December Technical Presentation

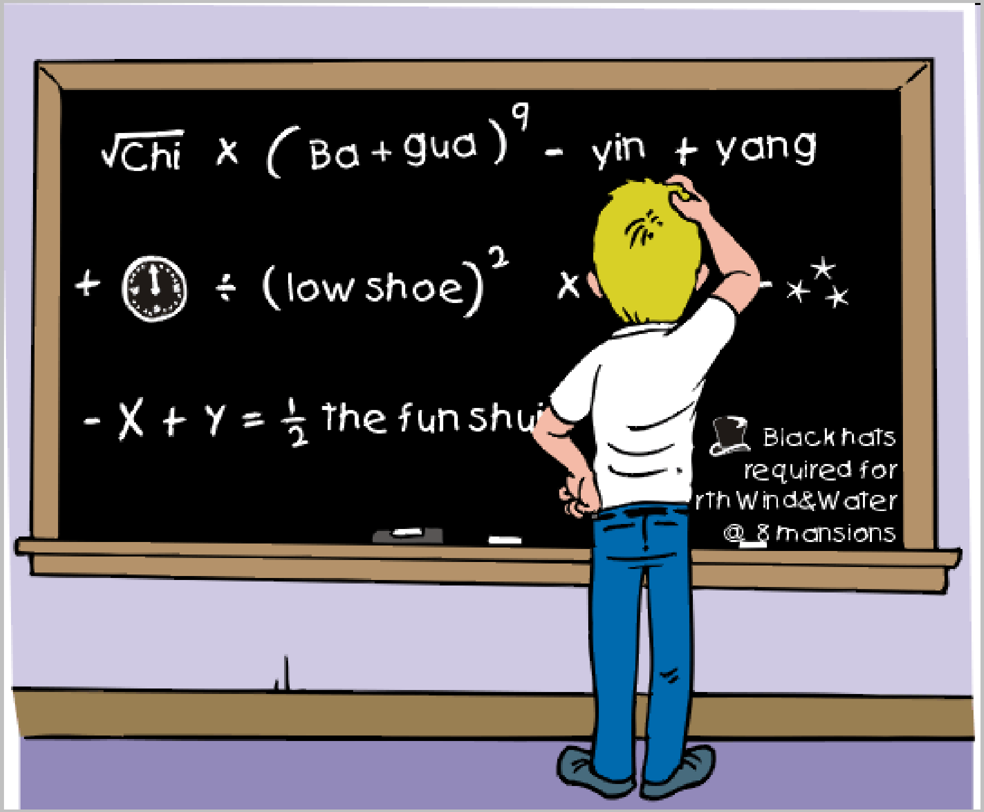
Practical Shell scripting tips

Or how to become even more lazier !

Linux User Group of Mauritius -- http://www.lugm.org/

(A) Not so obvious scripting problem

A simple problem: Run a binary after an automated SSH login and report to user if the binary fails to run


√Chi x (Ba + gua)⁹ - yin + yang
+ ÷ (low shoe)² x
- X + Y = ½ the fun shu
Black hats required for rth Wind&Water @ 8 mansions

(Re)viewing some basic concepts

```
logan@wawa:$ ls | less
```

The OS allocates a pipe (FIFO queue)
Takes the output from ls and sends it through the pipe
less in turn receives the output (now it's considered an input to less) and then Does some processing with it

```
$ mkfifo foo
$ ls > foo
$ less < foo
```

Cool pipe trick !

Download an mp3, Stream it, and save it (all in one)

```
$ Wget –O - http://www.archive.org/download/minirant3/MiniRantAndRamble3.mp3 | \
tee MiniRantAndRamble3.mp3 | \
mplayer -
```

But Unix pipes have a fundamental limitation !

The next step: 2-way pipes

Sockets are basically pipes that can communicate both ways

Cool socket trick

Testing if a port on a server is open without using netcat or telnet (think embedded environments)

```
exec 3<>/dev/tcp/www.google.mu/80
echo -e "GET / HTTP/1.1\n\n">&3
cat <&3
```

The next level: Going beyond bash v3.0

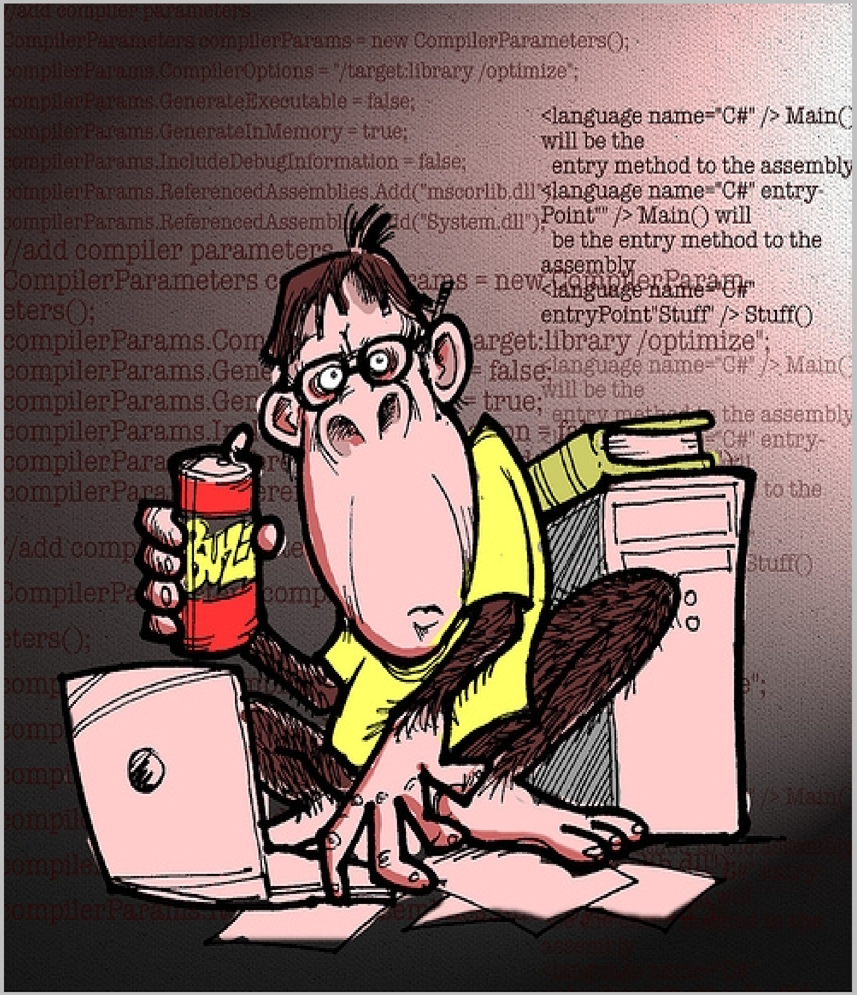
All shells provide the same basic constructs:
Variables, operators, arrays and control structures

coroutines are program components that generalize Subroutines to allow multiple entry points for suspending and resuming execution at certain locations

Solving part of the problem

Automate SSH

```
ssh logan@192.168.0.20 |&
print -p ‘ls‘
read -p
```



What about SSH sessions with passwords ?
How to account for failures ?
First time usage (Accept keys) ?

Enter a neat tool: Expect

```
#!/usr/bin/expect -f
set timeout -1
spawn chmod 700 fw.exp
spawn ssh logan@192.168.0.20 -p 222
expect {
"yes/no"  {send "yes\r" ; exp_continue }
"assword"  { send “xxxxxx\r" ; exp_continue }
"*$*"  { send "egrep 'wget|ftp|curl' /var/log/auth.log\r" ; interact}
default  { puts "OOPS" }

}
```

Another script

```
#!/usr/bin/expect -f
set timeout -1
spawn ssh logan@www.microsoft.com -D 1025
expect "*?assword:*"
send “xxxxx\r"
expect "*logan*"
send "htop\r"
```

Automate FTP

```
# Open an ftp session to a remote server, and wait for a username prompt.
spawn ftp $remote_server
expect "username:"
# Send the username, and then wait for a password prompt.
send "$my_user_id\r"
expect "password:"
# Send the password, and then wait for an ftp prompt.
send "$my_password\r"
expect "ftp>"
# Switch to binary mode, and then wait for an ftp prompt.
send "bin\r"
expect "ftp>"
# Turn off prompting.
send "prompt\r"
expect "ftp>"
# Get all the files
send "mget *\r"
expect "ftp>"
# Exit the ftp session, and wait for a special end-of-file character.
send "bye\r"
expect eof
```

Automate Telnet

```
#!/usr/bin/expect #Where the script should be run from.


set timeout 20 #If it all goes pear shaped the script will timeout after 20 seconds.

set name [lindex $argv 0] #First argument is assigned to the variable name

set user [lindex $argv 1] #Second argument is assigned to the variable user

set password [lindex $argv 2] #Third argument is assigned to the variable password


spawn telnet $name #This spawns the telnet program and connects it to the variable name


expect "login:" #The script expects login

send "$user " #The script sends the user variable

expect "Password:" #The script expects Password

send "$password " #The script sends the password variable

interact #This hands control of the keyboard over two you (Nice expect feature!)
```

Cool ideas that can be expect(ed)

Interactive malware deployment for script kiddies

Interactive server deployement & malware analysis using expect and regex for sysadmins.

Rapidshare/megavideo can be expected to become ``nag-free'' for the casual user

CISCO router reboot for button addicted netadmin

PATRICK FUGIT OLIVIA WILDE JOHN CHO and MATTHEW LILLARD
BICKFORD SHMECKLER'S
COOL IDEAS
Solving The Mysteries Of The Universe, One Girl At A Time

Zen sysadmin