shell 分批处理文件

作者：someone31950
原文链接：https://ld246.com/article/1535615017556
来源网站：链滴

脚本初步原型：具体应用可自行修补
[root@ppzCentos5 Multi_line_read_bin]# cat Multi_line_read.sh
#!/bin/bash
##如果没有参数直接退出
if [! -n "$1" -o ! -n "$2"]
then
echo "USAGE:$0 filename num"
exit
fi
##获取文件行数
MultiLines=<code>cat $1|wc -l</code> 1>&2 >/dev/null
#如何文件总行数和每次处理行数相等
if ["$MultiLines" -eq "$2"]
then
line_start=1
line_end=${MultiLines}
#echo {line_start}"-"{line_end}
#处理数据
sed -n "{line_start},{line_end}p" $1>./split/<c
de>date +%N</code> current_line.txt
exit
fi
##求文件行数和每次处理行数的余数
Remainder=(({MultiLines}%$2))
##文件行数减余数
reduce_Remainder=(({MultiLines}-${Remainder})

#参数 2 减 1
mtl=$(($2-1))
["$mtl" -eq 0]&&{
mtl=1
}
#先处理总行数减余数后的部分
<p>for((i=1;i<{reduce_Remainder};i+={mtl})){</p>
if [$i -gt 1]
then
((i++))
line_start=$i
line_end=(({i}+${mtl}))
else
line_start=$i
line_end=(({i}+${mtl}))
fi
#echo {line_start}"-"{line_end}
#处理数据
sed -n "{line_start},{line_end}p" $1>./split/<c
de>date +%N</code> current_line.txt
}
#总行数和每次处理行数除不尽，则处理余数部分
if ["MultiLines" -gt "reduce_Remainder"]
then
line_start=(({reduce_Remainder}+1))
line_end=${MultiLines}
#echo {line_start}"-"{line_end}

#处理数据
sed -n "{line_start},{line_end}p" $1>./split/<c
de>date +%N</code> current_line.txt
fi
处理 mongodb 数据的：把公共部分写成函数
[root@ppzCentos5 Multi_line_read_bin]# cat Multi_line_Contrast_id.sh
#!/bin/bash
##如果没有参数直接退出
if [! -n "$1" -o ! -n "$2"]
then
echo "USAGE:$0 filename num"
exit
fi
#############设置数据库连接信息
cmd=/root/mongodb3/bin/mongo
ip=192.168.179.132
port=27011
dbname=image
collection=fs.files
#############定义目及相关文件
dir=<code>pwd</code>
content={dir}/{collection}_content.txt
success={dir}/{collection}_success.log
fail={dir}/{collection}_fail.log
runtime={dir}/{collection}_runtime.log
current_lines={dir}/{collection}_current_lines.txt<
p>
file_ObjectId_1=${dir}/file_ObjectId_1
file_ObjectId_2=${dir}/file_ObjectId_2
#把 id 列表转换成 ObjectId 类型
sed -n 's#^"#ObjectId("#gp' $1 1>&2 >${file_ObjectId_1}
sed -n 's#">#"),#gp' {file_ObjectId_1} 1>&
>${file_ObjectId_2}
##获取文件行数
MultiLines=<code>cat ${file_ObjectId_2}|wc -l</code> 1>&2 >/dev/null
#############如果日志文件存在且不为空先清空
if [-s >success -o -s fail]
then
echo>$success
echo>$fail
fi
##定义公用函数
qureyHandle(){
sed -n "{line_start},{line_end}p" $1 1>&2
>$current_lines
>cmd ip:{port}/</
pan>dbname --quiet --eval "db.{collection}.find({_id:{\$in:[`cat
current_lines`]}},{_id:1,filename:1})" 1>&2 >$content
if [>? -eq 0 -a -s content]
then
echo "<code>cat $content</code>">>$success
else
echo "<code>cat $content</code>">>$fail
fi

```
<p>}</p>
<p>echo "开始时间:"<code>date +%F" "%T</code>&gt;$runtime</p>
<p>#如何文件总行数和每次处理行数相等</p>
<p>if [ "$MultiLines" -eq "$2" ]</p>
<p>then</p>
<p>line_start=1</p>
<p>line_end=${MultiLines}</p>
<p>#echo <span class="language-math">{line_start}"-"</span>{line_end}</p>
<p>#处理数据</p>
<p>qureyHandle ${file_ObjectId_2}</p>
<p>echo "结束时间:"<code>date +%F" "%T</code>&gt;&gt;$runtime</p>
<p>#处理完退出</p>
<p>exit</p>
<p>fi</p>
<p>##求文件行数和每次处理行数的余数</p>
<p>Remainder=<span class="language-math">((</span>{MultiLines}%$2))</p>
<p>##文件行数减余数</p>
<p>reduce_Remainder=<span class="language-math">((</span>{MultiLines}-${Remainder})
</p>
<p>#参数 2 减 1</p>
<p>mtl=$(($2-1))</p>
<p>[ "$mtl" -eq 0 ]&amp;amp;&amp;amp;{</p>
<p>mtl=1</p>
<p>}</p>
<p>#先处理总行数减余数后的部分</p>
<p>for((i=1;i&lt;<span class="language-math">{reduce_Remainder};i+=</span>{mtl})){</p>
<p>if [ $i -gt 1 ]</p>
<p>then</p>
<p>((i++))</p>
<p>line_start=$i</p>
<p>line_end=<span class="language-math">((</span>{i}+${mtl}))</p>
<p>else</p>
<p>line_start=$i</p>
<p>line_end=<span class="language-math">((</span>{i}+${mtl}))</p>
<p>fi</p>
<p>#echo <span class="language-math">{line_start}"-"</span>{line_end}</p>
<p>#处理数据</p>
<p>qureyHandle ${file_ObjectId_2}</p>
<p>}</p>
<p>#总行数和每次处理行数除不尽，则处理余数部分</p>
<p>if [ "<span class="language-math">MultiLines" -gt "</span>reduce_Remainder" ]</p>
<p>then</p>
<p>line_start=<span class="language-math">((</span>{reduce_Remainder}+1))</p>
<p>line_end=${MultiLines}</p>
<p>#echo <span class="language-math">{line_start}"-"</span>{line_end}</p>
<p>#处理数据</p>
<p>qureyHandle ${file_ObjectId_2}</p>
<p>fi</p>
<p>#处理查询到的结果</p>
<p>awk -F[()] '{print $2}' <span class="language-math">success&gt;</span>{success}_result
/p>
<p>awk '{print $0}' $1 <span class="language-math">{success}_result|sort|uniq -u&gt;</spa
>{dir}/<span class="language-math">{1}_</span>{success}_notexist</p>
<p>echo "结束时间:"<code>date +%F" "%T</code>&gt;&gt;$runtime</p>
```

[root@ppzCentos5 Multi_line_read_bin]