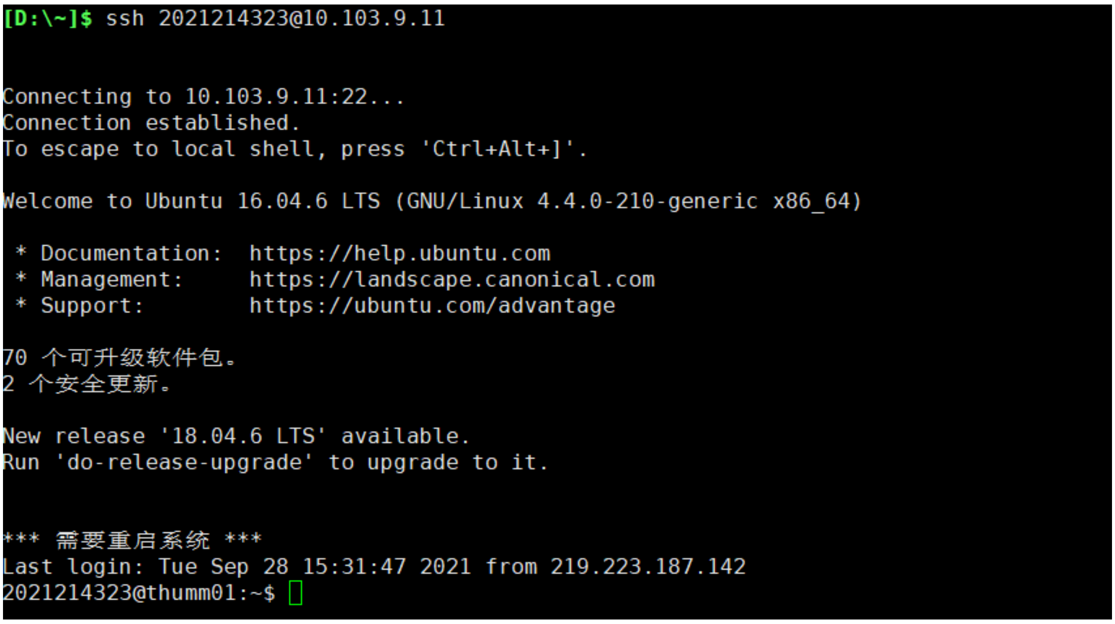
任务一:

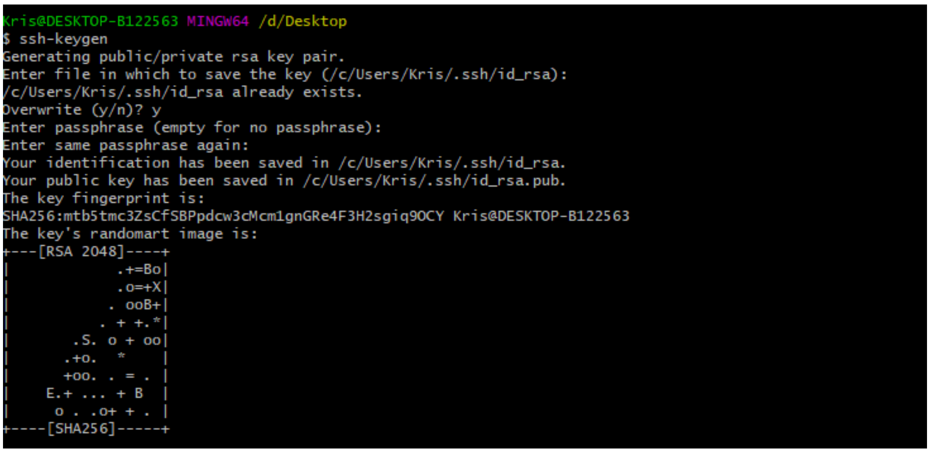


```
[D:\~]$ ssh 2021214323@10.103.9.11


Connecting to 10.103.9.11:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.

Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-210-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

70 个可升级软件包。
2 个安全更新。

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


*** 需要重启系统 ***
Last login: Tue Sep 28 15:31:47 2021 from 219.223.187.142
2021214323@thumm01:~$ 
```

输入指令，登录成功

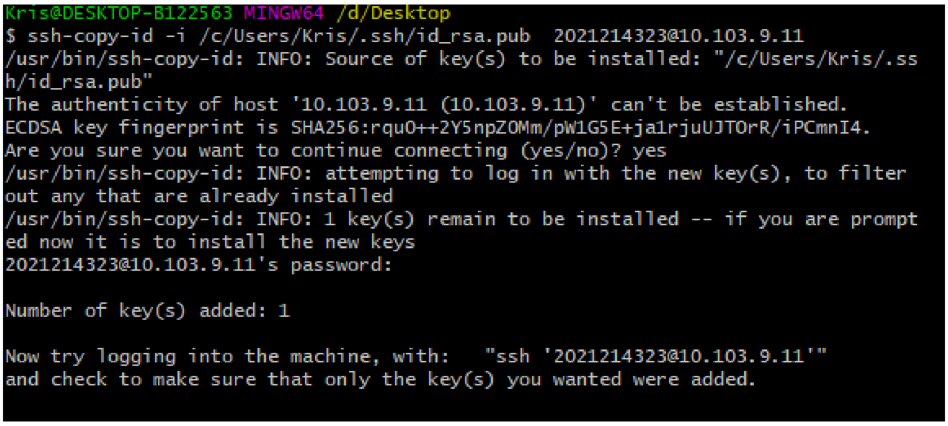
任务二:

在个人电脑上生成公钥和私钥



```
Kris@DESKTOP-B122563 MINGW64 /d/Desktop
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Kris/.ssh/id_rsa):
/c/Users/Kris/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Kris/.ssh/id_rsa.
Your public key has been saved in /c/Users/Kris/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:mtb5tmc3ZsCfSBPpdcw3cMcm1gnGRe4F3H2sgiq9OCY Kris@DESKTOP-B122563
The key's randomart image is:
+---[RSA 2048]----+
|            .+=Bo|
|            .o=+X|
|           . ooB+|
|          . + +.*|
|       .S. o + oo|
|      .+o.  *    |
|      +oo. . = . |
|    E.+ ... + B  |
|     o . .o+ + . |
+----[SHA256]-----+
```

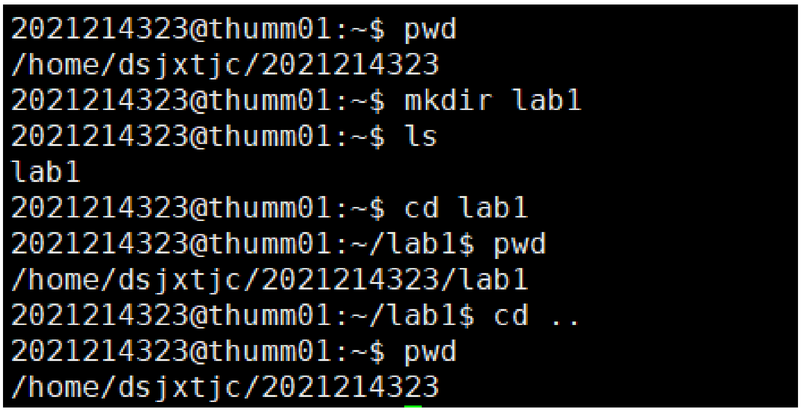
将公钥内容复制到服务器



```
Kris@DESKTOP-B122563 MINGW64 /d/Desktop
$ ssh-copy-id -i /c/Users/Kris/.ssh/id_rsa.pub  2021214323@10.103.9.11
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/c/Users/Kris/.ss
h/id_rsa.pub"
The authenticity of host '10.103.9.11 (10.103.9.11)' can't be established.
ECDSA key fingerprint is SHA256:rquO++2Y5npZOMm/pW1G5E+ja1rjuUJTOrR/iPCmnI4.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
2021214323@10.103.9.11's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh '2021214323@10.103.9.11'"
and check to make sure that only the key(s) you wanted were added.
```

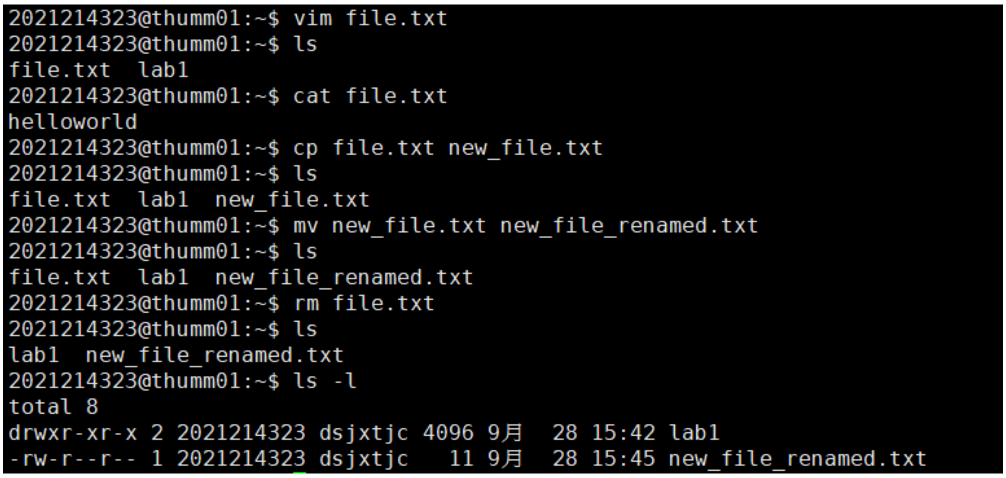
测试已可免密登录

任务三：pwd mkdir cd 等命令的使用



```
2021214323@thumm01:~$ pwd
/home/dsjxtjc/2021214323
2021214323@thumm01:~$ mkdir lab1
2021214323@thumm01:~$ ls
lab1
2021214323@thumm01:~$ cd lab1
2021214323@thumm01:~/lab1$ pwd
/home/dsjxtjc/2021214323/lab1
2021214323@thumm01:~/lab1$ cd ..
2021214323@thumm01:~$ pwd
/home/dsjxtjc/2021214323
```

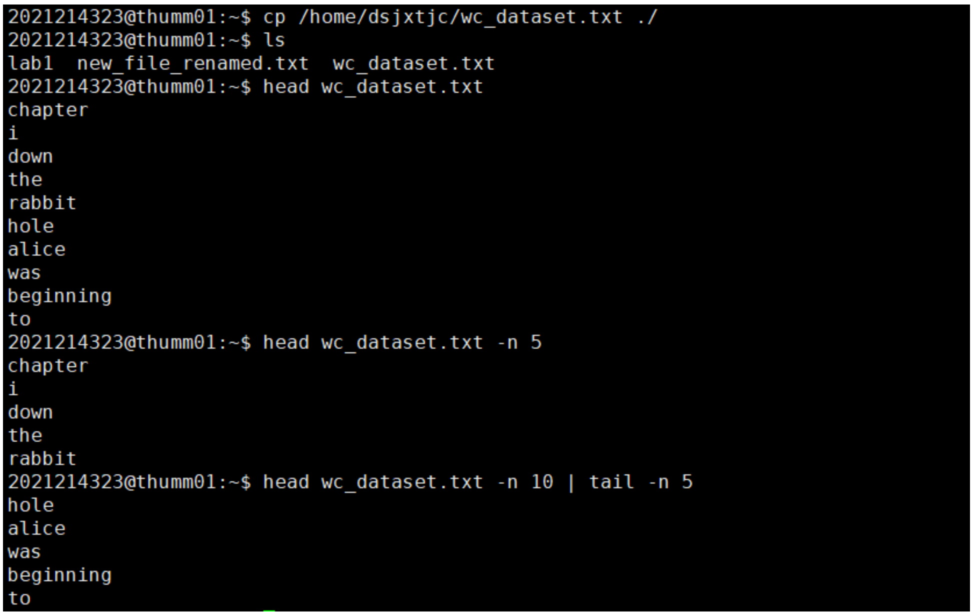
任务四 ： cp vim ls mv rm 等指令的使用



```
2021214323@thumm01:~$ vim file.txt
2021214323@thumm01:~$ ls
file.txt  lab1
2021214323@thumm01:~$ cat file.txt
helloworld
2021214323@thumm01:~$ cp file.txt new_file.txt
2021214323@thumm01:~$ ls
file.txt  lab1  new_file.txt
2021214323@thumm01:~$ mv new_file.txt new_file_renamed.txt
2021214323@thumm01:~$ ls
file.txt  lab1  new_file_renamed.txt
2021214323@thumm01:~$ rm file.txt
2021214323@thumm01:~$ ls
lab1  new_file_renamed.txt
2021214323@thumm01:~$ ls -l
total 8
drwxr-xr-x 2 2021214323 dsjxtjc 4096 9月  28 15:42 lab1
-rw-r--r-- 1 2021214323 dsjxtjc   11 9月  28 15:45 new_file_renamed.txt
```

任务五

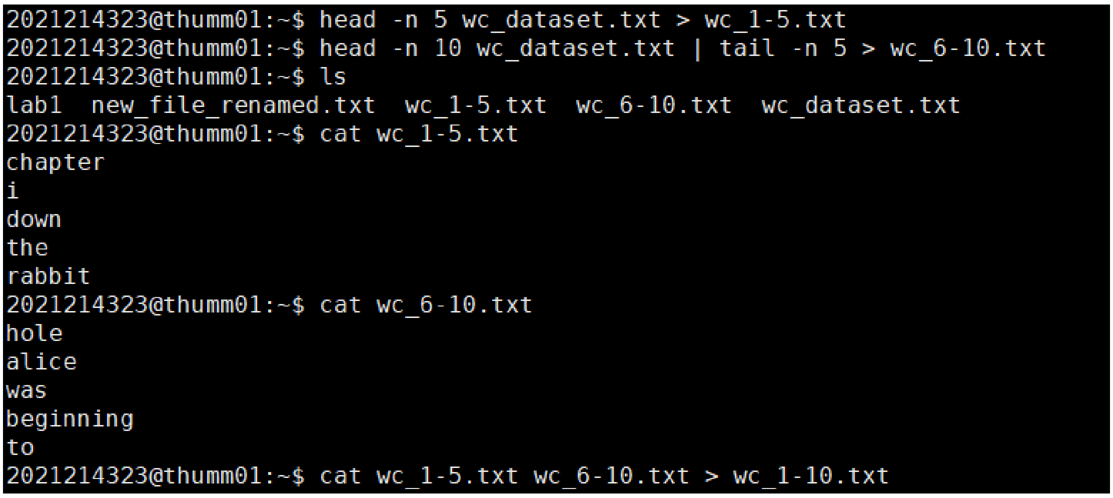
将数据集复制后用 head 和 tail 查看



```
2021214323@thumm01:~$ cp /home/dsjxtjc/wc_dataset.txt ./
2021214323@thumm01:~$ ls
lab1  new_file_renamed.txt  wc_dataset.txt
2021214323@thumm01:~$ head wc_dataset.txt
chapter
i
down
the
rabbit
hole
alice
was
beginning
to
2021214323@thumm01:~$ head wc_dataset.txt -n 5
chapter
i
down
the
rabbit
2021214323@thumm01:~$ head wc_dataset.txt -n 10 | tail -n 5
hole
alice
was
beginning
to
```

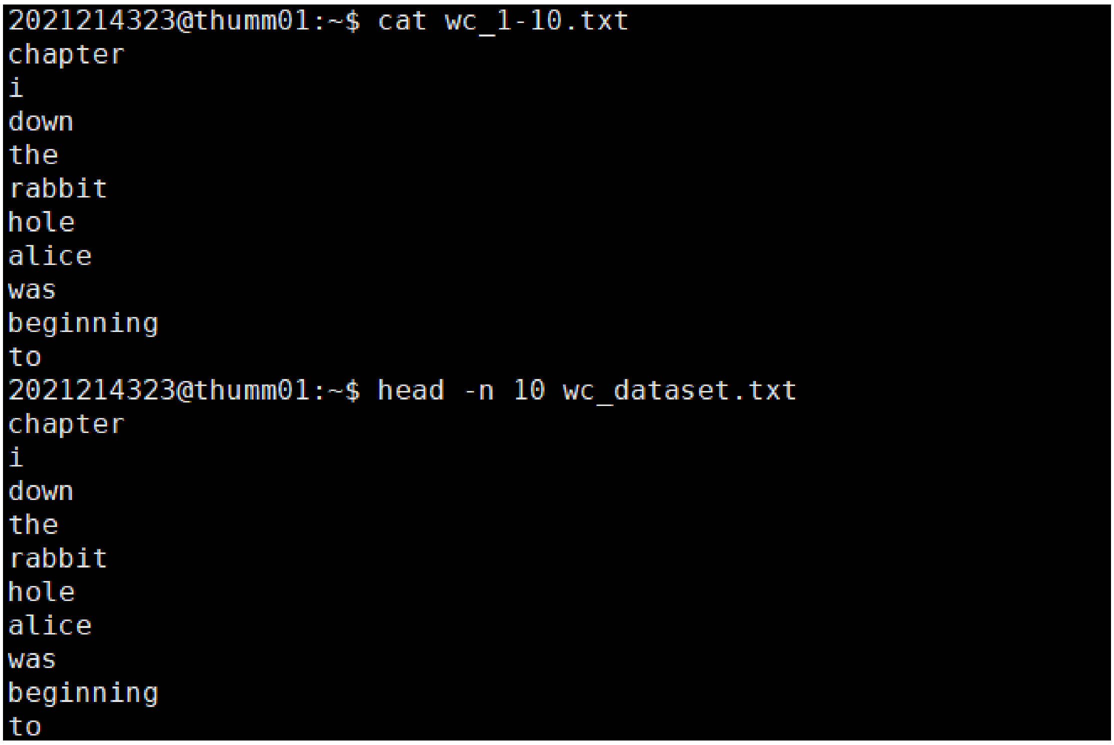
重定向符‘>’的使用 并使用 cat 查看结果

结合 cat 和重定向符 合并文件



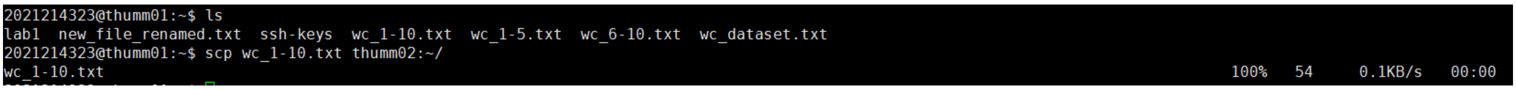
```
2021214323@thumm01:~$ head -n 5 wc_dataset.txt > wc_1-5.txt
2021214323@thumm01:~$ head -n 10 wc_dataset.txt | tail -n 5 > wc_6-10.txt
2021214323@thumm01:~$ ls
lab1  new_file_renamed.txt  wc_1-5.txt  wc_6-10.txt  wc_dataset.txt
2021214323@thumm01:~$ cat wc_1-5.txt
chapter
i
down
the
rabbit
2021214323@thumm01:~$ cat wc_6-10.txt
hole
alice
was
beginning
to
2021214323@thumm01:~$ cat wc_1-5.txt wc_6-10.txt > wc_1-10.txt
```

查看合并结果符合预期

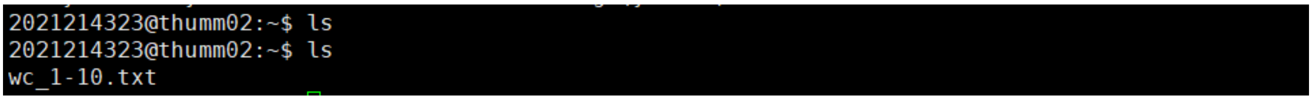


```
2021214323@thumm01:~$ cat wc_1-10.txt
chapter
i
down
the
rabbit
hole
alice
was
beginning
to
2021214323@thumm01:~$ head -n 10 wc_dataset.txt
chapter
i
down
the
rabbit
hole
alice
was
beginning
to
```

使用 scp 在两个主机间传文件

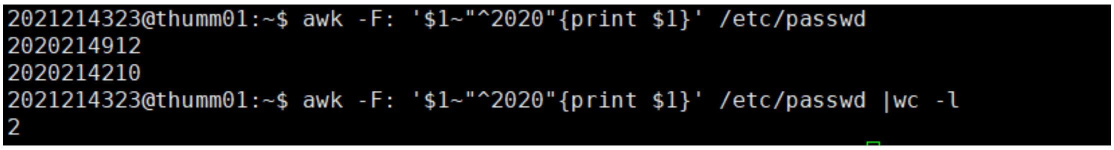


```
2021214323@thumm01:~$ ls
lab1  new_file_renamed.txt  ssh-keys  wc_1-10.txt  wc_1-5.txt  wc_6-10.txt  wc_dataset.txt
2021214323@thumm01:~$ scp wc_1-10.txt thumm02:~/
wc_1-10.txt                                                                                           100%   54     0.1KB/s   00:00
```



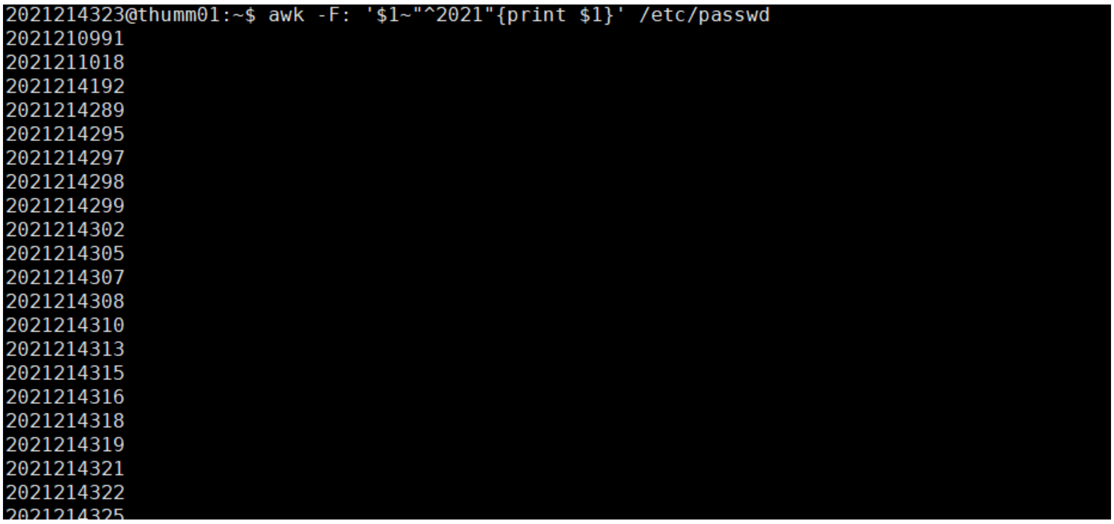
```
2021214323@thumm02:~$ ls
2021214323@thumm02:~$ ls
wc_1-10.txt
```

使用 awk 查看 2020 级的学生 以及人数

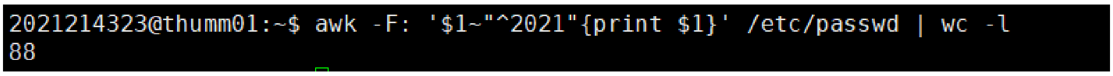


```
2021214323@thumm01:~$ awk -F: '$1~"^2020"{print $1}' /etc/passwd
2020214912
2020214210
2021214323@thumm01:~$ awk -F: '$1~"^2020"{print $1}' /etc/passwd |wc -l
2
```

使用 awk 查看 2021 级的学生 以及人数

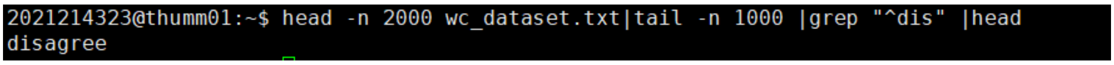


```
2021214323@thumm01:~$ awk -F: '$1~"^2021"{print $1}' /etc/passwd
2021210991
2021211018
2021214192
2021214289
2021214295
2021214297
2021214298
2021214299
2021214302
2021214305
2021214307
2021214308
2021214310
2021214313
2021214315
2021214316
2021214318
2021214319
2021214321
2021214322
2021214325
```



```
2021214323@thumm01:~$ awk -F: '$1~"^2021"{print $1}' /etc/passwd | wc -l
88
```

显示 1000~2000 行中所有以"dis"开头的单词（显示前 10 条）

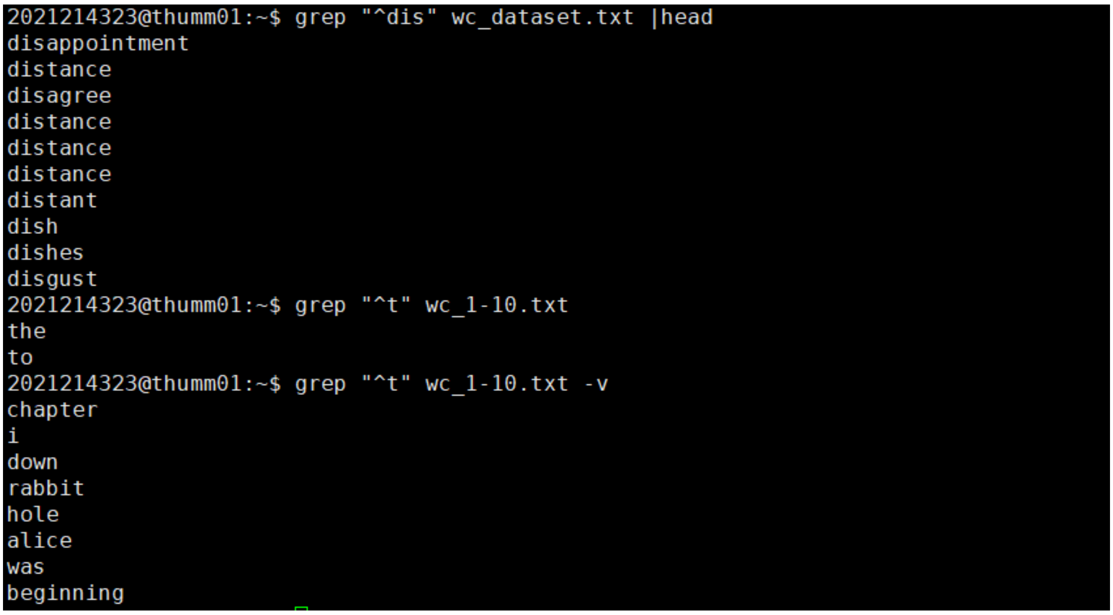


```
2021214323@thumm01:~$ head -n 2000 wc_dataset.txt|tail -n 1000 |grep "^dis" |head
disagree
```

显示所有以"dis"开头的单词（显示前 10 条）

查找 wc_1-10.txt 中以 t 字母开头的单词

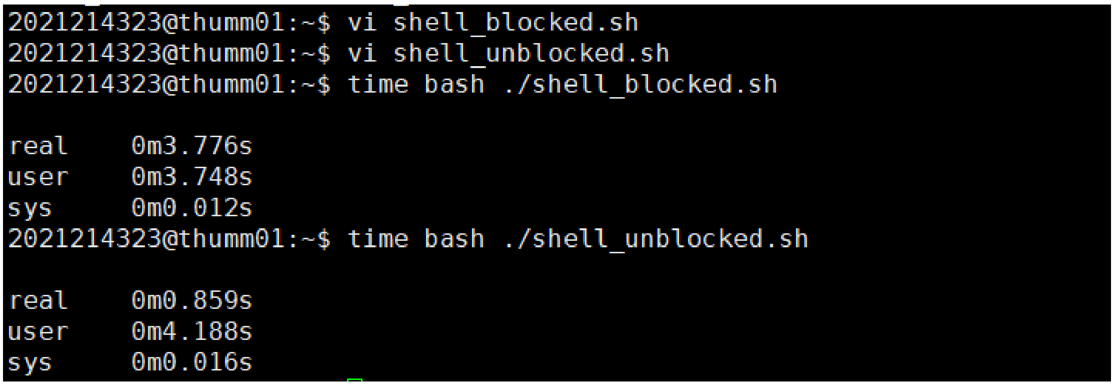
接着添加参数-v，过滤以 t 字母开头的单词



```
2021214323@thumm01:~$ grep "^dis" wc_dataset.txt |head
disappointment
distance
disagree
distance
distance
distance
distant
dish
dishes
disgust
2021214323@thumm01:~$ grep "^t" wc_1-10.txt
the
to
2021214323@thumm01:~$ grep "^t" wc_1-10.txt -v
chapter
i
down
rabbit
hole
alice
was
beginning
```

任务六 阻塞与非阻塞对比

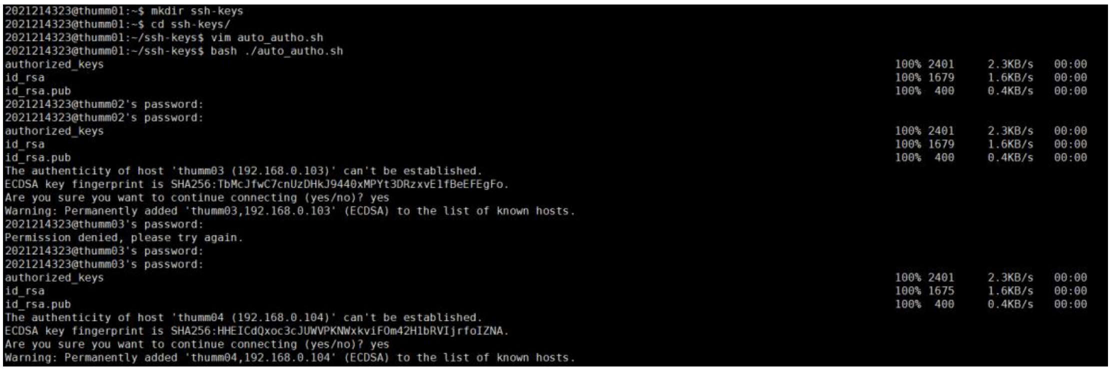
用户时间相差较少，非阻塞真实时间大幅减少



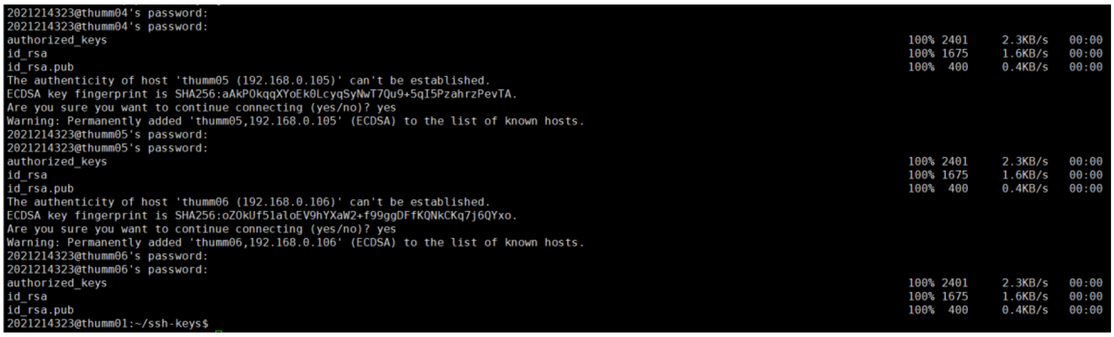
```
2021214323@thumm01:~$ vi shell_blocked.sh
2021214323@thumm01:~$ vi shell_unblocked.sh
2021214323@thumm01:~$ time bash ./shell_blocked.sh

real	0m3.776s
user	0m3.748s
sys	0m0.012s
2021214323@thumm01:~$ time bash ./shell_unblocked.sh

real	0m0.859s
user	0m4.188s
sys	0m0.016s
```

任务七

7.1 集群之间免密登录配置



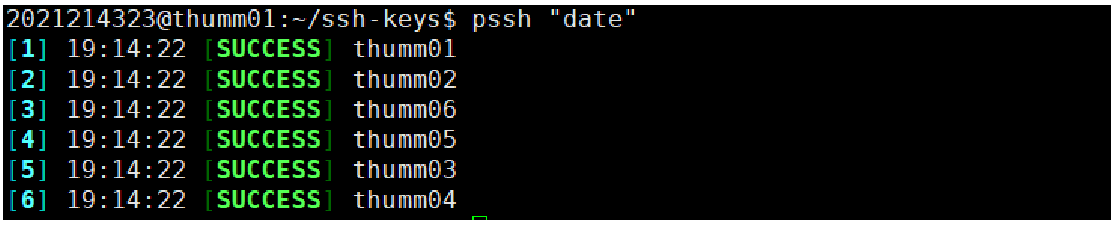
```
2021214323@thumm01:~$ mkdir ssh-keys
2021214323@thumm01:~$ cd ssh-keys/
2021214323@thumm01:~/ssh-keys$ vim auto_autho.sh
2021214323@thumm01:~/ssh-keys$ bash ./auto_autho.sh
authorized_keys                                                                  100% 2401     2.3KB/s   00:00
id_rsa                                                                           100% 1679     1.6KB/s   00:00
id_rsa.pub                                                                       100%  400     0.4KB/s   00:00
2021214323@thumm02's password:
2021214323@thumm02's password:
authorized_keys                                                                  100% 2401     2.3KB/s   00:00
id_rsa                                                                           100% 1679     1.6KB/s   00:00
id_rsa.pub                                                                       100%  400     0.4KB/s   00:00
The authenticity of host 'thumm03 (192.168.0.103)' can't be established.
ECDSA key fingerprint is SHA256:TbMcJfwC7cnUzDHkJ9440xMPYt3DRzxvElfBeEFEgFo.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'thumm03,192.168.0.103' (ECDSA) to the list of known hosts.
2021214323@thumm03's password:
Permission denied, please try again.
2021214323@thumm03's password:
2021214323@thumm03's password:
authorized_keys                                                                  100% 2401     2.3KB/s   00:00
id_rsa                                                                           100% 1675     1.6KB/s   00:00
id_rsa.pub                                                                       100%  400     0.4KB/s   00:00
The authenticity of host 'thumm04 (192.168.0.104)' can't be established.
ECDSA key fingerprint is SHA256:HHEICdQxoc3cJUWVPKNWxkviFOm42H1bRVIjrfoIZNA.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'thumm04,192.168.0.104' (ECDSA) to the list of known hosts.
```



```
2021214323@thumm04's password:
2021214323@thumm04's password:
authorized_keys                                                                  100% 2401     2.3KB/s   00:00
id_rsa                                                                           100% 1675     1.6KB/s   00:00
id_rsa.pub                                                                       100%  400     0.4KB/s   00:00
The authenticity of host 'thumm05 (192.168.0.105)' can't be established.
ECDSA key fingerprint is SHA256:aAkPOkqqXYoEk0LcyqSyNwT7Qu9+5qI5PzahrzPevTA.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'thumm05,192.168.0.105' (ECDSA) to the list of known hosts.
2021214323@thumm05's password:
2021214323@thumm05's password:
authorized_keys                                                                  100% 2401     2.3KB/s   00:00
id_rsa                                                                           100% 1675     1.6KB/s   00:00
id_rsa.pub                                                                       100%  400     0.4KB/s   00:00
The authenticity of host 'thumm06 (192.168.0.106)' can't be established.
ECDSA key fingerprint is SHA256:oZOkUf51aloEV9hYXaW2+f99ggDFfKQNkCKq7j6QYxo.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'thumm06,192.168.0.106' (ECDSA) to the list of known hosts.
2021214323@thumm06's password:
2021214323@thumm06's password:
authorized_keys                                                                  100% 2401     2.3KB/s   00:00
id_rsa                                                                           100% 1675     1.6KB/s   00:00
id_rsa.pub                                                                       100%  400     0.4KB/s   00:00
2021214323@thumm01:~/ssh-keys$
```

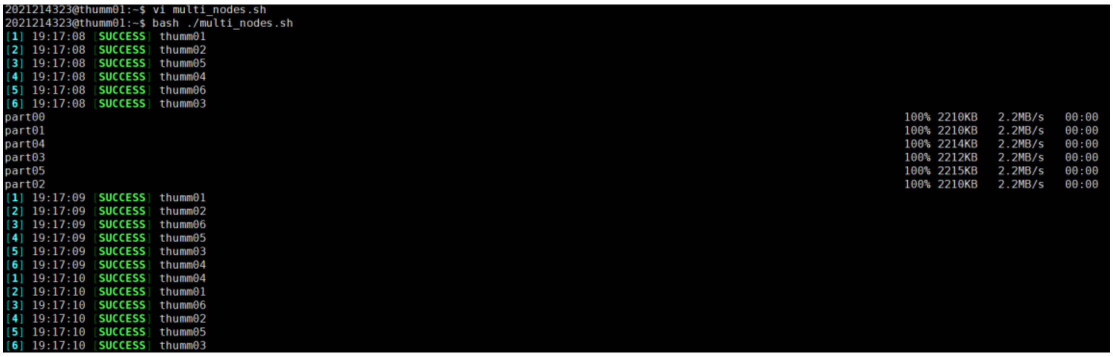
7.2 集群间批管理

使用 pssh 指令



```
2021214323@thumm01:~/ssh-keys$ pssh "date"
[1] 19:14:22 [SUCCESS] thumm01
[2] 19:14:22 [SUCCESS] thumm02
[3] 19:14:22 [SUCCESS] thumm06
[4] 19:14:22 [SUCCESS] thumm05
[5] 19:14:22 [SUCCESS] thumm03
[6] 19:14:22 [SUCCESS] thumm04
```

7.3 在多主机上并行执行任务



```
2021214323@thumm01:~$ vi multi_nodes.sh
2021214323@thumm01:~$ bash ./multi_nodes.sh
[1] 19:17:08 [SUCCESS] thumm01
[2] 19:17:08 [SUCCESS] thumm02
[3] 19:17:08 [SUCCESS] thumm05
[4] 19:17:08 [SUCCESS] thumm04
[5] 19:17:08 [SUCCESS] thumm06
[6] 19:17:08 [SUCCESS] thumm03
part00                                                              100% 2210KB   2.2MB/s   00:00
part01                                                              100% 2210KB   2.2MB/s   00:00
part04                                                              100% 2214KB   2.2MB/s   00:00
part03                                                              100% 2212KB   2.2MB/s   00:00
part05                                                              100% 2215KB   2.2MB/s   00:00
part02                                                              100% 2210KB   2.2MB/s   00:00
[1] 19:17:09 [SUCCESS] thumm01
[2] 19:17:09 [SUCCESS] thumm02
[3] 19:17:09 [SUCCESS] thumm06
[4] 19:17:09 [SUCCESS] thumm05
[5] 19:17:09 [SUCCESS] thumm03
[6] 19:17:09 [SUCCESS] thumm04
[1] 19:17:10 [SUCCESS] thumm04
[2] 19:17:10 [SUCCESS] thumm01
[3] 19:17:10 [SUCCESS] thumm06
[4] 19:17:10 [SUCCESS] thumm02
[5] 19:17:10 [SUCCESS] thumm05
[6] 19:17:10 [SUCCESS] thumm03
```

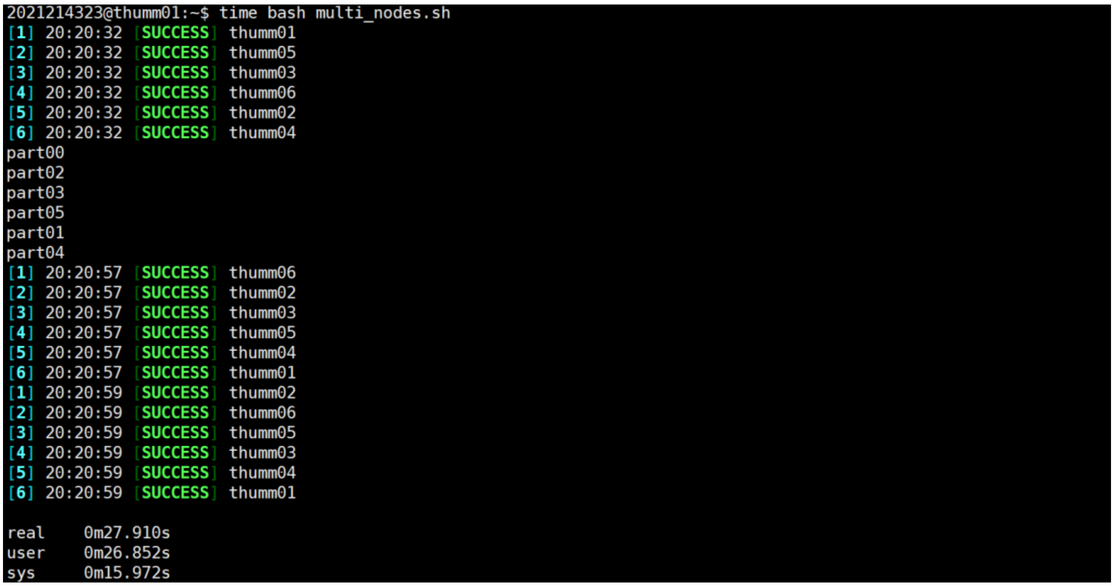
任务 8

改变 split 方法，将-l 参数改为-C，将文件按字节划分

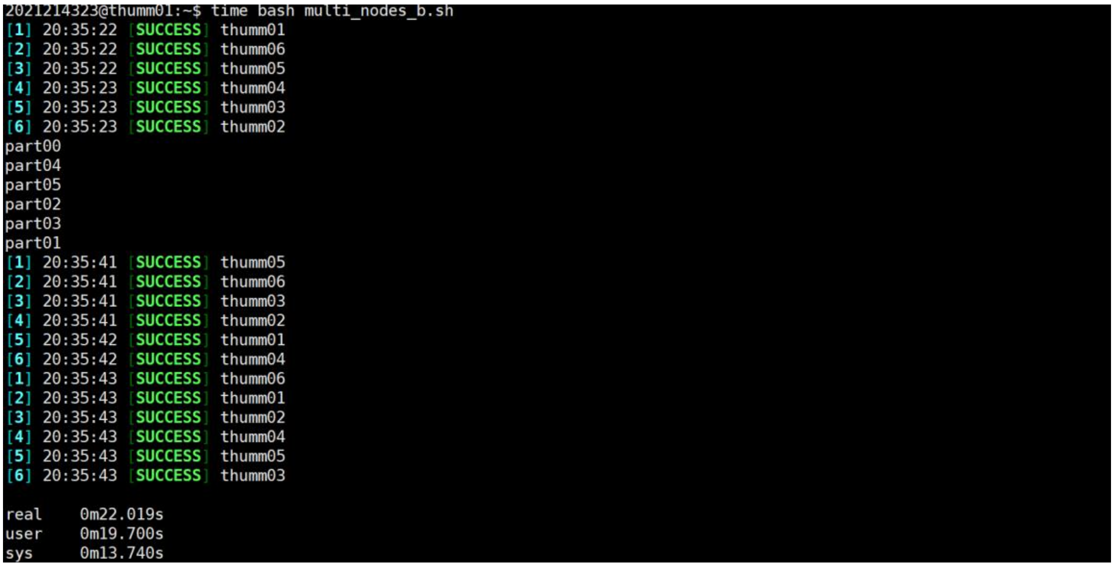
验证：在自制 wc_bigdata.txt 上执行类似任务 7 过程

用 time 进行时间对比

代码 1 为-l 参数以行数进行划分，代码 2 为-C 参数以字节进行划分，结果分别如下



```
2021214323@thumm01:~$ time bash multi_nodes.sh
[1] 20:20:32 [SUCCESS] thumm01
[2] 20:20:32 [SUCCESS] thumm05
[3] 20:20:32 [SUCCESS] thumm03
[4] 20:20:32 [SUCCESS] thumm06
[5] 20:20:32 [SUCCESS] thumm02
[6] 20:20:32 [SUCCESS] thumm04
part00
part02
part03
part05
part01
part04
[1] 20:20:57 [SUCCESS] thumm06
[2] 20:20:57 [SUCCESS] thumm02
[3] 20:20:57 [SUCCESS] thumm03
[4] 20:20:57 [SUCCESS] thumm05
[5] 20:20:57 [SUCCESS] thumm04
[6] 20:20:57 [SUCCESS] thumm01
[1] 20:20:59 [SUCCESS] thumm02
[2] 20:20:59 [SUCCESS] thumm06
[3] 20:20:59 [SUCCESS] thumm05
[4] 20:20:59 [SUCCESS] thumm03
[5] 20:20:59 [SUCCESS] thumm04
[6] 20:20:59 [SUCCESS] thumm01

real    0m27.910s
user    0m26.852s
sys     0m15.972s
```



```
2021214323@thumm01:~$ time bash multi_nodes_b.sh
[1] 20:35:22 [SUCCESS] thumm01
[2] 20:35:22 [SUCCESS] thumm06
[3] 20:35:22 [SUCCESS] thumm05
[4] 20:35:23 [SUCCESS] thumm04
[5] 20:35:23 [SUCCESS] thumm03
[6] 20:35:23 [SUCCESS] thumm02
part00
part04
part05
part02
part03
part01
[1] 20:35:41 [SUCCESS] thumm05
[2] 20:35:41 [SUCCESS] thumm06
[3] 20:35:41 [SUCCESS] thumm03
[4] 20:35:41 [SUCCESS] thumm02
[5] 20:35:42 [SUCCESS] thumm01
[6] 20:35:42 [SUCCESS] thumm04
[1] 20:35:43 [SUCCESS] thumm06
[2] 20:35:43 [SUCCESS] thumm01
[3] 20:35:43 [SUCCESS] thumm02
[4] 20:35:43 [SUCCESS] thumm04
[5] 20:35:43 [SUCCESS] thumm05
[6] 20:35:43 [SUCCESS] thumm03

real    0m22.019s
user    0m19.700s
sys     0m13.740s
```

代码 1

```
#!/bin/bash --login
pssh "mkdir -p ~/multi-nodes"           # 在 thumm01-thumm06节点的主目录下创建 multi-nodes 目录
cd multi-nodes


lines=`cat ../wc_bigdataset.txt | wc -l`     # 计算 wc_bigdataset.txt 的行数
lines_per_node=$(($lines/6+1))             # 将 wc_bigdataset.txt 划分为6部分，计算每部的行数
split -l $lines_per_node ../wc_bigdataset.txt -d part  # 划分 wc_dataset.txt 为 part00-part06

# 将不同的部分分别传至不同的节点
for ((i=0;i<6;i=i+1));do
    scp part0$i thumm0$(($i+1)):~/multi-nodes/part &
done
wait  # 等待节点传输完成

# 让每个节点运行任务，将结果保存在各自的~/multi-nodes/result 文件中
pssh "grep '^t' ~/multi-nodes/part > ~/multi-nodes/result"

# 将所有节点的计算结果传至 thumm01(当前操作的主机)
pslurp -L ~/multi-nodes/ ~/multi-nodes/result .

# 将所有结果整合成一个文件：t_head_multi_node.txt
rm -rf ~/multi-nodes/t_head_multi_node.txt
for ((i=1; i<=6; i=i+1)); do
    cat ~/multi-nodes/thumm0$i/result >> ~/multi-nodes/t_head_multi_node.txt
done
```

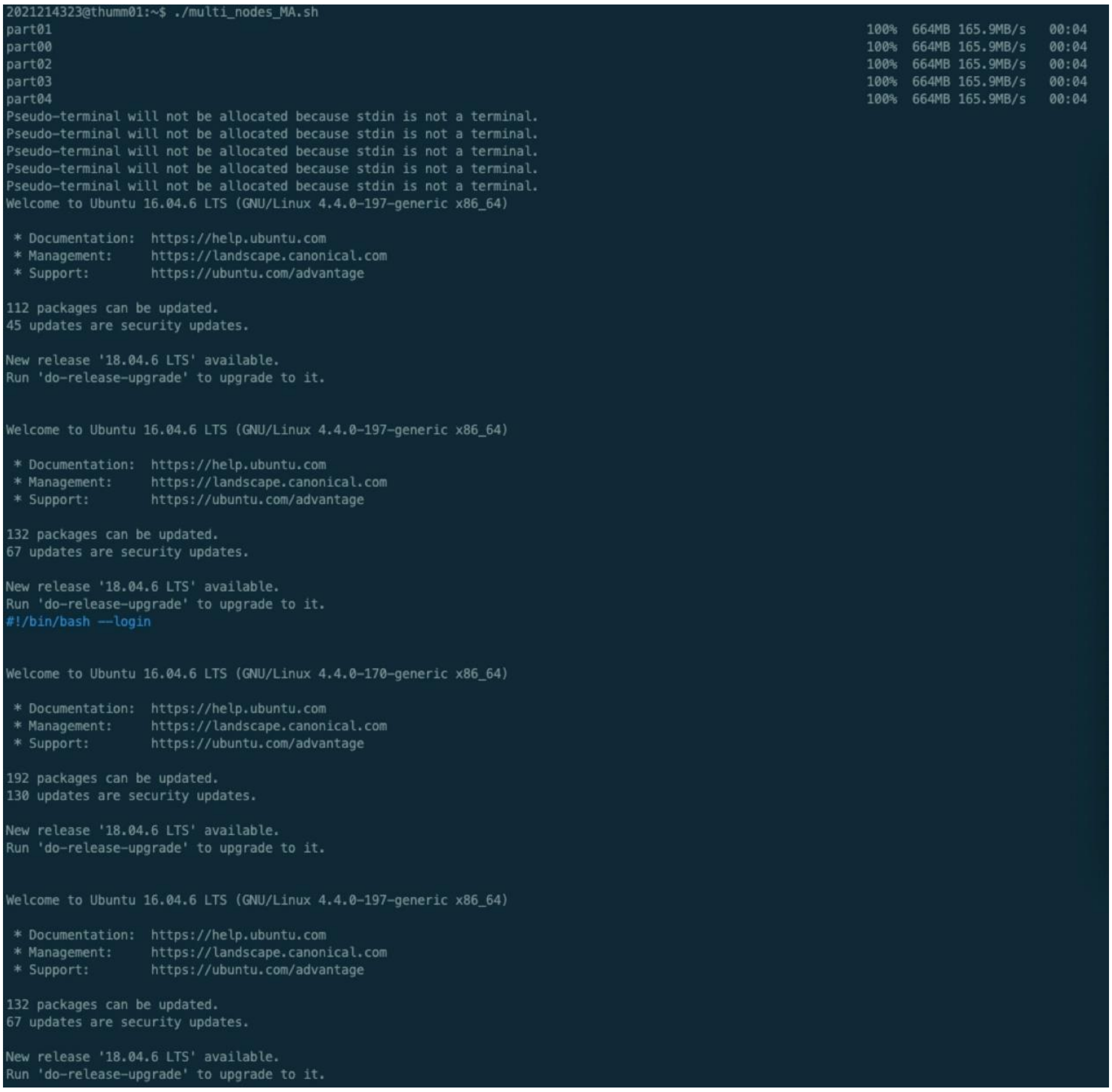
代码 2

```
#!/bin/bash --login
pssh "mkdir -p ~/multi-nodes"                    # 在 thumm01-thumm06 节点的主目录下创建 multi-nodes 目录
cd multi-nodes


bytes=`ls -l ../wc_bigdataset.txt | awk '{print $5}'`         # 计算 wc_bigdataset.txt 的比特数
bytes_per_node=$(($bytes/6+1))                        # 将 wc_bigdataset.txt 划分为 6 部分, 计算每部的比特数
split -C $bytes_per_node ../wc_bigdataset.txt -d part   # 划分 wc_dataset.txt 为 part00-part06

# 将不同的部分分别传至不同的节点
for ((i=0;i<6;i=i+1));do
    scp part0$i thumm0$(($i+1)):~/multi-nodes/part &
done
wait   # 等待节点传输完成

# 让每个节点运行任务，将结果保存在各自的~/multi-nodes/result 文件中
pssh "grep '^t' ~/multi-nodes/part > ~/multi-nodes/result"

# 将所有节点的计算结果传至 thumm01(当前操作的主机)
pslurp -L ~/multi-nodes/ ~/multi-nodes/result .

# 将所有结果整合成一个文件：t_head_multi_node.txt
rm -rf ~/multi-nodes/t_head_multi_node.txt
for ((i=1; i<=6; i=i+1)); do
    cat ~/multi-nodes/thumm0$i/result >> ~/multi-nodes/t_head_multi_node.txt
done
```

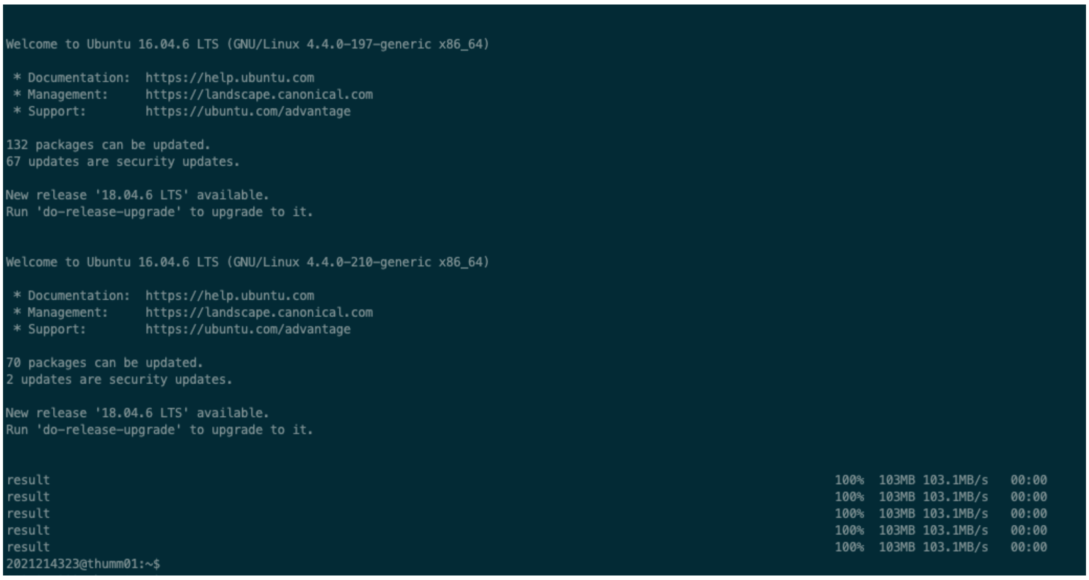
任务 9

Thumm01 作为 master，thumm02-06 作为 agent

Master 生成子任务，并负责分发数据，最后实验结果如下

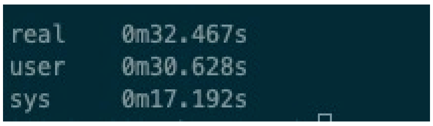


```
2021214323@thumm01:~$ ./multi_nodes_MA.sh
part01                                                                                                      100%  664MB 165.9MB/s   00:04
part00                                                                                                      100%  664MB 165.9MB/s   00:04
part02                                                                                                      100%  664MB 165.9MB/s   00:04
part03                                                                                                      100%  664MB 165.9MB/s   00:04
part04                                                                                                      100%  664MB 165.9MB/s   00:04
Pseudo-terminal will not be allocated because stdin is not a terminal.
Pseudo-terminal will not be allocated because stdin is not a terminal.
Pseudo-terminal will not be allocated because stdin is not a terminal.
Pseudo-terminal will not be allocated because stdin is not a terminal.
Pseudo-terminal will not be allocated because stdin is not a terminal.
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-197-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

112 packages can be updated.
45 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-197-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

132 packages can be updated.
67 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
#!/bin/bash --login


Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-170-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

192 packages can be updated.
130 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-197-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

132 packages can be updated.
67 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```



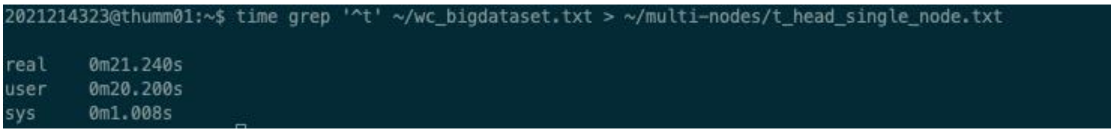
```

Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-197-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

132 packages can be updated.
67 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-210-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

70 packages can be updated.
2 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


result                                                                                                   100%  103MB 103.1MB/s   00:00
result                                                                                                   100%  103MB 103.1MB/s   00:00
result                                                                                                   100%  103MB 103.1MB/s   00:00
result                                                                                                   100%  103MB 103.1MB/s   00:00
result                                                                                                   100%  103MB 103.1MB/s   00:00
2021214323@thumm01:~$
```

多机部分见代码 3
多级执行时间



```
real	0m32.467s
user	0m30.628s
sys	0m17.192s
```

单机执行时间



```
2021214323@thumm01:~$ time grep '^t' ~/wc_bigdataset.txt > ~/multi-nodes/t_head_single_node.txt

real	0m21.240s
user	0m20.200s
sys	0m1.008s
```

多机时间竟然比单机久，分析可能原因

多机数据分割方法问题，可能可以按任务 7 方式更好的分割，提高效率

ssh 建立连接耗时，后续可改为其他方式

代码 3

```
#!/bin/bash --login
echo 'mkdir -p ~/multi-nodes' > agent.sh             # 在节点主目录下创建 multi-nodes 目录

# 让每个节点运行任务，将结果保存在各自的~/multi-nodes/result 文件中
echo "grep '^t' ~/multi-nodes/part > ~/multi-nodes/result" >> agent.sh

# 将所有节点的计算结果传至 thumm01(当前操作的主机)
echo "scp ~/multi-nodes/result thumm01:~/multi-nodes/" >> agent.sh

mkdir -p ~/multi-nodes
cd multi-nodes

for ((i=1;i<6;i=i+1));do
   mkdir -p thumm0$(($i+1))
done
wait

lines=`cat ../wc_bigdataset.txt | wc -l`     # 计算 wc_bigdataset.txt 的行数
lines_per_node=$(($lines/5+1))              # 将 wc_bigdataset.txt 分为5部分，计算每部的行数
split -l $lines_per_node ../wc_bigdataset.txt -d part  # 划分 wc_dataset.txt 为 part00-part05

# 将不同的部分分别传至不同的节点
for ((i=0;i<5;i=i+1));do
   scp part0$i thumm0$(($i+2)):~/multi-nodes/part &
done
wait  # 等待节点传输完成
for ((i=1;i<6;i=i+1));do
   ssh thumm0$(($i+1)) < ~/agent.sh &
done
wait

for ((i=1;i<6;i=i+1));do
   scp thumm0$(($i+1)):~/multi-nodes/result ~/multi-nodes/thumm0$(($i+1))/ &
done
wait

# 将所有结果整合成一个文件：t_head_multi_node.txt
rm -rf ~/multi-nodes/t_head_multi_node.txt
for ((i=2; i<=6; i=i+1)); do
   cat ~/multi-nodes/thumm0$i/result >> ~/multi-nodes/t_head_multi_node.txt
done
```