
DTU

DTU

Linux and C Programming (62558)

2nd Assignment

Mads Richardt (s224948)

What task should I automate...?

Backup a local directory to a remote host!!!

Pseudo code

Step 1) Make a compressed copy of local directory.

Step 2) Send compressed copy to remote host.

Step 3) Remove compressed copy from local machine.

Step 1 - (tar[1])

Create a compressed copy a directory using the tar utility.

Code Example:

```
madsrichardt@penguin:~$ ls
Arduino  arduino-1.8.19  Assignment  bin  Desktop  embed_sys  embSys
madsrichardt@penguin:~$ tar -czf my_archive.tar.gz Assignment
madsrichardt@penguin:~$ ls
Arduino  arduino-1.8.19  Assignment  bin  Desktop  embed_sys  embSys  my_archive.tar.gz
```

Flags:

- -c Create an archive.
- -z Compress archive using gzip.
- -f Allows the user to specify the name of the archive

[1] Gilmore, John; Fenlason, Jay, "Basic Tar Format", Free Software Foundation, February 2019

Step 2 and 3 - (ssh[1], rsync[2] and rm)

- Generate a RSA key pair.
  - $ ssh-keygen
- Copy the public key to a remote host.
  - $ ssh-copy-id userName@remoteHost
- Safely send file to remote host.
  - $ rsync -v --rsh="ssh -p hostPort" localFile username@remoteHost:destinationPath
- Remove file from local machine.
  - $ rm path/fileName

[1] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Protocol Architecture", RFC 4251, January 2006

[2] https://rsync.samba.org

Putting it all together...

Bash script:

```
#! /usr/bin/bash
PORT="22222"
DIR=$(pwd)
DEST="student@130.225.170.80:/home/student/backup_folder"
VERBOSE=0
# Get arguments
while getopts 'vp:s:d:' OPTION; do
    case "$OPTION" in
        p) PORT="$OPTARG" ;;
        s) DIR="$OPTARG" ;;
        d) DEST="$OPTARG" ;;
        v) VERBOSE=1 ;;
        ?) echo "script usage: $(basename $0) [-p port] [-s source_directory]
        [-d server:destination_folder] [-v verbose]" >&2
        exit 1 ;;
    esac
done
```

```
# Set Archive name
ARCHIVE_NAME=/tmp/$(basename $DIR).tar.gz
# Creat and Compress TAR archive
if [ $VERBOSE -eq 1 ]
then
    echo "Creating TAR archive from $DIR and compressing archive to $ARCHIVE_NAME"
    tar -cvzf $ARCHIVE_NAME  $DIR
    echo "... done!"
else
tar -czf $ARCHIVE_NAME  $DIR
fi
# Send compressed TAR archive
if [ $VERBOSE -eq 1 ]
then
    echo ""
    echo "Sending $ARCHIVE_NAME to $DEST on port $PORT"
    rsync -vh --rsh="ssh -p $PORT" $ARCHIVE_NAME  $DEST
    echo "... done!"
else
    rsync --rsh="ssh -p $PORT" $ARCHIVE_NAME  $DEST
fi
```

```
# Remove compressed TAR archive
if [ $VERBOSE -eq 1 ]
then
    echo ""
    echo "Removing $ARCHIVE_NAME from local machine"
    rm $ARCHIVE_NAME
    echo "... done!"
else
    rm $ARCHIVE_NAME
fi
```

- Save script in /home/userName/bin/backup.sh.
- cd to /home/userName/bin and make backup.sh executable ($ sudo chmod +x backup.sh).
- Make sure /home/userName/bin is in PATH - now backup.sh can be called from everywhere without specifying the full path to backup.sh.

Let’s test it!