42 -Finding files

find : Short examples: Searches recursively for files or directories through a directory tree.

Syntax:
`find startdirectory [search_criteria_options] [-exec command \;]`

Examples:

`find Path -name Datei` (Name should always be inside " " when * is used)
`find Path -regex "pattern"` Uses a regular expression to find the filename
`find Path -type f` (f= regular file, d = Directory, l = Symlink)
`find Path -user Benutzername`
`find Path -maxdepth Tiefe`
`find Path -xdev` (doesn't search inside mounted filesystems)
`find Path -cmin -Minuten`
`find Path -follow` (follow symlinks)
`find Path -perm +u+s` Finds all files having the SUID turned on.

More Examples:

`find . type d -maxdepth 1 | sort`
Finds all directories located in the current directory and sort them

`cd /etc/ ; find . -name "*XF*"`

Finds all files recursively in `/etc` directory of which their names includes the pattern 'XF'

`find /opt/kde3 -maxdepth 2 -type f -name "*edit"`
Searches in `/opt/kde3` and in 2 subdirectories deep, any file of which their name ends-up with the word '`edit`'

`find . -follow -cmin -5`
(Search the files that of which their properties have been changed less than 5 minutes ago)

`-cmin +5` Properties of file **c**hanged more than 5 minutes ago
`-amin -6` Content of file **a**ccessed less than 6 minutes ago
`-mmin +8` Content of file **m**odified more than 8 minutes ago

`-ctime +5` Properties of file **c**hanged more than 5 days ago
`-atime -7` Content of file **a**ccessed less than 7 days ago
`-mtime -3` Content of file **m**odified less than 3 days ago

`find /etc -type f -name '*.conf' -exec grep -H "hosts" {} \;`
Executes the `grep` on each found file. Each found line will be shown accompanied with the name of the file where it was found.

`find /etc -type f -name '*.conf' -ok grep -H "hosts" {} \;`
Same actions a as above except that `-ok` option asks `find` to prompt for confirmation(with **y** or **n**) of the command to do before executing it.

- **locate** : Locate files in the whole system based using a database of filenames.

Syntax:

`locate` *`filename`* Searched in the locate database for the *`filename`*.
This database is in `/var/lib/locatedb`
It is updated via the command: `updatedb [`*`options`*`]`
The configuration file for updatedb is `/etc/updatedb.conf`

`-d path, --database=path`
Instead of searching the default file name database, search the file name databases in path, which is a colon-separated list of database file names. You can also use the environment variable `LOCATE_PATH` to set the list of database files to search.
The option overrides the environment variable if both are used.

`-e, --existing` Only print out such names that currently exist (instead of such names that existed when the database was created).
Note that this may slow down the program a lot, if there are many matches in the database.

`-i, --ignore-case`
Ignore case distinctions in both the pattern and the file names.

- **slocate** : Secure locate of file in system

Syntax:

`slocate [options] filename`

Secure Locate provides the same features as the `locate` but it will also store fille permissions and ownership so that users will not see files they do not have access to. `slocate` database is not the same as `locate` database.
It needs to be built by issuing the `slocate` command with proper options:

Database Build Options:

`-u`	Create slocate database starting at path `/`
`-U <dir>`	Create slocate database starting at path <dir>
`-e <dir1,dir2,...>`	Exclude directories from `slocate` database .
`-f <fstype1,...>`	Exclude files on specific file systems from the `slocate` database.
`-c`	Parse `/etc/updatedb.conf` when updating the `slocate` database.
`-l <level>`	Security level: `0` Turns security checks off. This will make searchsfaster. `1` Turns security checks on. This is the default.
`-o <file>` `--output=<file>`	Specifies the database to create.
`-v, --verbose`	Verbose mode. Display files when creating database.

Slocate Search Options:

`-i`	Does a case insensitive search.
`-q`	Quiet mode. Error messages are suppressed.
`-n <num>`	Limit the amount of results shown to <num>.
`--regexp=<regexp>`	
`-r <regexp>`	Search the database using a basic POSIX regular expression.
`-d <path>`	Specifies the path of databasesto search in.
`--database=<path>`	

- **whereis**: Search for a program and possibly its man pages from a predefined path.

Syntax:

`whereis` *`filename`*	Searches a predefined (hard coded) list of directories for the filename and man pages. They must be in the path predifined during compilation of `whereis` program.

- **which** : Search for the first occurence of a program through the `PATH`.

Syntax:

`which` *`filename`*	Searches the `PATH` for the first occurence of the filename. The filename can be a list of files.
`type -p` *`filename`*	Same as above `which` *`filename`*