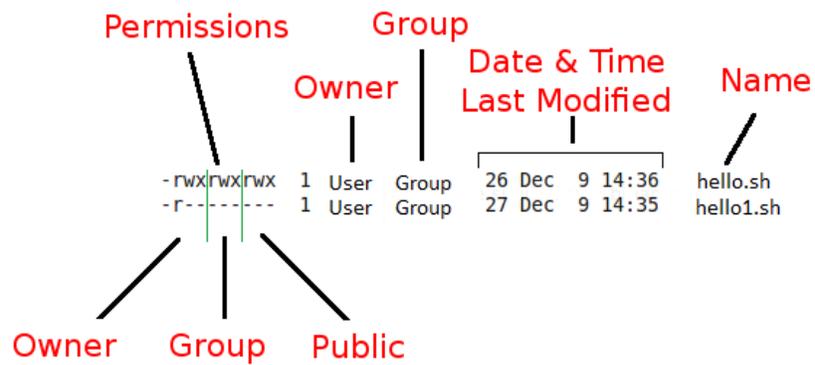


1

Getting Started and Working with Shell Scripting



2

Drilling Deep into Process Management, Job Control, and Automation

```
student@ubuntu:~$ pstree
init--NetworkManager--dhclient
                        |--dnsmasq
                        --2*[{NetworkManager}]
--accounts-daemon--{accounts-daemon}
--acpid
--anacron--sh--run-parts--apt--sleep
--atd
--avahi-daemon--avahi-daemon
--bamfdaemon--2*[{bamfdaemon}]
--bluetoothd
--colord--2*[{colord}]
--console-kit-dae--64*[{console-kit-dae}]
--cron
--cupsd
--2*[dbus-daemon]
--dbus-launch
--dconf-service--2*[{dconf-service}]
--gconfd-2
--geoclue-master
--6*[getty]
```

```
student@ubuntu:~$  
student@ubuntu:~$ ps  
  PID TTY          TIME CMD  
 2621 pts/0    00:00:00 bash  
 2797 pts/0    00:00:00 ps  
student@ubuntu:~$  
student@ubuntu:~$
```

```
student@ubuntu:~$ ps -f  
UID          PID  PPID  C  STIME TTY          TIME CMD  
student    2621  2610  0  19:14 pts/0    00:00:00 bash  
student    2864  2621  0  19:31 pts/0    00:00:00 ps -f  
student@ubuntu:~$  
student@ubuntu:~$
```

```
student@ubuntu:~$  
student@ubuntu:~$  
student@ubuntu:~$ ps -lf  
F S UID          PID  PPID  C  PRI  NI ADDR SZ WCHAN  STIME TTY          TIME CMD  
0 S student    2621  2610  0  80   0 - 1817 wait  19:14 pts/0    00:00:00 bash  
0 R student    2928  2621  0  80   0 - 1237 -    19:35 pts/0    00:00:00 ps -lf  
student@ubuntu:~$  
student@ubuntu:~$  
student@ubuntu:~$
```

```
student@ubuntu:~$
student@ubuntu:~$ ps -ef
UID      PID  PPID  C  STIME TTY          TIME CMD
root      1    0    0  19:06 ?           00:00:01 /sbin/init
root      2    0    0  19:06 ?           00:00:00 [kthreadd]
root      3    2    0  19:06 ?           00:00:00 [ksoftirqd/0]
root      5    2    0  19:06 ?           00:00:00 [kworker/0:0H]
root      7    2    0  19:06 ?           00:00:00 [rcu_sched]
root      8    2    0  19:06 ?           00:00:00 [rcu_bh]
root      9    2    0  19:06 ?           00:00:00 [migration/0]
root     10    2    0  19:06 ?           00:00:03 [watchdog/0]
root     11    2    0  19:06 ?           00:00:00 [khelper]
root     12    2    0  19:06 ?           00:00:00 [kdevtmpfs]
root     13    2    0  19:06 ?           00:00:00 [netns]
root     14    2    0  19:06 ?           00:00:00 [writeback]
root     15    2    0  19:06 ?           00:00:00 [kintegrityd]
root     16    2    0  19:06 ?           00:00:00 [bioset]
root     17    2    0  19:06 ?           00:00:00 [kworker/u17:0]
root     18    2    0  19:06 ?           00:00:00 [kblockd]
root     19    2    0  19:06 ?           00:00:00 [ata_sff]
root     20    2    0  19:06 ?           00:00:00 [khubd]
root     21    2    0  19:06 ?           00:00:00 [md]
root     22    2    0  19:06 ?           00:00:00 [devfreq_wq]
```

```
student@ubuntu:~$
student@ubuntu:~$ ps
  PID TTY          TIME CMD
 2621 pts/0    00:00:00 bash
 3796 pts/0    00:00:00 sleep
 3797 pts/0    00:00:00 ps
student@ubuntu:~$
student@ubuntu:~$
student@ubuntu:~$ kill 3796
[1]+  Terminated                  sleep 10000
student@ubuntu:~$
student@ubuntu:~$ ps
  PID TTY          TIME CMD
 2621 pts/0    00:00:00 bash
 3799 pts/0    00:00:00 ps
student@ubuntu:~$
```

```
student@ubuntu:~$  
student@ubuntu:~$ ps  
  PID TTY          TIME CMD  
 2621 pts/0    00:00:00 bash  
 3828 pts/0    00:00:00 sleep  
 3829 pts/0    00:00:00 ps  
student@ubuntu:~$  
student@ubuntu:~$ pkill sleep  
[1]+  Terminated                  sleep 10000  
student@ubuntu:~$  
student@ubuntu:~$ ps  
  PID TTY          TIME CMD  
 2621 pts/0    00:00:00 bash  
 3832 pts/0    00:00:00 ps  
student@ubuntu:~$  
student@ubuntu:~$
```

```
student@ubuntu:~$  
student@ubuntu:~$ ps  
  PID TTY          TIME CMD  
 2621 pts/0    00:00:00 bash  
 3868 pts/0    00:00:00 ps  
student@ubuntu:~$  
student@ubuntu:~$  
student@ubuntu:~$ sleep 10000  
  
^C  
student@ubuntu:~$  
student@ubuntu:~$
```

```
student@ubuntu:~$
student@ubuntu:~$ sleep 10000 &
[1] 3885
student@ubuntu:~$ sleep 20000 &
[2] 3887
student@ubuntu:~$ sleep 30000 &
[3] 3888
student@ubuntu:~$ sleep 40000 &
[4] 3890
student@ubuntu:~$ jobs
[1]  Running                sleep 10000 &
[2]  Running                sleep 20000 &
[3]- Running                sleep 30000 &
[4]+ Running                sleep 40000 &
student@ubuntu:~$
student@ubuntu:~$
```

```
student@ubuntu:~$ fg 3
sleep 30000

^Z
[3]+  Stopped                sleep 30000
student@ubuntu:~$
student@ubuntu:~$ jobs
[1]  Running                sleep 10000 &
[2]  Running                sleep 20000 &
[3]+  Stopped                sleep 30000
[4]-  Running                sleep 40000 &
student@ubuntu:~$
```

```

top - 22:05:50 up 2:58, 2 users, load average: 0.04, 0.03, 0.05
Tasks: 171 total, 2 running, 168 sleeping, 1 stopped, 0 zombie
Cpu(s): 0.3%us, 0.3%sy, 0.0%ni, 99.0%id, 0.3%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 2063524k total, 942860k used, 1120664k free, 98244k buffers
Swap: 2094076k total, 0k used, 2094076k free, 444288k cached

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1189	root	20	0	107m	65m	12m	S	1.0	3.3	0:22.54	Xorg
2264	student	20	0	250m	54m	31m	S	0.3	2.7	0:09.02	unity-2d-shell
2275	student	20	0	102m	30m	22m	S	0.3	1.5	0:20.20	vmtoolsd
2610	student	20	0	89996	19m	10m	S	0.3	1.0	0:05.55	gnome-terminal
4071	student	20	0	2856	1168	872	R	0.3	0.1	0:00.54	top
1	root	20	0	3768	2092	1284	S	0.0	0.1	0:01.51	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.22	ksoftirqd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H
7	root	20	0	0	0	0	S	0.0	0.0	0:00.68	rcu_sched
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
9	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
10	root	RT	0	0	0	0	S	0.0	0.0	0:03.96	watchdog/0
11	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	khelper
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
13	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	netns
14	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	writeback
15	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kintegrityd
16	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	bioaset

```
top - 22:05:50 up 2:58, 2 users, load average: 0.04, 0.03, 0.05
```

```
Tasks: 171 total, 2 running, 168 sleeping, 1 stopped, 0 zombie
```

```
Cpu(s): 0.3%us, 0.3%sy, 0.0%ni, 99.0%id, 0.3%wa, 0.0%hi, 0.0%si, 0.0%st
```

```
Mem: 2063524k total, 942860k used, 1120664k free, 98244k buffers
Swap: 2094076k total, 0k used, 2094076k free, 444288k cached
```

```

student@ubuntu:~$ iostat
Linux 3.13.0-32-generic (ubuntu)      02/04/2015      _i686_ (1 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.37    0.08   0.49   0.44    0.00   98.62

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sda                  3.05         38.06         10.34       478399     129968

```

```

student@ubuntu:~$ vmstat
procs -----memory----- --swap-- -----io----- -system-- ----cpu----
 r b swpd free buff cache si so bi bo in cs us sy id wa
 2 0  0 1124376 98496 444392 0 0 38 10 40 92 0 0 99 0
student@ubuntu:~$

```

```

student@ubuntu:~$ sar -u 2 3
Linux 3.13.0-32-generic (ubuntu)      02/04/2015      _i686_ (1 CPU)

10:44:37 PM    CPU    %user   %nice %system %iowait  %steal   %idle
10:44:39 PM    all     0.50    0.00   1.49   1.00    0.00   97.01
10:44:41 PM    all     0.51    0.00   0.51   0.00    0.00   98.99
10:44:43 PM    all     0.51    0.00   0.51   0.00    0.00   98.99
Average:         all     0.50    0.00   0.84   0.34    0.00   98.32
student@ubuntu:~$

```

```

*      *      *      *      *      command to be executed
-      -      -      -      -
|      |      |      |      |
|      |      |      |      +----- day of week (0 to 6) (Sunday=0)
|      |      |      +----- month (1 to 12)
|      |      +----- day of          month (1 to 31)
|      +----- hour (0 to 23)
+----- min (0 to 59)

```

3

Using Text Processing and Filters in Your Scripts

```
student@ubuntu: ~  
student@ubuntu:~$ cut -d: -f1,3 /etc/passwd  
root:0  
daemon:1  
bin:2  
sys:3  
sync:4  
games:5  
man:6  
lp:7  
mail:8  
news:9  
uucp:10  
proxy:13  
www-data:33  
backup:34  
list:38
```

```
student@ubuntu:~$ cut -d: -f1-5 /etc/passwd
root:x:0:0:root
daemon:x:1:1:daemon
bin:x:2:2:bin
sys:x:3:3:sys
sync:x:4:65534:sync
games:x:5:60:games
man:x:6:12:man
lp:x:7:7:lp
mail:x:8:8:mail
news:x:9:9:news
uucp:x:10:10:uucp
proxy:x:13:13:proxy
```

```
student@ubuntu:~/work$ touch file{1,2,3}
student@ubuntu:~/work$ ls
file1 file2 file3
student@ubuntu:~/work$
```

```
student@ubuntu:~/work$ mkdir directory{1,2,3}{a,b,c}
student@ubuntu:~/work$ ls
directory1a directory1c directory2b directory3a directory3c
directory1b directory2a directory2c directory3b
```

```
student@ubuntu:~/work$ touch file{a..z}
student@ubuntu:~/work$ ls
filea filed fileg filej filem filep files filev filey
fileb filee fileh filek filen fileq filet filew filez
filec filef filei filel fileo filer fileu filex
```

Char	Meaning	Example	Possible Output
>	Output Redirection	\$ ls > ls.out	Output of ls command is redirected(overwritten) to ls.out file
>>	Output Redirection (append)	\$ ls >> ls.out	Output of ls command is redirected(append) to ls.out file
<	Input Redirection	\$ tr 'a' 'A' < file1	The tr command read input from file1 instead of keyboard(stdin)
`cmd` or \$(cmd)	Command substitution	\$echo `date` or \$ echo \$(date)	The command date is substituted with the result and sent to echo for display
	OR Conditional Execution	\$ test \$x -gt 10 \$x -lt 15	Check whether x value is greater than 10 or less than 15
&&	AND Conditional Execution	\$ test \$x -gt 10 && \$x -lt 15	Check whether x value is greater than 10 and less than 15

4

Working with Commands

```
student@ubuntu:~$ echo $(cal)
April 2015 Su Mo Tu We Th Fr Sa 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
19 20 21 22 23 24 25 26 27 28 29 30
```

```
student@ubuntu:~$ echo "$(cal)"
April 2015
Su Mo Tu We Th Fr Sa
      1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30
```

```
student@ubuntu:~$ pwd
/home/student
student@ubuntu:~$
student@ubuntu:~$ dirname="$(basename $(pwd))"
student@ubuntu:~$
student@ubuntu:~$ echo $dirname
student
```

```
student@ubuntu:~$ w; date
15:57:23 up 8:02, 2 users, load average: 0.01, 0.03, 0.05
USER  TTY  FROM          LOGIN@  IDLE   JCPU   PCPU WHAT
student tty7                01:05   15:01m 15.40s 0.18s gnome-session --session=ubuntu
student pts/1          :0      01:05   3.00s  0.19s 0.00s w
Fri Mar 20 15:57:23 IST 2015
```

5

Exploring Expressions and Variables

```
SCRIPT BEGINS  
Hello student!  
  
Today's date and time:  
Tue Dec 15 17:14:39 IST 2015  
  
The value of my_num is 50  
The value of my_day is Sunday  
  
SCRIPT FINISHED!!
```

```
Earth  
Earth  
$planet  
$planet  
Enter some text  
Venus  
$planet now equals Venus
```

```

student@ubuntu:~/Desktop/work$
student@ubuntu:~/Desktop/work$ env
SSH_AGENT_PID=2251
GPG_AGENT_INFO=/tmp/keyring-uthSRq/gpg:0:1
TERM=xterm
SHELL=/bin/bash
XDG_SESSION_COOKIE=ca107c3f47929bd197ef224e00000002-1429721473.254990-367780197
WINDOWID=62914565
GNOME_KEYRING_CONTROL=/tmp/keyring-uthSRq
USER=student
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;0
1:or=40;31;01:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31
:*.tgz=01;31:*.arj=01;31:*.taz=01;31:*.lzh=01;31:*.lza=01;31:*.tlz=01;31:*.txz=01;31:*.
zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lz=01;31:*.xz=01;31:*.bz2=01;31:*.
bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=0
1;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31
:*.rz=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.p
pm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg
=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=0
1;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;3
5:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.a

```

Variable	Description
HOME	the user's home directory
PATH	the search path for commands
PWD	current working directory
IFS	the internal field separator; i.e., the character that separates individual arguments from each other
PS1	the primary shell prompt
PS2	the secondary shell prompt
PS3	the tertiary shell prompt (see select)
?	the exit status or (return value) of the most recent child process
\$	the process ID of the current shell itself
#	the number of arguments passed to the shell
0-9	argument 0 (usually the command itself), argument 1, and so on, as passed to the shell
*	all arguments (with the exception of argument 0) as separate words or arguments
@	all arguments (with the exception of argument 0) as separate words or arguments

```

Your Timezone is set to: America/Los_Angeles
Wed Apr 22 10:52:53 PDT 2015
Your Timezone is set to: Asia/Tokyo
Thu Apr 23 02:52:53 JST 2015
Your Timezone is set to: Asia/Kolkata
Wed Apr 22 23:22:53 IST 2015

```

```
student@ubuntu:~/Desktop/work$  
student@ubuntu:~/Desktop/work$ ./export1.sh  
  
The second variable bar  
student@ubuntu:~/Desktop/work$
```

```
student@ubuntu:~/Desktop/work$ bash set_02.sh  
The date is Thu Apr 23 00:36:53 IST 2015  
The month is Apr  
student@ubuntu:~/Desktop/work$ █
```

```
student@ubuntu:~/Desktop/work$ bash set_03.sh  
Executing script set_03.sh  
  
One two three in German are:  
eins  
zwei  
drei  
name phone address birthdate salary  
At this time $1 = name and $4 = birthdate
```

```
student@ubuntu:~/Desktop/work$ bash shift_02.sh 1 2 3 4 5 6 7 8 9 10 11 12 13
All parameters before shift
$#: 13
$@: 1 2 3 4 5 6 7 8 9 10 11 12 13
$*: 1 2 3 4 5 6 7 8 9 10 11 12 13

$1 $2 $9 $10 are: 1 2 9 10

All parameters after one shift
$#: 12
$@: 2 3 4 5 6 7 8 9 10 11 12 13
$*: 2 3 4 5 6 7 8 9 10 11 12 13

$1 $2 $9 are: 2 3 10
All parameters after shift 2
$#: 10
$@: 4 5 6 7 8 9 10 11 12 13
$*: 4 5 6 7 8 9 10 11 12 13

$1 $2 $9 are: 4 5 12
${10}: 13
```

```
student@ubuntu:~/Desktop/work$ ./default_argument_2.sh One Two
One
Two
student@ubuntu:~/Desktop/work$
student@ubuntu:~/Desktop/work$ ./default_argument_2.sh One
One
One
student@ubuntu:~/Desktop/work$ █
```

6

Neat Tricks with Shell Scripting

```
student@ubuntu:~/work$ ./file_08.sh
student@ubuntu:~/work$ cat output.txt
processor      : 0
vendor_id     : GenuineIntel
cpu family    : 6
model         : 60
model name    : Intel(R) Core(TM) i7-4600M CPU @ 2.90GHz
stepping      : 3
microcode     : 0x17
cpu MHz       : 2893.510
cache size    : 4096 KB
physical id   : 0
siblings      : 1
core id       : 0
cpu cores     : 1
apicid        : 0
initial apicid : 0
fdiv_bug     : no
f00f_bug     : no
coma_bug     : no
fpu          : yes
```

7

Performing Arithmetic Operations in Shell Scripts

```
student@ubuntu:~$ help let
let: let arg [arg ...]
    Evaluate arithmetic expressions.

    Evaluate each ARG as an arithmetic expression.  Evaluation is done in
    fixed-width integers with no check for overflow, though division by 0
    is trapped and flagged as an error.  The following list of operators is
    grouped into levels of equal-precedence operators.  The levels are listed
    in order of decreasing precedence.

    id++, id--      variable post-increment, post-decrement
    ++id, --id     variable pre-increment, pre-decrement
    -, +           unary minus, plus
    !, ~           logical and bitwise negation
    **            exponentiation
    *, /, %       multiplication, division, remainder
    +, -         addition, subtraction
    <<, >>       left and right bitwise shifts
    <=, >=, <, > comparison
    ==, !=       equality, inequality
    &            bitwise AND
    ^           bitwise XOR
    |           bitwise OR
    &&          logical AND
    ||          logical OR
    expr ? expr : expr
                  conditional operator
    =, *=, /=, %=,
    +=, -=, <<=, >>=,
    &=, ^=, |=  assignment

    Shell variables are allowed as operands.  The name of the variable
    is replaced by its value (coerced to a fixed-width integer) within
    an expression.  The variable need not have its integer attribute
    turned on to be used in an expression.

    Operators are evaluated in order of precedence.  Sub-expressions in
    parentheses are evaluated first and may override the precedence
    rules above.

    Exit Status:
```

8

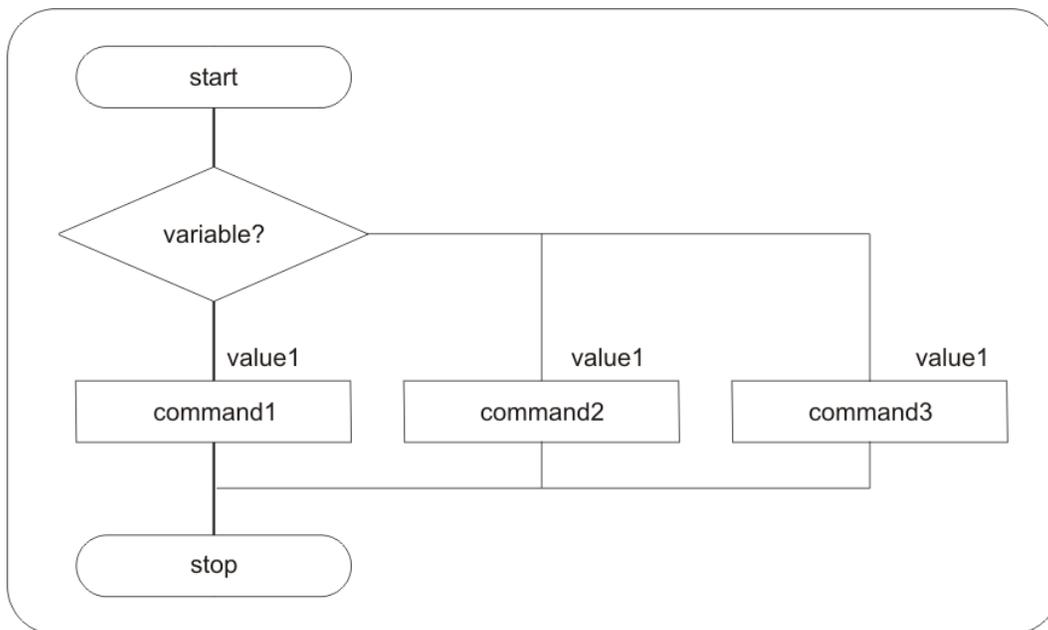
Automating Decision Making in Scripts

Test Operator	Tests True If
[integer_1 -eq integer_2]	integer_1 is equal to integer_2
[integer_1 -ne integer_2]	integer_1 is not equal to integer_2
[integer_1 -gt integer_2]	integer_1 is greater than integer_2
[integer_1 -ge integer_2]	integer_1 is greater than or equal to integer_2
[integer_1 -le integer_2]	integer_1 is less than integer_2
[integer_1 -lt integer_2]	integer_1 is less than or equal to integer_2

Test Operator	Tests True If
-b file_name	Check if file is Block special file
-c file_name	Check if file is Character special file
-d file_name	Check if Directory is existing
-e file_name	Check if File existence
-f file_name	Check if file is Regular file and not a directory
-G file_name	Check if file is existing and is owned by the effective group ID
-g file_name	Check if file has Set-group-ID set
-k file_name	Check if file has Sticky bit set
-L file_name	Check if file is a symbolic link
-p file_name	Check if file is a named pipe
-O file_name	Check if file exists and is owned by the effective user ID
-r file_name	Check if file is readable
-S file_name	Check if file is a socket
-s file_name	Check if file has nonzero size
-t fd	Check if file has fd (file descriptor) and is opened on a terminal
-u file_name	Check if file has Set-user-ID bit set
-w file_name	Check if file is writable
-x file_name	Check if file is executable

Test Operator	Tests True If
[file_1 -nt file_2]	Check if file is newer than file2
[file_1 -ot file_2]	Check if file is file1 is older than file2
[file_1 -ef file_2]	Check if file1 and file2 have the same device or inode numbers

Test Operator	Tests True If
[string_1 -a string_2]	Both string_1 and string_2 are true
[string_1 -o string_2]	Either string_1 or string_2 is true
[! string_1]	Not a string_1 match
[[pattern_1 && pattern_2]]	Both pattern_1 and pattern_2 are true
[[pattern_1 pattern_2]]	Either pattern_1 or pattern_2 is true
[[! pattern]]	Not a pattern match



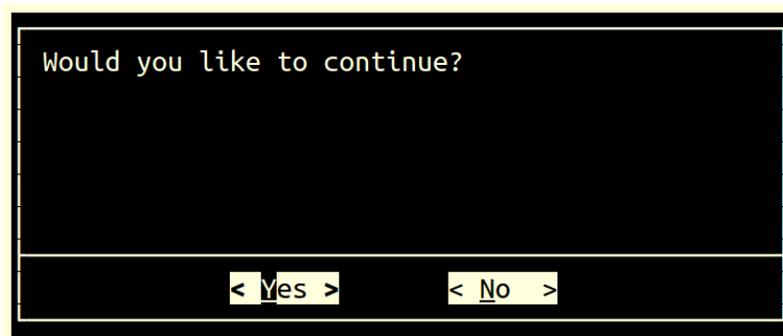
Multiple Branching with `case`

10

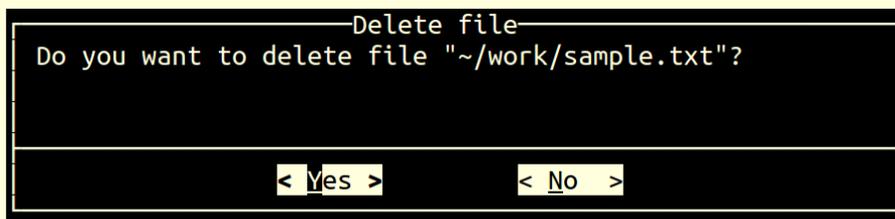
Using Advanced Functionality in Scripts

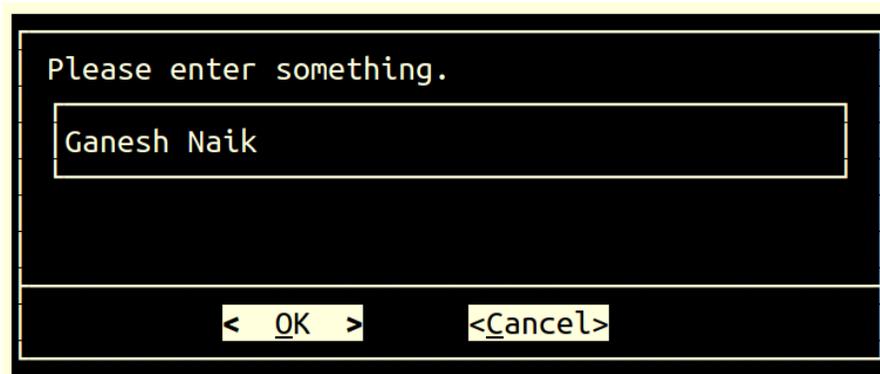
```
student@ubuntu:~/work$ kill -l
1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL      5) SIGTRAP
6) SIGABRT     7) SIGBUS      8) SIGFPE      9) SIGKILL     10) SIGUSR1
11) SIGSEGV    12) SIGUSR2    13) SIGPIPE    14) SIGALRM    15) SIGTERM
16) SIGSTKFLT  17) SIGCHLD    18) SIGCONT    19) SIGSTOP    20) SIGTSTP
21) SIGTTIN    22) SIGTTOU    23) SIGURG     24) SIGXCPU    25) SIGXFSZ
26) SIGVTALRM  27) SIGPROF    28) SIGWINCH   29) SIGIO       30) SIGPWR
31) SIGSYS     34) SIGRTMIN   35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
```



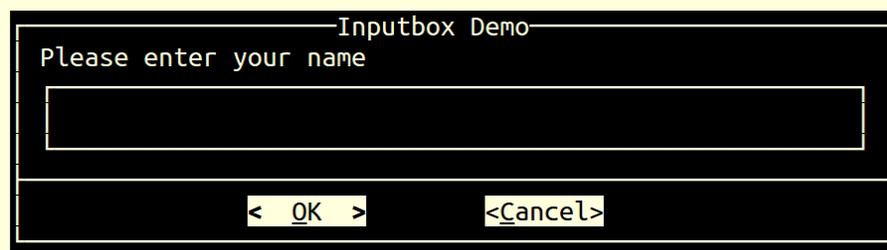


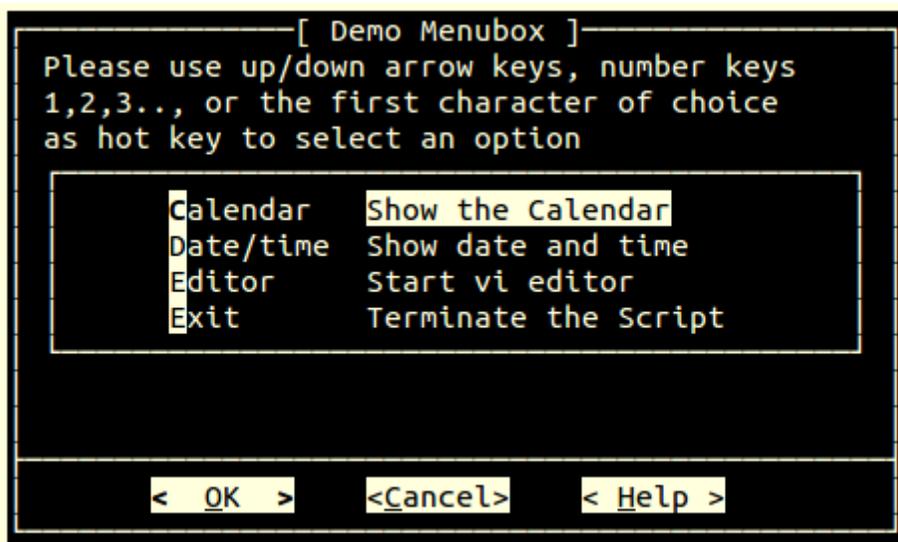
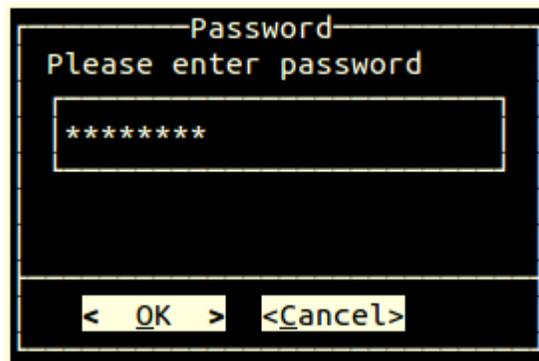
Learning Dialog Yes-No box





Learn Shell Scripting





This is a checklist

- a This is one option
- b This is the second option

< OK >

<Cancel>

This is a selective list, where only one option can be chosen

- a This is the first option
- b This is the second option

< OK >

<Cancel>

This is a progress bar

65%

12

Pattern Matching and Regular Expressions with sed and awk

