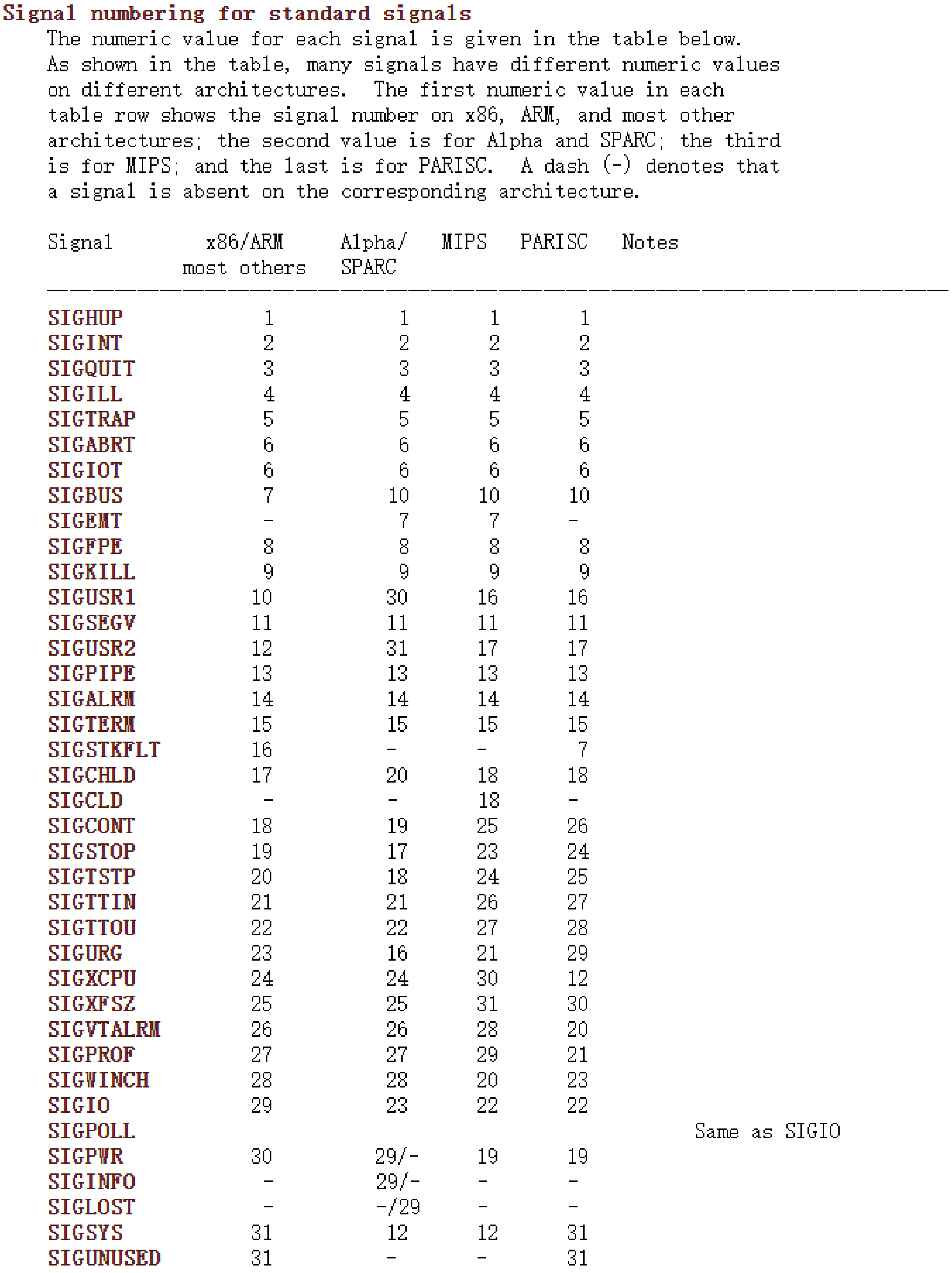
HW5 Solution

Author:金之航

StuID:2183411101

Problem

1.Job Control



Signal numbering for standard signals

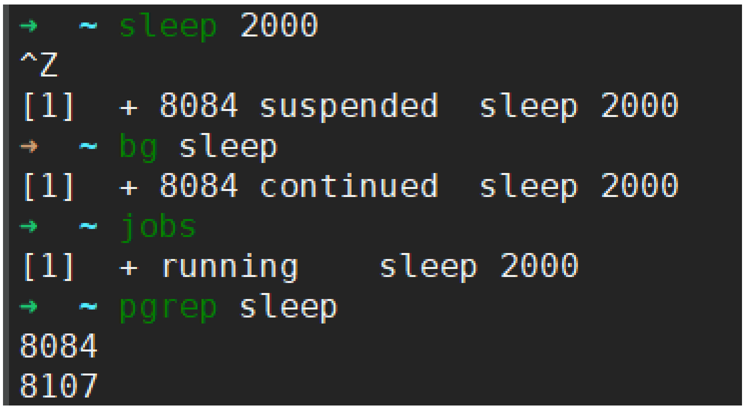
The numeric value for each signal is given in the table below. As shown in the table, many signals have different numeric values on different architectures. The first numeric value in each table row shows the signal number on x86, ARM, and most other architectures; the second value is for Alpha and SPARC; the third is for MIPS; and the last is for PARISC. A dash (-) denotes that a signal is absent on the corresponding architecture.

Signal	x86/ARM most others	Alpha/ SPARC	MIPS	PARISC	Notes
SIGHUP	1	1	1	1	
SIGINT	2	2	2	2	
SIGQUIT	3	3	3	3	
SIGILL	4	4	4	4	
SIGTRAP	5	5	5	5	
SIGABRT	6	6	6	6	
SIGIOT	6	6	6	6	
SIGBUS	7	10	10	10	
SIGEMT	-	7	7	-	
SIGFPE	8	8	8	8	
SIGKILL	9	9	9	9	
SIGUSR1	10	30	16	16	
SIGSEGV	11	11	11	11	
SIGUSR2	12	31	17	17	
SIGPIPE	13	13	13	13	
SIGALRM	14	14	14	14	
SIGTERM	15	15	15	15	
SIGSTKFLT	16	-	-	7	
SIGCHLD	17	20	18	18	
SIGCLD	-	-	18	-	
SIGCONT	18	19	25	26	
SIGSTOP	19	17	23	24	
SIGTSTP	20	18	24	25	
SIGTTIN	21	21	26	27	
SIGTTOU	22	22	27	28	
SIGURG	23	16	21	29	
SIGXCPU	24	24	30	12	
SIGXFSZ	25	25	31	30	
SIGVTALRM	26	26	28	20	
SIGPROF	27	27	29	21	
SIGWINCH	28	28	20	23	
SIGIO	29	23	22	22	
SIGPOLL					Same as SIGIO
SIGPWR	30	29/-	19	19	
SIGINFO	-	29/-	-	-	
SIGLOST	-	-/29	-	-	
SIGSYS	31	12	12	31	
SIGUNUSED	31	-	-	31	

-Terminal Multiplexes
-Dot files
-Remote Machines

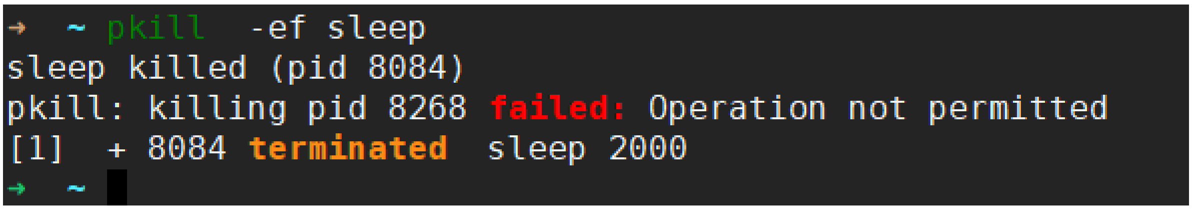
```
sleep 2000
nohup sleep 2000 &
jobs
bg %1 #fg?
kill -STOP %
jobs
kill -HUP %1
kill -HUP %2
kill -KILL %2 #note
```

```
sleep 2000
bg sleep
pgrep sleep
```



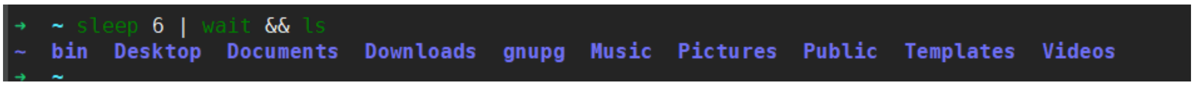
```
➜  ~ sleep 2000
^Z
[1]  + 8084 suspended  sleep 2000
➜  ~ bg sleep
[1]  + 8084 continued  sleep 2000
➜  ~ jobs
[1]  + running    sleep 2000
➜  ~ pgrep sleep
8084
8107
```

```
pkill  -ef sleep

-a  Include process ancestors in the match list.  By default, the current pgrep or pkill process and all of its ancestors are excluded (unless -v is used). #默认就是-a所以不用再写
-f  Match against full argument lists. The default is to match against process names.
```



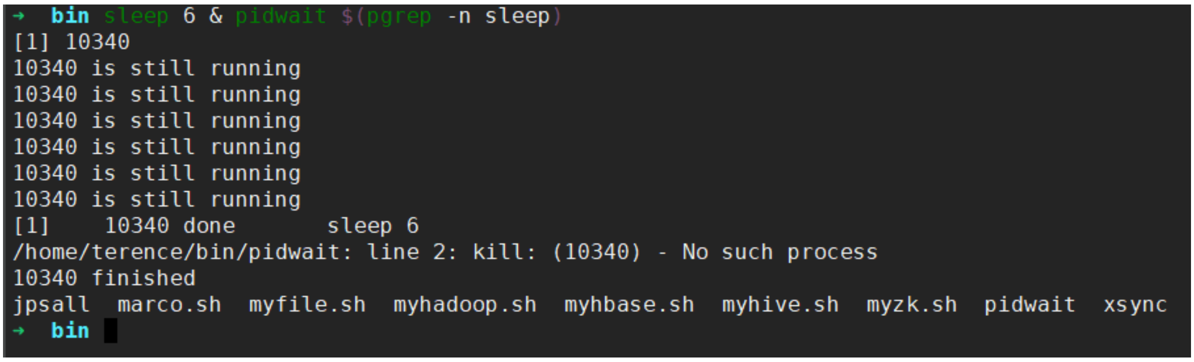
```
➜  ~ pkill  -ef sleep
sleep killed (pid 8084)
pkill: killing pid 8268 failed: Operation not permitted
[1]  + 8084 terminated  sleep 2000
➜  ~ 
```

```
sleep 60 | wait && ls
```



```
➜  ~ sleep 6 | wait && ls
~ bin  Desktop  Documents  Downloads  gnupg  Music  Pictures  Public  Templates  Videos
➜  ~
```

```
sleep 6 & pidwait $(pgrep -n sleep)

#pidwait:
#!/bin/bash
while kill -0 $1
do
echo "$1 is still running"
sleep 1
done
echo "$1 finished"
ls
```

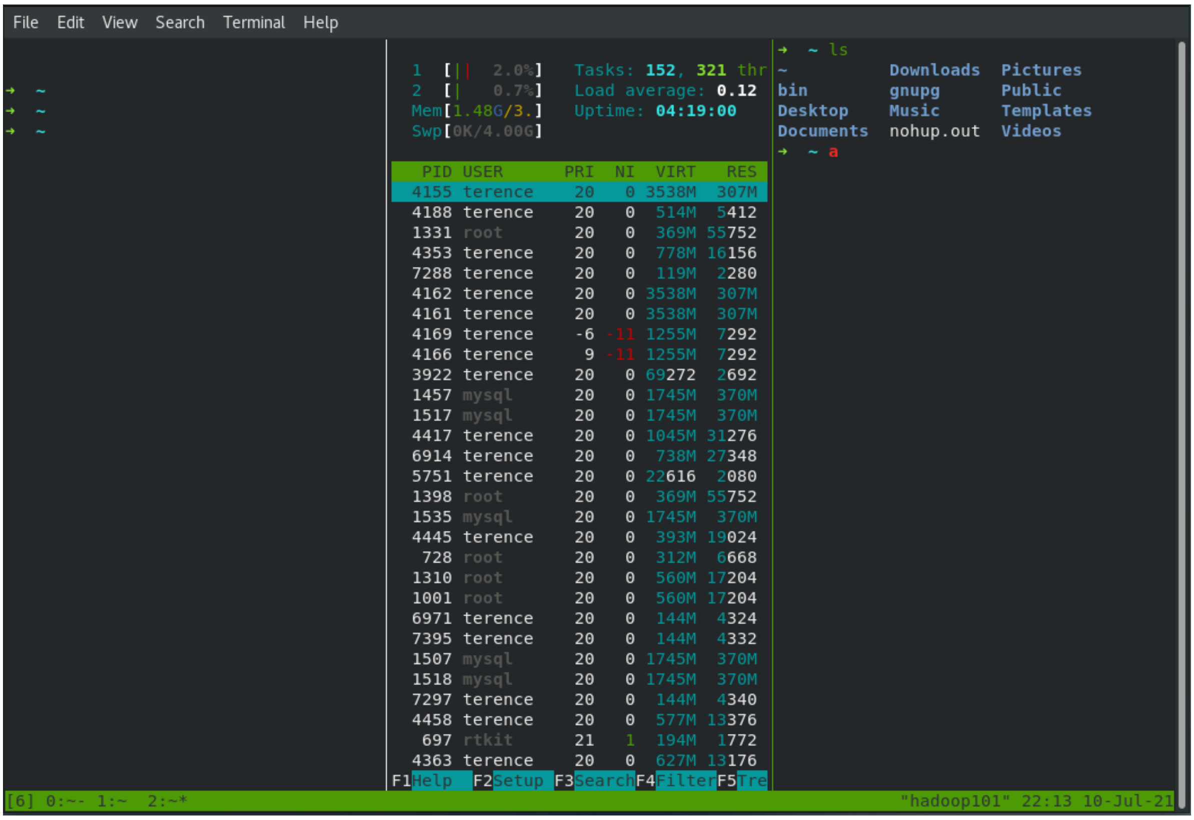


```
➜  bin sleep 6 & pidwait $(pgrep -n sleep)
[1] 10340
10340 is still running
10340 is still running
10340 is still running
10340 is still running
10340 is still running
10340 is still running
[1]    10340 done       sleep 6
/home/terence/bin/pidwait: line 2: kill: (10340) - No such process
10340 finished
jpsall  marco.sh  myfile.sh  myhadoop.sh  myhbase.sh  myhive.sh  myzk.sh  pidwait  xsync
➜  bin 
```

2.Mux

-Sessions
-Windows
-Panes

```
tmux new -t foobar
<C-b> N Go to the N th window. Note they are numbered
<C-b> p Goes to the previous window
<C-b> n Goes to the next window
<C-b> , Rename the current window
<C-b> w List current windows
<C-b> " Split the current pane horizontally
<C-b> % Split the current pane vertically
<C-b> <direction> Move to the pane in the specified direction. Direction here means arrow keys.
<C-b> z Toggle zoom for the current pane
<C-b> [ Start scrollback. You can then press <space> to start a selection and <enter> to copy that selection.
<C-b> <space> Cycle through pane arrangements.
```



```
➜  ~
➜  ~
➜  ~
```

```
  1  [||     2.0%]   Tasks: 152, 321 thr
  2  [|      0.7%]   Load average: 0.12
  Mem[1.48G/3.]      Uptime: 04:19:00
  Swp[0K/4.00G]

   PID USER      PRI  NI  VIRT   RES
  4155 terence    20   0 3538M  307M
  4188 terence    20   0  514M  5412
  1331 root       20   0  369M 55752
  4353 terence    20   0  778M 16156
  7288 terence    20   0  119M  2280
  4162 terence    20   0 3538M  307M
  4161 terence    20   0 3538M  307M
  4169 terence    -6 -11 1255M  7292
  4166 terence     9 -11 1255M  7292
  3922 terence    20   0 69272  2692
  1457 mysql      20   0 1745M  370M
  1517 mysql      20   0 1745M  370M
  4417 terence    20   0 1045M 31276
  6914 terence    20   0  738M 27348
  5751 terence    20   0 22616  2080
  1398 root       20   0  369M 55752
  1535 mysql      20   0 1745M  370M
  4445 terence    20   0  393M 19024
   728 root       20   0  312M  6668
  1310 root       20   0  560M 17204
  1001 root       20   0  560M 17204
  6971 terence    20   0  144M  4324
  7395 terence    20   0  144M  4332
  1507 mysql      20   0 1745M  370M
  1518 mysql      20   0 1745M  370M
  7297 terence    20   0  144M  4340
  4458 terence    20   0  577M 13376
   697 rtkit      21   1  194M  1772
  4363 terence    20   0  627M 13176
F1Help  F2Setup F3SearchF4FilterF5Tre
```

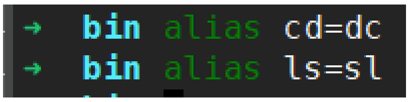
```
➜  ~ ls
~             Downloads  Pictures
bin           gnupg      Public
Desktop       Music      Templates
Documents     nohup.out  Videos
➜  ~ a
```

```
[6] 0:~- 1:~  2:~*                                 "hadoop101" 22:13 10-Jul-21
```

3.Aliases

wrong typing dc or ls

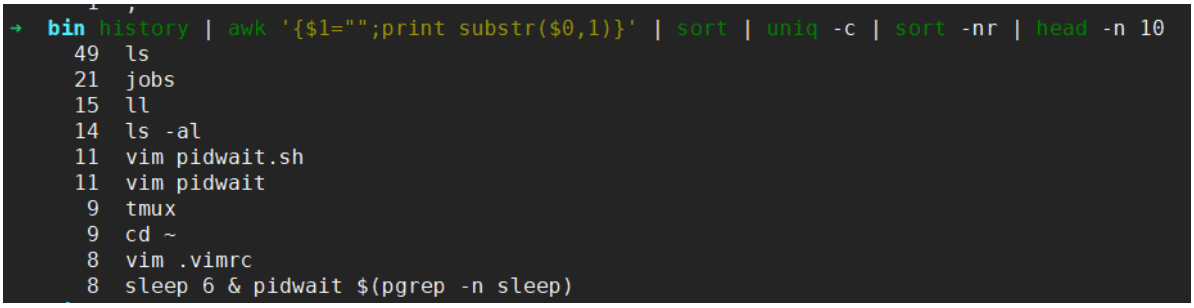
```
alias cd=dc
alias cd="rm -rf"
alias ls=sl
```



```
➜  bin alias cd=dc
➜  bin alias ls=sl
```

history sort

```
bin history | awk '{$1="";print substr($0,1)}' | sort | uniq -c | sort -nr |
head -n 10
别名过程省略了，具体就是alias a=b, 有空格的需要用()
```



```
➜  bin history | awk '{$1="";print substr($0,1)}' | sort | uniq -c | sort -nr | head -n 10
     49  ls
     21  jobs
     15  ll
     14  ls -al
     11  vim pidwait.sh
     11  vim pidwait
      9  tmux
      9  cd ~
      8  vim .vimrc
      8  sleep 6 & pidwait $(pgrep -n sleep)
```

4.Dotfiles

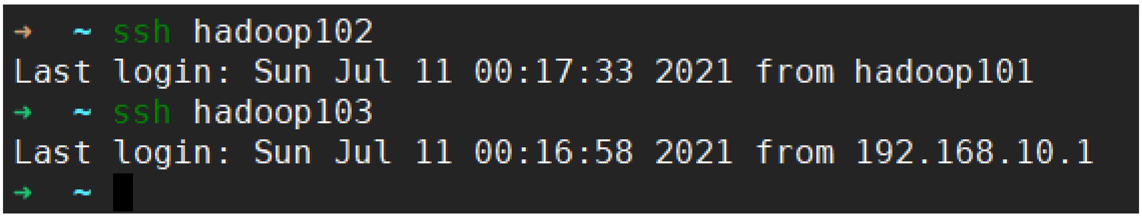
Some other examples of tools that can be configured through dotfiles are:

```
bash - ~/.bashrc, ~/.bash_profile
zsh - ~/.zshrc, ~/.bash_profile
git - ~/.gitconfig
vim - ~/.vimrc and ~/.vimrc and the ~/.vim folder
ssh - ~/.ssh/config
tmux - ~/.tmux.conf
```

5.Portability

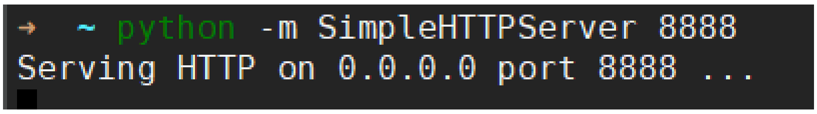
I have done ssh(secure shell) already.

```
#typing follow cmd on hadoop101
ssh hadoop102
ssh hadoop101
```



```
➜  ~ ssh hadoop102
Last login: Sun Jul 11 00:17:33 2021 from hadoop101
➜  ~ ssh hadoop103
Last login: Sun Jul 11 00:16:58 2021 from 192.168.10.1
➜  ~ 
```

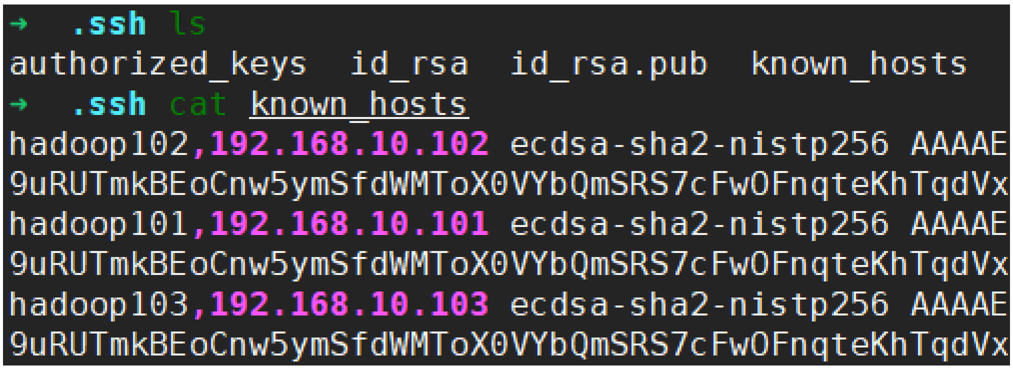
```
python -m SimpleHTTPServer 8888
curl localhost:9999
```



```
➜  ~ python -m SimpleHTTPServer 8888
Serving HTTP on 0.0.0.0 port 8888 ...
```

ssh config

```
cat known hosts #查看已经受信任的主机，以及他们的密钥
```

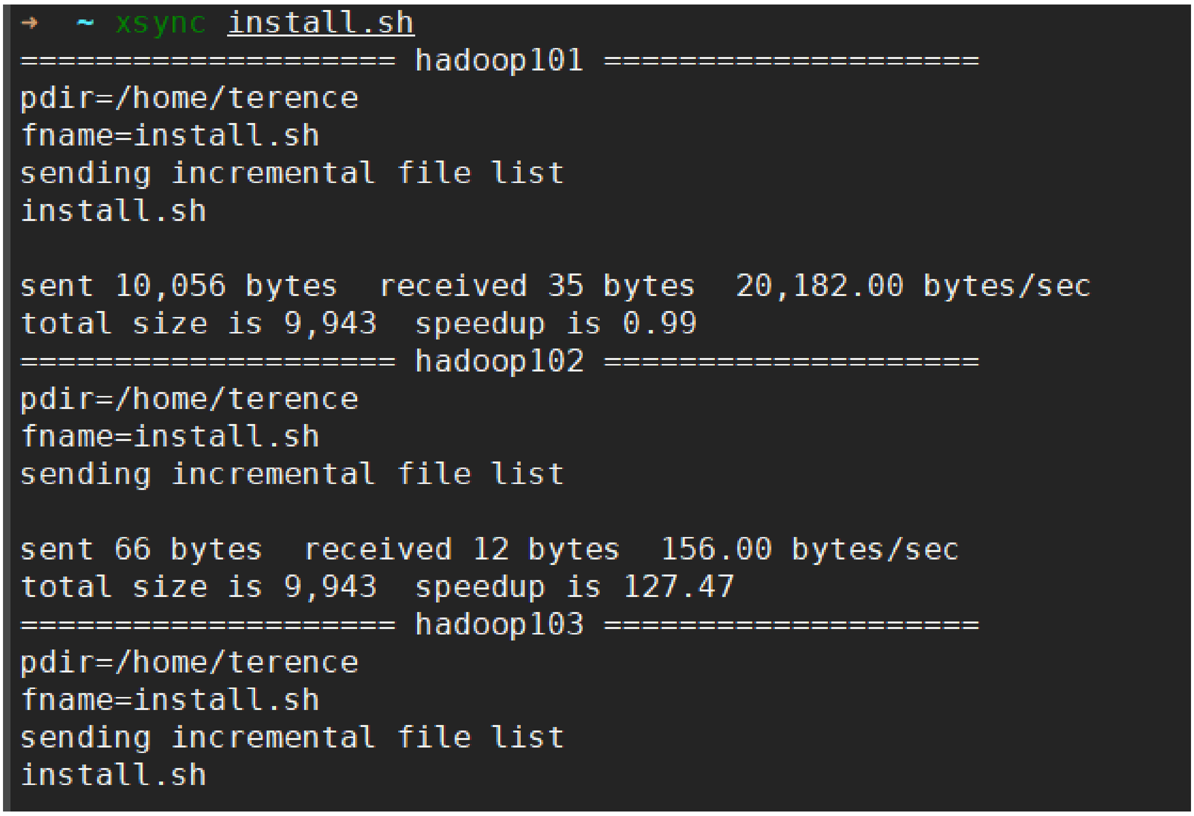


```
➜  .ssh ls
authorized_keys  id_rsa  id_rsa.pub  known_hosts
➜  .ssh cat known_hosts
hadoop102,192.168.10.102 ecdsa-sha2-nistp256 AAAAE
9uRUTmkBEoCnw5ymSfdWMToX0VYbQmSRS7cFwOFnqteKhTqdVx
hadoop101,192.168.10.101 ecdsa-sha2-nistp256 AAAAE
9uRUTmkBEoCnw5ymSfdWMToX0VYbQmSRS7cFwOFnqteKhTqdVx
hadoop103,192.168.10.103 ecdsa-sha2-nistp256 AAAAE
9uRUTmkBEoCnw5ymSfdWMToX0VYbQmSRS7cFwOFnqteKhTqdVx
```

可以进行文件的分发：

```
#!/bin/bash
```

```
#1 获取输入参数个数，如果没有参数，直接退出
pcount=$#
if [ $pcount -lt 1 ]
then
    echo Not Enough Arguement!
    exit;
fi

#2. 遍历集群所有机器
# 也可以采用：
# for host in hadoop{101..103};
for host in hadoop101 hadoop102 hadoop103
do
    echo ====================    $host    ====================
    #3. 遍历所有目录，挨个发送
    for file in $@
    do
        #4 判断文件是否存在
        if [ -e $file ]
        then
            #5. 获取父目录
            pdir=$(cd -P $(dirname $file); pwd)
            echo pdir=$pdir

            #6. 获取当前文件的名称
            fname=$(basename $file)
            echo fname=$fname

            #7. 通过ssh执行命令：在$host主机上递归创建文件夹（如果存在该文件夹）
            ssh $host "mkdir -p $pdir"

            #8. 远程同步文件至$host主机的$USER用户的$pdir文件夹下
            rsync -av $pdir/$fname $USER@$host:$pdir
        else
            echo $file does not exists!
        fi
    done
done

xsync install.sh
```



```
➜  ~ xsync install.sh
==================== hadoop101 ====================
pdir=/home/terence
fname=install.sh
sending incremental file list
install.sh

sent 10,056 bytes  received 35 bytes  20,182.00 bytes/sec
total size is 9,943  speedup is 0.99
==================== hadoop102 ====================
pdir=/home/terence
fname=install.sh
sending incremental file list

sent 66 bytes  received 12 bytes  156.00 bytes/sec
total size is 9,943  speedup is 127.47
==================== hadoop103 ====================
pdir=/home/terence
fname=install.sh
sending incremental file list
install.sh
```

6.Mosh

```
Mosh表示移动Shell(Mobile Shell)，是一个用于从客户端跨互联网连接远程服务器的命令行工具。它能用于SSH连接，但是比Secure Shell功能更多。
yum install mosh
```

7.SSh -N -f

```
-N      Do not execute a remote command.  This is useful for just forwarding ports.
-f      Requests ssh to go to background just before command execution.  This is useful if ssh is going to ask for passwords or passphrases, but the user wants it in the background.  This implies -n.  The recommended way to start X11 programs at a remote site is with something like ssh -f host xterm.

If the ExitOnForwardFailure configuration option is set to ``yes'', then a client started with -f will wait for all remote port forwards to be successfully established before placing itself in the background.

ssh -fN -L 9999:localhost:8888 hadoop102
```