

Contents

1	Sysvinit 项目测试案例分析	1
1.1	测试 wall 命令	1
1.1.1	编译 sysvinit 项目获得可执行文件 wall	1
1.1.2	执行 wall 命令加消息参数	1
1.1.3	打开新的 Terminal 窗口，再次执行该命令	1
1.2	测试 mesg 命令	2
1.2.1	使用 tty 命令查看终端名称 tty	2
1.2.2	使用 who 命令查看当前登录用户名 user	2
1.2.3	使用 write 命令给当前终端发消息	2
1.2.4	使用 mesg n 禁止消息接收功能	2
1.3	#测试 mesg 命令	3

1 Sysvinit 项目测试案例分析

1.1 测试 wall 命令

1.1.1 编译 sysvinit 项目获得可执行文件 wall

```
$ cd sysvinit-2.88dsf/
$ make
make -C src all
make[1]: Entering directory `/home/akaedu/Github/sysvinit/
sysvinit-2.88dsf/src'
make[1]: Nothing to be done for `all'.
make[1]: Leaving directory `/home/akaedu/Github/sysvinit/
sysvinit-2.88dsf/src'
$ ls src/wall -l
-rwxrwxr-x 1 akaedu akaedu 13243 Jun 22 14:49 src/wall
$
```

查看 src 目录下已经生成 wall 命令。

1.1.2 执行 wall 命令加消息参数

```
$ src/wall "hello msg"
$
Broadcast message from akaedu@ubuntu (pts/1) (Sun Jun 23 08:59:32 2013):

hello msg
```

能够看到有广播的消息显示在终端窗口。

1.1.3 打开新的 Terminal 窗口，再次执行该命令

```
$
Broadcast message from akaedu@ubuntu (pts/1) (Sun Jun 23 09:00:36 2013):

hello msg
```

此时，新打开的终端窗口也能够看到有广播的消息显示出来。

1.2 测试 mesg 命令

1.2.1 使用 tty 命令查看终端名称 tty

```
$ tty
/dev/pts/1
$
```

获得终端名称 pts/1

1.2.2 使用 who 命令查看当前登录用户名 user

```
$ who
akaedu    tty2          2013-06-22 18:44
akaedu    pts/1         2013-06-22 18:45 (:0.0)
$
```

1.2.3 使用 write 命令给当前终端发消息

```
$ write akaedu pts/1
```

```
Message from akaedu@ubuntu on pts/1 at 09:13 ...
hello msg
hello msg
test write cmd
test write cmd
EOF
$
```

可以看到 `write` 命令能够实现自己给自己当前的终端发消息。按 `ctrl+d` 结束输入。

1.2.4 使用 `mesg n` 禁止消息接收功能

```
$ mesg n
$ write akaedu pts/1
write: write: you have write permission turned off.

write: akaedu has messages disabled on pts/1
```

可以看到当前终端如果使用 `mesg n` 命令之后，就不再接收 `write` 发来的消息。但如果用 `wall` 命令发送仍然可以接收。

`write` 也支持给其他终端发消息，做法是打开新的 Terminal 窗口，同样需要查看登录用户名和终端名称。

1.3 #测试 `mesg` 命令