Tp-Note: markup enhanced granular note-taking

Save and edit your clipboard content as a note file

Jens Getreu

2024-04-07

Markup languages like *Markdown* [1] are perfectly suited for fast note-taking. Type your notes with your favourite text editor and observe the live rendered text in your web browser.

Tp-Note helps you to quickly get started writing notes with its powerful template system. If you like to keep your notes next to your files, and you care about expressive filenames, then *Tp-Note* might be the tool of your choice. As *Tp-Note* synchronizes the note's filename with its document title, you will find your notes more easily.

On Tue, 2023-12-19 at 12:58 +1100, Dev Rain wrote:

> *Found Tp-Note awhile back and it has become part of my daily workflow, and indeed part of my daily note-taking life. I wanted to extend my thanks; so thank you. dev.rain*

Tp-Note is available for Linux, Windows and MacOS. This manual illustrates its main use cases and how to get started:

1. Fast start note taking (when the lecture starts).
2. Take a note about an existing or downloaded file.
3. Bookmark and comment a hyperlink.
4. Copy and annotate a page from a book.
5. Best practice.
6. Note taking for system administrators.

If you want to customize *Tp-Note* with your own templates or if you want to use another markup language than Markdown, please consult Tp-Note's man-page for more technical details. It also explains how to change *Tp-Note*'s default text editor.

The project is hosted on Gitlab: getreu/tp-note. The project's webpage is on http://blog.getreu.net. The documentation of this project is dived into two parts:

- User manual

  Tp-Note user manual - html

  Tp-Note user manual - pdf

- Unix man-page (more technical)

 Tp-Note manual page - html

  Tp-Note manual page - pdf

1 How students take notes

A fellow student still uses paper and pen. I asked her why, and she replied with "It helps me concentrate better. My computer distracts me. I will do many other things and I wont remain concentrated on my task.".

This is certainly true. As far as I am concerned, I am not good at logistics. For me having all my documents and notes on one little machine is a blessing.
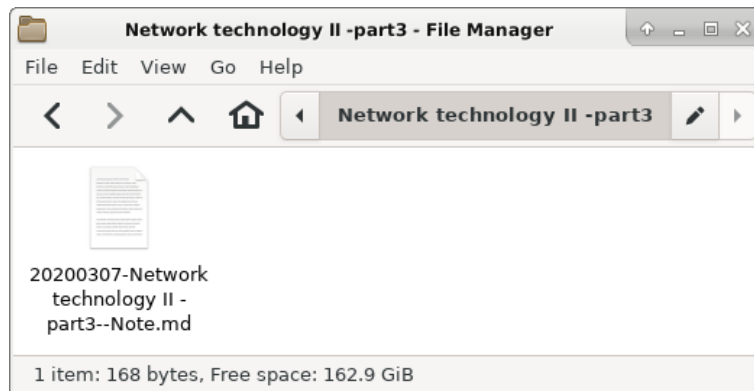
The following sections illustrate how to work with *Tp-Note* with my most common workflows.

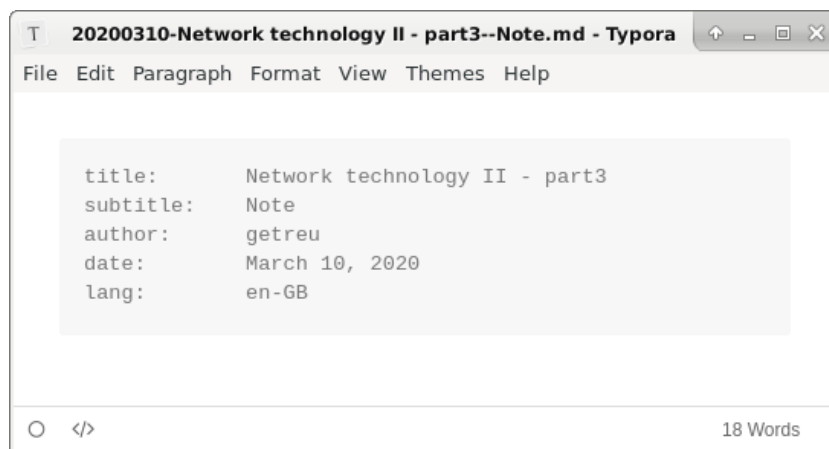## 1.1 Fast start note taking (when the lecture starts)
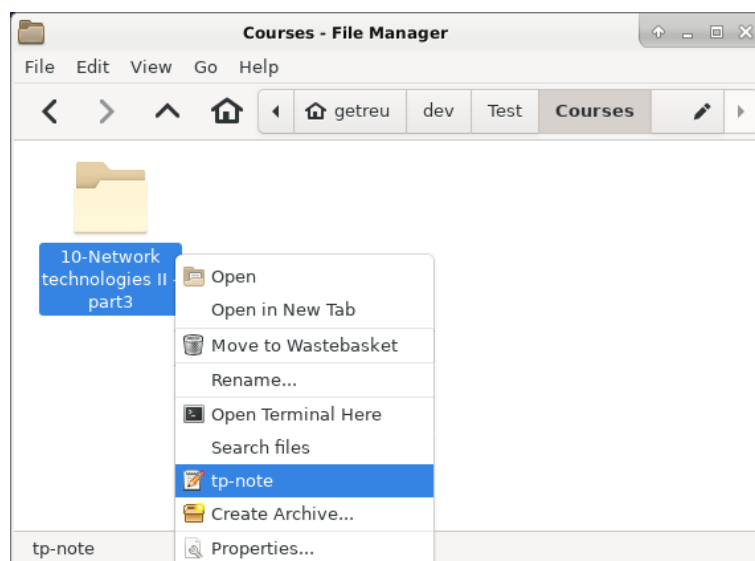
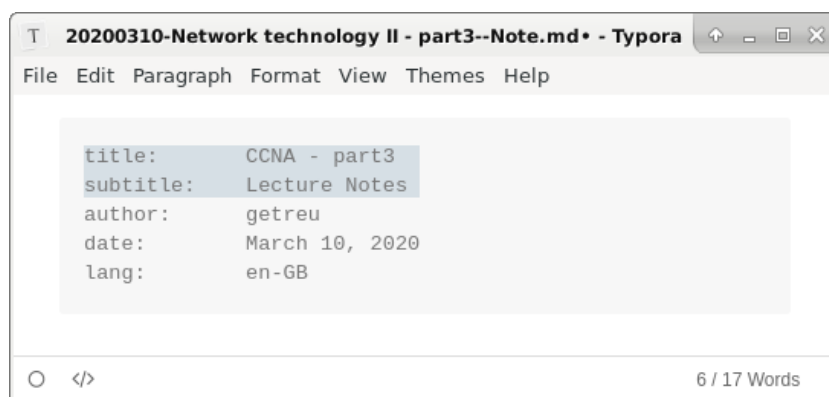The folder in which the new note will be created.

Alternatively you can open the folder where you want to create a new note and right-click on some empty white space.
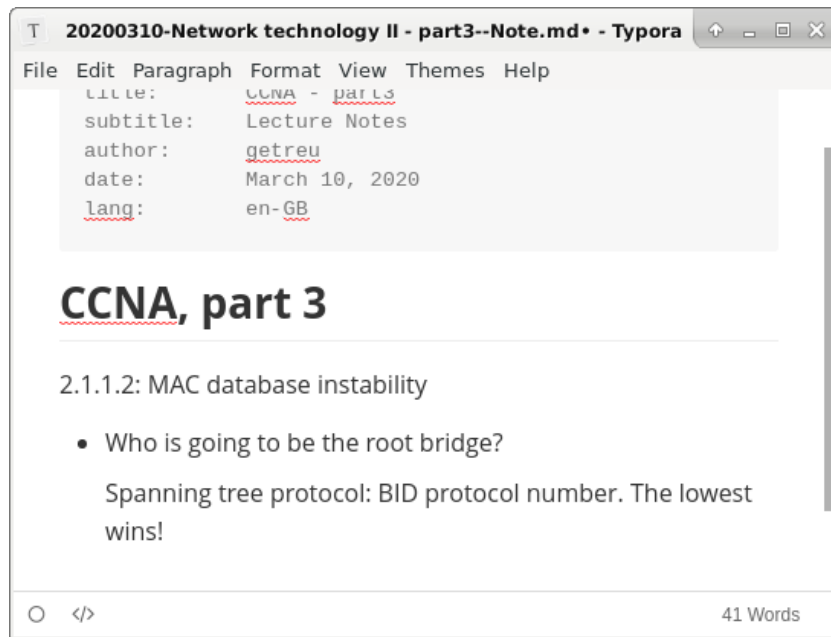

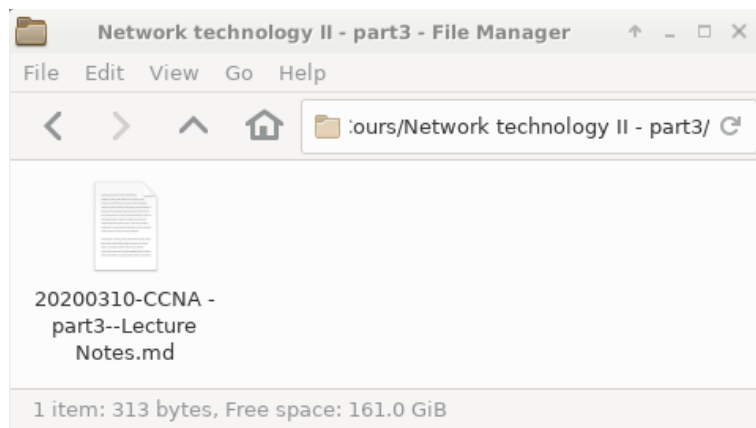The new unmodified note created by template on disk

The new unmodified note created by template

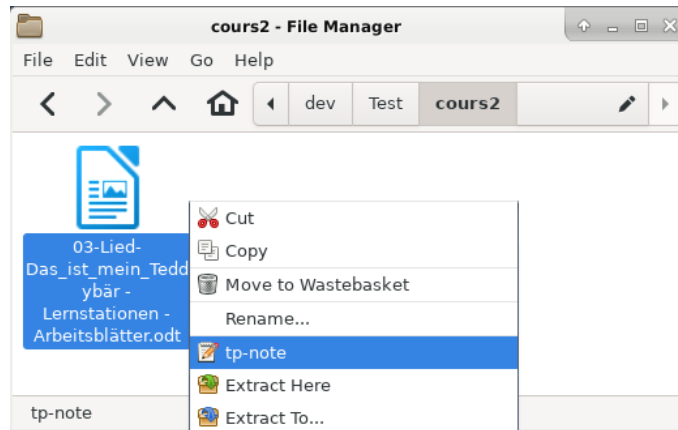Change the title

Add some text

The new note file on disk after closing the editor

> **Note**
>
> Before and after launching the editor *Tp-Note* renames the file to be in sync with the note's metadata (i.e. title and subtitle). For more details see How it works: Organize your files and notes with sort-tags.

1.2 Taking notes about a file


We want to take a note about a pdf

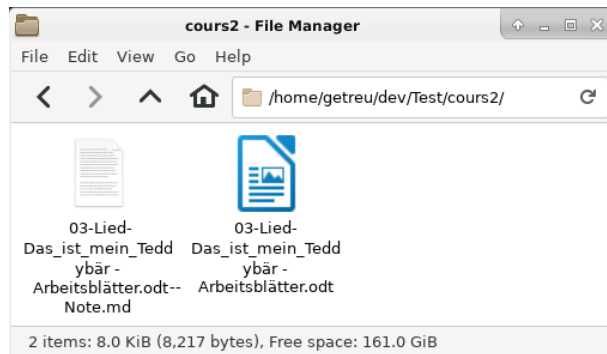The new unmodified note created automatically

The source code of the note shows the link with its target. The left-click, opens the `.odt` document.

```
---
title:      Lied-Das_ist_mein_Teddybär - Arbeitsblätter.odt
subtitle:   Note
author:     Getreu
date:       2023-09-21
lang:       en-GB
---

[03-Lied-Das_ist_mein_Teddybär - Arbeitsblätter.odt](<03-Lied-
Das_ist_mein_Teddybär - Arbeitsblätter.odt>)
```
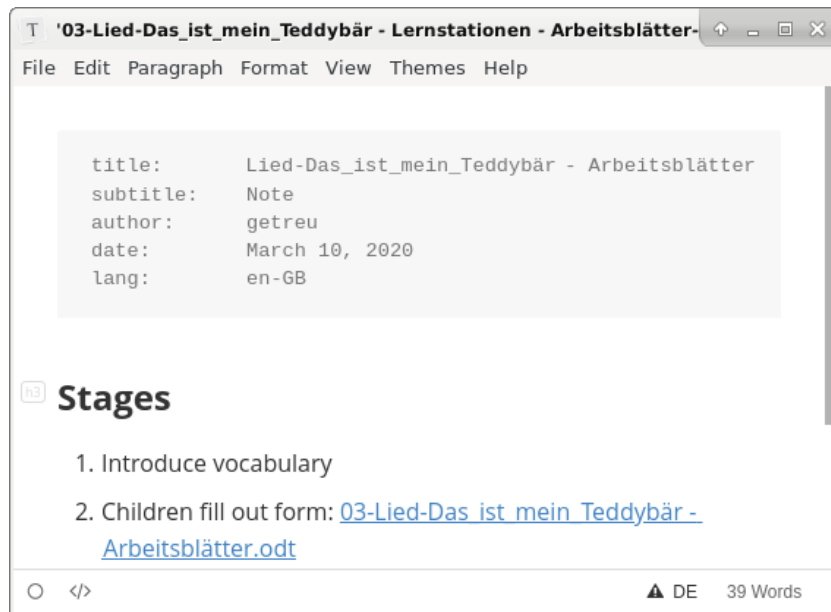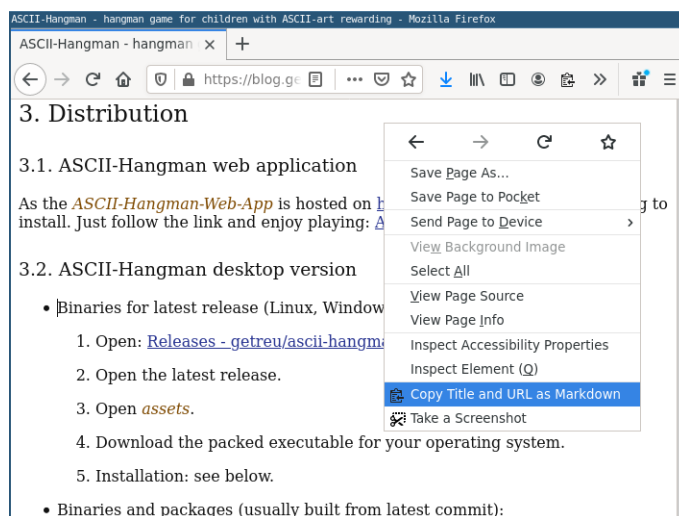
Annotate

The new note file on disk after closing the editor

1.3 Document the download location of a local file

The approach is similar to what we have seen in the previous chapter:

Copy the location of the download page as Markdown

Note: for convenience I use in this example the Firefox browser addon Copy Selection as Markdown to copy the hyperlink. If this addon is not available, you can also copy the URL directly from the search bar.
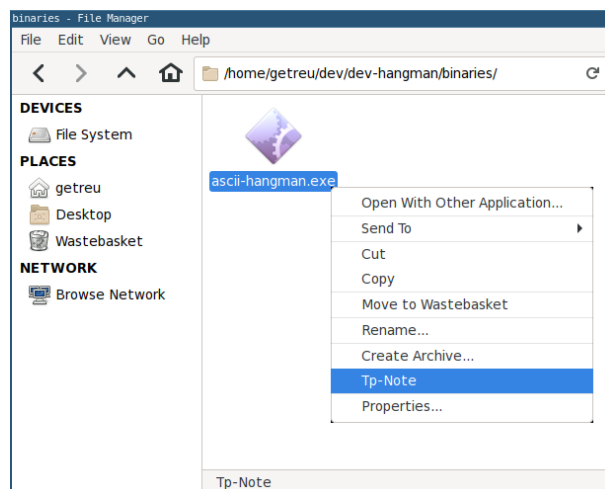
Select the file to annotate and start Tp-Note
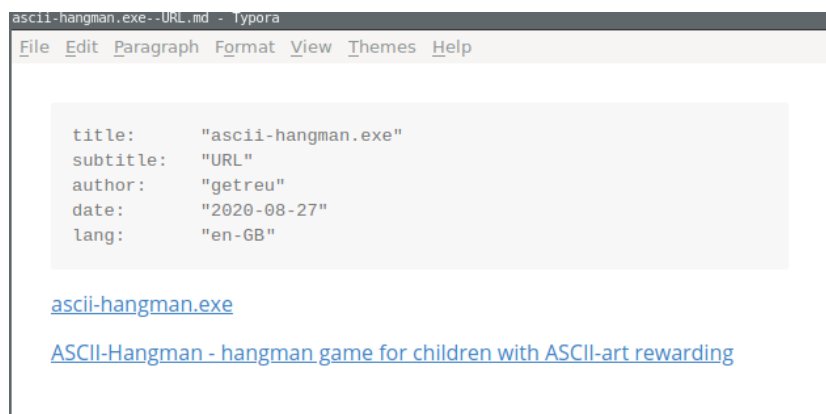
The new unmodified note created automatically

The source-code of the note shows the links with their targets.

```
---
title:      ascii-hangman.exe
subtitle:   URL
author:     Getreu
date:       2020-08-27
lang:       en-GB
---

[ascii-hangman.exe](<ascii-hangman.exe>)

[ASCII-Hangman - hangman game for children with ASCII-art
rewarding](<https://blog.getreu.net/projects/ascii-hangman/
#distribution>)
```

The new note file on disk after closing the editor

1.4 Bookmark and comment a hyperlink
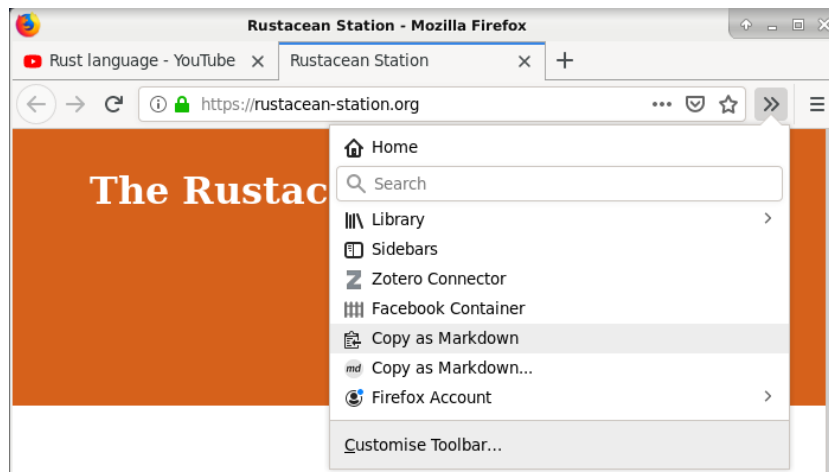
Copy a link in Markdown format

To copy a link in Markdown format a browser addon is needed. I re-
commend the addons Copy as Markdown and Copy Selection as Mark-
down available for Firefox.

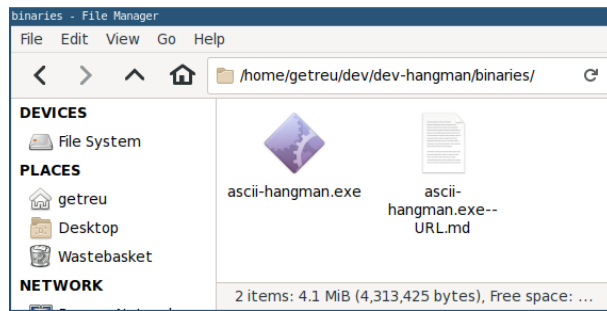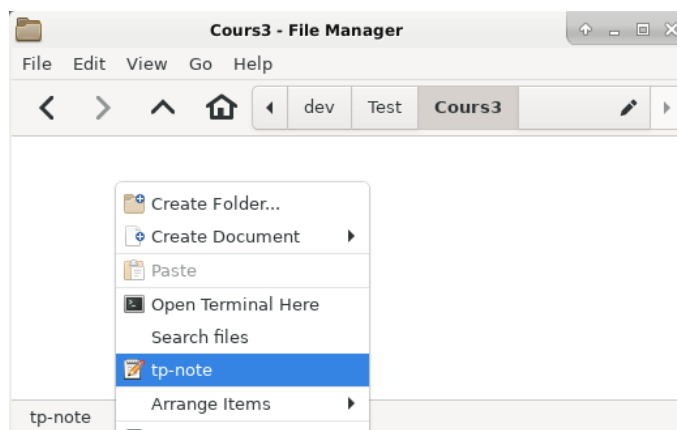Right-click on or in the new note's destination folder and
start Tp-Note
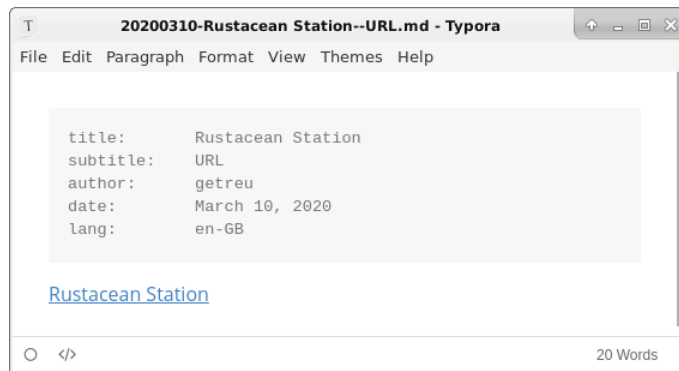
The new unmodified note created automatically

The source code of the note shows the link target:

```
---
title:      Rustacean Station
subtitle:   URL
author:     Getreu
date:       2023-09-21
lang:       en-GB
---

[Rustacean Station](<https://rustacean-station.org/>)
```

In this example we copied only one Markdown link "Rustacean Station". Furthermore, *Tp-Note* allows you also to insert a list of Markdown links in a template. For example with Copy as Markdown you could copy a link list of all open tabs. In this case, *Tp-Note* would retain only the name of the first link as document title, whereas the whole link list would appear in the body of the note.
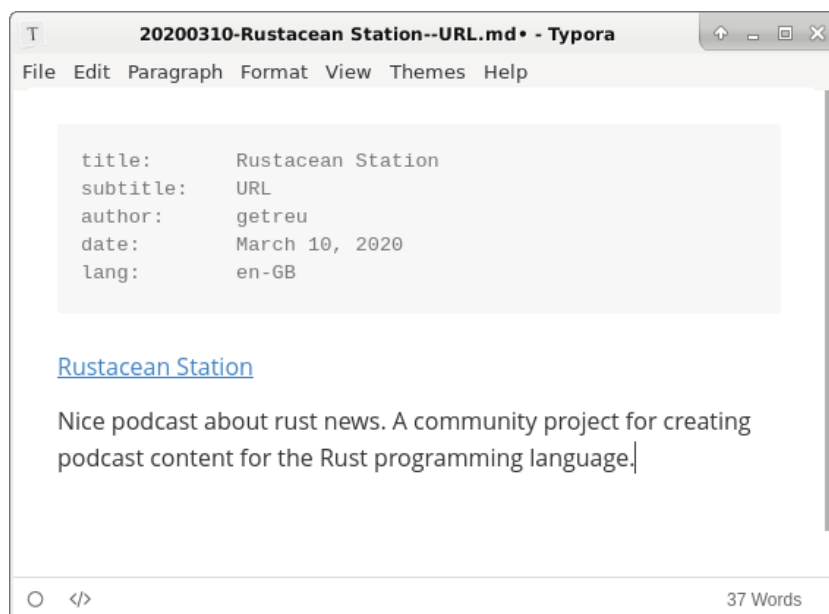
Annotate

The new note file on disk after closing the editor

1.5 Copy a page from a book
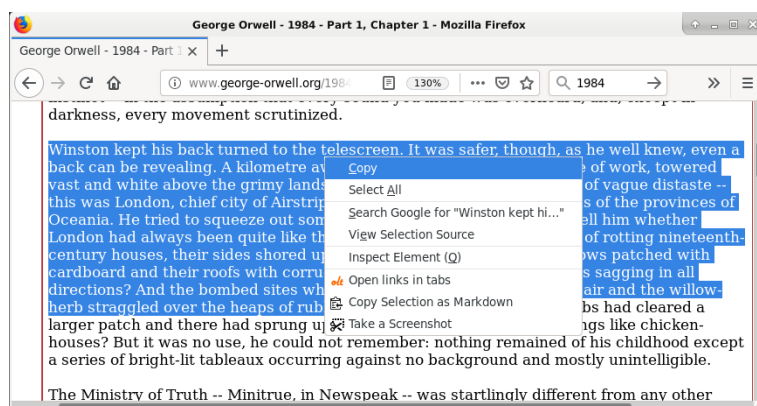
Copy some chapters
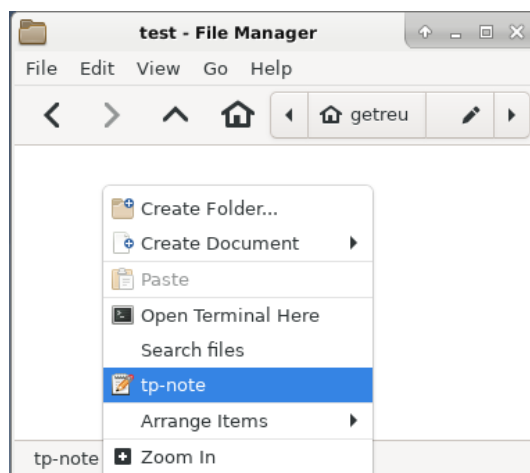
Right-click on or in the new note's destination folder and start Tp-Note
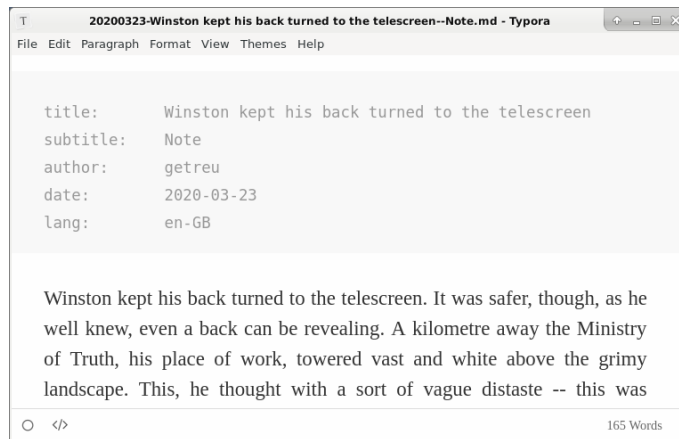
The new unmodified note created automatically

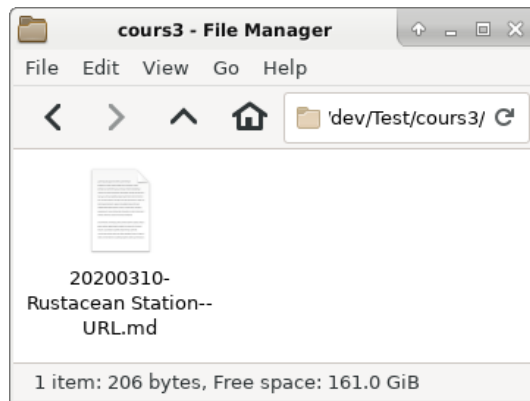The source code of the note shows the completed template:

```
---
title:      Winston kept his back turned to the telescreen
subtitle:   Note
author:     Getreu
date:       2020-03-23
lang:       en-GB
---


Winston kept his back turned to the telescreen. It was safer,
though, as he well
knew, even a back can be revealing. A kilometer away the Ministry
of Truth, his
place of work, towered vast and white above the grimy landscape...
```

In this example we copied only text. *Tp-Note* suggests the first sentence as title. This can be changed before saving as illustrated above. Here we just save and observe the file on the disk.
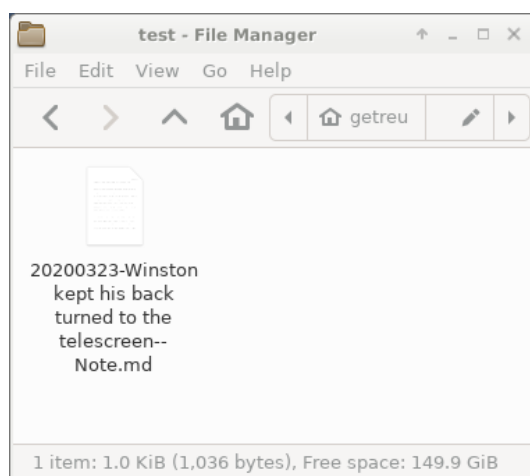
The new note file on disk after closing the editor

1.6 Best practice

Tp-Note's greatest advantage is it's flexibility. It easily integrates with your workflow. As people work differently, there is no best usage either. Nevertheless, after having used *Tp-Note* for some months now, here my personal preferences and configuration:

- Tp-Note
- Addon for Firefox: Copy Selection as Markdown
- Markdown editor : Apostrophe
- Integration with the file manager (start entry in context menue) as described below.

My favourite Markdown editor at the moment is Apostrophe. I like its clean and simple interface helping me to stay focused on the content while writing.[2]

When copying extracts from a web-page, I often need to preserve its hyperlinks. For this, the Firefox browser addon Copy Selection as Markdown is very handy. It precedes the copied extract with a hyperlink to the origin of the webpage. When *TP-Note* reads the extract from the clipboard, it uses the first Markdown hyperlink it can find for composing the note's title and its filename on disk. This is why the web page's name ends up automatically in the note's title and filename. Here a sample work flow:
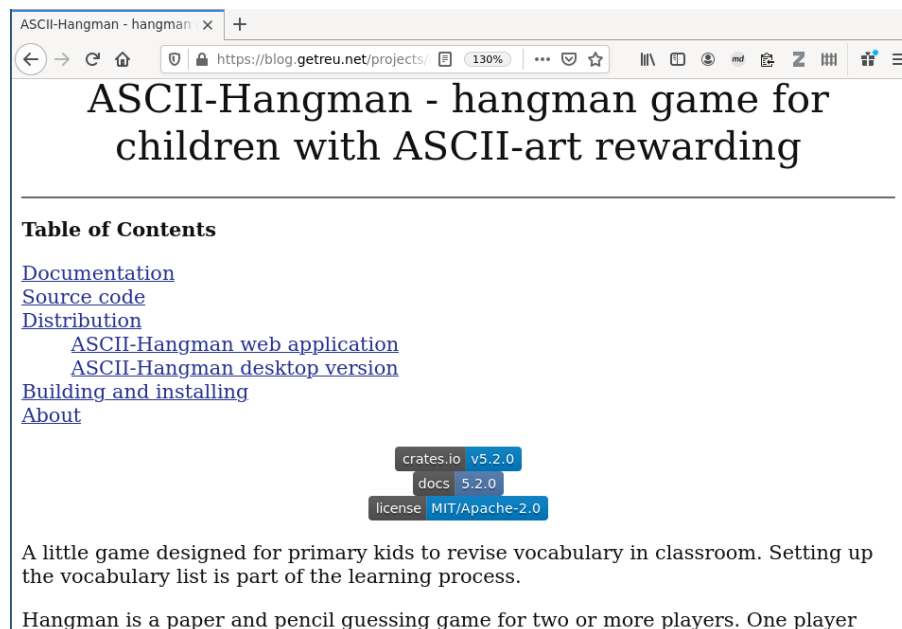
A webpage to copy extracts from
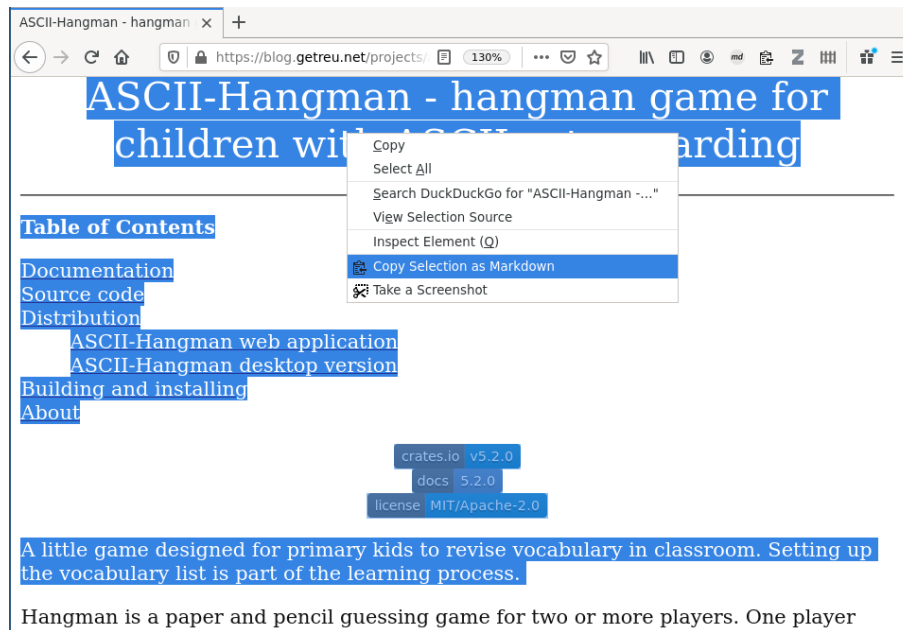
"Copy Selection as Markdown"
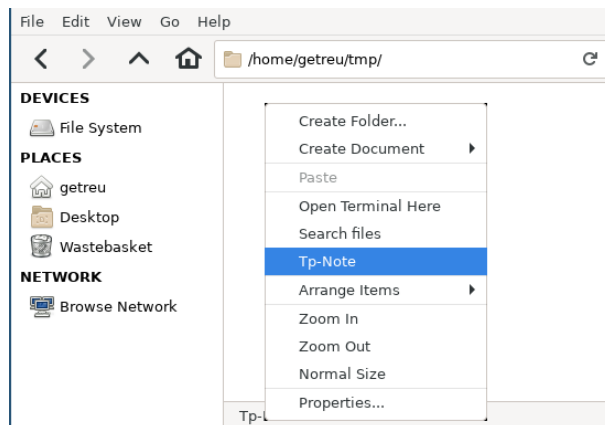
Right-click on or in the new note's destination folder and start Tp-Note
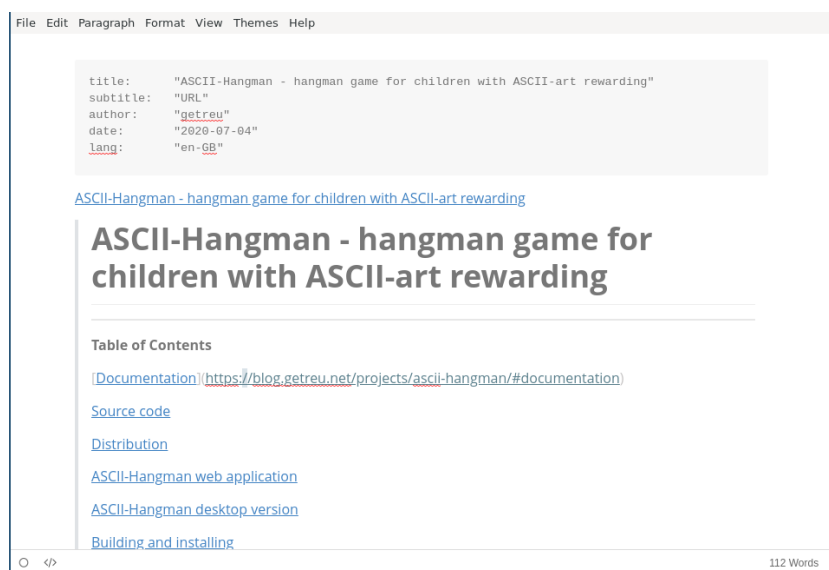
The new template generated note opened with Typora
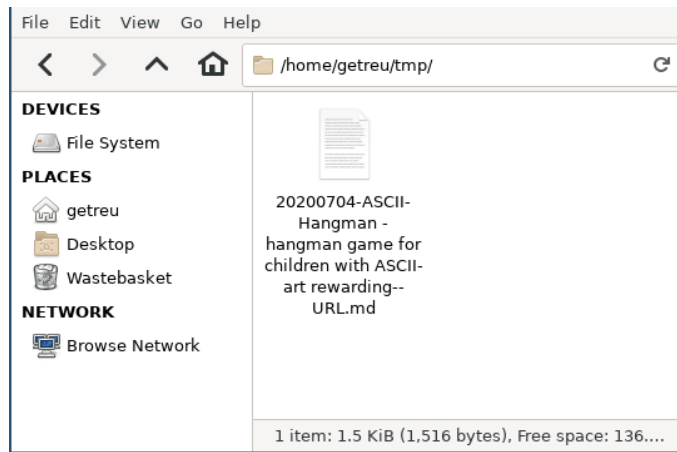
The new note file on disk after closing the editor

Note, no content or filename was edited manually in this example. *Tp-Note* takes care of interpreting the clipboard's content and generating the file on disk.

1.7 Note taking for system administrators (and console lovers)

As *Tp-Note* makes extensive use of the clipboard, it mainly targets desktop systems running a graphical environment. But also when working on the console *Tp-Note* can be useful with its built-in clipboard simulation: Instead of copying the content into your clipboard, pipe it into *Tp-Note*:

```
echo  "Some clipboard content" | tpnote
```

1.7.1 Typical workflows

The following examples work with the full featured version of *Tp-Note* as well as with the `--no-default-features` console only version.

- Document a downloaded file:

  Download the file i3-extensions.zip:

  ```
  wget "http://blog.getreu.net/_downloads/i3-extensions.zip"
  ```

  Document from where you downloaded the file:

  ```
  echo  "[download](<http://blog.getreu.net/_downloads/i3-extensions.zip>)" | tpnote i3-extensions.zip
  ```

  This creates the file `i3-extensions.zip--URL.md` with the following content:

```
---
title:      i3-extensions.zip
subtitle:   URL
author:     getreu
date:       2020-09-03
lang:       en-GB
---

[i3-extensions.zip](<i3-extensions.zip>)

[download](<http://blog.getreu.net/_downloads/i3-
extensions.zip>)
```

- Download a webpage, convert it to Markdown and insert the result into a *Tp-Note* file. The note's title is the name of the first hyperlink found in the webpage.

  Install `pandoc` and `curl`:

  ```
  sudo apt install pandoc curl
  ```

  Download and convert:

  ```
  curl 'https://blog.getreu.net' | pandoc -f html -t
  markdown_strict | tpnote
  ```

- Download a webpage while preserving its metadata:

  Same as above, but the following preserves the webpage's metadata, e.g. title, author, date... :

  ```
  curl 'https://blog.getreu.net' | pandoc --standalone -f html -t
  markdown_strict+yaml_metadata_block | tpnote
  ```

  creates the note file `20200910-Jens\ Getreu\'s\ blog.md` with the webpage's content.

- Generate a note for a given content with YAML header:

  ```
  echo -e "---\ntitle: Todo\nfile_ext: mdtxt\n---\n\nnothing" |
  tpnote
  ```

  creates the file `20200910-Todo.mdtxt` with the content:

  ```
  ---
  title:      Todo
  subtitle:   ''
  author:     getreu
  ```

```
date:        2020-09-13
lang:        en-GB

file_ext:    mdtxt
---

nothing
```

- Reformat the header of a note file:

```
mv "20200921-My Note.md" "20200921-My Note-(1).md"
cat "20200921-My Note-(1).md" | tpnote --batch
```

creates the file `20200921-My Note.md` with a rearranged header and the same body.

- Launch, for once only, a different text editor.
The external text editor, *Tp-Note* defaults to, is defined in the configuration file and can be changed there. If you want to use a different text editor just for a one-shot, type:

```
FILE=$(tpnote --batch); vi "$FILE"; tpnote --batch "$FILE"
```

- Create a new note overwriting the template's default for `subtitle`:

```
cd dev
echo -e "---\nsubtitle: Draft\n---\n# Draft" | tpnote
```

creates the note file `20200925-dev--Draft.md` with the content:

```
---
title:       dev
subtitle:    Draft
author:      Getreu
date:        2020-09-25
lang:        en-GB
---

# Draft
```

- Synchronize filenames and headers of all note files in the current directory:

```
find . -type f -name "*.md" -exec tpnote --batch {} \; >/dev/null
```

- Generate an HTML rendition of an existing note file in the same directory:

```
tpnote --export='./my_notes' './my_notes/20210209-debug--
Note.md'
```

or, equivalent but shorter:

```
tpnote --export= './my_notes/20210209-debug--Note.md'
```

or, even shorter:

```
tpnote -x '' './my_notes/20210209-debug--Note.md'
```

- Generate a PDF rendition of an existing note file :

Install the `wkhtmltopdf`-tool:

```
sudo apt install wkhtmltopdf
```

Generate the PDF rendition of the existing note `20210122-my--Note.md`:

```
tpnote -x - '20210122-my--Note.md' | wkhtmltopdf - '20210209-
debug--Note.md.pdf'
```

- View and follow hyperlinks in a note file:

When no graphical environment is available, *Tp-Note* disables the viewer feature with its internal HTTP server. As a workaround, use *Tp-Note*'s HTML export flag and pipe the result into a text based web browser.

Install the text based web browser `lynx`:

```
sudo apt install lynx
```

Convert the existing note `20210122-my_note.md` into HTML and open the rendition with `lynx`:

```
tpnote -x - '20210122-my_note.md' | lynx --stdin
```

Note, the above also works in case *Tp-Note* was compiled with `--no-default-features` which is recommended for headless systems.

2 How it works: Organize your files and notes with sort-tags

Consider the following *Tp-Note*-file:

```
20151208-Make this world a better place--Suggestions.md
```

The filename has 4 parts:

```
{{ fm_sort_tag }}-{{ fm_title }}--{{ fm_subtitle }}.
{{ fm_file_ext }}
```

A so called *sort-tag* is a numerical prefix at the beginning of the file-name. It is used to order files and notes in the file system. Besides numerical digits and whitespace, a *sort-tag* can be any combination of -_.[3] and is usually used as:
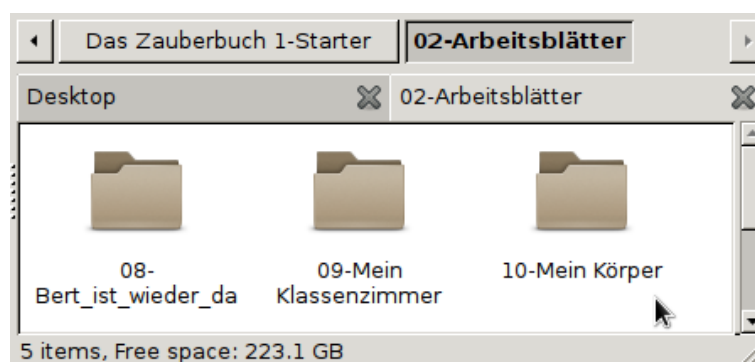
- *chronological sort-tag*

  ```
  20140211-Reminder.doc
  20151208-Manual.pdf
  2015-12-08-Manual.pdf
  ```
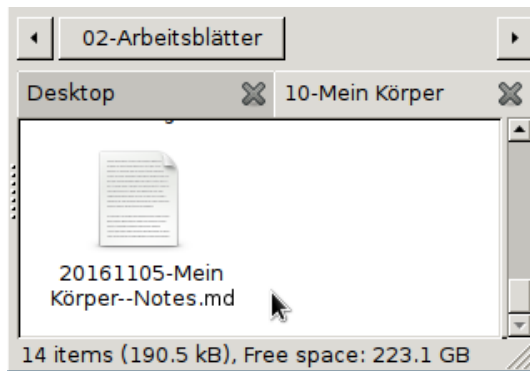
- or as a *sequence number sort-tag*.

  ```
  02-Invoices
  08-Tax documents
  09_02-Notes
  09.02-Notes
  ```

The figures below illustrate organizing files with *sort-tags".

Folders with sequence number sort-tag

File with chronological sort-tag

When *Tp-Note* creates a new note, it automatically prepends a *chronological sort-tag* of today. The `{{ fm_title }}` part is usually derived from the parent directory's name omitting its own *sort-tag*.

3 Installation

Depending on the operating system, the installation process is more or less automated and can be divided into two steps:

1. Minimum setup without file manager integration
   This step consists of downloading *Tp-Note*'s binary and copying it to your hard-disk. See section Distribution on *Tp-Note*'s project page for a list of available packages and binaries.

2. Optional integration with your file manager.

At the moment of this writing, an installer automating steps 1. and 2. is available for Windows only. Packages for Debian Linux and Ubuntu help you with step 1. For other operating systems check section Distribution for precompiled binaries or build Tp-Note yourself.

3.1 Minimum setup without file manager integration

Tp-Note's template engine can be tested and used without file manager integration. Download the appropriate binary for your architecture and place it in your PATH. See the Distribution in the section in Tp-Note's Readme-document for more details.

Bear in mind that the preferred way to install Tp-Note under Windows is the Windows installer package. However, if you do not have the right to install software on your computer, you can place the Tp-Note binary directly on your desktop.

3.2 Usage of the minimum setup

Once you have placed the `tpnote` binary in your `PATH` you can invoke Tp-Note on the command line by typing `tpnote` optionally followed by a directory path or a file path.

Having a copy (or symbolic link) of Tp-Note's binary `tpnote` on your desktop, enables you to execute the following workflow: To create a new note, either double-click on the *Tp-Note*-icon, or drag and drop a folder or file and drop it on the *Tp-Note*-icon. This opens an editor with your new note.

Anyway, for more comfort, I recommend integrating *Tp-Note* into the file manager's context menu. See section Optional integration with your file manager for more details. There you also will find a list of tested Markdown editors, if you wish to use one. *Tp-Note* works with every Unicode text editor and Markdown editor (see section Optional customization and man-page for more details).

Tp-Note's note files can be printed directly from the viewer (webbrowser) window or first converted into `.html` with `tpnote -x ''` `mynote.md`. For other formats e.g. `.docx`, `.odt` and `.pdf` use Pandoc or `wkhtmltopdf`.

3.3 Troubleshooting

3.3.1 Incompatible configuration files

While upgrading *Tp-Note*, new features may cause a change in *Tp-Notes*'s configuration file structure and the program may fail to start displaying an error message. Please consult the following section Upgrading in the project's Readme document for more information about incompatible configuration files.

3.3.2 Debugging

`Tp-Note`'s logging feature is controlled with the command line-options: `--debug` and `--popup` or by the corresponding configuration file variables: `arg_default.debug` and `arg_default.popup`.

Please consult *Tp-Note*'s manual page for more information about the debugging options `--debug` and `--popup` and how to use them.

BTW: Under Windows, when you see only Chinese characters in the *Notepad* text editor, update Windows to the latest version or install Notepad++.

3.4 Optional customization

3.4.1 Chose your favourite text editor and make it default

- Your preferred text editor is not *Notepad*? Change it![4]

  Note taking with *Tp-Note* is more fun with a good markup (Markdown) text editor, although any Unicode text editor will do (even Notepad >= Windows 10-update 1903). *Tp-Note* is preconfigured to work with:

  - Apostrophe | Flathub
  - VSCodium | Flathub
  - Visual Studio Code | Flathub
  - ReText — Simple but powerful editor for Markdown and reStructuredText

  Please refer to Tp-Note's man-page to learn how to register your text editor with *Tp-Note*'s configuration file.

3.4.2 Integrate *Tp-Note* with your file manager

- You prefer working in a desktop environment instead of working on a shell?

  Read the following section Optional integration with your file manager to learn how to configure your file manager's context menu to launch *Tp-Note*.

3.4.3 Multilingual customization

- Do you write your notes in multiple languages?

  Tp-Note integrates complex linguistic heuristics to determine in what natural language a new note is authored and stores the result in the `lang:` header variable of the new note.

  This process can be configured in various ways. The most important is to provide *Tp-Note* with a list of language candidates you write your notes. C.f. the variable `tmpl.filter.get_lang` in Tp-Note's configuration file.

You may also want to indicate the default region codes of your preferred languages. C.f. the variable `tmpl.filter.map_lang` in Tp-Note's configuration file.

Please refer to *customization* section in <u>Tp-Note's man-page</u> to learn how to configure *Tp-Note*'s natural language processing.

3.4.4 Choose the web browser for note viewing and make it your default

- Is your preferred web browser is not *Firefox*? Change it![5]

After opening the text editor, *Tp-Note* internally renders the note file and opens a web browser to display the note's content. Which web browser on your system will be launched, depends on which of them *Tp-Note* finds first by searching through a configurable list of well known web browsers.
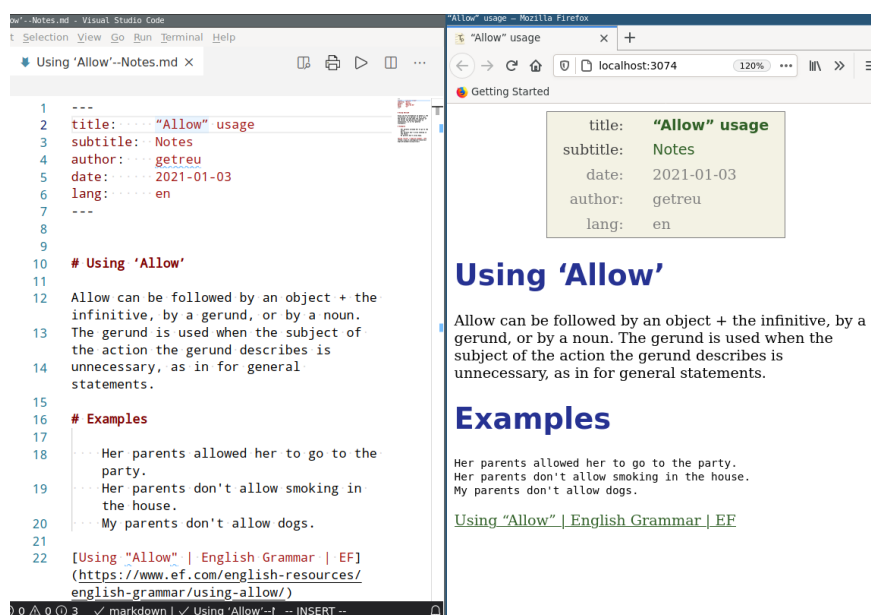
Tp-Note with open text editor (left) and viewer (right)

Please refer to <u>Tp-Note's man-page</u> to learn how change which web browser *Tp-Note* launches as note viewer.

3.4.5 Customize the way how *Tp-Note*'s viewer renders the note's content

The way the note will appear in your web browser depends on:

- which of *Tp-Note*'s internal markup renderer is used and

- the HTML template, that defines the visual appearance (colours, fonts etc.) of the rendition.

Please refer to Tp-Note's man-page to learn how to register a file extension with a particular markup renderer or to learn how to change the HTML-template that renders the note's content.

### 3.4.6 Change the default markup language

- Your preferred markup language is not *Markdown*, but ReStructuredText, Asciidoc, T2t, Textile, Wiki* or *Mediawiki*? Change it!

  Tp-Note's core function is a template system and as such it is markup language agnostic. The default templates largely abstain from markup specific code, which makes it easy to switch the default new note's markup language. Please refer to Tp-Note's man-page to learn how to change its templates in the configuration file.

  In addition, *Tp-Note* comes with a build in note viewer which is optional and independent from its core functionality. When *Tp-Note* opens a note file, it detects the markup language through the note file extension and launches the associated builtin markup renderer. The whole process can be customized in *Tp-Note*'s configuration file. Please refer to Tp-Note's man-page for details.

4 Optional integration with your file manager

This section shows how to integrate *Tp-Note* in the context menu of your file manager. The context menu appears, when you click right on a file icon, on a directory icon or on the white space in between (cf. figure below). In the following we will configure the file manager to launch *Tp-Note* with the path to the selected icon.
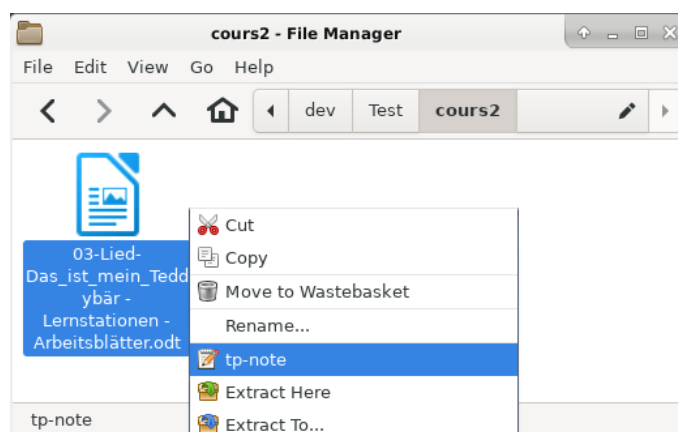
Tp-note in the context-menu (right-click menu)

4.1 Windows file explorer configuration

Tp-Note is distributed with a Microsoft Windows Installer package `tpnote-x.x.x-x86_64.msi`, which automates the following key registration. Omit this section if you have installed *Tp-Note* through this `.msi` package!

1. Make the directory `C:\Program Files\tpnote\bin\` with Administrator rights and move the binary `tpnote.exe` into it.

2. Open the *notepad* text editor and paste the following registry key into it.

```
Windows Registry Editor Version 5.00

[HKEY_CLASSES_ROOT\Directory\Background\shell\New Tp-Note]

[HKEY_CLASSES_ROOT\Directory\Background\shell\New Tp-
Note\command]
@="\"C:\\Program Files\\tpnote\\bin\\tpnote.exe\""

[HKEY_CLASSES_ROOT\*\OpenWithList\tpnote.exe]
@=""




[HKEY_CLASSES_ROOT\SystemFileAssociations\.txt\shell\edit.tpnote
@="Edit Tp-Note"



[HKEY_CLASSES_ROOT\SystemFileAssociations\.txt\shell\edit.tpnote
@="\"C:\\Program Files\\tpnote\\bin\\tpnote.exe\" \"%1\""



[HKEY_CLASSES_ROOT\SystemFileAssociations\.txt\shell\view.tpnote
@="View Tp-Note"



[HKEY_CLASSES_ROOT\SystemFileAssociations\.txt\shell\view.tpnote
@="\"C:\\Program Files\\tpnote\\bin\\tpnote.exe\" \"-v\" \"-
n\" \"%1\""



[HKEY_CLASSES_ROOT\SystemFileAssociations\.txt\shell\export.tpno
@="Export Tp-Note"
```

```
[HKEY_CLASSES_ROOT\SystemFileAssociations\.txt\shell\export.tpno
 @="\"C:\\Program Files\\tpnote\\bin\\tpnote.exe\" \"--
export=\" \"%1\""


[HKEY_CLASSES_ROOT\SystemFileAssociations\.md\shell\edit.tpnote
 @="Edit Tp-Note"


[HKEY_CLASSES_ROOT\SystemFileAssociations\.md\shell\edit.tpnote
 @="\"C:\\Program Files\\tpnote\\bin\\tpnote.exe\" \"%1\""


[HKEY_CLASSES_ROOT\SystemFileAssociations\.md\shell\view.tpnote
 @="View Tp-Note"


[HKEY_CLASSES_ROOT\SystemFileAssociations\.md\shell\view.tpnote
 @="\"C:\\Program Files\\tpnote\\bin\\tpnote.exe\" \"-v\" \"-
n\" \"%1\""


[HKEY_CLASSES_ROOT\SystemFileAssociations\.md\shell\export.tpno
 @="Export Tp-Note"


[HKEY_CLASSES_ROOT\SystemFileAssociations\.md\shell\export.tpno
 @="\"C:\\Program Files\\tpnote\\bin\\tpnote.exe\" \"--
export=\" \"%1\""
```
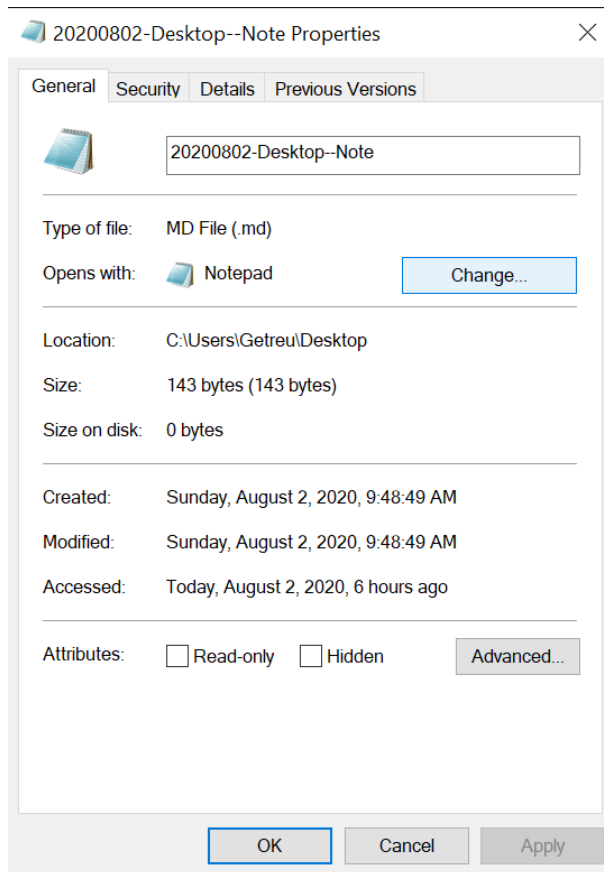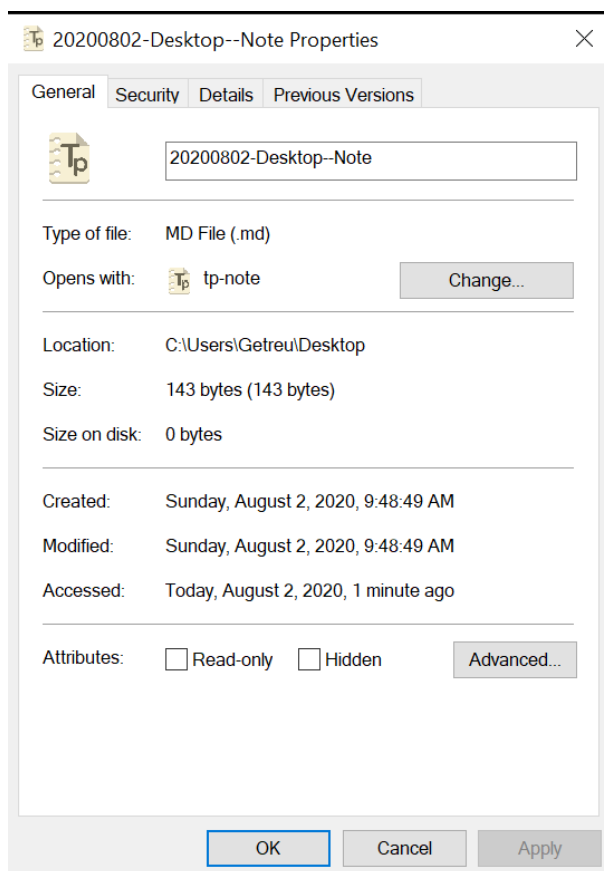
3. Save the file as:

   - File name: `tpnote.reg`
   - Save as type: `All files`
   - Encoding: `UTF-16 LE`

4. Double-click on `tpnote.reg` and confirm several times.

5. Assign `tpnote` as default application for `.md`-files

Click-right on some .md file to open file properties

Press "Change ..." and choose "Tp-Note"

4.2 Linux file manager configuration

To simplify the configuration we first place the binary *Tp-Note* in our
`$PATH`:

```
cd /usr/local/bin
sudo wget https://blog.getreu.net/projects/tp-note/_downloads/
x86_64-unknown-linux-gnu/release/tpnote
sudo chmod 755 tpnote
```

Debian and *Ubuntu* user can also download [Debian/Ubuntu package]
and install it with:

```
sudo dpkg -i tpnote_latest_amd64.deb
```
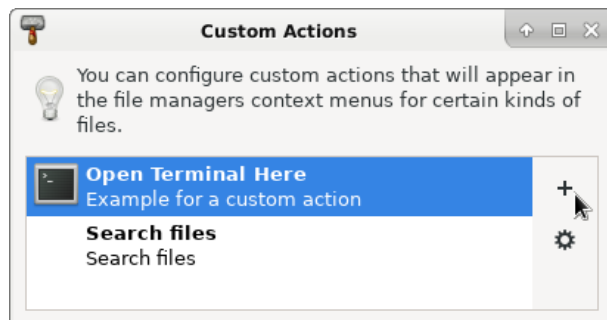
4.2.1 Configure Thunar's custom actions

Most file manager allow extending the context menu. As an example,
the following images show the configuration of the *Thunar* file manger.

4.2.1.1 Add context menu entry: Edit Tp-Note

In *Thunar*'s menu go to:

```
Edit -> Configure custom actions...
```


Thunar's custom action configuration

Edit custom action

Appearance Condition

Thunar's custom action configuration with Tp-Note

4.2.1.2 Add context menu entry: View Tp-Note

The following context menu entry allows us to view the rendered note in the system's default web browser. This is very handy when your note contains hyperlinks.

In Thunar, we add a custom action the same way as we did before:
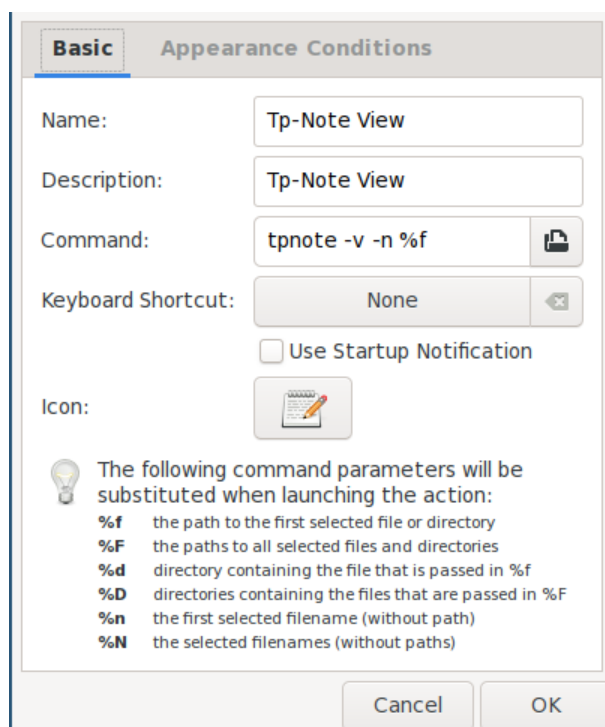
Edit custom action
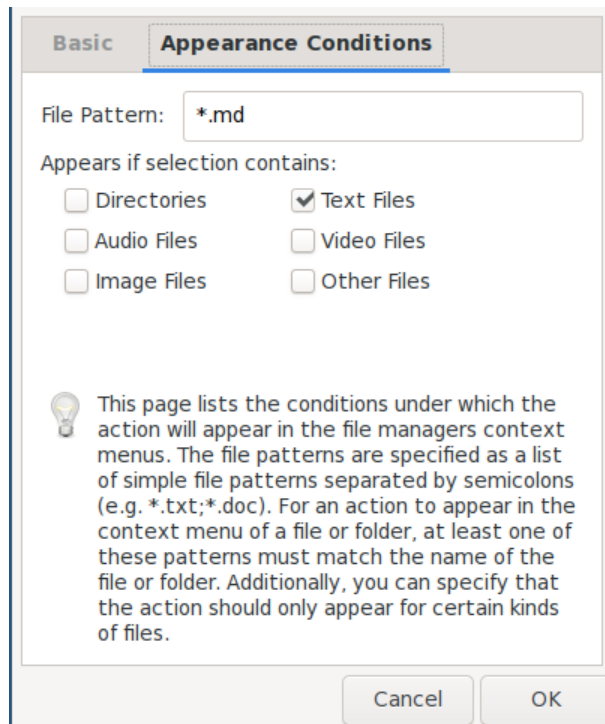
Appearance Condition

4.2.2 Configure Thunar's custom actions system-wide

Alternatively, instead of manually adding custom actions for each user, you can do this system-wide:

```
sudo nano /etc/xdg/Thunar/uca.xml
```

Search for `</actions>` and replace it with:

```xml
<action>
  <icon>accessories-text-editor</icon>
  <name>Tp-Note</name>
  <command>tpnote %f</command>
  <description>Tp-Note</description>
  <patterns>*</patterns>
  <directories/>
  <audio-files/>
  <image-files/>
  <other-files/>
  <text-files/>
  <video-files/>
</action>
<action>
  <icon>accessories-text-editor</icon>
  <name>Tp-Note View</name>
  <command>tpnote -v -n %f</command>
```

```xml
        <description>Tp-Note View</description>
        <patterns>*.txt; *.md;*.rst;*.adoc;*.txtnote</patterns>
        <text-files/>
      </action>
    </actions>
```

The change becomes effective only after the user deletes his own configuration file in `~/.config/Thunar/uca.xml`:

```
killall thunar
rm ~/.config/Thunar/uca.xml
thunar
```

Optional bonus: add a menu entry "Download webpage as Markdown"

In addition to the above, the following adds a context menu entry for fast downloading and converting a webpage to a Markdown Tp-Note file.

First install some helper programs:

```
sudo apt install xclip curl pandoc
```

Then edit the system-wide Thunar configuration file:

```
sudo nano /etc/xdg/Thunar/uca.xml
```

Search for `</actions>` and replace it with:

```xml
      <action>
        <icon>accessories-text-editor</icon>
        <name>Download URL here</name>
        <command>curl $(xclip -o)| pandoc --standalone -f html -t
markdown_strict+yaml_metadata_block+pipe_tables | tpnote  %F</
command>
        <description>Download URL</description>
        <patterns>*</patterns>
        <directories/>
      </action>
    </actions>
```

The change becomes effective only after the user deletes his own configuration file in `~/.config/Thunar/uca.xml`:

```
killall thunar
rm ~/.config/Thunar/uca.xml
thunar
```

Optional bonus 2: add a menu entry "Export note as Pdf"

First install the `weasyprint` filter program:

```
sudo apt install weasyprint
```

Then edit the system-wide Thunar configuration file:

```
sudo nano /etc/xdg/Thunar/uca.xml
```

Search for `</actions>` and replace it with: [6]

```
        <action>
          <icon>accessories-text-editor</icon>
          <name>Tp-Note Export</name>
            <command>tpnote --export=- %f | weasyprint - %f.pdf</command>
          <description>Tp-Note Export</description>
          <patterns>*.txt; *.md;*.rst;*.adoc;*.txtnote</patterns>
          <text-files/>
        </action>
      </actions>
```

The change becomes effective only after the user deletes his own configuration file in `~/.config/Thunar/uca.xml`:

```
killall thunar
rm ~/.config/Thunar/uca.xml
thunar
```

4.2.3 Configure Pcmanfm's custom actions system-wide

Pcmanfm is the default file manager in *Lubuntu* and in *Raspbian* on the Raspberry Pi.

Create the configuration file:

```
sudo nano /usr/local/share/file-manager/actions/tpnote.desktop
```

with the following content:

```
[Desktop Entry]
Type=Action
Name[en]=Tp-Note
```

```
Tooltip=Tp-Note
Icon=package-x-generic
Profiles=profile-zero;

[X-Action-Profile profile-zero]
Name[en]=Default profile
Exec=tpnote %f
```

The above creates the custom context menu item *Tp-Note*.

4.2.3.1 Note viewer

Create the configuration file:

```
sudo nano /usr/local/share/file-manager/actions/tpnote-
view.desktop
```

with the following content:

```
[Desktop Entry]
Type=Action
Name[en]=Tp-Note View
Tooltip=Tp-Note View
Icon=package-x-generic
Profiles=profile-zero;

[X-Action-Profile profile-zero]
Name[en]=Default profile
Exec=tpnote -v -n %f
```

The above creates the custom context menu item *Tp-Note View*.

4.2.4 Configure the text based file manager MidnightCommander

The Ncurses library based file manager *MidnightCommander* `mc` enjoys
great popularity among people working on the console. As *Tp-Note*
stores the note's content in UTF-8 encoded plain text, `mc` can be used for
full text searches in all note files of a directory. Start the full text search
with the keys `[Esc] [?]`.

The following instructions configure `mc`'s `[F3]`-key to open `.md` files for
viewing. This is where *Tp-Note* generates the HTML rendition of the
note file and opens the rendition with the *Lynx* web browser. The
`[Enter]`-key runs *Tp-Note* in editing mode.

  1. First install the *Midnight Commander* and the *Lynx* web browser:

```
              sudo apt install mc lynx
```

2. Edit `mc`'s configuration file `/etc/mc/mc.ext`:

```
          sudo nano /etc/mc/mc.ext
```

3. Find the following lines (Debian 11+):

```
shell/i/.md
        Include=editor
```

and disable them:

```
#shell/i/.md
#       Include=editor
```

4. Replace the line `default/*` with:

```
          regex/i/.(txt|md|rst|adoc|txtnote)$
              Open=tpnote %f
              View=if HTML=`tpnote -b -n -x - %f`; then (echo $"HTML" |
    lynx --stdin); else less    %f; fi


          default/*
```

5. Restart all instances of `mc` :

```
          sudo killall mc
          mc
```

To test the configuration, navigate to some `.md` note file and press `[F3]` or `[Enter]`.

---

1. *Tp-Note* is preconfigured to work with many well-known external text editors: e.g.: `code`, `atom`, `retext`, `geany`, `gedit`, `mousepad`, `leafpad`, `nvim-qt`, and `gvim` under Linux and `notpad++` and `notepad` under Windows. To register your own text editor, please consult the man-page. For best user experience, I recommend text editors with internal markup previewer.↩

2. At the moment I do not recommend using Typora because of this bug: Errornous rendering when the value of the last header var is the empty string · Issue #4633 · typora/typora-issues↩

3. The compulsory trailing - separator is not considered to be part of a sort-tag, although dashes within the sort-tag are allowed.↩

4. *Tp-Note* is preconfigured to work with many well-known external text editors: e.g.: `code`, `atom`, `retext`, `geany`, `gedit`, `mousepad`, `leafpad`, `nvim-qt`, and `gvim` under Linux and `notpad++` and `notepad` under Windows. To register your own text editor, please consult the man-page. For best user experience, I recommend text editors with internal markup previewer.↩

5. *Tp-Note* is preconfigured to work with many well-known external text editors: e.g.: `code`, `atom`, `retext`, `geany`, `gedit`, `mousepad`, `leafpad`, `nvim-qt`, and `gvim` under Linux and `notpad++` and `notepad` under Windows. To register your own text editor, please consult the man-page. For best user experience, I recommend text editors with internal markup previewer.↩

6. The `sed` filter is a workaround of a [bug in wkhtmltopdf](). Once this is solved, you can remove the `sed` command. Then the line 4 of the above listing becomes: `<command>tpnote --export=- %f | wkhtmltopdf --footer-center "[page]/[topage]" -B 2cm -L 2cm -R 2cm -T 2cm - %f.pdf</command>`↩