SCRIPT ADMINISTRADOR 1

#!/bin/bash

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

# Sathish Arthar (Sathisharthar {at} gmail.com) - Jan 2014

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

#simple menu driven shell script to to get information about your Linux server / desktop and Do some Users and File operations Quickly.

# Define variables

LSB=/usr/bin/lsb\_release

# Purpose: Display pause prompt

# $1-> Message (optional)

function pause(){

local message="$@"

[ -z $message ] && message="Press [Enter] key to continue..."

read -p "$message" readEnterKey

}

# Purpose - Display a menu on screen

function show\_menu(){

date

echo "---------------------------"

echo " Main Menu"

echo "---------------------------"

echo "1. Operating system info"

echo "2. Hostname and dns info"

echo "3. Network info"

echo "4. Who is online"

echo "5. Last logged in users"

echo "6. Free and used memory info"

echo "7. Get my ip address"

echo "8. My Disk Usage"

echo "9. Process Usage"

echo "10. Users Operations"

echo "11. File Operations"

echo "12. Exit"

}

# Purpose - Display header message

# $1 - message

function write\_header(){

local h="$@"

echo "---------------------------------------------------------------"

echo " ${h}"

echo "---------------------------------------------------------------"

}

# Purpose - Get info about your operating system

function os\_info(){

write\_header " System information "

echo "Operating system : $(uname)"

[ -x $LSB ] && $LSB -a || echo "$LSB command is not insalled (set \$LSB variable)"

#pause "Press [Enter] key to continue..."

pause

}

# Purpose - Get info about host such as dns, IP, and hostname

function host\_info(){

local dnsips=$(sed -e '/^$/d' /etc/resolv.conf | awk '{if (tolower($1)=="nameserver") print $2}')

write\_header " Hostname and DNS information "

echo "Hostname : $(hostname -s)"

echo "DNS domain : $(hostname -d)"

echo "Fully qualified domain name : $(hostname -f)"

echo "Network address (IP) : $(hostname -i)"

echo "DNS name servers (DNS IP) : ${dnsips}"

pause

}

# Purpose - Network inferface and routing info

function net\_info(){

devices=$(netstat -i | cut -d" " -f1 | egrep -v "^Kernel|Iface|lo")

write\_header " Network information "

echo "Total network interfaces found : $(wc -w <<<${devices})"

echo "\*\*\* IP Addresses Information \*\*\*"

ip -4 address show

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

echo "\*\*\* Network routing \*\*\*"

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

netstat -nr

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

echo "\*\*\* Interface traffic information \*\*\*"

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

netstat -i

pause

}

# Purpose - Display a list of users currently logged on

# display a list of receltly loggged in users

function user\_info(){

local cmd="$1"

case "$cmd" in

who) write\_header " Who is online "; who -H; pause ;;

last) write\_header " List of last logged in users "; last ; pause ;;

esac

}

# Purpose - Display used and free memory info

function mem\_info(){

write\_header " Free and used memory "

free -m

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

echo "\*\*\* Virtual memory statistics \*\*\*"

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

vmstat

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

echo "\*\*\* Top 5 memory eating process \*\*\*"

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

ps auxf | sort -nr -k 4 | head -5

pause

}

# Purpose - Get Public IP address form your ISP

function ip\_info(){

cmd='curl -s'

write\_header " Public IP information "

ipservice=checkip.dyndns.org

pipecmd=(sed -e 's/.\*Current IP Address: //' -e 's/<.\*$//') #using brackets to use it as an array and avoid need of scaping

$cmd "$ipservice" | "${pipecmd[@]}"

pause

}

# purpose - Get Disk Usage Information

function disk\_info() {

usep=$(echo $output | awk '{ print $1}' | cut -d'%' -f1)

partition=$(echo $output | awk '{print $2}')

write\_header " Disk Usage Info"

if [ "$EXCLUDE\_LIST" != "" ] ; then

df -H | grep -vE "^Filesystem|tmpfs|cdrom|${EXCLUDE\_LIST}" | awk '{print $5 " " $6}'

else

df -H | grep -vE "^Filesystem|tmpfs|cdrom" | awk '{print $5 " " $6}'

fi

pause

}

#Purpose of Process Usage Information

function proc\_info() {

write\_header " Process Usage Info"

txtred=$(tput setaf 1)

txtgrn=$(tput setaf 2)

txtylw=$(tput setaf 3)

txtblu=$(tput setaf 4)

txtpur=$(tput setaf 5)

txtcyn=$(tput setaf 6)

txtrst=$(tput sgr0)

COLUMNS=$(tput cols)

center() {

w=$(( $COLUMNS / 2 - 20 ))

while IFS= read -r line

do

printf "%${w}s %s\n" ' ' "$line"

done

}

centerwide() {

w=$(( $COLUMNS / 2 - 30 ))

while IFS= read -r line

do

printf "%${w}s %s\n" ' ' "$line"

done

}

while :

do

clear

echo ""

echo ""

echo "${txtcyn}(please enter the number of your selection below)${txtrst}" | centerwide

echo ""

echo "1. Show all processes" | center

echo "2. Kill a process" | center

echo "3. Bring up top" | center

echo "4. ${txtpur}Return to Main Menu${txtrst}" | center

echo "5. ${txtred}Shut down${txtrst}" | center

echo ""

read processmenuchoice

case $processmenuchoice in

1 )

clear && echo "" && echo "${txtcyn}(press ENTER or use arrow keys to scroll list, press Q to return to menu)${txtrst}" | centerwide && read

ps -ef | less

;;

2 )

clear && echo "" && echo "Please enter the PID of the process you would like to kill:" | centerwide

read pidtokill

kill -2 $pidtokill && echo "${txtgrn}Process terminated successfully.${txtrst}" | center || echo "${txtred}Process failed to terminate. Please check the PID and try again.${txtrst}" | centerwide

echo "" && echo "${txtcyn}(press ENTER to continue)${txtrst}" | center && read

;;

3 )

top

;;

4 )

clear && echo "" && echo "Are you sure you want to return to the main menu? ${txtcyn}y/n${txtrst}" | centerwide && echo ""

read exitays

case $exitays in

y | Y )

clear && exit

;;

n | N )

clear && echo "" && echo "Okay. Nevermind then." | center && echo "" && echo "${txtcyn}(Press ENTER to continue.)${txtrst}" | center && read

;;

\* )

clear && echo "" && echo "${txtred}Please make a valid selection.${txtrst}" | center && echo "" && echo "${txtcyn}(Press ENTER to continue.)${txtrst}" | center && read

esac

;;

5 )

clear && echo "" && echo "Are you sure you want to shut down? ${txtcyn}y/n${txtrst}" | centerwide && echo ""

read shutdownays

case $shutdownays in

y | Y )

clear && shutdown -h now

;;

n | N )

clear && echo "" && echo "Okay. Nevermind then." | center && echo "" && echo "${txtcyn}(Press ENTER to continue.)${txtrst}" | center && read

;;

\* )

clear && echo "" && echo "${txtred}Please make a valid selection." | center && echo "" && echo "${txtcyn}(Press ENTER to continue.)${txtrst}" | center && read

;;

esac

;;

\* )

clear && echo "" && echo "${txtred}Please make a valid selection." | center && echo "" && echo "${txtcyn}(Press ENTER to continue.)${txtrst}" | center && read

;;

esac

done

pause

}

#Purpose - For getting User operation and infrmations

function user\_infos() {

write\_header "User Operations"

txtred=$(tput setaf 1)

txtgrn=$(tput setaf 2)

txtylw=$(tput setaf 3)

txtblu=$(tput setaf 4)

txtpur=$(tput setaf 5)

txtcyn=$(tput setaf 6)

txtrst=$(tput sgr0)

COLUMNS=$(tput cols)

center() {

w=$(( $COLUMNS / 2 - 20 ))

while IFS= read -r line

do

printf "%${w}s %s\n" ' ' "$line"

done

}

centerwide() {

w=$(( $COLUMNS / 2 - 30 ))

while IFS= read -r line

do

printf "%${w}s %s\n" ' ' "$line"

done

}

while :

do

clear

echo ""

echo ""

echo "${txtcyn}(please enter the number of your selection below)${txtrst}" | centerwide

echo ""

echo "1. Create a user" | center

echo "2. Change the group for a user" | center

echo "3. Create a group" | center

echo "4. Delete a user" | center

echo "5. Change a password" | center

echo "6. ${txtpur}Return to Main Menu${txtrst}" | center

echo "7. ${txtred}Shut down${txtrst}" | center

echo ""

read usermenuchoice

case $usermenuchoice in

1 )

clear && echo "" && echo "Please enter the new username below: ${txtcyn}(NO SPACES OR SPECIAL CHARACTERS!)${txtrst}" | centerwide && echo ""

read newusername

echo "" && echo "Please enter a group for the new user: ${txtcyn}(STILL NO SPACES OR SPECIAL CHARACTERS!)${txtrst}" | centerwide && echo ""

read newusergroup

echo "" && echo "What is the new user's full name? ${txtcyn}(YOU CAN USE SPACES HERE IF YOU WANT!)${txtrst}" | centerwide && echo ""

read newuserfullname

echo "" && echo ""

groupadd $newusergroup

useradd -g $newusergroup -c "$newuserfullname" $newusername && echo "${txtgrn}New user $newusername created successfully.${txtrst}" | center || echo "${txtred}Could not create new user.${txtrst}" | center

echo "" && echo "${txtcyn}(press ENTER to continue)${txtrst}" | center

read

;;

2 )

clear && echo "" && echo "Which user needs to be in a different group? ${txtcyn}(USER MUST EXIST!)${txtrst}" | centerwide && echo ""

read usermoduser

echo "" && echo "What should be the new group for this user? ${txtcyn}(NO SPACES OR SPECIAL CHARACTERS!)${txtrst}" | centerwide && echo ""

read usermodgroup

echo "" && echo ""

groupadd $usermodgroup

usermod -g $usermodgroup $usermoduser && echo "${txtgrn}User $usermoduser added to group $usermodgroup successfully.${txtrst}" | center || echo "${txtred}Could not add user to group. Please check if user exists.${txtrst}" | centerwide

echo "" && echo "${txtcyn}(press ENTER to continue)${txtrst}" | center

read

;;

3 )

clear && echo "" && echo "Please enter a name for the new group below: ${txtcyn}(NO SPACES OR SPECIAL CHARACTERS!)${txtrst}" | centerwide && echo ""

read newgroup

echo "" && echo ""

groupadd $newgroup && echo "${txtgrn}Group $newgroup created successfully.${txtrst}" | center || echo "${txtred}Failed to create group. Please check if group already exists.${txtrst}" | centerwide

echo "" && echo "${txtcyn}(press ENTER to continue)${txtrst}" | center

read

;;

4 )

clear && echo "" && echo "Please enter the username to be deleted below: ${txtcyn}(THIS CANNOT BE UNDONE!)${txtrst}" | centerwide && echo ""

read deletethisuser

echo "" && echo "${txtred}ARE YOU ABSOLUTELY SURE YOU WANT TO DELETE THIS USER? SERIOUSLY, THIS CANNOT BE UNDONE! ${txtcyn}y/n${txtrst}" | centerwide

read deleteuserays

echo "" && echo ""

case $deleteuserays in

y | Y )

userdel $deletethisuser && echo "${txtgrn}User $deletethisuser deleted successfully." | center || echo "${txtred}Failed to delete user. Please check username and try again.${txtrst}" | centerwide

;;

n | N )

echo "Okay. Nevermind then." | center

;;

\* )

echo "${txtred}Please make a valid selection.${txtrst}" | center

;;

esac

echo "" && echo "${txtcyn}(press ENTER to continue)${txtrst}" | center

read

;;

5 )

clear && echo "" && echo "Which user's password should be changed?" | centerwide

read passuser

echo ""

passwd $passuser && echo "${txtgrn}Password for $passuser changed successfully.${txtrst}" | center || echo "${txtred}Failed to change password.${txtrst}" | center

echo "" && echo "${txtcyn}(press ENTER to continue)${txtrst}" | center

read

;;

6 )

clear && echo "" && echo "Are you sure you want to return to the main menu? ${txtcyn}y/n${txtrst}" | centerwide && echo ""

read exitays

case $exitays in

y | Y )

clear && exit

;;

n | N )

clear && echo "" && echo "Okay. Nevermind then." | center && echo "" && echo "${txtcyn}(Press ENTER to continue.)${txtrst}" | center && read

;;

\* )

clear && echo "" && echo "${txtred}Please make a valid selection.${txtrst}" | center && echo "" && echo "${txtcyn}(Press ENTER to continue.)${txtrst}" | center && read

;;

esac

;;

7 )

clear && echo "" && echo "Are you sure you want to shut down? ${txtcyn}y/n${txtrst}" | centerwide && echo ""

read shutdownays

case $shutdownays in

y | Y )

clear && shutdown -h now

;;

n | N )

clear && echo "" && echo "Okay. Nevermind then." | center && echo "" && echo "${txtcyn}(Press ENTER to continue.)${txtrst}" | center && read

;;

\* )

clear && echo "" && echo "${txtred}Please make a valid selection.${txtrst}" | center && echo "" && echo "${txtcyn}(Press ENTER to continue.)${txtrst}" | center && read

;;

esac

;;

\* )

clear && echo "" && echo "${txtred}Please make a valid selection.${txtrst}" | center && echo "" && echo "${txtcyn}(Press ENTER to continue.)${txtrst}" | center && read

;;

esac

done

pause

}

#Purpose - For File Opertios

function file\_info() {

write\_header "File OPerations"

txtred=$(tput setaf 1)

txtgrn=$(tput setaf 2)

txtylw=$(tput setaf 3)

txtblu=$(tput setaf 4)

txtpur=$(tput setaf 5)

txtcyn=$(tput setaf 6)

txtrst=$(tput sgr0)

COLUMNS=$(tput cols)

center() {

w=$(( $COLUMNS / 2 - 20 ))

while IFS= read -r line

do

printf "%${w}s %s\n" ' ' "$line"

done

}

centerwide() {

w=$(( $COLUMNS / 2 - 30 ))

while IFS= read -r line

do

printf "%${w}s %s\n" ' ' "$line"

done

}

while :

do

clear

echo ""

echo ""

echo "${txtcyn}(please enter the number of your selection below)${txtrst}" | centerwide

echo ""

echo "1. Create a file" | center

echo "2. Delete a file" | center

echo "3. Create a directory" | center

echo "4. Delete a directory" | center

echo "5. Create a symbolic link" | center

echo "6. Change ownership of a file" | center

echo "7. Change permissions on a file" | center

echo "8. Modify text within a file" | center

echo "9. Compress a file" | center

echo "10. Decompress a file" | center

echo "11. ${txtpur}Return to main menu${txtrst}" | center

echo "12. ${txtred}Shut down${txtrst}" | center

echo ""

read mainmenuchoice

case $mainmenuchoice in

1 )

clear && echo "" && echo "Current working directory:" | center && pwd | center

echo "" && echo "Please enter the ${txtcyn}full path${txtrst} and filename for the new file:" | centerwide && echo ""

echo "${txtcyn}(if file exists, it will be touched)${txtrst}" | center && echo ""

read touchfile

echo "" && echo ""

touch $touchfile && echo "${txtgrn}File $touchfile touched successfully.${txtrst}" | centerwide || echo "${txtred}Touch failed. How did you even do that?${txtrst}" | center

echo "" && echo "${txtcyn}(press ENTER to continue)${txtrst}" | center && read

;;

2 )

clear && echo "" && echo "Current working directory:" | center && pwd | center && echo "" && ls && echo ""

echo "Please enter the ${txtcyn}full path${txtrst} and filename for the file to be deleted:" | centerwide && echo ""

read rmfile

echo "" && echo ""

rm -i $rmfile && echo "${txtgrn}File removed successfully.${txtrst}" | center || echo "${txtred}File removal failed.${txtrst}" | center

echo "" && echo "${txtcyn}(press ENTER to continue)${txtrst}" | center && read

;;

3 )

clear && echo "" && echo "Current working directory:" | center && pwd | center && echo "" && ls && echo ""

echo "Please enter the ${txtcyn}full path${txtrst} for the directory to be created:" | centerwide && echo ""

read mkdirdir

echo "" && echo ""

mkdir $mkdirdir && echo "${txtgrn}Directory $mkdirdir created successfully.${txtrst}" | centerwide || echo "${txtred}Failed to create directory.${txtrst}" | center

echo "" && echo "${txtcyn}(press ENTER to continue)${txtrst}" | center && read

;;

4 )

clear && echo "" && echo "Current working directory:" | center && pwd | center && echo "" && ls && echo ""

echo "Please enter the ${txtcyn}full path${txtrst} for the directory to be removed: ${txtcyn}(MUST BE EMPTY!)${txtrst}" | centerwide && echo ""

read rmdirdir

echo "" && echo ""

rmdir $rmdirdir && echo "${txtgrn}Directory $rmdirdir removed successfully.${txtrst}" | centerwide || echo "${txtred}Failed to remove directory. Please ensure directory is empty.${txtrst}" | centerwide

echo "" && echo "${txtcyn}(press ENTER to continue)${txtrst}" | center && read

;;

5 )

clear && echo "" && echo "Please enter the input file for the symbolic link: ${txtcyn}(FULL PATH!)${txtrst}" | centerwide && echo ""

read symlinfile

echo "" && echo "Please enter the output file for the symbolic link: ${txtcyn}(SERIOUSLY, FULL PATH!)${txtrst}" | centerwide && echo ""

read symloutfile

echo "" && echo ""

ln -s $symlinfile $symloutfile

cat $symloutfile && clear && echo "" && echo "${txtgrn}Symbolic link created successfully at $symloutfile${txtrst}" | centerwide || echo "${txtred}Failed to create symbolic link. Please check paths and filenames and try again.${txtrst}" | centerwide && rm -f $symloutfile

echo "" && echo "${txtcyn}(press ENTER to continue)${txtrst}" | center && read

;;

6 )

clear && echo "" && echo "Which file's ownership should be changed? ${txtcyn}(MUST EXIST, USE FULL PATH!)${txtrst}" | centerwide && echo ""

read chownfile

echo "" && echo "Please enter the username for the new owner of $chownfile: ${txtcyn}(USER MUST EXIST)${txtrst}" | centerwide && echo ""

read chownuser

echo "" && echo "Please enter the new group for $chownfile: ${txtcyn}(GROUP MUST EXIST)${txtrst}" | centerwide && echo ""

read chowngroup

echo "" && echo ""

chown $chownuser.$chowngroup $chownfile && echo "${txtgrn}Ownership of $chownfile changed successfully.${txtrst}" | centerwide || echo "${txtred}Failed to change ownership. Please check if user, group and file exist.${txtrst}" | center

echo "" && echo "${txtcyn}(press ENTER to continue)${txtrst}" | center && read

;;

7 )

clear && echo "" && echo "Which file's permissions should be changed? ${txtcyn}(MUST EXIST, USE FULL PATH!)${txtrst}" | centerwide && echo ""

read chmodfile

echo "" && echo "Please enter the three-digit numeric string for the permissions you would like to set:" | centerwide

echo ""

echo "${txtcyn}( format is [owner][group][all] | ex: ${txtrst}777${txtcyn} for full control for everyone )${txtrst}" | centerwide

echo ""

echo "${txtcyn}4 = read${txtrst}" | center

echo "${txtcyn}2 = write${txtrst}" | center

echo "${txtcyn}1 = execute${txtrst}" | center

echo ""

read chmodnum

echo "" && echo ""

chmod $chmodnum $chmodfile && echo "${txtgrn}Permissions for $chmodfile changed successfully.${txtrst}" | centerwide || echo "${txtred}Failed to set permissions.${txtrst}" | center

echo "" && echo "${txtcyn}(press ENTER to continue)${txtrst}" | center && read

;;

8 )

clear && echo "" && echo "Please enter the full path and filename for the file you wish to edit:" | centerwide && echo ""

read editfile

echo "Which program would you like to use to edit this file?" | centerwide && echo ""

echo "${txtcyn}(please enter the number of your selection below)${txtrst}" | centerwide

echo "1. vim" | center

echo "2. nano" | center

echo "3. mcedit" | center

echo "4. emacs" | center

echo "5. pico" | center

echo ""

read editapp

echo ""

case $editapp in

1 )

vim $editfile || echo "${txtred}Failed to open vim. Please check if it is installed.${txtrst}" | centerwide

;;

2 )

nano $editfile || echo "${txtred}Failed to open nano. Please check if it is installed.${txtrst}" | centerwide

;;

3 )

mcedit $editfile || echo "${txtred}Failed to open mcedit. Please check if it is installed.${txtrst}" | centerwide

;;

4 )

emacs $editfile || echo "${txtred}Failed to open emacs. Please check if it is installed.${txtrst}" | centerwide

;;

5 )

pico $editfile || echo "${txtred}Failed to open pico. Please check if it is installed.${txtrst}" | centerwide

;;

\* )

echo "${txtred}Please make a valid selection.${txtrst}" | center

;;

esac

echo "" && echo "${txtcyn}(press ENTER to continue)${txtrst}" | center && read

;;

9 )

clear && echo "" && echo "Please enter the ${txtcyn}full path${txtrst} and filename for the file you wish to compress:" | centerwide && echo ""

read pressfile

echo "" && echo "Which method of compression would you like to use?" | centerwide && echo ""

echo "${txtcyn}(please enter the number of your selection below)${txtrst}" | centerwide

echo ""

echo "1. gzip" | center

echo "2. bzip2" | center

echo "3. compress" | center

echo ""

read pressmethod

echo ""

case $pressmethod in

1 )

gzip $pressfile && echo "${txtgrn}File compressed successfully.${txtrst}" | center || echo "${txtred}File failed to compress.${txtrst}" | center

;;

2 )

bzip2 $pressfile && echo "${txtgrn}File compressed successfully.${txtrst}" | center || echo "${txtred}File failed to compress.${txtrst}" | center

;;

3 )

compress $pressfile && echo "${txtgrn}File compressed successfully.${txtrst}" | center || echo "${txtred}File failed to compress.${txtrst}" | center

;;

\* )

echo "${txtred}Please make a valid selection.${txtrst}" | center

;;

esac

echo "" && echo "${txtcyn}(press ENTER to continue)${txtrst}" | center && read

;;

10 )

clear && echo "" && echo "Please enter the ${txtcyn}full path${txtrst} and filename for the file you wish to decompress:" | centerwide && echo ""

read depressfile

case $depressfile in

\*.gz | \*.GZ )

gunzip $depressfiles && echo "${txtgrn}File decompressed successfully.${txtrst}" | center || echo "${txtred}File failed to decompress.${txtrst}" | center

;;

\*.bz2 | \*.BZ2 )

bunzip2 $depressfile && echo "${txtgrn}File decompressed successfully.${txtrst}" | center || echo "${txtred}File failed to decompress.${txtrst}" | center

;;

\*.z | \*.Z )

uncompress $depressfile && echo "${txtgrn}File decompressed successfully.${txtrst}" | center || echo "${txtred}File failed to decompress.${txtrst}" | center

;;

\* )

echo "${txtred}File does not appear to use a valid compression method (gzip, bzip2, or compress). Please decompress manually.${txtrst}" | centerwide

esac

echo "" && echo "${txtcyn}(press ENTER to continue)${txtrst}" | center && read

;;

11 )

clear && echo "" && echo "Are you sure you want to return to the main menu? ${txtcyn}y/n${txtrst}" | centerwide && echo ""

read exitays

case $exitays in

y | Y )

clear && exit

;;

n | N )

clear && echo "" && echo "Okay. Nevermind then." | center && echo "" && echo "${txtcyn}(Press ENTER to continue.)${txtrst}" | center && read

;;

\* )

clear && echo "" && echo "${txtred}Please make a valid selection.${txtrst}" | center && echo "" && echo "${txtcyn}(Press ENTER to continue.)${txtrst}" | center && read

esac

;;

12 )

clear && echo "" && echo "Are you sure you want to shut down? ${txtcyn}y/n${txtrst}" | centerwide && echo ""

read shutdownays

case $shutdownays in

y | Y )

clear && shutdown -h now

;;

n | N )

clear && echo "" && echo "Okay. Nevermind then." | center && echo "" && echo "${txtcyn}(Press ENTER to continue.)${txtrst}" | center && read

;;

\* )

clear && echo "" && echo "${txtred}Please make a valid selection.${txtrst}" | center && echo "" && echo "${txtcyn}(Press ENTER to continue.)${txtrst}" | center && read

;;

esac

;;

\* )

clear && echo "" && echo "${txtred}Please make a valid selection.${txtrst}" | center && echo "" && echo "${txtcyn}(Press ENTER to continue.)${txtrst}" | center && read

;;

esac

done

pause

}

# Purpose - Get input via the keyboard and make a decision using case..esac

function read\_input(){

local c

read -p "Enter your choice [ 1 -12 ] " c

case $c in

1) os\_info ;;

2) host\_info ;;

3) net\_info ;;

4) user\_info "who" ;;

5) user\_info "last" ;;

6) mem\_info ;;

7) ip\_info ;;

8) disk\_info ;;

9) proc\_info ;;

10) user\_infos ;;

11) file\_info ;;

12) echo "Bye!"; exit 0 ;;

\*)

echo "Please select between 1 to 12 choice only."

pause

esac

}

# ignore CTRL+C, CTRL+Z and quit singles using the trap

trap '' SIGINT SIGQUIT SIGTSTP

# main logic

while true

do

clear

show\_menu # display memu

read\_input # wait for user input

done