**包括函数定义、举例应用：**

**定义函数：**

function FUNC\_NAME {

BODY

}

或

FUNC\_NAME() {

BODY

}

例子1：

showecho() {

echo “hi”

}

for I in {1..10}; do

echo $I

showecho

done

例子2：

showmenu() {

cat << EOF

u|U show UID

g|G show GID

s|S show shell

q|Q quit;

EOF

}

showmenu

read –p “your choice” CHOICE

case $CHOICE in

u|U)

id –u root;;

g|G)

id –g root;;

s|S)

echo “/bin/bash”;;

\*)

showmenu

read –p “Reselect:” CHOIEC

esac

**函数返回值**：默认最后语句的执行结果

**自定义返回值**：return #

test.h:

JUSER() {

id –u root&> /dev/null;then

return 0

else

return 1

fi

}

JUSER &> /dev/null

echo $?

**接受参数：**

例子1：

SUM() {

local SUMARY=0

SUMARY=$[$1+$2]

echo $SUMARY

}

SUM 5 6

//输出11

例子2：接受start和stop作为参数启动或者停止服务

start() {

/usr/sbin/vsftpd &

[ $? –eq 0 ] && return 0 || return 1

}

stop() {

killall vsftpd

[ $? –eq 0 ] && return 0 || return 1

}

case $1 in

start)

start

[ $? –eq 0 ] && echo “starting vsftpd”

;;

stop)

stop ;;

\*)

echo “Usage:’basename $0’ {start|stop}” //basename 基名，最后一段的名字

改进：进程已存在就提示：

/var/lock/subsys/vsftpd文件

例子：（**有缺陷**，但基本可以体现含义）

LOCKFILE=/var/lock/subsys/vsftpd

start() {

**[ ! –e $LOCKFILE ] &&** /usr/sbin/vsftpd &

[ $? –eq 0 ] &**&touch $LOCKFILE &**& return 0 || return 1

}

stop() {

**[ ! –e $LOCKFILE ] &&** killall vsftpd

[ $? –eq 0 ] && return 0 || return 1

}

case $1 in

start)

start

[ $? –eq 0 ] && echo “starting vsftpd”

;;

stop)

stop

if [ $? –eq 0 ]; then

echo –e “stop ok”

else

echo –e “stop failed”

fi

;;

restart)

stop

sleep 1

start

;;

\*)

echo “Usage:’basename $0’ {start|stop}” //basename 基名，最后一段的名字

;;

esac

改进：将输出放到函数中

**系统自带的很好的函数学习例子：**

/etc/rc.d/init.d/functions

用less查看，关键的daemon() killproc() pidofproc() stasus()和一堆的echo\_success等函数