Johannes’ mirroring-page

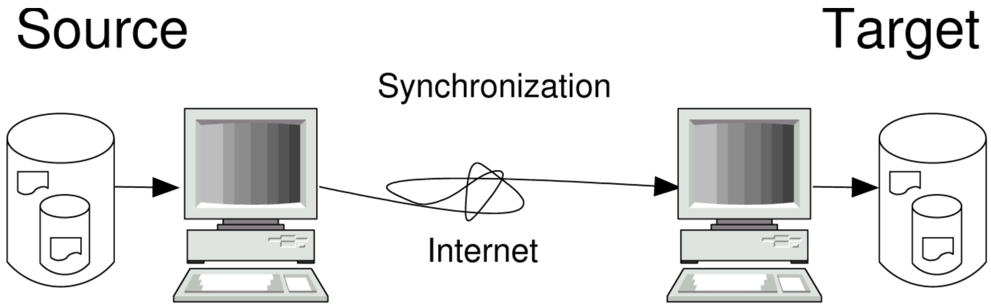
Johannes Franken
<jfranken@jfranken.de>

On this page I show you some ways to copy a directory structure using internet protocols and to keep it up to date.

Contents

1. Overview
   a) Requirements concerning copying programs
   b) Assessment of some copying programs
2. Details on the copying programs
   a) wget
   b) rsync
   c) tar
   d) cp

Overview



Requirements concerning copying programs

- The **transport** must be feasible in the given infrastructure:
  - as **local** transport I denote any operations on volumes, that can be mounted into the filesystem. This concerns:
    - any block-devices (e.g. connected over PCI, IDE, SCSI, USB, PCMCIA, Floppy disks, CD/DVD drives, Flashcards)
    - fileserver’s shares (e.g. over nfs, smbfs, ncpfs, gfs, coda)
    - file systems on loop-devices
  - Over **http(s),ftp** you access Web- and FTP-servers.
  - A **pipe** allows you to delegate communications to another program (e.g. ssh, rsh, rexec, netcat).
- The **scope** of a copy tells its congruence with the original. Besides the filename and -content, the following criteria might be interesting:
  - **delete:** At synchronization, will any objects be deleted from the copy, which meanwhile have been deleted from the source?
  - **filedate:** Will the files’ timestamps be adopted?
  - **permissions:** will user, group and authorization be adopted?
- The **performance** of a synchronization process can be improved by the following actions:
  - A **differential** copying process transfers only those files, which have changed since the last synchronization.
  - A **partial** copying process transfers only the bytes or blocks within those files, which have actually changed. .
  - The files can be **compressed** before transferring and then be decompressed on the receiving side.

Assessment of some copying programs

Program	Transport				Scope			Performance		
	local	http(s)	ftp	pipe	delete	filedate	permissions	differential	partial	compression
wget	-	X	X	-	-	X	-	X	X	-
cp -duRp	X	-	-	-	-	X	X	X	-	-
tar	X	-	-	X	-	X	X	-	-	X
rsync	X	-	-	X	X	X	X	X	X	X

Summary: rsync satisfies most requirements. Use wget, if you need to communicate with the server over http or ftp.

Details on the copying programs

wget

Description

wget copies files over http, https or ftp from the corresponding servers. Additionally it can

- (for http(s):) follow the links contained in HTML-files and
- (for ftp:) grab any subdirectories.

If the server conveys filedates, wget will adopt them for the files it receives. This way it can avoid retransmission of files already available.
The copy differs from the original at the following details:

- Files that have been deleted from the server, stay alive in the copy.
- Files not pointed at by a link are missing (for http).
- No permissions (owner, group, authorization) are transferred.

Usage

```
wget options URL
```

Options of interest are:

Option	implication
`-N`	Do not download files that are already available locally and match the server’s filedate.
`-nH --cut-dirs=2`	In recursive mode, wget normally creates a subdirectory for the hostname and any directories mentioned in the URL. The option **`-nH`** suppresses the creation of hostdirs, and **`--cut-dirs=2`** the creation of the first two directories. For example: **`wget -r -nH --cut-dirs=2 http://www.jfranken.de/homepages/johannes/vortraege`** will create the directory **`vortraege`**.
`-k`	turns absolute URLs to relative ones in HTML-files. Caution, this does not work in any situations.
`-r -np`	(recursive, no-parent): If the given URL provides a html-file, wget will also fetch any elements referenced (in particular links and graphics) and repeat this procedure for them. The option **`-np`** avoids ascending to the parent directory. wget ignores references to other hosts, except if you set the parameter **`-H`**.
`-p -l 10`	The parameter **`-l 10`** limits the recursion depth for **`-r`** to 10 levels. The default depth is 5. If you set **`-l 0`**, it downloads at infinitive depth, which can cause filesystem problems on cyclic links.
`-H -Djfranken.de,our-isp.org`	Also follow links to different servers, if they belong to the domain **`jfranken.de`** or **`our-isp.org`**.
`-nv`	Avoids output of debugging messages.

wget wil direct its ftp- or http-requests automatically to your proxyserver, if the environment varibale **`http_proxy`** oder **`ftp_proxy`** are set, e.g. by

```
$ export http_proxy=http://jfranken:secret@proxy.jfranken.de:3128/
$ export ftp_proxy=$http_proxy
```

Links

- wget project page
- wget(1) manpage

rsync

Description

rsync can synchronize files or directories over

- the local filesystem (hard- and floppy disks, CD/DVD drives, flashcards, ...)
- fileservers (e.g. NFS, Windows, Novell), that are mounted into the filesystem.
- a remote shell (rsh, ssh)
- a tcp-connection to the rsyncd

When using a remote shell or rsyncd, the other side must be local.
If called with appropriate parameters, rsync will only transfer the changed parts within files, preserve file attributes and delete any files from the target directory, that have been deleted from the source.
Usage:

```
rsync options source(s) target
```

Interesting options are:

Option	implication
`-a`	(archive mode): copies any subdirectories, most attributes (symlinks, permissions, filedate, group, devices) and (if you’re root) the owner of the files.
`-v --progress`	(verbose): **`-v`** prints a list of files during transfer. If you additionally set **`--progress`**, it will continuously show you the number of bytes transferred and the progress in percent.
`-n`	(dry-run): Don’t write, just simulate the procedure.
`-z -e` program	If there is a colon in the source or target, rsync interprets the part before it as hostname and communicates over the program specified by **`-e`**, to which it passes the following parameters: ● as source: **`hostname rsync --server --sender . Sourcedir`** ● as target: **`hostname rsync --server . Targetdir`** **`ssh`** works great as a program for **`-e`**. If you want to insert your own parameters at the beginning of the parameter list, you need to put them in double quotes with the program. The parameter **`-z`** makes rsync compress any data it transfers.

`--delete --force`
`--delete-excluded`
	deletes any entries from the target directory, which have been deleted from the source meanwhile.

`--partial`	Don’t delete the partial files if the connection is lost. This way rsync can resume the transfer next time.
`--exclude=`pattern	Ignore any files which match the given pattern, e.g. **`--exclude *~`** . Read more on rsync’s extensive exluding capabilities in the rsync(1) manpage.
`-x`	excludes any files on filesystems, that are mounted into the source directory.

Explanation for the notation of source and target:

- If the source ends on a slash (`/`), rsync will not copy that directory itself, but any objects it contains.
- If they don’t have any colon, rsync interprets them as path in the filesystem.
- If they have one colon, rsync interprets the part before it as hostname and commnicates over the program specified by **`-e`**.
- If there are two successive colons in the source or target, rsync interprets the part before them as hostname and talks to port 873 on that host, whose inetd should then call another rsync with the parameter **`--daemon`**. If at this time the environment variable **`RSYNC_PROXY`** is set to the name and port of your proxy server, rsync will tunnel any communications as https-CONNECT-stream through your https-proxy server.

Links

- rsync project page
- rsync(1) manpage
- rsyncd.conf(5) manpage
- The rsync algorithm

tar

tar was originally designed for writing disk backups on magnetic tapes. It can convert directories into streams and vice versa. If you pipe that stream to another host, you can make full copies of directories over an internet.
The copy differs from the original at the following detail:

- Files that have been deleted from the server, stay alive in the copy.

The following command will copy the **`mydir`** directory into the **`/tmp`**-directory on the computer **`gate`**

```
$ tar cf - mydir/ | ssh gate ’cd /tmp && tar xpvf -’
```

In my talk about netcat I show tunneling of **`tar`** over arbitrary tcp connections, even without a remote shell.

cp

`cp` is the standard command for copying files. Called with the parameters **`-duRp`**, it will copy any links, rights, timestamps and subdirectories, and only for those files, that have been added or changed since the last synchronization.
Read more on cp in the cp(1) manpage