系统环境变量：/etc/profile

用户环境变量：find –type f –name ‘.bash\_profile’ source .bash\_profile

给shell文件赋权限：chmod 777 xx.sh

常用系统变量：$HOME $PWD $USER $SHELL set 显示

自定义变量：

A=5 echo $A unset A 变量提升为全局变量 export A

变量赋值等号两侧不能有空格，变量默认都是字符串类型，无法直接进行数值运算。变量的值中如果有空格需要用单引号或者双引号包起来。

$0 该脚本的名字 $1-9 输入的变量 变量个数超过10 需要加{} ${10} echo ‘$1 $2 $3’

$# 输入变量的个数 常用于循环

$\* 将输入的所有变量作为一个整体看待

$@ 将输入的每一个变量分开看待。

$? 返回最后一个命令的返回状态 成功为0 其他为不成功。

运算符： + - \\* / % 加 减 乘 除 取余

运算： 1.$((运算式)) $[运算式] 2.expr 运算符间要有空格

(2+3)\*4 expr `expr 2 + 3` \\* 4 $[(2+3)\*4]

条件判断：[ condition ] 注意前后有空格 如果条件为变量则非空即为true []为false

1.整数比较：-lt –le -gt -ge -eq -ne

2.字符串比较：=

3.文件权限判断：-r -w -x 读 写 执行

4.文件类型判断： -f -d -e 文件 文件夹 是否存在

5.多条件判断： && 前一条命令执行成功才执行后一条命令 || 前一条命令执行失败才执行后一条命令

[ condition ] && echo ok || echo notok 类似于if else

[ -x demo.sh ] echo $?

流程控制：

if:

if [ $1 –eq ‘1’ ]

then

echo ‘输入了1’

elif [ $1 –eq ‘2’ ]

then

echo ‘输入了2’

fi

case:

case $1 in

‘1’)

echo ‘输入了1’

;;

‘2’)

echo ‘输入了2’

;;

\*)

echo ‘输入了其他’

;;

esac

for:

for i in “$\*” “$1$2…..$n”

do

echo “$i”

done

for i in “$@” ‘$1’’$2’…..’$n’

do

echo “$i”

done

while:

s=0

i=1

while [ $i –le 100]

do

s=$[ $s+$i ]

$i=$[ $i+1 ]

done

常用的函数：

1.自定义函数：

function sum()

{

s=0

s=$[ $1+$2 ]

echo “$s”

}

read –p “please enter the number1:” n1

read –p “please enter the number2:” n2

sum n1 n2

2.basename:  
 basename /home/a/b.txt b.txt

basename /home/a/b.txt .txt b

3.dirname

dirname /home/a/b.txt /home/a

4.cut 作用是按照指定的分隔符选列

echo $PATH

/usr/lib64/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/home/fsdm/bin

echo $PATH | cut -d: -f 2-

/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/home/fsdm/bin

5.sed