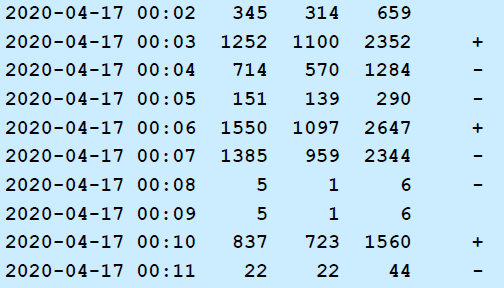
**第一题：生成TCP活动状况报告**

netstat --statistics命令可以列出tcp等协议的统计信息。编写shell脚本程序，每隔1分钟生成1行信息：当前时间；这一分钟内TCP发送了多少报文；接收了多少报文；收发报文总数；行尾给出符号+或-或空格（+表示这分钟收发报文数比上分钟多10包以上，差别在10包或以内用空格，否则用符号-）。运行示例如下：



分析知：

|  |  |
| --- | --- |
| TCP发送报文数 | netstat --statistics | awk '/segments sent out/{printf("%s\n",$1)}' |
| TCP接收报文数 | netstat --statistics | awk '/[0-9] segments received/{printf("%s\n",$1)}' |

**源代码：**

last\_snt=$(netstat --statistics | awk '/segments sent out/{printf("%s\n",$1)}')

last\_rcv=$(netstat --statistics | awk '/[0-9] segments received/{printf("%s\n",$1)}')

last\_tot=0

while true

do

time=$(date +"%F %H:%M")

snt=$(netstat --statistics | awk '/segments sent out/{printf("%s\n",$1)}')

rcv=$(netstat --statistics | awk '/[0-9] segments received/{printf("%s\n",$1)}')

dlt\_snt=$(($snt-$last\_snt))

dlt\_rcv=$(($rcv-$last\_rcv))

tot=$(($dlt\_snt+$dlt\_rcv))

dlt\_tot=$(($tot-$last\_tot))

if [ $dlt\_tot -gt 10 ]; then

ch="+"

elif [ $dlt\_tot -ge -10 ]; then

ch=" "

else

ch="-"

fi

last\_snt=$snt

last\_rcv=$rcv

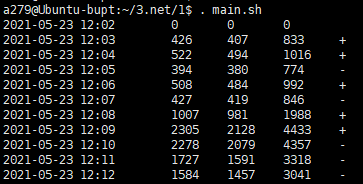
last\_tot=$tot

echo -e "$time\t$dlt\_snt\t$dlt\_rcv\t$tot\t$ch"

sleep 60

done

**实验结果截图：**



**第二题：下载bing图库中图片**

访问https://bing.ioliu.cn/?p=23可以看到bing图库第23页的内容（见下一PPT中的图片），这个Web页有多个图片小样，将鼠标放到某个小样上，如右上角，可见中文说明信息

“野花草甸上的一只欧亚雕鸮，德国莱茵兰-普法尔茨”和日期信息2019-08-03，点击一下，此图片就可以下载。

编写脚本程序bing.sh，将这个图库中照片全部下载下来存放到本地bing目录中，上面URL中p=23可以换成p=1到p=126可访问126个页，每页有12个图，每个图的日期，中文说明信息和下载地址及文件名html文件中可提取。要求下载后的文件命名为“日期 中文说明.jpg”例如：2019-08-03 野花草甸上的一只欧亚雕鸮，德国莱茵兰-普法尔茨.jpg。

要求

◼ 命令行参数

◆ ./bing.sh后面可以跟两个参数，通过指定页号区间限定下载范围，没有参数时页号区间为1-126

◆ 要允许多个程序并发，例如：一个终端上启动 ./bing.sh 1 63，另一个终端上启动./bing.sh 64 126，这样在两个终端上同时下载，以加快任务完成的速度。

◼ 不重复下载已下载的图片

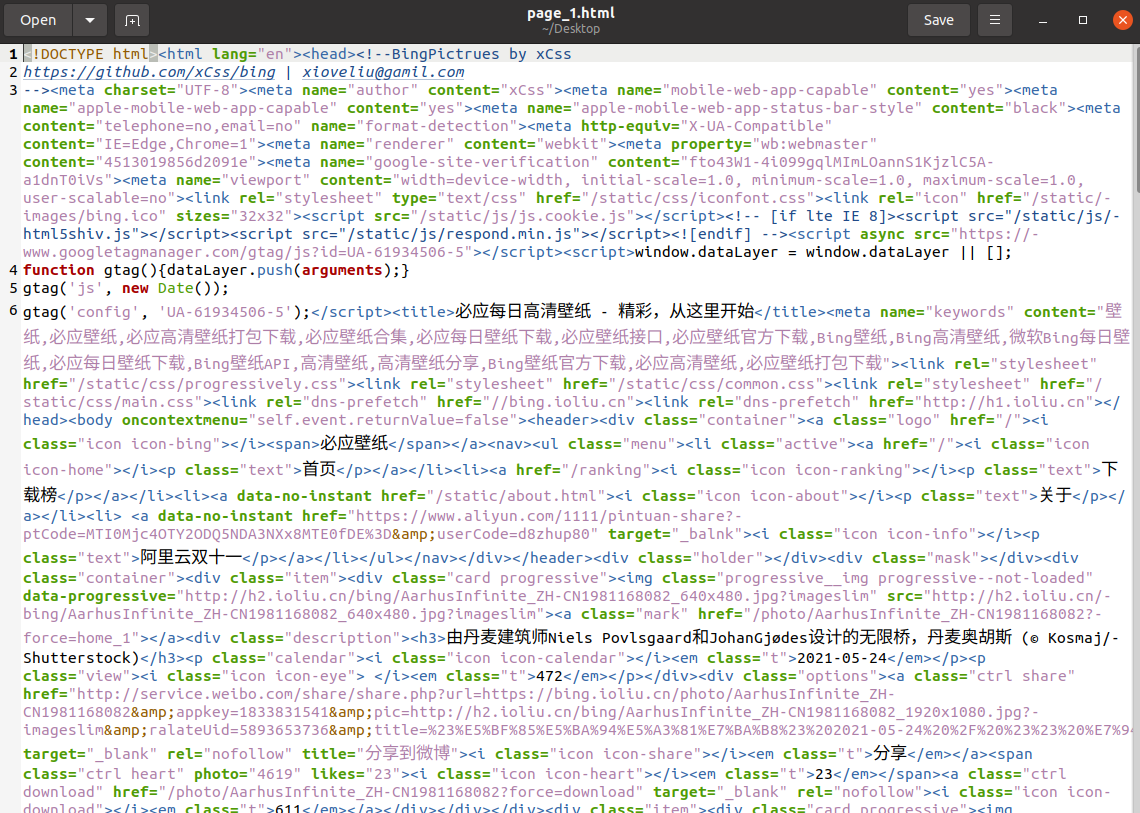
◆ 检查图片是否已下载，如果已下载，则不再下载，这样在一定程度上支持批量任务在被中断后可以从断点继续

◼ 考虑下载文件出现故障的情况

◆ 如果一个图片有5MB，接收1.5MB后网络断开，则残存一个不完整的图片文件。避免这种现象发生的一种方法是，wget下载时使用一个临时文件名。判断wget是否成功 ，若成功则将文件改名为正式名称；若失败，删除临时文件。临时文件名的选取要考虑前述的并发问题，至少不可以固定一个名字导致两进程的争夺。

由于之前尝试在老师给的Linux服务器上使用wget，总是提示403 forbidden，所以这个实验我换到了自己的ubuntu上来做，使用的是ubuntu 20.03 LTS。

首先使用wget -O page\_1.html https://bing.ioliu.cn/?p=1，观察第一页的网页源代码



根据此，进行简单的预处理

|  |  |
| --- | --- |
| 命令 | 含义 |
| cat page\_$i.html | 序号为$i的原始网页 |
| sed -e 's/ //g' | 去除空格 |
| sed -e 's/src="/>\n\n/g' | 在图片地址前换行 |
| sed -e 's/?imageslim">//g' | 去除图片地址后缀 |
| sed -e 's/class="t">/class="t"> /g' | 在日期前添加空格，方便后续分隔 |
| sed -e 's/<h3>/<h3> /g' | 在中文说明前添加空格，方便后续分隔 |
| sed -e 's/<[^<>]\*>//g' | 去除超文本标记符 |
| sed -e 's/([^()]\*)//g' | 去除说明中用”()”包围的作者信息，同样例输出 |
| sed -e 's/©//g' | 去除字符© |
| awk '/^http:/{printf("%s %s %s\n",$1,$2,$3)}' | 选取以”http:”开头的行的1~3列，组成”地址 中文 日期”的格式 |
| > page\_post\_$i.html | 将处理后的信息存储到page\_post\_$i.html文件中 |

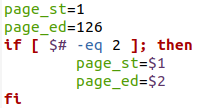
故得到命令

cat page\_$i.html | sed -e 's/ //g' | sed -e 's/src="/>\n\n/g' | sed -e 's/?imageslim">//g' | sed -e 's/class="t">/class="t"> /g' | sed -e 's/<h3>/<h3> /g' | sed -e 's/<[^<>]\*>//g' | sed -e 's/([^()]\*)//g' | sed -e 's/©//g' | awk '/^http:/{printf("%s %s %s\n",$1,$2,$3)}' > page\_post\_$i.html

执行后得到12条记录，如下图，符合预期。

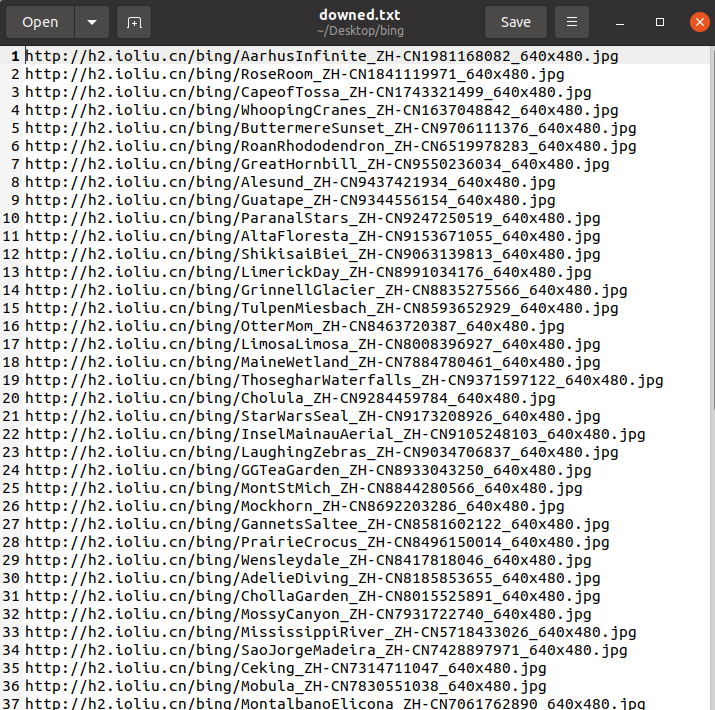


**对于要求——命令行参数：**



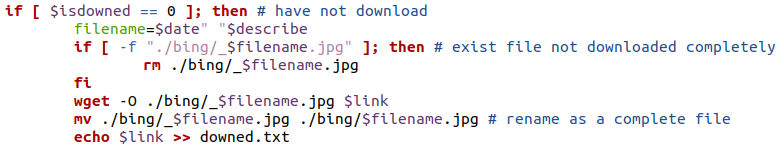
如上，在程序开始时判断输入参数个数，如果有2个参数，则依次将2个变量赋值为输入的2个参数；否则，使用默认的1和126作为变量值。

**对于要求——不重复下载已下载的图片：**



如上图，使用downed.txt这个文件来记录所有已下载的图片的链接，当将要下载的链接位于该文件中时，则跳过。当图片下载完成后，将图片链接加入到该文件中。

**对于要求——考虑下载文件出现故障的情况：**



当使用wget下载图片时，先在其文件名前加上一个”\_”符号，以标识未下载完成。当下载前，会检查bing目录下是否存在同样文件名，但前面有”\_”符号的文件，如果存在，则删除该临时文件，并重新下载。当下载完成后，将其重命名，去除该符号。

**对于选作：**

由于使用的是图片的下载链接来判断是否重复，因而可以做到内容相同时只保留一个文件。

**源代码：**

page\_st=1

page\_ed=126

if [ $# -eq 2 ]; then

page\_st=$1

page\_ed=$2

fi

if [ ! -x "bing" ]; then # if folder "bing" not exist

mkdir bing

fi

if [ ! -f "downed.txt" ]; then # if file "downed.txt" not exist

touch downed.txt

fi

IFS=$'\n\n'

for i in `seq $page\_st $page\_ed`

do

wget -O page\_$i.html https://bing.ioliu.cn/?p=$i

cat page\_$i.html | sed -e 's/ //g' | sed -e 's/src="/>\n\n/g' | sed -e 's/?imageslim">//g' | sed -e 's/class="t">/class="t"> /g' | sed -e 's/<h3>/<h3> /g' | sed -e 's/<[^<>]\*>//g' | sed -e 's/([^()]\*)//g' | sed -e 's/©//g' | awk '/^http:/{printf("%s %s %s\n",$1,$2,$3)}' > page\_post\_$i.html

for record in $(cat page\_post\_$i.html)

do

IFS\_OLD=$IFS

IFS=' '

set -- $record; link=$1;describe=$2;date=$3;

IFS=$IFS\_OLD

isdowned=0

for downlink in $(cat downed.txt) # loop check if downloaded

do

if [ $link == $downlink ]; then

isdowned=1

fi

done

if [ $isdowned == 0 ]; then # have not download

filename=$date" "$describe

if [ -f "./bing/\_$filename.jpg" ]; then # exist file not downloaded completely

rm ./bing/\_$filename.jpg

fi

wget -O ./bing/\_$filename.jpg $link

mv ./bing/\_$filename.jpg ./bing/$filename.jpg # rename as a complete file

echo $link >> downed.txt

fi

done

rm page\_post\_$i.html

rm page\_$i.html

done

**实验结果截图（部分）：**

使用. main.sh 1 5和. main.sh 4 7同时执行，获取前7页内容，获取到的图片如下

