By Matthew Strunk

November 30th, 2015  Unix Systems Programming

Ip Phase 3

Contents

[UNIX Programming 2](#_Toc437370347)

[UNIX Processes (Week 2) 5](#_Toc437370348)

[Differences between Daemons and Processes 9](#_Toc437370349)

[Introduction to Perl (Week 3) TBD 10](#_Toc437370350)

[Writing Perl Programs (Week 4) TBD 11](#_Toc437370351)

[UNIX Tools (Week 5) TBD 12](#_Toc437370352)

[References 13](#_Toc437370353)

[References 13](#_Toc437370354)

# UNIX Programming

#!/bin/bash

######################################################################################

# This script will prompt the system administrator for all valid input parameters.

# Ask if a wall needs to be sent, and if yes, sent the message.

# Ask if this is a shutdown or reboot, and process appropriately.

#

######################################################################################

echo "Do you want to send a wall"

read vwall

string=${vwall,,}

if [ "$string" == "yes" ]; then

 echo "Sending message to all users"|wall

 exit

fi

echo "Do you want to shutdown or restart the server?"

read i

string=${i,,}

if [ "$string" == "shutdown" ]; then

 echo "Rebooting the server"

 shutdown -h +5 "Warning: the server is going down to shut down in 5 minutes."

elif [ "$string" == "restart" ]; then

 echo "Restarting the server"

 shutdown -r +5 "Warning: the server will restart in 5 minutes."

else

 echo "Goodbye"

fi

Shutdown – r will restart the server if you add +5 it will restart the server in 5 minutes.

Vwall will allow you to broadcast a message to all users on the server. Wall will also allow you to broadcast a message to all users on the server as well to. The zero level is the system halt and can be powered down. Number one is single user mode which is very rarely used. 2 is Multie users but no NFS. The third type is command line interface. 4 is for the users. Five is for multiple users or gui mode. The sixth mode is used when rebooting the system.

# UNIX Processes (Week 2)

bin/sh

#Commands for project

read –p ‘Enter database name : ‘ dbasename

read – p ‘Enter user name’ : ‘ username1

read – s –p ‘Enter Password’ : ‘ passwrd1

#Above promps sysadmin for valid input parameters

echo “Please select a menu option”

#moving on!

#1/bin/sh

amenu=”1. Do this thing” ;

bmenu=”2. Do that thing” ;

cmenu=”3. Do some other thing” ;

dmenu=”4. Create a group” ;

emenu=”5. Create a User” ;

fmenu=”6. Drop group” ;

gmenu=”7. Kill User processes “ ;

hmenu=”8. Extras” ;

imenu=” ” ;

jmenu=” “ ;

kmenu=” “ ;

lmenu=” “ ;

wrongchoice () { MSG=”Choose another option” ; }

apick () { thing1 ; }

bpick () { thing2 ; }

cpick () { thing3 ; }

dpick () { cgrp ; }

epick () { cuser ; }

fpick () { dgrp ; }

gpick () { kuproc ; }

hpick () { ext ; }

ipick () { wrongchoice ; }

jpick () { wrongchoice ; }

kpick () { wrongchoice ; }

lpick () { wrongchoice ; }

themenu () {

clear

echo “\t\tPlease select an option”

echo

echo “\t\t\t” $amenu

echo “\t\t\t” $bmenu

echo “\t\t\t” $cmenu

echo “\t\t\t” $dmenu

echo “\t\t\t” $emenu

echo “\t\t\t” $fmenu

echo “\t\t\t” $gmenu

echo “\t\t\t” $hmenu

echo “\t\t\t” $imenu

echo “\t\t\t” $jmenu

echo “\t\t\t” $kmenu

echo “\t\t\t” $lmenu

echo $MSG

}

MSG=

while true

do

themenu

read answer

MSG=

case $answer in

 a|A) apick;;

 b|B) bpick;;

 c|C) cpick;;

 d|D) dpick;;

 e|E) epick;;

 f|F) fpick;;

 g|G) gpick;;

 h|H) hpick;;

 i|I) ipick;;

 j|J) jpick;;

 k|K) kpick;;

 l|L) lpick;;

 x|X) break;;

\*) wrongchoice;;

#breaks on user press X

#menu display above

#1/bin/sh

groupadd -g THISGID $thisgroup

#1/bin/sh

groupdel $thisgroup

#1/bin/sh

#create a User file with users names to input in a list

# the file will be called userslist.txt

if grep –q “^${group}:” /etc/group

for I in ‘more userslist.txt ‘

do

echo $i

adduser $i

done

#now run it

./userslist.txt

#1/bin/sh

for ttt in host1

do

echo $tt

ssh $ttt “rmuser –p thisuser”

done

#remove or drop user

#1/bin/sh

#kill proc

read – p ‘Enter user name’ : ‘ username1

top –U username1

read – p ‘Enter process group name’ : ‘ procgrpname

while [true]

ps –aux

do

echo “Enter the process ID to kill”

read procnum

echo “Are you sure you want to kill $procnum?”

echo “1 for yes, 2 for no”

read chosen

if [ $chosen –eq 1]

then

kill -9 $procnum

fi

done

#Admin user info kill processes

##  Differences between Daemons and Processes

 While processes and daemons are similar in scope and function, they are typically different in one major way, daemons are started when the computer starts and tend not to be stopped unless killed by a process or manually, or the computer shuts off. Processes, and more specifically USER PROCESSES, are individual jobs that the daemon will call to fulfill the request of the user. These are threads that are ended when the job completes and are not kept running as daemons are. Killing a daemon or all the daemons running will cause errors such as inability to access the server via SSH, and most often will require a reboot of the system to restart those daemons required to manage the system.

# Introduction to Perl (Week 3)

#!/usr/bin/perl

 use strict;

 use warnings;

 my $option = 0;

 my @choices = ("null", "date", "who", "pwd", "ls");

 while ($option ne 5) {

 &PrintMenu;

 my $result = `$choices[$option]`;

 print "\n$result\n";

 $option = 0;

 }

 exit(0);

sub PrintMenu {

 while ($option < 1 || $option > 5) {

 print " \n";

 print " Please select from one of the following options: \n";

 print " 1) Show current date and time \n";

 print " 2) Show users currently logged in \n";

 print " 3) Show name of the working directory \n";

 print " 4) Show contents of the working directory \n";

 print " 5) Exit \n";

 print " \n";

 print " Option --> ";

 $option = <STDIN>;

 die ("\nThanks for playing! \n\n") if ($option == 5);

 print "\n\*\*\* Error \*\*\*\* Unknown intput: $option" if ($option < 1 || $option > 5);

 }

}

This is where the options are inserted in order to put in the values of the script. You must choose an option in order for it to work.. The script will allow you to choose any of the 5 options before you exit. The while statement while ($option ne 5) { &PrintMenu; my $result = $choices[$option]`; print "\n$result\n"; $option = 0; }allows you to select the menu and enter a result in the script. These ar the statement tthat are inserted for the value of my choices my $option = 0; my @choices = ("null", "date", "who", "pwd", "ls"); If you choose option one to 5 it will give you a selection.

# Writing Perl Programs (Week 4) TBD

# UNIX Tools (Week 5) TBD

# References

# References

*Perl Introduction*. (2015). Retrieved from Tutorialpoint: http://www.tutorialspoint.com/perl/perl\_introduction.htm