.1”，将这个文件删除，只保留原来的“beijing.html”文件，现在我们可以使用 cat 命令查看这个文件：

```
0% 空气质量</td></tr></thead>
<tr><td>奥体中心</td><td>28</td><td>
</td><td>19μg/m³</td><td>20μg/m³</td></tr>
<tr><td>昌平镇</td><td>45</td><td> <
</td><td>31μg/m³</td><td>35μg/m³</td></tr>
<tr><td>大兴旧宫</td><td>28</td><td>
</td><td>19μg/m³</td><td>23μg/m³</td></tr>
<tr><td>定陵(对照点)</td><td>28</td><td> </td><td>8μg/m³</td><td>11μg/m³</td></tr>
<tr><td>东四</td><td>23</td><td> </t
d><td>16μg/m³</td><td>21μg/m³</td></tr>
<tr><td>房山燕山</td><td>57</td><td> </td><td>40μg/m³</td><td>42μg/m³</td></tr>
<tr><td>丰台小屯</td><td>28</td><td>
</td><td>19μg/m³</td><td>21μg/m³</td></tr>
```

在这里我们看见了我们想要的监测站点的各个数据，并且以表格方式存储，但是时间不包含在其中，继续查找，我们可以找到时间的位置：

```
<div style="text-align:center">
<h3 style="font-size:14px; font-weight:bold">北京实时空气质量指数</h3></div>
<div class="remark">更新：2022年03月19日 10时</div>
<div style="background:url(http://www.86pm25.com/images/aqi-ruler-140616.png) no
tion: 1px 0">
```

- d) 可以看出，时间放在了 div 的盒子里，而根据这个 div 的类“class=“remark””，因此，再加上之前我们看到的每一个观测点的开头为<tr><td>，故可以创建一个 awk1.txt 文件，将时间行，与观测点行选择出来：

```
b285@Ubuntu-bupt: ~
/class="remark"/ {print}
/^(<tr><td>)/ {print}
```

```
b285@Ubuntu-bupt:~$ cat beijing.html | awk -f awk1.txt
<div class="remark">更新: 2022年03月19日 10时</div>
<tr><td>奥体中心</td><td>28</td><td>
</td><td>19μg/m³</td><td>20μg/m³</td></tr>
<tr><td>昌平镇</td><td>45</td><td> <
</td><td>31μg/m³</td><td>35μg/m³</td></tr>
```

这样我们就看到我们需要的数据已经整合在一起了。

- e) 接下来我们需要去掉所有的标签，因为标签都是使用一对尖括号括起来的，所以可以使用 sed 语句，将尖括号及其内部的内容整体替换为空格：

```
b285@Ubuntu-bupt:~$ cat beijing.html | awk -f awk1.txt | sed 's/<[^<>]*>/ /g'
更新: 2022年03月19日 10时
奥体中心 28      19μg/m³  20μg/m³
昌平镇 45      31μg/m³  35μg/m³
大兴旧宫 28      19μg/m³  23μg/m³
定陵(对照点) 28      8μg/m³  11μg/m³
东四 23      16μg/m³  21μg/m³
房山燕山 57      40μg/m³  42μg/m³
丰台小屯 28      19μg/m³  21μg/m³
```

其中的正则表达式<[^<>]\*>的含义为以<开头，以>结尾，中间任意个不为<>的字符若干个组成的字符串，这样一来可以看到数据清晰了许多。

- f) 下一步可以将时间信息与每一行进行合并，方便我们使用 sed 对每一行进行格式化操作，新建文件 awk2.txt：

```
b285@Ubuntu-bupt: ~
```

```
/2022/ {data = $1; time = $2;}
/g/ {printf("%s %s %s\n", data, time, $0);}
```

执行后得到的输出为：

```
b285@Ubuntu-bupt:~$ cat beijing.html | awk -f awk1.txt | sed 's/<[^<>]*>/ /g' |
awk -f awk2.txt
更新: 2022年03月19日 10时 奥体中心 28      19μg/m³  20μg/m³
更新: 2022年03月19日 10时 昌平镇 45      31μg/m³  35μg/m³
更新: 2022年03月19日 10时 大兴旧宫 28      19μg/m³  23μg/m³
更新: 2022年03月19日 10时 定陵(对照点) 28      8μg/m³  11μg/m³
更新: 2022年03月19日 10时 东四 23      16μg/m³  21μg/m³
更新: 2022年03月19日 10时 房山燕山 57      40μg/m³  42μg/m³
更新: 2022年03月19日 10时 丰台小屯 28      19μg/m³  21μg/m³
```

这样我们就将我们需要的数据放在了同一行中，便于接下来使用 sed 进行每一行字符串的格式化。

- g) 首先需要将每一行最开始的“更新:”这三个字符和污染指数的单位“μg/m³”删除，然后将每一行中各个数据间的空格数量置为 1，最后进行时间的格式化操作，我可以新建一个 sed.txt 文件来存放刚才的这三个步骤：

```

b285@Ubuntu-bupt: ~
s/ */ /g
s/更新: //g
s/μg/m³//g
s/\([0-9][0-9]*\)年\([0-9][0-9]*\)月\([0-9][0-9]*\)日 \([0-9][0-9]*\)时/\1-\2-\3
\4:00:00/g
~
~

```

这样执行后得到了如下输出：

```

b285@Ubuntu-bupt:~$ cat beijing.html | awk -f awk1.txt | sed 's/<[^<>]*>/ /g' |
awk -f awk2.txt | sed -f sed.txt | more
2022-03-19 10:00:00 奥体中心 28 19 20
2022-03-19 10:00:00 昌平镇 45 31 35
2022-03-19 10:00:00 大兴旧宫 28 19 23
2022-03-19 10:00:00 定陵(对照点) 28 8 11
2022-03-19 10:00:00 东四 23 16 21
2022-03-19 10:00:00 房山燕山 57 40 42
2022-03-19 10:00:00 丰台小屯 28 19 21
2022-03-19 10:00:00 古城 33 23 27

```

- h) 再次使用 awk 命令选择我们想要的 pm2.5 的值(由原网站可知第五列数据为 pm2.5 的值)，并在各个属性之间加上逗号，最后重定向到一个 beijing.csv 文件中：

```

2022-03-19 10:00:00,通州东关,17
2022-03-19 10:00:00,万寿西宫,16
2022-03-19 10:00:00,延庆石河营,20
2022-03-19 10:00:00,延庆夏都,23
b285@Ubuntu-bupt:~$ cat beijing.html | awk -f awk1.txt | sed 's/<[^<>]*>/ /g' |
awk -f awk2.txt | sed -f sed.txt | awk '{printf("%s %s,%s,%s\n",$1,$2,$3,$5);}'
> beijing.csv
b285@Ubuntu-bupt:~$ ls
awk1.txt  awk2.txt  beijing.csv  beijing.html  sed.txt

```

- i) 最后我们查看这个文件内容：

```

b285@Ubuntu-bupt:~$ cat beijing.csv
2022-03-19 10:00:00,奥体中心,19
2022-03-19 10:00:00,昌平镇,31
2022-03-19 10:00:00,大兴旧宫,19
2022-03-19 10:00:00,定陵(对照点),8
2022-03-19 10:00:00,东四,16
2022-03-19 10:00:00,房山燕山,40
2022-03-19 10:00:00,丰台小屯,19
2022-03-19 10:00:00,古城,23
2022-03-19 10:00:00,官园,18
2022-03-19 10:00:00,海淀万柳,22
2022-03-19 10:00:00,怀柔新城,24
2022-03-19 10:00:00,怀柔镇,23
2022-03-19 10:00:00,门头沟三家店,22
2022-03-19 10:00:00,密云新城,14
2022-03-19 10:00:00,密云镇,11
2022-03-19 10:00:00,农展馆,18
2022-03-19 10:00:00,平谷新城,10
2022-03-19 10:00:00,顺义新城,16
2022-03-19 10:00:00,天坛,16
2022-03-19 10:00:00,通州东关,17
2022-03-19 10:00:00,万寿西宫,16
2022-03-19 10:00:00,延庆石河营,20
2022-03-19 10:00:00,延庆夏都,23
b285@Ubuntu-bupt:~$

```

### 3. 实验总结

本次实验让我熟悉了 Linux 中正则表达式的操作以及 sed、awk 等命令的使用，最重要的训练了写正则表达式的能力以及处理一段文本时的逻辑顺序，在实验期间也走了不少弯路导致处理数据起来十分繁琐，在几次修改后已经得到简化，然而还有一些问题没有得到解决，比如在写 awk2.txt 文件的时候一开始我将日期设置为变量 data，但是到了最后输出的时候却看不见 data 的值，如图：

```
awk -e '/2022/ {data=$0;}' -e '/g/ {print data, $0}'  
奥体中心 28      19µg/m³  20µg/m³  
昌平镇 45       31µg/m³  35µg/m³  
大兴旧宫 28     19µg/m³  23µg/m³  
定陵(对照点) 28   8µg/m³   11µg/m³  
东四 23        16µg/m³  21µg/m³  
房山燕山 57     40µg/m³  42µg/m³
```

然而将 data 与 \$0 的顺序调换后发现又可以输出了：

```
awk -e '/2022/ {data=$0;}' -e '/g/ {print $0, data}'  
奥体中心 28      19µg/m³  20µg/m³  更新: 2022年03月19日 10时  
昌平镇 45       31µg/m³  35µg/m³  更新: 2022年03月19日 10时  
大兴旧宫 28     19µg/m³  23µg/m³  更新: 2022年03月19日 10时  
定陵(对照点) 28   8µg/m³   11µg/m³  更新: 2022年03月19日 10时  
东四 23        16µg/m³  21µg/m³  更新: 2022年03月19日 10时  
房山燕山 57     40µg/m³  42µg/m³  更新: 2022年03月19日 10时  
丰台小屯 28     19µg/m³  21µg/m³  更新: 2022年03月19日 10时  
古城 33       23µg/m³  27µg/m³  更新: 2022年03月19日 10时
```

最后测试发现第一次不是没有输出日期而是监测点的信息将时间信息覆盖掉了：

```
b285@Ubuntu-bupt:~$ cat beijing.html | awk -f awk1.txt | sed 's/<[^<>]*>/g' |  
awk -f awk2.txt  
奥体中心 28 03月19日 10时  
昌平镇 452年03月19日 10时  
大兴旧宫 28 03月19日 10时  
定陵(对照点) 28 19日 10时  
东四 23022年03月19日 10时  
房山燕山 57 03月19日 10时  
丰台小屯 28 03月19日 10时  
古城 33022年03月19日 10时  
官园 26022年03月19日 10时  
海淀万柳 32 03月19日 10时
```

从结果来看更像是使用变量 data 输出 \$0 的时候光标没有发生移动，导致数据之间覆盖，这个问题还没有解决。

通过本次实验，我认识到了正则表达式在处理字符串的强大之处，如果能更加熟练掌握这个技术想必在之后的学习中发现不少捷径。