Sample Scripts

????

- ip.sh - ?? Public IP ????
- whoamifuck.sh - Linux ??????

Auto-rar.sh

????????? (*.part0XX.rar)

```
#!/bin/bash

echo "-> Started: "`date +%m/%d/%y\ %H:%M\ %Z`
echo "-> As:"`whoami`

i=1
ARCHIVES="$@"
for f in $ARCHIVES; do
  PARTCHECK=$(expr "$f" : ".*\([Pp][Aa][Rr][Tt][0-9]\+\.[Rr][Aa][Rr]\)")
  RARCHECK=$(expr "$f" : "\(.*\.[Rr][Aa][Rr]\)")
  echo "-> Processing($i of $#): $f"

  if [ "$PARTCHECK" ] && [ -f $f ]; then
    echo "-> Extracting Multipart Archive"
    unrar x $f
    if [ $? -eq 0 ]; then
        FILES=$(expr "$f" : "\(.*[Pp][Aa][Rr][Tt]\).*")
        echo "-> Extraction Successful"
        echo "-> Removing $FILES*.rar"
        rm $FILES*.rar
    else
        echo "-> **Extraction Failed"
        exit 1
    fi
  else
```

```
    if [ "$RARCHECK" ] && [ -f $f ]; then
      echo "-> Extracting Single Archive"
      unrar x $f
      if [ $? -eq 0 ]; then
        echo "-> Extraction Successful"
        echo "-> Removing $f"
        rm $f
      else
        echo "-> **Extraction Failed"
        exit 1
      fi
    fi
  fi
    echo ""
    i=$(( i+1 ))
done
```

Domaincheck.sh

```
#!/bin/bash

#Specify all the domains you want to check
DOMAINS="google.com dribbble.com facebook.com youtube.com"

current_epoch=`date '+%s'`
for dm in $DOMAINS
do
  expiry_date=`whois $dm | egrep -i "Expiration Date:|Expires on"| head -1 | awk '{print $NF}'`
  echo -n " $dm - Expires on $expiry_date "
  expiry_epoch=`date --date="$expiry_date" '+%s'`
  epoch_diff=`expr $expiry_epoch - $current_epoch`
  days=`expr $epoch_diff / 86400`
  echo " $days days remaining. "
done
```

Scripting Basics

BASH SCRIPTING BASICS

script.sh

```bash
#!/bin/bash

username="Jay"
filename=$3

read -p "Enter your username: " user
echo "Username: $user"

if [ "$EUID" -ne 0 ]; then
    echo "You are not running this script as the root user."
else
    echo "You are running this script as the root user."
fi

echo "Counting to 5:"
for i in {1..5}; do
    echo "$i"
done


function greet() {
    echo "Hello, $1!"
}
greet "Alice"

echo "Enter a number between 1 and 2: "
read num
case $num in
    1) echo "You chose one." ;;
    2) echo "You chose two." ;;
    *) echo "Invalid choice." ;;
esac

if [ -e "$filename" ]  && [ -d "$filename" ]; then
    echo "File exists and is a directory."
else
    echo "File does not exist or is not a directory."
fi

echo "First argument: $1"
echo "Second argument: $2"

cat nonexistent-file.txt 2> /dev/null
echo "Exit status: $?"

fruits=("Apple" "Orange" "Banana")
echo "Fruits: ${fruits[0]}"

declare -A capitals
capitals[USA]="Washington D.C."
capitals[France]="Paris"
echo "Capital of France: ${capitals[France]}"

current_date=$(date)
echo "Today's date is: $current_date"

echo "This is a sample text." > example.txt
find / -name hello.txt &> /dev/null

result=$( expr 15 - 2 )
echo $result

SRC="/path/to/foo.cpp"
BASEPATH=${SRC##*/}
echo $BASEPATH

trap 'echo "Received SIGTERM signal. Cleaning up..."; exit' SIGTERM

# This is a single line comment

:' this a multiline
 commment'
```

Shebang Line

Variables

User Input

Conditional if Statement

For Loop

Functions

bash script.sh

Conditional Case Statement

File Operations

Command Line Arguments

Exit Status Codes

Indexed Arrays

Associative Arrays

Command Substitution

Command Line Redirections

Arithmetic Operations

Parameter Expansion

Process Signal Handling

Comments

sysxplore.com

Revision #5
Created 1 June 2020 07:44:00 by Admin
Updated 11 November 2024 19:15:58 by Admin