

Downloading Files and Packages

Category	Experience Level	Author
Linux	Novice	Nick Ruffles

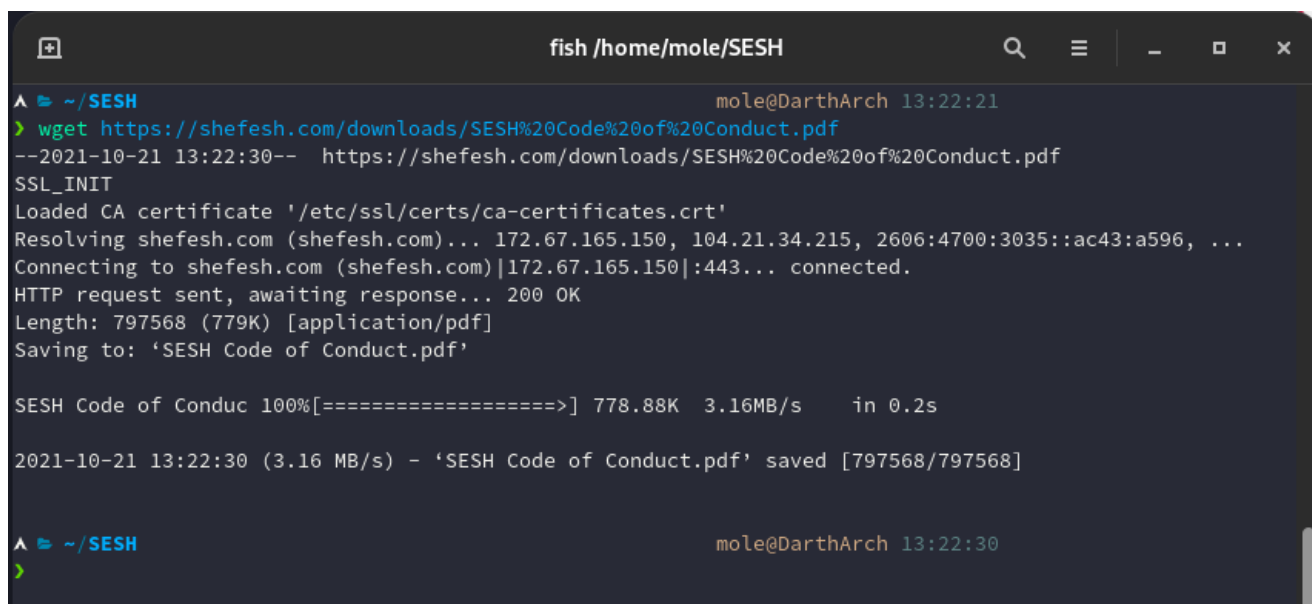
Contents

- [Downloading files](#)
- [Downloading packages](#)
- [Installing packages manually](#)
- [deb packages](#)
- [Using make](#)

Downloading files

In Linux there are two main utilities for downloading files, and both have a very similar sort of syntax. The go-to for downloading files is `wget`, it is available on (almost all) Linux distributions as well as windows.

To use `wget`, simply type `wget [url of file/page]`.



```
fish /home/mole/SESH
mole@DarthArch 13:22:21
> wget https://shefesh.com/downloads/SESH%20Code%20of%20Conduct.pdf
--2021-10-21 13:22:30-- https://shefesh.com/downloads/SESH%20Code%20of%20Conduct.pdf
SSL_INIT
Loaded CA certificate '/etc/ssl/certs/ca-certificates.crt'
Resolving shefesh.com (shefesh.com)... 172.67.165.150, 104.21.34.215, 2606:4700:3035::ac43:a596, ...
Connecting to shefesh.com (shefesh.com)|172.67.165.150|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 797568 (779K) [application/pdf]
Saving to: 'SESH Code of Conduct.pdf'

SESH Code of Conduc 100%[=====] 778.88K  3.16MB/s   in 0.2s

2021-10-21 13:22:30 (3.16 MB/s) - 'SESH Code of Conduct.pdf' saved [797568/797568]

mole@DarthArch 13:22:30
>
```

This will reach out to the website using http/s and download the file into the current working directory.

There is another command that works for downloading files called `curl`. `curl` is different to `wget` as it's default place to 'save' the file is the standard output (displayed on the terminal). To change this we can use the command `curl [url of file/page] -o /path/to/save/file`


```
fish /home/mole
mole@ ~> sudo apt update
[sudo] password for mole:
Hit:1 http://gb.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://gb.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://gb.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://gb.archive.ubuntu.com/ubuntu focal-security InRelease
Get:5 https://download.docker.com/linux/ubuntu focal InRelease [57.7 kB]
Fetched 57.7 kB in 0s (186 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
mole@ ~> sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  alsa-ucm-conf
1 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 26.9 kB of archives.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://gb.archive.ubuntu.com/ubuntu focal-updates/main amd64 alsa-ucm-conf all 1.2.2-1ubuntu0.11 [
26.9 kB]
Fetched 26.9 kB in 0s (191 kB/s)
(Reading database ... 110467 files and directories currently installed.)
Preparing to unpack ../alsa-ucm-conf_1.2.2-1ubuntu0.11_all.deb ...
Unpacking alsa-ucm-conf (1.2.2-1ubuntu0.11) over (1.2.2-1ubuntu0.10) ...
Setting up alsa-ucm-conf (1.2.2-1ubuntu0.11) ...
mole@ ~> █
```

If you want to find packages and how to install them on your distribution, you can head over to command-not-found.com, the website allows you to search for packages based on the command, for example the `ls` command is part of the `coreutils` package.

Installing packages manually

Manually installing packages means that you have to type some extra commands to install them, and you have to maintain them yourself (as the package managers generally won't update them for you).

Some applications make it relatively simple to install the program by giving you an executable or a bash script you can run to install the program. It's as simple as opening a terminal and running the script.

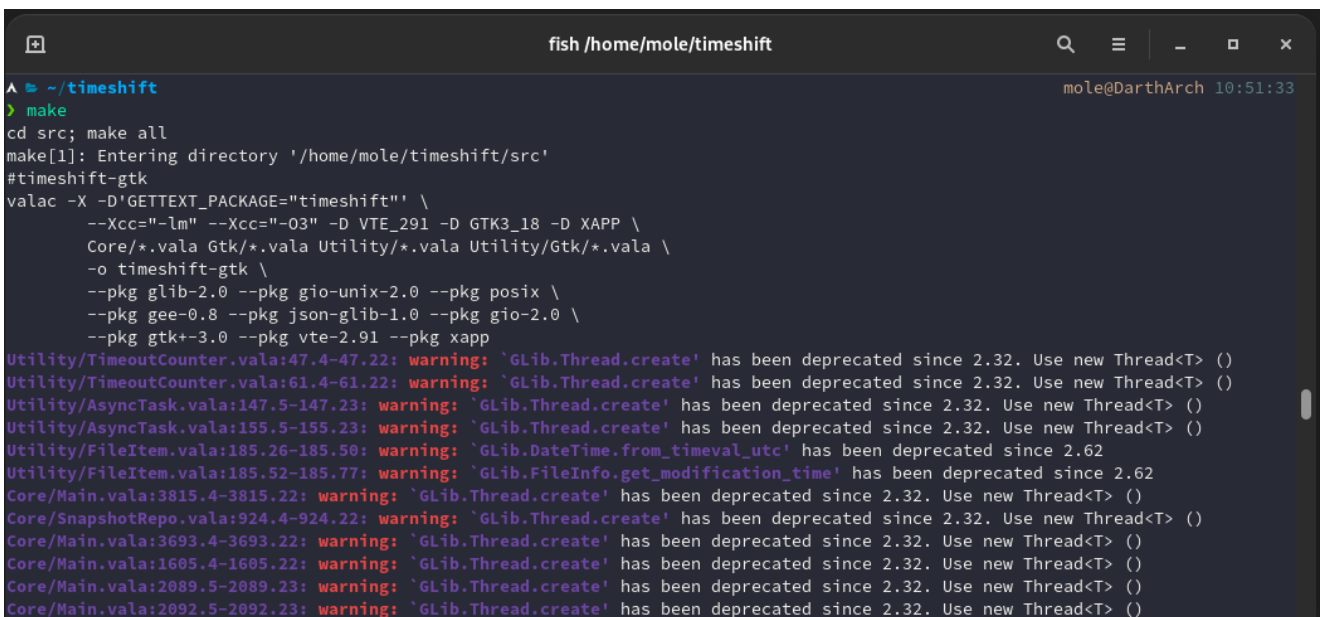
deb packages

Some packages come in the form of `.deb` files, these can be installed using the `dpkg` or `apt` package manager, with `sudo dpkg -i ./filename.deb` and `sudo apt install ./filename.deb` respectively. This will install the package using the package manager, however you may have to manually update the package.

Using `make`

Some packages may not be installable via a package manager at all, and instead you need to 'build' the application. In this scenario the developer of the application will generally give you build instructions for you to use to compile the project yourself.

Generally following the install instructions given by the developer will work well, it's normally stored in the `README.md` file in the repository. In most cases you will be using `make` to install the software. First check to see if there is a `makefile` file in the repository, as this will contain how Linux should build the software. For example, check out the makefile for [Timeshift](#) repository. First we can run `make` to build the software, then we will use `make install` to move the compiled binaries into the correct location for Linux to use (normally in the `/bin` and `/usr/bin` directories).



```
fish /home/mole/timeshift
^ | ~ /timeshift
> make
cd src; make all
make[1]: Entering directory '/home/mole/timeshift/src'
#timeshift-gtk
valac -X -D'GETTEXT_PACKAGE="timeshift"' \
  --Xcc="-lm" --Xcc="-O3" -D VTE_291 -D GTK3_18 -D XAPP \
  Core/*.vala Gtk/*.vala Utility/*.vala Utility/Gtk/*.vala \
  -o timeshift-gtk \
  --pkg glib-2.0 --pkg gio-unix-2.0 --pkg posix \
  --pkg gee-0.8 --pkg json-glib-1.0 --pkg gio-2.0 \
  --pkg gtk+-3.0 --pkg vte-2.91 --pkg xapp
Utility/TimeoutCounter.vala:47.4-47.22: warning: `GLib.Thread.create' has been deprecated since 2.32. Use new Thread<T> ()
Utility/TimeoutCounter.vala:61.4-61.22: warning: `GLib.Thread.create' has been deprecated since 2.32. Use new Thread<T> ()
Utility/AsyncTask.vala:147.5-147.23: warning: `GLib.Thread.create' has been deprecated since 2.32. Use new Thread<T> ()
Utility/AsyncTask.vala:155.5-155.23: warning: `GLib.Thread.create' has been deprecated since 2.32. Use new Thread<T> ()
Utility/FileItem.vala:185.26-185.50: warning: `GLib.DateTime.from_timeval_utc' has been deprecated since 2.62
Utility/FileItem.vala:185.52-185.77: warning: `GLib.FileInfo.get_modification_time' has been deprecated since 2.62
Core/Main.vala:3815.4-3815.22: warning: `GLib.Thread.create' has been deprecated since 2.32. Use new Thread<T> ()
Core/SnapshotRepo.vala:924.4-924.22: warning: `GLib.Thread.create' has been deprecated since 2.32. Use new Thread<T> ()
Core/Main.vala:3693.4-3693.22: warning: `GLib.Thread.create' has been deprecated since 2.32. Use new Thread<T> ()
Core/Main.vala:1605.4-1605.22: warning: `GLib.Thread.create' has been deprecated since 2.32. Use new Thread<T> ()
Core/Main.vala:2089.5-2089.23: warning: `GLib.Thread.create' has been deprecated since 2.32. Use new Thread<T> ()
Core/Main.vala:2092.5-2092.23: warning: `GLib.Thread.create' has been deprecated since 2.32. Use new Thread<T> ()
```