Version : **2024.01**

Last update : 2024/11/27 13:29

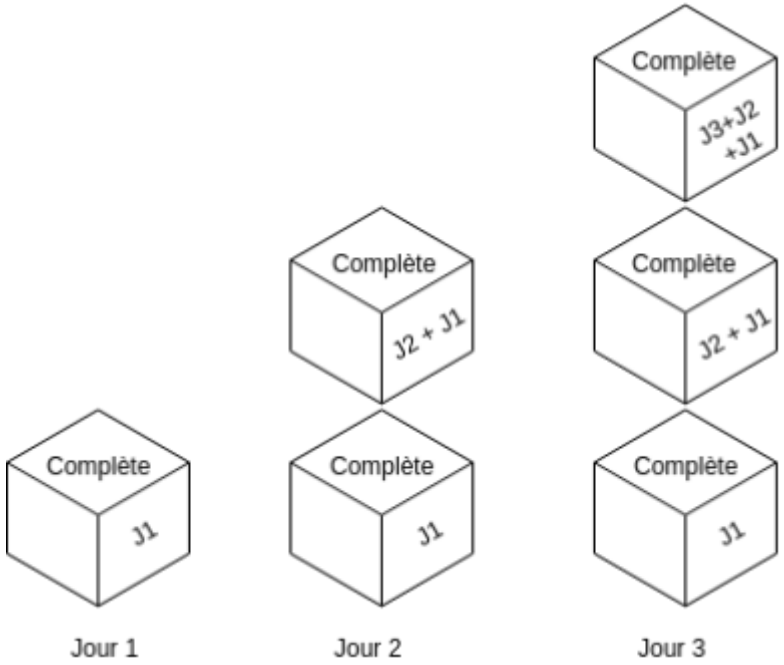
RH12409 – Archiving and Compression

Contents

Backup types

Full backup

In this case, all files and folders, the entire partition or the entire disk are backed up:



Restoring requires :

- The last full backup.

Advantage:

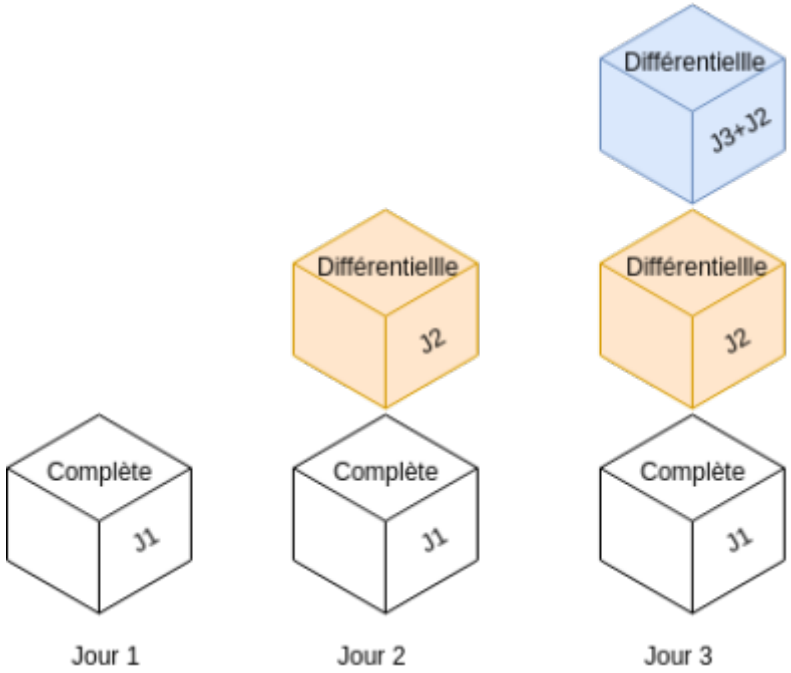
- Faster and simpler restoration than restoring from differential and incremental backups.

Disadvantage:

- Requires more storage space than differential and incremental backups.

Differential backup

In this case, modified or newly added data is backed up on the basis of the **last full backup**:



Restoration requires:

- The last full backup,
- The last differential backup.

Advantage :

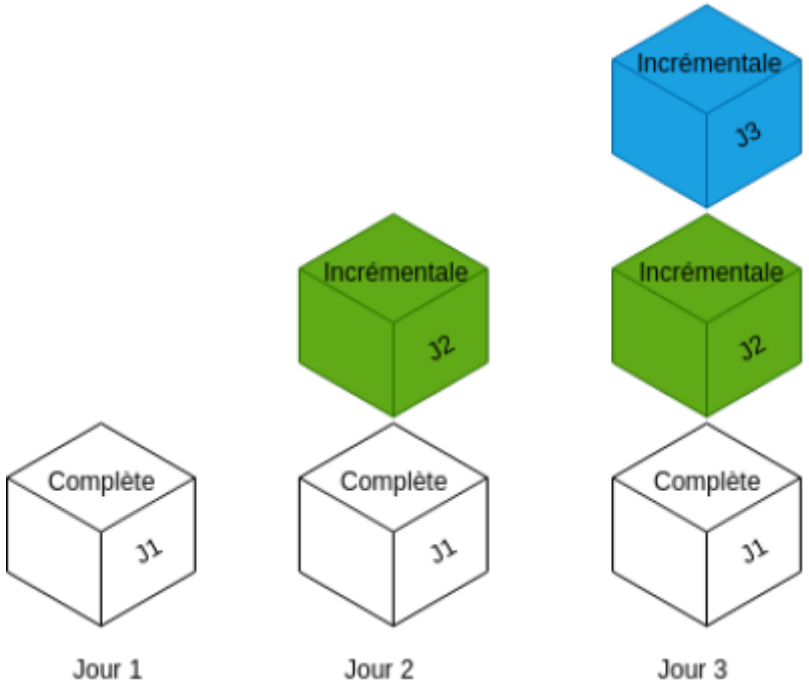
- Faster restoration than restoring from an incremental backup.

Disadvantage:

- Requires more storage space than an incremental backup.

Incremental backup

In this case, only what has changed since the last backup (full, differential or incremental) is backed up:



Restoration requires:

- The last full backup,
- All incremental backups made since, and in order.

Advantage:

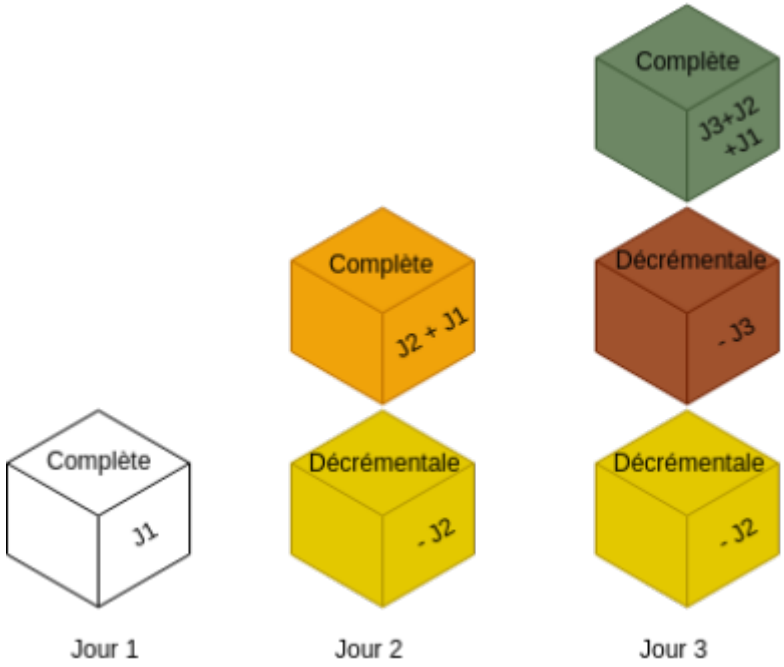
- Faster backup than full or differential backup,
- Requires less storage space than differential backup.

Disadvantage:

- Takes longer to restore.

Decremental backup

In this case, all the files are backed up and then a backup is produced of what has changed since the last full backup. This backup is called a decremental backup:



Restoration requires:

- The last full backup.

Restoring D-1 requires:

- The last full backup,
- The decremental D backup.

For example, to obtain a system status of D1 when you are at D2, you need to restore the full backup of D2, which is actually D2+D1, and then the

decremental backup -D2. In this way, and mathematically, we obtain : D2 + D1 -D2 = D1.

Advantages :

- Faster and simpler restoration than restoring from differential and incremental backups.

Disadvantage:

- Requires more data manipulation for each backup.

Classic backup tools

Preparation

Before proceeding further, you need to create some files and directories to backup and archive:

```
[root@redhat9 ~]# mkdir -p /test/dirY; mkdir /test/dirZ
[root@redhat9 ~]# cd /test/dirY; touch Y1 Y2 Y3
[root@redhat9 dirY]# cd /test/dirZ; touch Z1 Z2
[root@redhat9 dirZ]# ls -lR /test
/test:
total 0
drwxr-xr-x. 2 root root 36 Sep 27 07:51 dirY
drwxr-xr-x. 2 root root 26 Sep 27 07:51 dirZ

/test/dirY:
total 0
-rw-r--r--. 1 root root 0 Sep 27 07:51 Y1
-rw-r--r--. 1 root root 0 Sep 27 07:51 Y2
-rw-r--r--. 1 root root 0 Sep 27 07:51 Y3

/test/dirZ:
```

```
total 0
-rw-r--r--. 1 root root 0 Sep 27 07:51 Z1
-rw-r--r--. 1 root root 0 Sep 27 07:51 Z2
```

The tar Command

Overview

The **tar** program was originally intended for backing up to magnetic tape, hence its name from **tape archiver**.

The **tar** command can be used to archive or back-up files to:

- a special file, for example the name of a tape drive,
- an ordinary file on disk,
- standard output for use in a pipe.

LAB #1 - Working with the tar command

You can now proceed with the back-up of the directory **test** and it's contents to an ordinary file:

```
[root@redhat9 dirZ]# tar cvf /tmp/test.tar /test
tar: Removing leading `/' from member names
/test/
/test/dirY/
/test/dirY/Y1
/test/dirY/Y2
/test/dirY/Y3
/test/dirZ/
/test/dirZ/Z1
/test/dirZ/Z2
```

To view the **table of contents** of your backup, use the following command:

```
[root@redhat9 dirZ]# tar tvf /tmp/test.tar
drwxr-xr-x root/root 0 2024-09-27 07:51 test/
drwxr-xr-x root/root 0 2024-09-27 07:51 test/dirY/
-rw-r--r-- root/root 0 2024-09-27 07:51 test/dirY/Y1
-rw-r--r-- root/root 0 2024-09-27 07:51 test/dirY/Y2
-rw-r--r-- root/root 0 2024-09-27 07:51 test/dirY/Y3
drwxr-xr-x root/root 0 2024-09-27 07:51 test/dirZ/
-rw-r--r-- root/root 0 2024-09-27 07:51 test/dirZ/Z1
-rw-r--r-- root/root 0 2024-09-27 07:51 test/dirZ/Z2
```

In order to create an incremental back-up, you now need to create an empty file to be used as a time reference file. All files modified or created after the creation of this file will be included in the incremental archive:

```
[root@redhat9 dirZ]# touch /tmp/dateref
```

Now modify two of the files in your **test** tree:

```
[root@redhat9 dirZ]# echo ‘CentOS is great!’ > /test/dirY/Y1

[root@redhat9 dirZ]# echo ‘RHEL is wonderful!’ > /test/dirZ/Z1
```

To perform your incremental backup, you need to back up only the files modified or created since the creation of your **/tmp/dateref** file.

Enter the following command:

```
[root@redhat9 dirZ]# tar -cvf /tmp/incremental.tar -N /tmp/dateref /test
tar: Removing leading `/' from member names
/test/
/test/dirY/
/test/dirY/Y1
tar: /test/dirY/Y2: file is unchanged; not dumped
tar: /test/dirY/Y3: file is unchanged; not dumped
```

```
/test/dirZ/
/test/dirZ/Z1
tar: /test/dirZ/Z2: file is unchanged; not dumped
```

Important - Note the use of the **-N** option with the **/tmp/dateref** argument, which identifies files modified or created since **/tmp/dateref** was created.

Now check the contents of the **/tmp/incremental.tar** archive:

```
[root@redhat9 dirZ]# tar tvf /tmp/incremental.tar
drwxr-xr-x root/root 0 2024-09-27 07:51 test/
drwxr-xr-x root/root 0 2024-09-27 07:51 test/dirY/
-rw-r--r-- root/root 20 2024-09-27 07:58 test/dirY/Y1
drwxr-xr-x root/root 0 2024-09-27 07:51 test/dirZ/
-rw-r--r-- root/root 21 2024-09-27 07:58 test/dirZ/Z1
```

Now delete the contents of the **test** directory:

```
[root@redhat9 dirZ]# rm -rf /test/*
```

Important - Note that the system allows you to delete the **/test/dirZ** directory, yet you are located in this very directory!

In order to restore the files from your first backup, go to the root of your system and restore the contents of your **test** directory by entering the following tar command:

```
[root@redhat9 ~]# cd /
[root@redhat9 /]# tar xvf /tmp/test.tar
```

```
test/
test/dirY/
test/dirY/Y1
test/dirY/Y2
test/dirY/Y3
test/dirZ/
test/dirZ/Z1
test/dirZ/Z2
```

Using the **ls** command, you can check that the contents of the **test** directory have been restored:

```
root@redhat9 /]# ls -lR /test
/test:
total 0
drwxr-xr-x. 2 root root 36 Sep 27 07:51 dirY
drwxr-xr-x. 2 root root 26 Sep 27 07:51 dirZ

/test/dirY:
total 0
-rw-r--r--. 1 root root 0 Sep 27 07:51 Y1
-rw-r--r--. 1 root root 0 Sep 27 07:51 Y2
-rw-r--r--. 1 root root 0 Sep 27 07:51 Y3

/test/dirZ:
total 0
-rw-r--r--. 1 root root 0 Sep 27 07:51 Z1
-rw-r--r--. 1 root root 0 Sep 27 07:51 Z2
```

Important - Note that at this point the **/test/dirY/Y1** and **/test/dirZ/Z1** files are empty.

Now restore your incremental archive:

```
[root@redhat9 /]# tar xvf /tmp/incremental.tar
test/
test/dirY/
test/dirY/Y1
test/dirZ/
test/dirZ/Z1
```

Using the **ls** command, you can check that the contents of incremental.tar have been restored:

```
[root@redhat9 /]# ls -lR /test
/test:
total 0
drwxr-xr-x. 2 root root 36 Sep 27 07:51 dirY
drwxr-xr-x. 2 root root 26 Sep 27 07:51 dirZ

/test/dirY:
total 4
-rw-r--r--. 1 root root 20 Sep 27 07:58 Y1
-rw-r--r--. 1 root root 0 Sep 27 07:51 Y2
-rw-r--r--. 1 root root 0 Sep 27 07:51 Y3

/test/dirZ:
total 4
-rw-r--r--. 1 root root 21 Sep 27 07:58 Z1
-rw-r--r--. 1 root root 0 Sep 27 07:51 Z2
```

Important - Note that the **/test/dirY/Y1** and **/test/dirZ/Z1** files are now non-empty.

The GPL tar command and compression

Lastly, the tar command can archive using compression algorithms:

Algorithm	tar command option
gzip	z
bzip2	j
lzma	J

Command Line Switches

The command line switches for the tar command are:

```
[root@redhat9 dirZ]# tar --help
Usage: tar [OPTION...] [FILE]...
GNU ‘tar’ saves many files together into a single tape or disk archive, and can
restore individual files from the archive.

Examples:
  tar -cf archive.tar foo bar # Create archive.tar from files foo and bar.
  tar -tvf archive.tar # List all files in archive.tar verbosely.
  tar -xf archive.tar # Extract all files from archive.

 Main operation mode:
  -A, --catenate, --concatenate append tar files to an archive
  -c, --create create a new archive
      --delete delete from the archive (not on mag tapes!)
  -d, --diff, --compare find differences between archive and file system
  -r, --append append files to the end of an archive
      --test-label test the archive volume label and exit
  -t, --list list the contents of an archive
  -u, --update only append files newer than copy in archive
```

```
 -x, --extract, --get extract files from an archive

Operation modifiers:

     --check-device check device numbers when creating incremental
                           archives (default)
 -g, --listed-incremental=FILE handle new GNU-format incremental backup
 -G, --incremental handle old GNU-format incremental backup
     --hole-detection=TYPE technique to detect holes
     --ignore-failed-read do not exit with nonzero on unreadable files
     --level=NUMBER dump level for created listed-incremental archive
     --no-check-device do not check device numbers when creating
                           incremental archives
     --no-seek archive is not seekable
 -n, --seek archive is seekable
     --occurrence[=NUMBER] process only the NUMBERth occurrence of each file
                           in the archive; this option is valid only in
                           conjunction with one of the subcommands --delete,
                           --diff, --extract or --list and when a list of
                           files is given either on the command line or via
                           the -T option; NUMBER defaults to 1
     --sparse-version=MAJOR[.MINOR]
                           set version of the sparse format to use (implies
                           --sparse)
 -S, --sparse handle sparse files efficiently

Local file name selection:
     --add-file=FILE add given FILE to the archive (useful if its name
                           starts with a dash)
 -C, --directory=DIR change to directory DIR
     --exclude=PATTERN exclude files, given as a PATTERN
     --exclude-backups exclude backup and lock files
     --exclude-caches exclude contents of directories containing
                           CACHEDIR.TAG, except for the tag file itself
```

```
      --exclude-caches-all exclude directories containing CACHEDIR.TAG
      --exclude-caches-under exclude everything under directories containing
                            CACHEDIR.TAG
      --exclude-ignore=FILE read exclude patterns for each directory from
                            FILE, if it exists
      --exclude-ignore-recursive=FILE
                            read exclude patterns for each directory and its
                            subdirectories from FILE, if it exists
      --exclude-tag=FILE exclude contents of directories containing FILE,
                            except for FILE itself
      --exclude-tag-all=FILE exclude directories containing FILE
      --exclude-tag-under=FILE exclude everything under directories
                            containing FILE
      --exclude-vcs exclude version control system directories
      --exclude-vcs-ignores read exclude patterns from the VCS ignore files
      --no-null disable the effect of the previous --null option
      --no-recursion avoid descending automatically in directories
      --no-unquote do not unquote input file or member names
      --no-verbatim-files-from -T treats file names starting with dash as
                            options (default)
      --null -T reads null-terminated names; implies
                            --verbatim-files-from
      --recursion recurse into directories (default)
  -T, --files-from=FILE get names to extract or create from FILE
      --unquote unquote input file or member names (default)
      --verbatim-files-from -T reads file names verbatim (no escape or option
                            handling)
  -X, --exclude-from=FILE exclude patterns listed in FILE

File name matching options (affect both exclude and include patterns):

      --anchored patterns match file name start
      --ignore-case ignore case
      --no-anchored patterns match after any ‘/’ (default for
```

```
                            exclusion)
     --no-ignore-case case sensitive matching (default)
     --no-wildcards verbatim string matching
     --no-wildcards-match-slash wildcards do not match ‘/’ (default for exclusion)
     --wildcards use wildcards (default for exclusion)
     --wildcards-match-slash wildcards match ‘/’ (default)

Overwrite control:

     --keep-directory-symlink preserve existing symlinks to directories when
                            extracting
     --keep-newer-files don't replace existing files that are newer than
                            their archive copies
 -k, --keep-old-files don't replace existing files when extracting,
                            treat them as errors
     --no-overwrite-dir preserve metadata of existing directories
     --one-top-level[=DIR] create a subdirectory to avoid having loose files
                            extracted
     --overwrite overwrite existing files when extracting
     --overwrite-dir overwrite metadata of existing directories when
                            extracting (default)
     --recursive-unlink empty hierarchies prior to extracting directory
     --remove-files remove files after adding them to the archive
     --skip-old-files don't replace existing files when extracting,
                            silently skip over them
 -U, --unlink-first remove each file prior to extracting over it
 -W, --verify attempt to verify the archive after writing it

Select output stream:

     --ignore-command-error ignore exit codes of children
     --no-ignore-command-error treat non-zero exit codes of children as
                            error
 -O, --to-stdout extract files to standard output
```

```
      --to-command=COMMAND pipe extracted files to another program

Handling of file attributes:

      --atime-preserve[=METHOD] preserve access times on dumped files, either
                            by restoring the times after reading
                            (METHOD='replace'; default) or by not setting the
                            times in the first place (METHOD='system')
      --clamp-mtime only set time when the file is more recent than
                            what was given with --mtime
      --delay-directory-restore delay setting modification times and
                            permissions of extracted directories until the end
                            of extraction
      --group=NAME force NAME as group for added files
      --group-map=FILE use FILE to map file owner GIDs and names
      --mode=CHANGES force (symbolic) mode CHANGES for added files
      --mtime=DATE-OR-FILE set mtime for added files from DATE-OR-FILE
  -m, --touch don't extract file modified time
      --no-delay-directory-restore
                            cancel the effect of --delay-directory-restore
                            option
      --no-same-owner extract files as yourself (default for ordinary
                            users)
      --no-same-permissions apply the user's umask when extracting permissions
                            from the archive (default for ordinary users)
      --numeric-owner always use numbers for user/group names
      --owner=NAME force NAME as owner for added files
      --owner-map=FILE use FILE to map file owner UIDs and names
  -p, --preserve-permissions, --same-permissions
                            extract information about file permissions
                            (default for superuser)
      --same-owner try extracting files with the same ownership as
                            exists in the archive (default for superuser)
      --sort=ORDER directory sorting order: none (default), name or
```

```
                              inode
 -s, --preserve-order, --same-order
                              member arguments are listed in the same order as
                              the files in the archive

Handling of extended file attributes:

     --acls Enable the POSIX ACLs support
     --no-acls Disable the POSIX ACLs support
     --no-selinux Disable the SELinux context support
     --no-xattrs Disable extended attributes support
     --selinux Enable the SELinux context support
     --xattrs Enable extended attributes support
     --xattrs-exclude=MASK specify the exclude pattern for xattr keys
     --xattrs-include=MASK specify the include pattern for xattr keys

Device selection and switching:

     --force-local archive file is local even if it has a colon
 -f, --file=ARCHIVE use archive file or device ARCHIVE
 -F, --info-script=NAME, --new-volume-script=NAME
                              run script at end of each tape (implies -M)
 -L, --tape-length=NUMBER change tape after writing NUMBER x 1024 bytes
 -M, --multi-volume create/list/extract multi-volume archive
     --rmt-command=COMMAND use given rmt COMMAND instead of rmt
     --rsh-command=COMMAND use remote COMMAND instead of rsh
     --volno-file=FILE use/update the volume number in FILE

Device blocking:

 -b, --blocking-factor=BLOCKS BLOCKS x 512 bytes per record
 -B, --read-full-records reblock as we read (for 4.2BSD pipes)
 -i, --ignore-zeros ignore zeroed blocks in archive (means EOF)
     --record-size=NUMBER NUMBER of bytes per record, multiple of 512
```

```
Archive format selection:

 -H, --format=FORMAT create archive of the given format

FORMAT is one of the following:
   gnu GNU tar 1.13.x format
   oldgnu GNU format as per tar <= 1.12
   pax POSIX 1003.1-2001 (pax) format
   posix same as pax
   ustar POSIX 1003.1-1988 (ustar) format
   v7 old V7 tar format

     --old-archive, --portability
                            same as --format=v7
     --pax-option=keyword[[:]=value][,keyword[[:]=value]]...
                            control pax keywords
     --posix same as --format=posix
 -V, --label=TEXT create archive with volume name TEXT; at
                            list/extract time, use TEXT as a globbing pattern
                            for volume name

Compression options:

 -a, --auto-compress use archive suffix to determine the compression
                            program
 -I, --use-compress-program=PROG
                            filter through PROG (must accept -d)
 -j, --bzip2 filter the archive through bzip2
 -J, --xz filter the archive through xz
     --lzip filter the archive through lzip
     --lzma filter the archive through xz --format=lzma
     --lzop filter the archive through lzop
     --no-auto-compress do not use archive suffix to determine the
                            compression program
```

```
      --zstd filter the archive through zstd
  -z, --gzip, --gunzip, --ungzip filter the archive through gzip
  -Z, --compress, --uncompress filter the archive through compress

Local file selection:

      --backup[=CONTROL] backup before removal, choose version CONTROL
      --hard-dereference follow hard links; archive and dump the files they
                            refer to
  -h, --dereference follow symlinks; archive and dump the files they
                            point to
  -K, --starting-file=MEMBER-NAME
                            begin at member MEMBER-NAME when reading the
                            archive
      --newer-mtime=DATE compare date and time when data changed only
  -N, --newer=DATE-OR-FILE, --after-date=DATE-OR-FILE
                            only store files newer than DATE-OR-FILE
      --one-file-system stay in local file system when creating archive
  -P, --absolute-names don't strip leading ‘/’s from file names
      --suffix=STRING backup before removal, override usual suffix (‘~’
                            unless overridden by environment variable
                            SIMPLE_BACKUP_SUFFIX)

File name transformations:

      --strip-components=NUMBER strip NUMBER leading components from file
                            names on extraction
      --transform=EXPRESSION, --xform=EXPRESSION
                            use sed replace EXPRESSION to transform file
                            names

Informative output:

      --checkpoint[=NUMBER] display progress messages every NUMBERth record
```

```
                            (default 10)
      --checkpoint-action=ACTION execute ACTION on each checkpoint
      --full-time print file time to its full resolution
      --index-file=FILE send verbose output to FILE
  -l, --check-links print a message if not all links are dumped
      --no-quote-chars=STRING disable quoting for characters from STRING
      --quote-chars=STRING additionally quote characters from STRING
      --quoting-style=STYLE set name quoting style; see below for valid STYLE
                            values
  -R, --block-number show block number within archive with each message
      --show-defaults show tar defaults
      --show-omitted-dirs when listing or extracting, list each directory
                            that does not match search criteria
      --show-snapshot-field-ranges
                            show valid ranges for snapshot-file fields
      --show-transformed-names, --show-stored-names
                            show file or archive names after transformation
      --totals[=SIGNAL] print total bytes after processing the archive;
                            with an argument - print total bytes when this
                            SIGNAL is delivered; Allowed signals are: SIGHUP,
                            SIGQUIT, SIGINT, SIGUSR1 and SIGUSR2; the names
                            without SIG prefix are also accepted
      --utc print file modification times in UTC
  -v, --verbose verbosely list files processed
      --warning=KEYWORD warning control
  -w, --interactive, --confirmation
                            ask for confirmation for every action

Compatibility options:

  -o when creating, same as --old-archive; when
                            extracting, same as --no-same-owner

Other options:
```

```
  -?, --help give this help list
      --restrict disable use of some potentially harmful options
      --usage give a short usage message
      --version print program version

Mandatory or optional arguments to long options are also mandatory or optional
for any corresponding short options.

The backup suffix is ‘~’, unless set with --suffix or SIMPLE_BACKUP_SUFFIX.
The version control may be set with --backup or VERSION_CONTROL, values are:

  none, off never make backups
  t, numbered make numbered backups
  nil, existing numbered if numbered backups exist, simple otherwise
  never, simple always make simple backups

Valid arguments for the --quoting-style option are:

  literal
  shell
  shell-always
  shell-escape
  shell-escape-always
  c
  c-maybe
  escape
  local
  clocale

*This* tar defaults to:
--format=gnu -f- -b20 --quoting-style=escape --rmt-command=/etc/rmt
--rsh-command=/usr/bin/ssh
```

The cpio Command

Overview

The **cpio** (Copy Input To Output) command. cpio can handle archives in **tar** format. The major difference between tar and cpio is that the latter stores the paths to the saved files at the same time as the files themselves. This means that if the absolute path was specified at the time of backup, it is impossible to restore a file to a location other than its original location.

You will now use the **cpio** software to perform backups and restores.

LAB #2 - Working with the cpio command

As a first step, you need to use the **find** command to build a list of files to back up:

```
[root@redhat9 /]# find /test > /tmp/cpio.list
[root@redhat9 /]# cat /tmp/cpio.list
/test
/test/dirY
/test/dirY/Y2
/test/dirY/Y3
/test/dirY/Y1
/test/dirZ
/test/dirZ/Z2
/test/dirZ/Z1
```

Now back up the files and directories referenced by the **/tmp/cpio.list** file:

```
[root@redhat9 /]# cpio -ov < /tmp/cpio.list > /tmp/test.cpio
/test
/test/dirY
/test/dirY/Y2
```

```
/test/dirY/Y3
/test/dirY/Y1
/test/dirZ
/test/dirZ/Z2
/test/dirZ/Z1
1 block
```

Now look at the **table of contents** of your backup:

```
[root@redhat9 /]# cpio -it < /tmp/test.cpio
/test
/test/dirY
/test/dirY/Y2
/test/dirY/Y3
/test/dirY/Y1
/test/dirZ
/test/dirZ/Z2
/test/dirZ/Z1
1 block
```

Now delete the **/test/dirY** directory and its contents:

```
[root@redhat9 /]# rm -rf /test/dirY
```

Check that the deletion is successful:

```
[root@redhat9 /]# ls -lR /test
/test:
total 0
drwxr-xr-x. 2 root root 26 Sep 27 07:51 dirZ

/test/dirZ:
total 4
-rw-r--r--. 1 root root 21 Sep 27 07:58 Z1
```

```
-rw-r--r--. 1 root root 0 Sep 27 07:51 Z2
```

Restore the deleted files with the following command:

```
[root@redhat9 /]# cpio -ivdum '/test/dirY/*' < /tmp/test.cpio
/test/dirY/Y2
/test/dirY/Y3
/test/dirY/Y1
1 block
```

Important - Note the use of the string **'/test/dirY/*'** which searches only the **dirY** directory as well as the **Y1**, **Y2** and **Y3** files in the test.cpio archive.

Using the **ls** command, you can check that the contents of repY have been restored:

```
[root@redhat9 /]# ls -lR /test
/test:
total 0
drwxr-xr-x. 2 root root 36 Sep 27 08:10 dirY
drwxr-xr-x. 2 root root 26 Sep 27 07:51 dirZ

/test/dirY:
total 4
-rw-r--r--. 1 root root 20 Sep 27 07:58 Y1
-rw-r--r--. 1 root root 0 Sep 27 07:51 Y2
-rw-r--r--. 1 root root 0 Sep 27 07:51 Y3

/test/dirZ:
total 4
-rw-r--r--. 1 root root 21 Sep 27 07:58 Z1
-rw-r--r--. 1 root root 0 Sep 27 07:51 Z2
```

Command Line Switches

The command line switches for the **cpio** command are:

```
[root@redhat9 /]# cpio --help
Usage: cpio [OPTION...] [destination-directory]
GNU `cpio' copies files to and from archives

Examples:
  # Copy files named in name-list to the archive
  cpio -o < name-list [> archive]
  # Extract files from the archive
  cpio -i [< archive]
  # Copy files named in name-list to destination-directory
  cpio -p destination-directory < name-list

 Main operation mode:
  -i, --extract Extract files from an archive (run in copy-in
                              mode)
  -o, --create Create the archive (run in copy-out mode)
  -p, --pass-through Run in copy-pass mode
  -t, --list Print a table of contents of the input

 Operation modifiers valid in any mode:

      --block-size=BLOCK-SIZE Set the I/O block size to BLOCK-SIZE * 512
                              bytes
  -B Set the I/O block size to 5120 bytes
  -c Identical to ‘-H newc’, use the new (SVR4)
                              portable format. If you wish the old portable
                              (ASCII) archive format, use ‘-H odc’ instead.
  -C, --io-size=NUMBER Set the I/O block size to the given NUMBER of
                              bytes
```

```
 -D, --directory=DIR Change to directory DIR
     --force-local Archive file is local, even if its name contains
                           colons
 -H, --format=FORMAT Use given archive FORMAT
     --quiet Do not print the number of blocks copied
 -R, --owner=[USER][:.][GROUP] Set the ownership of all files created to the
                           specified USER and/or GROUP
 -v, --verbose Verbosely list the files processed
 -V, --dot Print a '.' for each file processed
 -W, --warning=FLAG Control warning display. Currently FLAG is one of
                           none', "truncate", "all". Multiple options
                           accumulate.

Operation modifiers valid in copy-in and copy-out modes

 -F, --file=[[USER@]HOST:]FILE-NAME
                           Use this FILE-NAME instead of standard input or
                           output. Optional USER and HOST specify the user
                           and host names in case of a remote archive
 -M, --message=STRING Print STRING when the end of a volume of the
                           backup media is reached
     --rsh-command=COMMAND Use COMMAND instead of rsh

Operation modifiers valid only in copy-in mode:

 -b, --swap Swap both halfwords of words and bytes of
                           halfwords in the data. Equivalent to -sS
 -f, --nonmatching Only copy files that do not match any of the given
                           patterns
 -I [[USER@]HOST:]FILE-NAME Archive filename to use instead of standard input.
                           Optional USER and HOST specify the user and host
                           names in case of a remote archive
 -n, --numeric-uid-gid In the verbose table of contents listing, show
                           numeric UID and GID
```

```
 -r, --rename Interactively rename files
 -s, --swap-bytes Swap the bytes of each halfword in the files
 -S, --swap-halfwords Swap the halfwords of each word (4 bytes) in the
                            files
     --to-stdout Extract files to standard output

 -E, --pattern-file=FILE Read additional patterns specifying filenames to
                            extract or list from FILE
     --only-verify-crc When reading a CRC format archive, only verify the
                            checksum of each file in the archive, don't
                            actually extract the files

Operation modifiers valid only in copy-out mode:

 -A, --append Append to an existing archive.
     --device-independent, --reproducible
                            Create device-independent (reproducible) archives
     --ignore-devno Don't store device numbers
 -O [[USER@]HOST:]FILE-NAME Archive filename to use instead of standard
                            output. Optional USER and HOST specify the user
                            and host names in case of a remote archive
     --renumber-inodes Renumber inodes

Operation modifiers valid only in copy-pass mode:

 -l, --link Link files instead of copying them, when
                            possible

Operation modifiers valid in copy-in and copy-out modes:

     --absolute-filenames Do not strip file system prefix components from
                            the file names
     --no-absolute-filenames Create all files relative to the current
                            directory
```

```
 Operation modifiers valid in copy-out and copy-pass modes:

  -0, --null Filenames in the list are delimited by null
                             characters instead of newlines
  -a, --reset-access-time Reset the access times of files after reading
                             them
  -L, --dereference Dereference symbolic links (copy the files
                             that they point to instead of copying the links).

 Operation modifiers valid in copy-in and copy-pass modes:

  -d, --make-directories Create leading directories where needed
  -m, --preserve-modification-time
                             Retain previous file modification times when
                             creating files
      --no-preserve-owner Do not change the ownership of the files
      --sparse Write files with large blocks of zeros as sparse
                             files
  -u, --unconditional Replace all files unconditionally

  -?, --help give this help list
      --usage give a short usage message
      --version print program version

Mandatory or optional arguments to long options are also mandatory or optional
for any corresponding short options.

Report bugs to <bug-cpio@gnu.org>.
```

The dd command

Overview

The **dd** command is not actually a backup command.

The **dd** command copies the file passed as input to the output file, limiting the number of bytes copied by using two command line switches:

- **count**
  - the number of blocks
- **bs**
  - the size of each block

LAB #3 - Working with the dd command

You will now use **dd** to make a backup of your MBR and partition table.

Make a backup of your MBR, which is located in the first 446 bytes of your **/dev/sda** disk:

```
[root@redhat9 /]# dd if=/dev/sda of=/tmp/mbr.save bs=1 count=446
446+0 records in
446+0 records out
446 bytes copied, 0.00114645 s, 389 kB/s
```

Now make a backup of your partition table which is in the 64 bytes after the 446 previously saved:

```
[root@redhat9 /]# dd if=/dev/sda of=/tmp/tblpart.save bs=1 count=64 skip=446
64+0 records in
64+0 records out
64 bytes copied, 0.000282251 s, 227 kB/s
```

Important - Note the use of the **skip** option, which positions the start of the backup at the 447th byte.

Command Line Switches

The **dd** Command Line Switches are:

```
[root@redhat9 /]# dd --help
Usage: dd [OPERAND]...
  or: dd OPTION
Copy a file, converting and formatting according to the operands.

  bs=BYTES read and write up to BYTES bytes at a time (default: 512);
                 overrides ibs and obs
  cbs=BYTES convert BYTES bytes at a time
  conv=CONVS convert the file as per the comma separated symbol list
  count=N copy only N input blocks
  ibs=BYTES read up to BYTES bytes at a time (default: 512)
  if=FILE read from FILE instead of stdin
  iflag=FLAGS read as per the comma separated symbol list
  obs=BYTES write BYTES bytes at a time (default: 512)
  of=FILE write to FILE instead of stdout
  oflag=FLAGS write as per the comma separated symbol list
  seek=N skip N obs-sized blocks at start of output
  skip=N skip N ibs-sized blocks at start of input
  status=LEVEL The LEVEL of information to print to stderr;
                 none' suppresses everything but error messages,
                 noxfer' suppresses the final transfer statistics,
                 progress' shows periodic transfer statistics

N and BYTES may be followed by the following multiplicative suffixes:
c=1, w=2, b=512, kB=1000, K=1024, MB=1000*1000, M=1024*1024, xM=M,
GB=1000*1000*1000, G=1024*1024*1024, and so on for T, P, E, Z, Y.
Binary prefixes can be used, too: KiB=K, MiB=M, and so on.

Each CONV symbol may be:
```

```
  ascii from EBCDIC to ASCII
  ebcdic from ASCII to EBCDIC
  ibm from ASCII to alternate EBCDIC
  block pad newline-terminated records with spaces to cbs-size
  unblock replace trailing spaces in cbs-size records with newline
  lcase change upper case to lower case
  ucase change lower case to upper case
  sparse try to seek rather than write all-NUL output blocks
  swab swap every pair of input bytes
  sync pad every input block with NULs to ibs-size; when used
            with block or unblock, pad with spaces rather than NULs
  excl fail if the output file already exists
  nocreat do not create the output file
  notrunc do not truncate the output file
  noerror continue after read errors
  fdatasync physically write output file data before finishing
  fsync likewise, but also write metadata

Each FLAG symbol may be:

  append append mode (makes sense only for output; conv=notrunc suggested)
  direct use direct I/O for data
  directory fail unless a directory
  dsync use synchronized I/O for data
  sync likewise, but also for metadata
  fullblock accumulate full blocks of input (iflag only)
  nonblock use non-blocking I/O
  noatime do not update access time
  nocache Request to drop cache.  See also oflag=sync
  noctty do not assign controlling terminal from file
  nofollow do not follow symlinks
  count_bytes treat ‘count=N’ as a byte count (iflag only)
  skip_bytes treat ‘skip=N’ as a byte count (iflag only)
  seek_bytes treat ‘seek=N’ as a byte count (oflag only)
```

```
Sending a USR1 signal to a running ‘dd’ process makes it
print I/O statistics to standard error and then resume copying.

Options are:

      --help display this help and exit
      --version output version information and exit

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation <https://www.gnu.org/software/coreutils/dd>
or available locally via: info ‘(coreutils) dd invocation’
```

Dump and restore commands

Overview

The **dump** and **restore** commands are based on the data storage format (ext3). For this reason, it is not possible to backup directories within a file system, only complete file systems.

It is important to note that the file system must not be used during the dump process. For this reason it is normally advisable to unmount the file system.

There are 10 possible dump levels from **0** to **9**. When a dump is performed, the level is specified. Each time a dump is performed, this information is saved in the /etc/dumpdates file.

By definition, a level **0** dump is a full backup, while a level 1 dump is an incremental backup.

Note that files are backed up with relative names. This means that you must position yourself in the file system when restoring with the **restore** command.

Advanced Backup Tools

Unidirectional Backup Tools

These tools back up files to a local or remote directory in one direction only.

This first table compares the tools in terms of the characteristics and capabilities of the backup operations:

Tool	Backend	Differential backup	Incremental backup	Decremental backup	Built-in scheduling	Built-in restore	Encryption	Compression	Site
AMANDA	tar, gzip	✗	✓	✗	✗	✓	✓	✓	**AMANDA**
Areca Backup	Aucun	✓	✓	✗	✓	✓	✓	✓	**Areca Backup**
bacula (Bareos)	MySQL	✓	✓	✗	✓	✓	✓	✓	**bacula**
BackInTime	rsync, diff, meld	✗	✓	✗	✓	✓	✗	✗	**BackInTime**
BackupPC	rsync, samba, tar	✗	✓	✗	✓	✓	✗	✓	**BackupPC**
Dar	Aucun	✓	✓	✓	✓	✓	✓	✓	**Dar**
Déjà Dup	duplicity	✗	✓	✗	✓	✓	✓	✗	**Duplicity**
Grsync	rsync	✗	✓	✗	✓	✓	✗	✗	**Grsync**
luckyBackup	rsync	✗	✓	✗	✓	✓	✗	✗	**luckyBackup**
SBackup	Aucun	✗	✓	✗	✓	✓	✗	✓	**SBackup**

To Do - See **List of backup software** on Wikipedia for more information.

This second table compares tools sorted by date of the last known version:

Tool	Written in	Licence	Last Version	Version Date	Linux	Windows	Mac	Website
Dar	C++	GPL	2.6.2	09/02/2019	✓	✓	✓	**Dar**
Bacula (Bareos)	C, C++	GNU Affero General Public License v3.0	9.4.2	04/02/2019	✓	✓	✓	**bacula**
BackupPC	Perl	GPL v3.0	4.3.0	25/11/2018	✓	✓	✓	**BackupPC**
luckyBackup	C++	GPL v3.0	0.5.0	18/11/2018	✓	✓	✓	**luckyBackup**
Déjà Dup (Duplicity)	Python	GPL	0.7.18.2	17/11/2018	✓	✓	✓	**Duplicity**
AMANDA	C, Perl	GPL, LGPL, Apache, Amanda License	3.5.1	01/12/2017	✓	✓	✓	**AMANDA**
BackInTime	Python3	GPL	1.1.24	07/11/2017	✓	✗	✗	**BackInTime**
Grsync	GTK	GPL	1.2.6	15/03/2016	✓	✓	✓	**Grsync**
Areca Backup	Java	GPLv2	7.5	26/08/2015	✓	✓	✓	**Areca Backup**
SBackup	GTK	GPLv2	0.11.6	24/02/2014	✓	✗	✗	**SBackup**

Multidirectional Backup Tools

These tools synchronise files between two servers.

This first table compares the tools in relation to the characteristics and capabilities of the backup operations:

Tool	Backend	Differential backup	Incremental backup	Decremental backup	Built-in scheduling	Built-in restore	Encryption	Compression
FullSync	smb, ftp, sftp	✗	✓	✗	✓	S/O	✗	✗
FreeFileSync	Aucun	✗	✓	✗	✓	S/O	✗	✗

Tool	Backend	Differential backup	Incremental backup	Decremental backup	Built-in scheduling	Built-in restore	Encryption	Compression
unison	SSH, RSH	✗	✓	✗	✓	S/O	✗	✗
Synkron	Aucun	✗	✓	✗	✓	✓	✗	✗

To Do - See the **Comparison of file synchronization software** page on Wikipedia for more information...

This second table compares tools sorted by date of the last known version:

Tool	Written in	License	Last Version	Version Date	Linux	Windows	Mac	Website
FreeFileSync	Divers	GPL v3.0	10.8	05/01/2019	✓	✓	✓	**FreeFileSync**
unison	OCaml	GPL v3.0	2.51.2	27/01/2018	✓	✓	✓	**unison**
FullSync	Java	GPLv2	0.10.4	05/04/2016	✓	✓	✗	**FullSync**
Synkron	C++	GPL v2	1.6.2	25/01/2011	✓	✓	✓	**Synkron**

Partition Backup Tools

This first table compares the tools in relation to the characteristics and capabilities of the backup operations:

Tool	Backend	Differential backup	Incremental backup	Decremental backup	Built-in scheduling	Built-in restore	Encryption	Compression
CloneZilla	SSH, samba, NFS	✗	✗	✗	✗	✓	✓	✓
Partclone	Aucun	✗	✗	✗	✗	✓	✗	✓

Tool	Backend	Differential backup	Incremental backup	Decremental backup	Built-in scheduling	Built-in restore	Encryption	Compression
partimage	Aucun	✗	✗	✗	✗	✓	✗	✗

To Do - See the **Comparison of disk cloning software** page on Wikipedia for more information.

This second table compares tools sorted by the date of last known version:

Tool	Written in	Licence	Last Version	Version Date	Linux	Windows	Mac	Website
CloneZilla	Perl, Unix shell	GPL	2.6.0-37	10/01/2019	✓	✓	✓	CloneZilla
Partclone	C	GPL	0.2.89	05/07/2016	✓	✓	✓	Partclone
partimage	C	GPL	0.6.9	25/07/2010	✓	✓	✗	Partimage

LAB #4 - What to Backup

Backup the Package List

Remove the lock files from the RPM database:

```
root@redhat9 /]# rm -f /var/lib/rpm/__db*
```

Back up the RPM databases:

```
[root@redhat9 /]# tar czvf $(hostname).rpmdatabase.tar.gz /var/lib/rpm
tar: Removing leading `/' from member names
/var/lib/rpm/
```

```
/var/lib/rpm/rpmdb.sqlite
/var/lib/rpm/rpmdb.sqlite-wal
/var/lib/rpm/rpmdb.sqlite-shm
/var/lib/rpm/.rpm.lock
```

To backup a list of the installed RPM packages, use the following command:

```
[root@redhat9 /]# rpm -qa > list-of-packages_`hostname`_`date +%Y-%m-%d-%H-%M`
```

Consult the contents of the file:

```
[root@redhat9 /]# more list-of-packages_redhat9.ittraining.loc_2024-09-27-08-15
fonts-filesystem-2.0.5-7.el9.1.noarch
xkeyboard-config-2.33-2.el9.noarch
abattis-cantarell-fonts-0.301-4.el9.noarch
yelp-xsl-40.2-1.el9.noarch
mozilla-filesystem-1.9-30.el9.x86_64
google-noto-fonts-common-20201206-4.el9.noarch
foomatic-db-filesystem-4.0-72.20210209.el9.noarch
adobe-mappings-cmap-20171205-12.el9.noarch
subscription-manager-rhsm-certificates-20220623-1.el9.noarch
libreport-filesystem-2.15.2-6.el9.noarch
adobe-mappings-cmap-deprecated-20171205-12.el9.noarch
adobe-source-code-pro-fonts-2.030.1.050-12.el9.1.noarch
dejavu-without-mono-fonts-2.37-18.el9.noarch
dejavu-without-fonts-2.37-18.el9.noarch
langpacks-core-font-en-3.0-16.el9.noarch
google-droid-sans-fonts-20200215-11.el9.2.noarch
thai-scalable-fonts-common-0.7.2-5.el9.noarch
redhat-indexhtml-9-4.el9_2.noarch
poppler-data-0.4.9-9.el9.noarch
mobile-broadband-provider-info-20210805-2.el9.noarch
man-pages-overrides-9.0.0.0-1.el9.noarch
hunspell-filesystem-1.7.0-11.el9.x86_64
```

```
hplip-common-3.21.2-6.el9.x86_64
gawk-all-langpacks-5.1.0-6.el9.x86_64
adwaita-cursor-theme-40.1.1-3.el9.noarch
adobe-mappings-pdf-20180407-10.el9.noarch
vim-filesystem-8.2.2637-20.el9_1.noarch
rhsm-icons-6-1.el9.noarch
filesystem-3.16-2.el9.x86_64
urw-base35-fonts-common-20200910-6.el9.noarch
basesystem-11-13.el9.noarch
quota-nls-4.06-6.el9.noarch
publicsuffix-list-dafsa-20210518-3.el9.noarch
pkgconf-m4-1.7.3-10.el9.noarch
popt-1.18-8.el9.x86_64
xz-libs-5.2.5-8.el9_0.x86_64
libxcrypt-4.4.18-3.el9.x86_64
bzip2-libs-1.0.8-8.el9.x86_64
libzstd-1.5.1-2.el9.x86_64
libpng-1.6.37-12.el9.x86_64
libcap-ng-0.8.2-7.el9.x86_64
libicu-67.1-9.el9.x86_64
libunistring-0.9.10-15.el9.x86_64
libgpg-error-1.42-5.el9.x86_64
libseccomp-2.5.2-2.el9.x86_64
lcms2-2.12-3.el9.x86_64
readline-8.1-4.el9.x86_64
libwayland-client-1.21.0-1.el9.x86_64
libwayland-server-1.21.0-1.el9.x86_64
jansson-2.14-1.el9.x86_64
libxkbcommon-1.0.3-4.el9.x86_64
libwayland-egl-1.21.0-1.el9.x86_64
keyutils-libs-1.6.3-1.el9.x86_64
libdhash-0.5.0-53.el9.x86_64
libXau-1.0.9-8.el9.x86_64
--More--(4%)
```

```
[q]
```

Important - Next, the two files **list-of-packages_*** and **$(hostname).rpmdatabase.tar.gz** should be backed up on external media.

In order to restore the backups, retrieve the two files **package-list_*** and **$(hostname).rpmdatabase.tar.gz** from the external media and place them at the root of the file system:

```
[root@redhat9 ~]# cp list-of-packages_redhat9.ittraining.loc_2024-09-27-08-15
redhat9.ittraining.loc.rpmdatabase.tar.gz /
```

Go to the root of the file system and restore the RPM databases:

```
[root@redhat9 /]# tar xvf redhat9.ittraining.loc.rpmdatabase.tar.gz
var/lib/rpm/
var/lib/rpm/rpmdb.sqlite
var/lib/rpm/rpmdb.sqlite-wal
var/lib/rpm/rpmdb.sqlite-shm
var/lib/rpm/.rpm.lock
```

Use dnf to restore packages:

```
[root@redhat9 /]# dnf -y install -y $(cat list-of-packages_redhat9.ittraining.loc_2024-09-27-08-15)
```

Backing up the System Hard Disk Structure

```
[root@redhat9 /]# cd ~

[root@redhat9 ~]# fdisk -l /dev/sda > structure.list
```

```
[root@redhat9 ~]# cat structure.list
Disk /dev/sda: 50 GiB, 53687091200 bytes, 104857600 sectors
Disk model: QEMU HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xd00dfc8a

Device Boot Start End Sectors Size Id Type
/dev/sda1 * 2048 2099199 2097152 1G 83 Linux
/dev/sda2 2099200 104857599 102758400 49G 8e Linux LVM
```

Important - The **structure.list** file should then be backed up on external media.

Backing up the System Hard Disk Mount Points

Enter the following command:

```
[root@redhat9 ~]# df -h | grep "^/dev/" > mounts.list
[root@redhat9 ~]# cat mounts.list
/dev/mapper/rhel-root   44G  7.8G   37G  18% /
/dev/sda1             1014M  398M  617M  40% /boot
```

Important - The **mounts.list** file should then be saved on external media.

Bootloader backup

GRUB Legacy

```
# cp /boot/grub/menu.lst grubmenu.lst
```

GRUB 2 with BIOS

Enter the following commands:

```
[root@redhat9 ~]# cp /boot/grub2/grub.cfg grub.cfg
[root@redhat9 ~]# cp /boot/grub2/device.map device.map
```

GRUB 2 with EFI

```
# cp /boot/efi/EFI/redhat/grub.cfg grub.cfg
# cp /boot/grub2/device.map device.map
```

Important - You should then save the **menu.lst** or **grub.cfg** file and the **device.map** file on external media.

Backing up User Folders

```
[root@redhat9 ~]# cp -apv /home/ .
‘/home/’ -> ‘./home’
```

```
‘/home/trainee’ -> ‘./home/trainee’
/home/trainee/training' -> ’./home/trainee/training
/home/trainee/training/f1' -> ’./home/trainee/training/f1
‘/home/trainee/training/f2’ -> ‘./home/trainee/training/f2’
‘/home/trainee/training/f3’ -> ‘./home/trainee/training/f3’
/home/trainee/training/f4' -> ’./home/trainee/training/f4
/home/trainee/training/f5' -> ’./home/trainee/training/f5
/home/trainee/training/f52' -> ’./home/trainee/training/f52
/home/trainee/training/f62' -> ’./home/trainee/training/f62
‘/home/trainee/training/a100’ -> ‘./home/trainee/training/a100’
‘/home/trainee/training/f’ -> ‘./home/trainee/training/f’
‘/home/trainee/training/f.txt’ -> ‘./home/trainee/training/f.txt’
/home/trainee/training/f123.txt' -> ’./home/trainee/training/f123.txt
‘/home/trainee/training/f123123.txt’ -> ‘./home/trainee/training/f123123.txt’
/home/trainee/training/f123123123.txt' -> “./home/trainee/training/f123123123.txt”
/home/trainee/training/file' -> ’./home/trainee/training/file
/home/trainee/training/user_check' -> ’./home/trainee/training/user_check
/home/trainee/bin' -> ’./home/trainee/bin
/home/trainee/bin/myscript' -> ’./home/trainee/bin/myscript
/home/trainee/Desktop' -> ’./home/trainee/Desktop
/home/trainee/Documents' -> ’./home/trainee/Documents
/home/trainee/.config' -> ’./home/trainee/.config
/home/trainee/.config/ibus' -> ’./home/trainee/.config/ibus
/home/trainee/.config/ibus/bus' -> ’./home/trainee/.config/ibus/bus
‘/home/trainee/.config/ibus/bus/5a35a3eb625c45cea1d33535723e791f-unix-wayland-0’ ->
‘./home/trainee/.config/ibus/bus/5a35a3eb625c45cea1d33535723e791f-unix-wayland-0’
‘/home/trainee/.config/user-dirs.dirs’ -> ‘./home/trainee/.config/user-dirs.dirs’
/home/trainee/.config/user-dirs.locale' -> ’./home/trainee/.config/user-dirs.locale
/home/trainee/.config/evolution' -> ’./home/trainee/.config/evolution
/home/trainee/.config/evolution/sources' -> ’./home/trainee/.config/evolution/sources
/home/trainee/.config/evolution/sources/system-proxy.source' -> ’./home/trainee/.config/evolution/sources/system-
proxy.source
...
```

Important - The **/root/home** folder should then be backed up on external media.

The Rsync command

Overview

Rsync or *Remote Sync* is a file synchronisation utility that uses an algorithm that minimises the amount of data copied by only copying the parts of files that have been modified.

LAB #5 - Working with the rsync command

Create the directories **/test/dirA** and mkdir **/test/dirB** :

```
[root@redhat9 ~]# mkdir -p /test/dirA; mkdir /test/dirB
```

Now create 20 empty files in the **/test/dirA** directory:

```
[root@redhat9 ~]# touch /test/dirA/file{1..20}
[root@redhat9 ~]# ls -l /test/dirA/
total 0
-rw-r--r--. 1 root root 0 Sep 27 11:43 file1
-rw-r--r--. 1 root root 0 Sep 27 11:43 file10
-rw-r--r--. 1 root root 0 Sep 27 11:43 file11
-rw-r--r--. 1 root root 0 Sep 27 11:43 file12
-rw-r--r--. 1 root root 0 Sep 27 11:43 file13
-rw-r--r--. 1 root root 0 Sep 27 11:43 file14
-rw-r--r--. 1 root root 0 Sep 27 11:43 file15
```

```
-rw-r--r--. 1 root root 0 Sep 27 11:43 file16
-rw-r--r--. 1 root root 0 Sep 27 11:43 file17
-rw-r--r--. 1 root root 0 Sep 27 11:43 file18
-rw-r--r--. 1 root root 0 Sep 27 11:43 file19
-rw-r--r--. 1 root root 0 Sep 27 11:43 file2
-rw-r--r--. 1 root root 0 Sep 27 11:43 file20
-rw-r--r--. 1 root root 0 Sep 27 11:43 file3
-rw-r--r--. 1 root root 0 Sep 27 11:43 file4
-rw-r--r--. 1 root root 0 Sep 27 11:43 file5
-rw-r--r--. 1 root root 0 Sep 27 11:43 file6
-rw-r--r--. 1 root root 0 Sep 27 11:43 file7
-rw-r--r--. 1 root root 0 Sep 27 11:43 file8
-rw-r--r--. 1 root root 0 Sep 27 11:43 file9
```

To synchronise files from **/test/dirA** to the **/test/dirB** directory, use the **-r** option of the rsync command :

```
[root@redhat9 ~]# rsync -r /test/dirA/ /test/dirB
[root@redhat9 ~]# ls -l /test/dirB/
total 0
-rw-r--r--. 1 root root 0 Sep 27 11:45 file1
-rw-r--r--. 1 root root 0 Sep 27 11:45 file10
-rw-r--r--. 1 root root 0 Sep 27 11:45 file11
-rw-r--r--. 1 root root 0 Sep 27 11:45 file12
-rw-r--r--. 1 root root 0 Sep 27 11:45 file13
-rw-r--r--. 1 root root 0 Sep 27 11:45 file14
-rw-r--r--. 1 root root 0 Sep 27 11:45 file15
-rw-r--r--. 1 root root 0 Sep 27 11:45 file16
-rw-r--r--. 1 root root 0 Sep 27 11:45 file17
-rw-r--r--. 1 root root 0 Sep 27 11:45 file18
-rw-r--r--. 1 root root 0 Sep 27 11:45 file19
-rw-r--r--. 1 root root 0 Sep 27 11:45 file2
-rw-r--r--. 1 root root 0 Sep 27 11:45 file20
-rw-r--r--. 1 root root 0 Sep 27 11:45 file3
-rw-r--r--. 1 root root 0 Sep 27 11:45 file4
```

```
-rw-r--r--. 1 root root 0 Sep 27 11:45 file5
-rw-r--r--. 1 root root 0 Sep 27 11:45 file6
-rw-r--r--. 1 root root 0 Sep 27 11:45 file7
-rw-r--r--. 1 root root 0 Sep 27 11:45 file8
-rw-r--r--. 1 root root 0 Sep 27 11:45 file9
```

Important - Note that the timestamp of the synchronized files has not been preserved.

Delete the files in **/test/dirB** :

```
[root@redhat9 ~]# rm -rf /test/dirB/*
[root@redhat9 ~]# ls -l /test/dirB/
total 0
```

To synchronize files from **/test/dirA** to the **/test/dirB** directory, use the **-a** option of the rsync command :

```
[root@redhat9 ~]# rsync -a /test/dirA/ /test/dirB
[root@redhat9 ~]# ls -l /test/dirB/
total 0
-rw-r--r--. 1 root root 0 Sep 27 11:43 file1
-rw-r--r--. 1 root root 0 Sep 27 11:43 file10
-rw-r--r--. 1 root root 0 Sep 27 11:43 file11
-rw-r--r--. 1 root root 0 Sep 27 11:43 file12
-rw-r--r--. 1 root root 0 Sep 27 11:43 file13
-rw-r--r--. 1 root root 0 Sep 27 11:43 file14
-rw-r--r--. 1 root root 0 Sep 27 11:43 file15
-rw-r--r--. 1 root root 0 Sep 27 11:43 file16
-rw-r--r--. 1 root root 0 Sep 27 11:43 file17
-rw-r--r--. 1 root root 0 Sep 27 11:43 file18
-rw-r--r--. 1 root root 0 Sep 27 11:43 file19
-rw-r--r--. 1 root root 0 Sep 27 11:43 file2
```

```
-rw-r--r--. 1 root root 0 Sep 27 11:43 file20
-rw-r--r--. 1 root root 0 Sep 27 11:43 file3
-rw-r--r--. 1 root root 0 Sep 27 11:43 file4
-rw-r--r--. 1 root root 0 Sep 27 11:43 file5
-rw-r--r--. 1 root root 0 Sep 27 11:43 file6
-rw-r--r--. 1 root root 0 Sep 27 11:43 file7
-rw-r--r--. 1 root root 0 Sep 27 11:43 file8
-rw-r--r--. 1 root root 0 Sep 27 11:43 file9
```

Important - Note that not only does the **-a** option of the **rsync** command synchronize files in a recursive manner, as does the **-r** option, but it also preserves special files, symbolic links, permissions, owners, groups as well as file modification dates.

Again, delete the files in the **/test/dirB** directory:

```
[root@redhat9 ~]# rm -rf /test/dirB/*
[root@redhat9 ~]# ls -l /test/dirB/
total 0
```

Execute the following command and check the result:

```
[root@redhat9 ~]# rsync -a /test/dirA /test/dirB
[root@redhat9 ~]# ls -l /test/dirB/
total 4
drwxr-xr-x. 2 root root 4096 Sep 27 11:43 dirA
[root@redhat9 ~]# ls -l /test/dirB/dirA/
total 0
-rw-r--r--. 1 root root 0 Sep 27 11:43 file1
-rw-r--r--. 1 root root 0 Sep 27 11:43 file10
-rw-r--r--. 1 root root 0 Sep 27 11:43 file11
-rw-r--r--. 1 root root 0 Sep 27 11:43 file12
```

```
-rw-r--r--. 1 root root 0 Sep 27 11:43 file13
-rw-r--r--. 1 root root 0 Sep 27 11:43 file14
-rw-r--r--. 1 root root 0 Sep 27 11:43 file15
-rw-r--r--. 1 root root 0 Sep 27 11:43 file16
-rw-r--r--. 1 root root 0 Sep 27 11:43 file17
-rw-r--r--. 1 root root 0 Sep 27 11:43 file18
-rw-r--r--. 1 root root 0 Sep 27 11:43 file19
-rw-r--r--. 1 root root 0 Sep 27 11:43 file2
-rw-r--r--. 1 root root 0 Sep 27 11:43 file20
-rw-r--r--. 1 root root 0 Sep 27 11:43 file3
-rw-r--r--. 1 root root 0 Sep 27 11:43 file4
-rw-r--r--. 1 root root 0 Sep 27 11:43 file5
-rw-r--r--. 1 root root 0 Sep 27 11:43 file6
-rw-r--r--. 1 root root 0 Sep 27 11:43 file7
-rw-r--r--. 1 root root 0 Sep 27 11:43 file8
-rw-r--r--. 1 root root 0 Sep 27 11:43 file9
```

Important - Note that in this case, the **/** character is missing after **dirA** in the **rsync -a /test/dirA /test/dirB** command. The result is the synchronisation of the **directory /test/dirA** to **/test/dirB**.

In order to test a synchronisation, rsync allows the use of dry runs by using the **-n** and **-r** switches.

Again, delete the files in the **/test/dirB** directory:

```
[root@redhat9 ~]# rm -rf /test/dirB/*
[root@redhat9 ~]# ls -l /test/dirB/
total 0
```

Now execute the following command:

```
[root@redhat9 ~]# rsync -anv /test/dirA/ /test/dirB
sending incremental file list
./
file1
file10
file11
file12
file13
file14
file15
file16
file17
file18
file19
file2
file20
file3
file4
file5
file6
file7
file8
file9

sent 387 bytes received 79 bytes 932.00 bytes/sec
total size is 0 speedup is 0.00 (DRY RUN)

[root@redhat9 ~]# ls -l /test/dirB/
total 0
```

Important - Note that in this case, the result of the synchronisation is to send the **contents** of the **/test/dirA** directory to **/test/dirB**.

Now execute the following command:

```
[root@redhat9 ~]# rsync -anv /test/dirA /test/dirB
sending incremental file list
dirA/
dirA/file1
dirA/file10
dirA/file11
dirA/file12
dirA/file13
dirA/file14
dirA/file15
dirA/file16
dirA/file17
dirA/file18
dirA/file19
dirA/file2
dirA/file20
dirA/file3
dirA/file4
dirA/file5
dirA/file6
dirA/file7
dirA/file8
dirA/file9

sent 397 bytes received 80 bytes 954.00 bytes/sec
total size is 0 speedup is 0.00 (DRY RUN)

[root@redhat9 ~]# ls -l /test/dirB/
total 0
```

Important - Note that in this case, the result of the synchronisation is to send the

/test/dirA directory to **/test/dirB**.

Command Line Switches

The command line switches for the rsync command are:

```
[root@redhat9 ~]# rsync --help
rsync version 3.2.3 protocol version 31
Copyright (C) 1996-2020 by Andrew Tridgell, Wayne Davison, and others.
Web site: https://rsync.samba.org/
Capabilities:
    64-bit files, 64-bit inums, 64-bit timestamps, 64-bit long ints,
    socketpairs, hardlinks, hardlink-specials, symlinks, IPv6, atimes,
    batchfiles, inplace, append, ACLs, xattrs, optional protect-args, iconv,
    symtimes, prealloc, stop-at, no crtimes
Optimisations:
    SIMD, asm, openssl-crypto
Checksum list:
    md5 md4 none
Compress list:
    zstd lz4 zlibx zlib none

rsync comes with ABSOLUTELY NO WARRANTY.  This is free software, and you
are welcome to redistribute it under certain conditions.  See the GNU
General Public Licence for details.

rsync is a file transfer program capable of efficient remote update
via a fast differencing algorithm.

Usage: rsync [OPTION]... SRC [SRC]... DEST
  or rsync [OPTION]... SRC [SRC]... [USER@]HOST:DEST
```

```
  or rsync [OPTION]... SRC [SRC]... [USER@]HOST::DEST
  or rsync [OPTION]... SRC [SRC]... rsync://[USER@]HOST[:PORT]/DEST
  or rsync [OPTION]... [USER@]HOST:SRC [DEST]
  or rsync [OPTION]... [USER@]HOST::SRC [DEST]
  or rsync [OPTION]... rsync://[USER@]HOST[:PORT]/SRC [DEST]
The ‘:’ usages connect via remote shell, while ‘::’ & ‘rsync://’ usages connect
to an rsync daemon, and require SRC or DEST to start with a module name.

Options
--verbose, -v increase verbosity
--info=FLAGS fine-grained informational verbosity
--debug=FLAGS fine-grained debug verbosity
--stderr=e|a|c change stderr output mode (default: errors)
--quiet, -q suppress non-error messages
--no-motd suppress daemon-mode MOTD
--checksum, -c skip based on checksum, not mod-time & size
--archive, -a archive mode; equals -rlptgoD (no -H,-A,-X)
--no-OPTION turn off an implied OPTION (e.g. --no-D)
--recursive, -r recurse into directories
--relative, -R use relative path names
--no-implied-dirs don't send implied dirs with --relative
--backup, -b make backups (see --suffix & --backup-dir)
--backup-dir=DIR make backups into hierarchy based in DIR
--suffix=SUFFIX backup suffix (default ~ w/o --backup-dir)
--update, -u skip files that are newer on the receiver
--inplace update destination files in-place
--append append data onto shorter files
--append-verify --append w/old data in file checksum
--dirs, -d transfer directories without recursing
--mkpath create the destination's path component
--links, -l copy symlinks as symlinks
--copy-links, -L transform symlink into referent file/dir
--copy-unsafe-links only ‘unsafe’ symlinks are transformed
--safe-links ignore symlinks that point outside the tree
```

```
--munge-links munge symlinks to make them safe & unusable
--copy-dirlinks, -k transform symlink to dir into referent dir
--keep-dirlinks, -K treat symlinked dir on receiver as dir
--hard-links, -H preserve hard links
--perms, -p preserve permissions
--executability, -E preserve executability
--chmod=CHMOD affect file and/or directory permissions
--acls, -A preserve ACLs (implies --perms)
--xattrs, -X preserve extended attributes
--owner, -o preserve owner (super-user only)
--group, -g preserve group
--devices preserve device files (super-user only)
--copy-devices copy device contents as regular file
--specials preserve special files
-D same as --devices --specials
--times, -t preserve modification times
--atimes, -U preserve access (use) times
--open-noatime avoid changing the atime on opened files
--crtimes, -N preserve create times (newness)
--omit-dir-times, -O omit directories from --times
--omit-link-times, -J omit symlinks from --times
--super receiver attempts super-user activities
--fake-super store/recover privileged attrs using xattrs
--sparse, -S turn sequences of nulls into sparse blocks
--preallocate allocate dest files before writing them
--write-devices write to devices as files (implies --inplace)
--dry-run, -n perform a trial run with no changes made
--whole-file, -W copy files whole (w/o delta-xfer algorithm)
--checksum-choice=STR choose the checksum algorithm (aka --cc)
--one-file-system, -x don't cross filesystem boundaries
--block-size=SIZE, -B force a fixed checksum block-size
--rsh=COMMAND, -e specify the remote shell to use
--rsync-path=PROGRAM specify the rsync to run on remote machine
--existing skip creating new files on receiver
```

```
--ignore-existing skip updating files that exist on receiver
--remove-source-files sender removes synchronized files (non-dir)
--del an alias for --delete-during
--delete delete extraneous files from dest dirs
--delete-before receiver deletes before xfer, not during
--delete-during receiver deletes during the transfer
--delete-delay find deletions during, delete after
--delete-after receiver deletes after transfer, not during
--delete-excluded also delete excluded files from dest dirs
--ignore-missing-args ignore missing source args without error
--delete-missing-args delete missing source args from destination
--ignore-errors delete even if there are I/O errors
--force force deletion of dirs even if not empty
--max-delete=NUM don't delete more than NUM files
--max-size=SIZE don't transfer any file larger than SIZE
--min-size=SIZE don't transfer any file smaller than SIZE
--max-alloc=SIZE change a limit relating to memory alloc
--partial keep partially transferred files
--partial-dir=DIR put a partially transferred file into DIR
--delay-updates put all updated files into place at end
--prune-empty-dirs, -m prune empty directory chains from file-list
--numeric-ids don't map uid/gid values by user/group name
--usermap=STRING custom username mapping
--groupmap=STRING custom groupname mapping
--chown=USER:GROUP simple username/groupname mapping
--timeout=SECONDS set I/O timeout in seconds
--contimeout=SECONDS set daemon connection timeout in seconds
--ignore-times, -I don't skip files that match size and time
--size-only skip files that match in size
--modify-window=NUM, -@ set the accuracy for mod-time comparisons
--temp-dir=DIR, -T create temporary files in directory DIR
--fuzzy, -y find similar file for basis if no dest file
--compare-dest=DIR also compare destination files relative to DIR
--copy-dest=DIR ... and include copies of unchanged files
```

```
--link-dest=DIR hardlink to files in DIR when unchanged
--compress, -z compress file data during the transfer
--compress-choice=STR choose the compression algorithm (aka --zc)
--compress-level=NUM explicitly set compression level (aka --zl)
--skip-compress=LIST skip compressing files with suffix in LIST
--cvs-exclude, -C auto-ignore files in the same way CVS does
--filter=RULE, -f add a file-filtering RULE
-F same as --filter='dir-merge /.rsync-filter'
                         repeated: --filter='- .rsync-filter'
--exclude=PATTERN exclude files matching PATTERN
--exclude-from=FILE read exclude patterns from FILE
--include=PATTERN don't exclude files matching PATTERN
--include-from=FILE read include patterns from FILE
--files-from=FILE read list of source-file names from FILE
--from0, -0 all *-from/filter files are delimited by 0s
--protect-args, -s no space-splitting; wildcard chars only
--copy-as=USER[:GROUP] specify user & optional group for the copy
--address=ADDRESS bind address for outgoing socket to daemon
--port=PORT specify double-colon alternate port number
--sockopts=OPTIONS specify custom TCP options
--blocking-io use blocking I/O for the remote shell
--outbuf=N|L|B set out buffering to None, Line, or Block
--stats give some file-transfer stats
--8-bit-output, -8 leave high-bit chars unescaped in output
--human-readable, -h output numbers in a human-readable format
--progress show progress during transfer
-P same as --partial --progress
--itemize-changes, -i output a change-summary for all updates
--remote-option=OPT, -M send OPTION to the remote side only
--out-format=FORMAT output updates using the specified FORMAT
--log-file=FILE log what we're doing to the specified FILE
--log-file-format=FMT log updates using the specified FMT
--password-file=FILE read daemon-access password from FILE
--early-input=FILE use FILE for daemon's early exec input
```

```
--list-only list the files instead of copying them
--bwlimit=RATE limit socket I/O bandwidth
--stop-after=MINS Stop rsync after MINS minutes have elapsed
--stop-at=y-m-dTh:m Stop rsync at the specified point in time
--write-batch=FILE write a batched update to FILE
--only-write-batch=FILE like --write-batch but w/o updating dest
--read-batch=FILE read a batched update from FILE
--protocol=NUM force an older protocol version to be used
--iconv=CONVERT_SPEC request charset conversion of filenames
--checksum-seed=NUM set block/file checksum seed (advanced)
--ipv4, -4 prefer IPv4
--ipv6, -6 prefer IPv6
--version, -V print the version + other info and exit
--help, -h (*) show this help (* -h is help only on its own)

Use ‘rsync --daemon --help’ to see the daemon-mode command-line options.
Please see the rsync(1) and rsyncd.conf(5) man pages for full documentation.
See https://rsync.samba.org/ for updates, bug reports, and answers
```

Compression

The gzip command

Overview

The **gzip** command is a compression utility under GNU/Linux. The **gunzip** command is a GNU/Linux decompression utility.

LAB #6 - Working with the gzip command

Use **gzip** to compress your tar file:

```
[root@redhat9 ~]# gzip /tmp/test.tar
```

Note the size of the **test.tar.gz** file:

```
[root@redhat9 ~]# ls -l /tmp/test.tar.gz
-rw-r--r--. 1 root root 222 Sep 27 07:57 /tmp/test.tar.gz
```

Important - Note that the compressed file was created in the same directory as the source file and that the source file has disappeared.

Decompress the test.tar.gz file:

```
[root@redhat9 ~]# gunzip /tmp/test.tar.gz
```

Command Line Switches

The **gzip** Command Line Switches are:

```
[root@redhat9 ~]# gzip --help
Usage: gzip [OPTION]... [FILE]...
Compress or uncompress FILEs (by default, compress FILES in-place).

Mandatory arguments to long options are mandatory for short options too.

  -c, --stdout write on standard output, keep original files unchanged
  -d, --decompress decompress
  -f, --force force overwrite of output file and compress links
  -h, --help give this help
```

```
  -k, --keep keep (don't delete) input files
  -l, --list list compressed file contents
  -L, --license display software license
  -n, --no-name do not save or restore the original name and timestamp
  -N, --name save or restore the original name and timestamp
  -q, --quiet suppress all warnings
  -r, --recursive operate recursively on directories
      --rsyncable make rsync-friendly archive
  -S, --suffix=SUF use suffix SUF on compressed files
      --synchronous synchronous output (safer if system crashes, but slower)
  -t, --test test compressed file integrity
  -v, --verbose verbose mode
  -V, --version display version number
  -1, --fast compress faster
  -9, --best compress better

With no FILE, or when FILE is -, read standard input.

Report bugs to <bug-gzip@gnu.org>.
```

The command line switches for the **gunzip** command are:

```
[root@redhat9 ~]# gunzip --help
Usage: /usr/bin/gunzip [OPTION]... [FILE]...
Uncompress FILEs (by default, in-place).

Mandatory arguments to long options are mandatory for short options too.

  -c, --stdout write on standard output, keep original files unchanged
  -f, --force force overwrite of output file and compress links
  -k, --keep keep (don't delete) input files
  -l, --list list compressed file contents
  -n, --no-name do not save or restore the original name and timestamp
  -N, --name save or restore the original name and timestamp
```

```
  -q, --quiet suppress all warnings
  -r, --recursive operate recursively on directories
  -S, --suffix=SUF use suffix SUF on compressed files
      --synchronous synchronous output (safer if system crashes, but slower)
  -t, --test test compressed file integrity
  -v, --verbose verbose mode
      --help display this help and exit
      --version display version information and exit

With no FILE, or when FILE is -, read standard input.

Report bugs to <bug-gzip@gnu.org>.
```

The bzip2 command

Overview

The **bzip2** command is a compression utility under GNU/Linux. The **bunzip2** command is a decompression utility under GNU/Linux.

LAB #7 - Working with the bzip2 command

Use **bzip2** to compress your tar file:

```
[root@redhat9 ~]# bzip2 /tmp/test.tar
```

Note the size of the **tar.bz2** file:

```
[root@redhat9 ~]# ls -l /tmp | grep test.tar.bz
-rw-r--r--. 1 root root 207 Sep 27 07:57 test.tar.bz2
```

Important - Note that the compressed file was created in the same directory as the source file and the source file has disappeared.

Decompress the tar.bz2 file:

```
[root@redhat9 ~]# bunzip2 /tmp/test.tar.bz2
```

Command Line Switches

The command line switches for the **bzip2** command are:

```
[root@redhat9 ~]# bzip2 --help
bzip2, a block-sorting file compressor.  Version 1.0.8, 13-Jul-2019.

   usage: bzip2 [flags and input files in any order]

   -h --help print this message
   -d --decompress force decompression
   -z --compress force compression
   -k --keep keep (don't delete) input files
   -f --force overwrite existing output files
   -t --test test compressed file integrity
   -c --stdout output to standard out
   -q --quiet suppress noncritical error messages
   -v --verbose be verbose (a 2nd -v gives more)
   -L --license display software version & license
   -V --version display software version & license
   -s --small use less memory (at most 2500k)
   -1 .. -9 set block size to 100k .. 900k
   --fast alias for -1
```

```
   --best alias for -9

   If invoked as `bzip2', default action is to compress.
             as `bunzip2', default action is to decompress.
             as `bzcat', default action is to decompress to stdout.

   If no file names are given, bzip2 compresses or decompresses
   from standard input to standard output.  You can combine
   short flags, so `-v -4' means the same as -v4 or -4v, &c.
```

The command line switches for the **bunzip2** command are:

```
[root@redhat9 ~]# bunzip2 --help
bzip2, a block-sorting file compressor.  Version 1.0.8, 13-Jul-2019.

   usage: bunzip2 [flags and input files in any order]

   -h --help print this message
   -d --decompress force decompression
   -z --compress force compression
   -k --keep keep (don't delete) input files
   -f --force overwrite existing output files
   -t --test test compressed file integrity
   -c --stdout output to standard out
   -q --quiet suppress noncritical error messages
   -v --verbose be verbose (a 2nd -v gives more)
   -L --license display software version & license
   -V --version display software version & license
   -s --small use less memory (at most 2500k)
   -1 .. -9 set block size to 100k .. 900k
   --fast alias for -1
   --best alias for -9

   If invoked as `bzip2', default action is to compress.
```

```
              as `bunzip2', default action is to decompress.
              as `bzcat', default action is to decompress to stdout.

   If no file names are given, bzip2 compresses or decompresses
   from standard input to standard output.  You can combine
   short flags, so `-v -4' means the same as -v4 or -4v, &c.0
```

The xz command

Overview

The **xz** command is a compression utility under GNU/Linux. Other commands are:

- **unxz** - equivalent to **xz –decompress**.
- **xzcat** - equivalent to **xz –decompress –stdout**.
- **lzma** - equivalent to **xz –format=lzma**.
- **unlzma** - equivalent to **xz –format=lzma –decompress**.
- **lzcat** - equivalent to **xz –format=lzma –decompress –stdout**.

The xz command will not compress the file if :

- the file is not of standard type
- the file is a symbolic link
- the file is a physical link
- the file has a sticky bit, SUID bit or SGID bit
- the file already has an .xz or .lzma extension

The xz command will not decompress the file if :

- the file does not have an .xz or .lzma extension

LAB #8 - Working with the xz command

Use **xz** to compress your tar file :

```
[root@redhat9 ~]# xz /tmp/test.tar
```

Important - Note that the default operation of the command is identical to that of the **-z** option.

Note the presence of the **test.tar.xz** file:

```
[root@redhat9 ~]# ls -l /tmp | grep test.tar.xz
-rw-r--r--. 1 root root 236 Sep 27 07:57 test.tar.xz
```

Important - Note that the compressed file was created in the same directory as the source file and that the source file has disappeared. The source file can be maintained if the **–keep** option is specified. If the test.tar.xz file had already existed, the command would have failed with an error message. The file extension is **.xz**, however the command can also handle the **.lzma** extension.

Decompress the test.tar.xz file:

```
[root@redhat9 ~]# xz -d /tmp/test.tar.xz
[root@redhat9 ~]# ls -l /tmp | grep test
-rw-r--r--. 1 root root 160 Sep 25 16:05 greptest
-rw-r--r--. 1 root root 49 Sep 25 16:05 greptest1
-rw-r--r--. 1 root root 687556 Sep 25 16:08 sedtest
-rw-r--r--. 1 root root 512 Sep 27 08:08 test.cpio
```

```
-rw-r--r--. 1 root root 10240 Sep 27 07:57 test.tar
```

Command Line Switches

The command line switches for the **xz** command are:

```
[root@redhat9 ~]# xz --help
Usage: xz [OPTION]... [FILE]...
Compress or decompress FILEs in the .xz format.

  -z, --compress force compression
  -d, --decompress force decompression
  -t, --test test compressed file integrity
  -l, --list list information about .xz files
  -k, --keep keep (don't delete) input files
  -f, --force force overwrite of output file and (de)compress links
  -c, --stdout write to standard output and don't delete input files
  -0 ... -9 compression preset; default is 6; take compressor *and*
                      decompressor memory usage into account before using 7-9!
  -e, --extreme try to improve compression ratio by using more CPU time;
                      does not affect decompressor memory requirements
  -T, --threads=NUM use at most NUM threads; the default is 1; set to 0
                      to use as many threads as there are processor cores
  -q, --quiet suppress warnings; specify twice to suppress errors too
  -v, --verbose be verbose; specify twice for even more verbose
  -h, --help display this short help and exit
  -H, --long-help display the long help (lists also the advanced options)
  -V, --version display the version number and exit

With no FILE, or when FILE is -, read standard input.

Report bugs to <lasse.collin@tukaani.org> (in English or Finnish).
```

```
XZ Utils home page: <https://tukaani.org/xz/>
```

Other Utilities

There are other utilities for compression, each producing a file with a specific extension:

Tool	Extension	Compression command	Decompression command
compress	.Z	compress	uncompress
rar	.rar	rar	unrar
zip	.zip	zip	unzip
