### Backup the data

The ``rsync`` command is used to synchronize entire directory trees. Basically, it copies file as the ``cp`` command does. In addition, ``rsync`` checks if the file being copied already exists. If the file exists and there is no change in size or modification time, ``rsync`` will avoid an unnecessary copy and save time. Furthermore, because rsync copies only the parts of files that have actually changed, it can be very fast.

The ``rsync`` is very efficient when recursively copying one directory tree via network, because only the differences are transmitted. One often synchronizes the destination directory tree with the origin, using the ``rsync -r`` option to recursively walk down the directory tree copying all files and directories below the one listed as the source.

```

# rsync -ravzh project\_ABC /data/backups

sending incremental file list

project\_ABC/

project\_ABC/file1.txt

project\_ABC/file2.txt

project\_ABC/file3.txt

project\_ABC/file4.txt

sent 636 bytes received 92 bytes 1.46K bytes/sec

total size is 452 speedup is 0.62

```

### Compress the data

File data is often compressed to save disk space and reduce the time it takes to transmit files over networks. Linux uses a number of methods to perform this compression.

|Command|Usage|

|-------|-----------|

|gzip |The most frequently used Linux compression utility|

|bzip2 |Produces files significantly smaller than those produced by gzip|

|xz |The most space efficient compression utility used in Linux. It is now used by kernel.org to store archives of the Linux kernel.|

|zip |Is often required to examine and decompress archives from other operating systems|

These techniques vary in the efficiency of the compression (how much space is saved) and in how long they take to compress; generally the more efficient techniques take longer. Decompression time doesn't vary as much across different methods.

### Archiving data

The ``tar`` command allows you to create or extract files from an archive file, often called a tarball. At the same time you can optionally compress while creating the archive, and decompress while extracting its contents.

Here are some examples of the use of tar:

|Command|Usage|

|-------|-----------|

|tar xvf mydir.tar|Extract all the files in mydir.tar into the mydir directory|

|tar zcvf mydir.tar.gz mydir|Create the archive and compress with gzip|

|tar jcvf mydir.tar.bz2 mydir|Create the archive and compress with bz2|

|tar xvf mydir.tar.gz|Extract all the files in mydir.tar.gz into the mydir directory.|

|tar cvf mydir.tar|show the content into the mydir directory|

### Copying disks

The ``dd`` command is very useful for making copies of raw disk space. For example, to back up the Master Boot Record (MBR) (the first 512 byte sector on the disk that contains a table describing the partitions on that disk), use:

```

# dd if=/dev/sda of=sda.mbr bs=512 count=1

```

To use dd to make a copy of one disk onto another, deleting everything that previously existed on the second disk, use:

```

# dd if=/dev/sda of=/dev/sdb

```

An exact copy of the first disk device is created on the second disk device.

The ``dd`` command is usefull to duplicate a bootable disk as a Compact Flash card, a Micro SD card or a bootable USB dongle. Insert the CF Card to be copied into the system and make a copy

```

# dd if=/dev/sdb of=./backup.img

```

Remove the CF Card, insert a new one and make a new copy

```

# dd if=./backup.img of=/dev/sdc

```