

## Backing up data and systems

### Andy Pepperdine

These notes result from the discussions at the U3A in Bath FOSS session of 22 September 2011, and from the notes provided by members in the mailing list before the meeting.

The idea is to give some indication of what you should consider for backing up, and why. An indication of what I do will also be given.

For the purpose of this paper, I will consider only the normal private user with their won data to preserve. If you are trying to manage a company's data, there are other things to imagine that may be important, especially if you have a database that must be kept up to date in real time. The latter case is beyond this paper.

### *Types of data*

Most of the data you consider important will be things like your e-mails, your documents that you've created and edited, your pictures, videos, music, and other files that you have made in some way and that cannot be replaced from any other source. These you do not wish to lose.

Another type of data is what we might call the customisation of the system you work on, and these are usually found in the settings of the applications you use – things like language dictionaries, Firefox add-ons, details of financial accounts, etc. In general, they are stored by the application in its own save area, and these areas are always to be found in some directory under your account data somewhere. For example, Thunderbird keeps all its settings in a hidden directory in your home directory.

If you had to install an application, then you might consider how you would backup that installation. However, these are not created by you, and so they could be retrieved again if necessary from their original source. Only if they are no longer supported would you need a copy yourself, and even then you would also be concerned with maintaining it. In our cases, I suspect that cannot be the case and you would have moved all the uses of such an application to another one before they became obsolete.

Finally, we have the system itself. Do you need to save a copy of the system completely?

### *What are we guarding against?*

This question is important, since it will determine how you keep the data, how you would expect to recover and under what circumstances.

The most common case is that you have accidentally deleted or badly edited a file you were working on, and in this case, you will wish to recover from a previous known good version. You need to ask yourself how much work you might have to re-do to restore the file again. In particular, are you prepared to do a day's work again, or try to reduce it. How often are you actually working on your computer in a continuous way which might have to be re-done?

The next most common case is of a file system corruption which may mean that a number of files will have to be reset. In practice nowadays, these occurrences are very rare, so how much regular work do you want to do to guard against something that will rarely happen? Or could you treat this

case like the next one.

In the worst case, you lose everything currently on your machine through a disk crash, destroying the system, file system and all the data. What will you now do? These are now also fortunately rare, but when they do occur are disastrous if you have not taken steps to manage the possibility. It will mean that you will have to re-construct the whole system again, much like installing the next version of your operating system, applications and restoring your own data. If you think of it like this, it may help you to understand your real requirements and how you should treat it.

### ***Where to keep the backups***

Again, preparation and forethought is the key to managing the situation. Are you thinking only of errors to the data; or are you thinking in terms of catastrophic external events like fires and flood? The latter will mean that you will need a copy elsewhere, and not in the cupboard under your desk. This also affects the type of removable device you may consider keeping, and how you will store it.

### ***How frequently do you need to take a backup?***

This is closely related to the question: How frequently do you update the data? There is no point in copying every hour things which are not changed from one month to the next. But you will want to copy them soon after they have been changed.

### ***What is practical?***

First, let's look at the data you have created. This in many ways is the most important stuff to keep. Computers can be replaced if necessary; but your photographs and letters cannot.

All of the data you have created, and all your personal settings for your applications, are held as files and sub-directories under a single data directory that is restricted to your access. Every type of system has a place where all this is kept. On Linux (or indeed almost every Unix-type system) it is under `/home/username`, where *username* is the name of your account. Copying the contents of this directory will copy everything that is yours.

System data and applications are held elsewhere.

### **Relatively fixed data**

If you are like me, then you will not be modifying your pictures, and these will be added to in a collection, but not deleted. In addition, new pictures are added infrequently, and these files are large. Copying them often is to be avoided.

Depending on what backup method is employed, there are ways of separating this rarely changed data from the normal, and then ignoring it when taking regular backups. On Linux, you could create another directory, outside your own home directory, and set up a link from within your own data area to it. In this way it appears that it is in your data area to normal applications, but in fact the data is elsewhere and can be passed over when backing up.

For example, you could place the data in the home space, but under a different name, like this:

The `/home` directory is owned by root, and so you will need to modify it and create the new space as the administrator:

```
sudo mkdir /home/Userdata
```

The new directory will be owned by root, and this is not what you want, so the next thing is to transfer the ownership to yourself (which I will refer to by the name username).

```
sudo chown username:username /home/Userdata
```

You can now create under the new place the directories you need; for example:

```
mkdir /home/Userdata/Videos /home/Userdata/Pictures
```

Now copy over data from your existing directories into the new places. I use a copy here, and not a move, to make sure it is in the right place before deleting the originals – one step at a time.

```
cd
cp -r Videos/* /home/Userdata/Videos
cp -r Pictures/* /home/Userdata/Pictures
```

You could have used the graphical interface for this, and dragged and dropped the directories if you wished.

When you are satisfied that the data is correct and in the right place, give the original directory a new name:

```
mv Videos OrigVideos
mv Pictures OrigPictures
```

Now set up a link to the new place:

```
ln -s /home/Username/Videos Videos
ln -s /home/Username/Pictures Pictures
```

Now you will find you can access Pictures just as before, but you will be seeing them in the new data area.

**Note: Always** use the `-s` option under with the `ln` command. It is almost certain that any normal user will need this. But if you do not put it in, then it will apparently work until some considerable time later when it may not give the results you were expecting. The reasons are technical, but remember “`ln -s`” is an idiom that must be kept together.

Now you are in a position to delete the original directories, saving space under your own home directory:

```
rm -r OrigVideos OrigPictures
```

Of course, if you have a second device with lots of space for your data, then you could have put the Videos and Pictures there instead. You just need to create the target directories and set up the links.

The new areas can now be saved separately, either at rather infrequent but regular intervals, or only after significant updates to the data there. Either would keep the backups you need for just those things.

## Normal data in use every day

For the normal case, though, you will want to take a backup of your data fairly frequently. You can either do this by keeping a “mirror” or “snapshot” of your current data, or taking a copy of those files that have changed since the last backup, or making a complete copy. Which you choose depends on how often you want to do it, and what devices you have available.

Strictly speaking, a *mirror* is a copy of a file system that is updated simultaneously with any update to the original. Mirrors are used in large database operations to ensure that there is always a correct copy at any time, and if one of the disks should break, then the other can take over until the broken one is fixed. For the private user, this is ludicrous overkill. For a company with realtime data, it is essential.

On the other hand a snapshot is useful to anyone. It is a copy of a file system taken at a particular time, and may be what you already consider a backup to be. The snapshot may either be a complete pristine copy of what exists at a time; or it might be a merging of an older snapshot with what has changed on the current system since it was taken. Keeping an updated snapshot is useful and relatively quick, but if you keep the two exactly synchronised, then any deletions will also be transferred and old files may disappear for good earlier than you expect. A mixture of new copies and updated current snapshots is probably more suitable for the home user.

### **Making a synchronised copy**

There is a script showing how I do this. I called it, probably erroneously, *mirror*, but you could give it any name you like. It is a bash script to be run on a Linux system,.

The extended version, *backup-own*, also takes a complete backup into a new directory, and compresses an older one, deleting the very oldest compressed one.

They are included at the end of this document. As with all scripts passed on from anyone else, there is no warranty with them, and only use them once you understand what they are doing!

### ***Infrequently updated data***

For data that is infrequently updated, or even never changed, but more files may be added, then one strategy would be to take a copy onto a separate device only after such an infrequent update. It does mean that you will have to remember to do it; you cannot rely on it being incorporated into the standard backup regime, although if you only ever took incremental backups (saving only what has changed) it would not add much to the process in time, but could add considerably to the space used if you had several such copies.

### ***How many backup devices to use***

What would happen if you should find that your backup device has been lost or destroyed? Where do keep it?

I use two USB external hard disks which I swap each Sunday. So that daily mirrors are taken onto the current device, and then at the weekend, they are swapped and a full backup taken. The device not in current use is held away from the machines so I have at least a copy not more than one week old of everything, in case I have to start again.

### ***When to do it?***

Since you may not be very regular in your timetable, trying to get a backup taken automatically at a given time may not be practical. Such a scheme would work if you had a server that is always on – it could be easily set up to take a copy each day at the same time, but it is not good for a desk machine.

My solution to this is to get into a habit each morning of reading e-mail, news and any other standard activity. Then close down all applications and run the mirror. This means that I will have a backup at most a few working hours old (but may be days on the calendar). For me it's more convenient this way, since I do not then have to hang around for the backup when I log out.

### ***Further information***

Backups seem to be such personal things, and every system differs. I'm not sure what to recommend for general users for information other than to ask around among the people you know.

The most important thing is that you are satisfied with what you are doing, after you've considered the possible things that could go wrong.

### ***Scripts***

If you do not know how to read or execute these scripts, then let me know.

#### **Mirror**

```
#!/bin/bash

# Update the mirror only

DISK=' '
LOC="andy/Lenovo"
MIRROR="$LOC/DeskMirror"
ERRLOG=rsync.error.txt

# Clear out error log file
rm -f $ERRLOG
if [ -e $ERRLOG ]; then
    echo "Cannot remove error log file $ERRLOG - terminating"
    exit 3
fi
touch $ERRLOG

# Find the mirror on an attached disk

for d in $(mount | grep '^/' | awk '{print $3}'); do
    if [ -d "$d/$MIRROR" ]; then
        DISK="$d"
```

```
    fi
done

# Check we have found one
if [ -z "$DISK" ]; then
    echo "Cannot find $MIRROR on a mounted device"
    exit 1
fi

# Say what is happening
echo "Mirror found on $DISK"
echo ''

# Synchronise with a current snapshot
echo "rsync with $MIRROR"
rsync -av "$HOME/" "$DISK/$MIRROR" 2>> $ERRLOG
RC=$?
if [ $RC -ne 0 ]; then
    echo "Error during rsync, return code $RC" | tee $ERRLOG
fi
echo "==== end of rsync with $MIRROR" >> $ERRLOG
echo ''

echo "Error log follows:"
echo ''
cat $ERRLOG

exit 0
```

### Backup-own

```
#!/bin/bash

# Backup a single home directory onto an installed portable disk

DISK=''
LOC="andy/Lenovo"
```

```
MIRROR="$LOC/DeskMirror"
ERRLOG=rsync.error.txt

# Clear out error log file
rm -f $ERRLOG
if [ -e $ERRLOG ]; then
    echo "Cannot remove error log file $ERRLOG - terminating"
    exit 3
fi
touch $ERRLOG

# Find the name of the attached disk to take backups
# by looking for the mirror to contain the first sync

for d in $(mount | grep '^/' | awk '{print $3}'); do
    if [ -d "$d/$MIRROR" ]; then
        DISK="$d"
    fi
done

# Check we have found one
if [ -z "$DISK" ]; then
    echo "Cannot find $MIRROR on a mounted device"
    exit 1
fi

# Say what is happening
echo "Backups to be taken onto $DISK"
echo ''

# Synchronise with a default mirror as first go
echo "rsync with $MIRROR"
rsync -av "$HOME/" "$DISK/$MIRROR" 2>> $ERRLOG
RC=$?
if [ $RC -ne 0 ]; then
    echo "Error during rsync, return code $RC" | tee $ERRLOG
```

```
fi
echo "=====  
end of rsync with $MIRROR" >> $ERRLOG
echo ''

# Create name of next backup
NEXT=$USER$(date '+%y%m%d')

# Make a new backup
echo "rsync with $LOC/$NEXT"
rsync -av "$HOME/" "$DISK/$LOC/$NEXT" 2>> $ERRLOG
RC=$?
if [ $RC -ne 0 ]; then
    echo "Error during rsync with $LOC/$NEXT, return code $RC" |
tee $ERRLOG
fi
echo "=====  
end of rsync with $LOC/$NEXT" >> $ERRLOG
echo ''

# Move working directory to the backup area
cd $DISK/$LOC

# Find oldest compressed file and remove it

# How many straight directories do we have?
# If there are more than 3, then we will compress the oldest

ND="$(ls | grep -v 'tgz$' | wc -l)"
if [ "$ND" -gt 3 ]; then

    # How many compressed directories do we have?
    # If there are more than 3, then we remove the oldest

    NZ="$(ls | grep 'tgz$' | wc -l)"
    if [ "$NZ" -gt 3 ]; then

        echo "Find oldest compressed directory and remove it"
```

```
OLD="$(ls -rt *.tgz | head -1)"
if [ ! -z "$OLD" ]; then
    echo "Removing $OLD"
    rm -f "$OLD"
    echo ''
fi

else
    echo Only "$NZ" compressed directories found
fi

# Find oldest backup directory
echo "Find oldest backup directory and compress it"
OLDDIR="$(ls -rt | grep '[0-9]$(ls -rt | head -1)'"

# Compress it and remove it
if [ ! -z "$OLDDIR" ]; then
    echo "Compressing $OLDDIR"
    cd "$OLDDIR"
    tar cvzf "../$OLDDIR.tgz" .
    RC=$?
    if [ "$RC" -ne 0 ]; then
        echo "Error compressing $OLDDIR, return code $RC"
    fi
    cd ..
    echo ''
    echo "Removing $OLDDIR"
    chmod -R u+w "$OLDDIR"
    rm -fr "$OLDDIR"
fi

else
    echo Only "$ND" backup directories found
fi

echo ''
```

**2011-09-22**

**Backups**

**U3A in Bath**

```
echo "Backups completed"  
echo ''  
echo "Error log follows:"  
echo ''  
cat $ERRLOG  
  
exit 0
```