

spmenu(1) 0.4.1 | fancy dynamic menu

spmenu

spmenu is a program for X11 that reads standard input and allows the user to select items separated by a newline. It is a fork of [suckless's dmenu](#) which is a good more minimal alternative to spmenu.

What makes spmenu different from all the dmenu forks is that spmenu has features like color support, Vim-like modes, image support, proper mouse/keybind configuration, and more.

It should be noted that most of these features may be disabled by the user, either during compile time or through configuration.

Usage

On runtime, spmenu reads from standard input (stdin). spmenu items are separated by a newline (`\n`). When (by default) Enter is pressed, the selected item will be piped to stdout.

This allows things like `printf "Apple\nOrange\nPear\n" | spmenu`. This command will spawn an spmenu window with three items, 'Apple', 'Orange' and 'Pear'. This can be used in shell scripts to create interactive menus.

On top of this, you can specify arguments to change the behavior of spmenu. See a list below for a list.

Arguments

You may use long, descriptive arguments or the shorter arguments.

- mh, --line-height height
Set spmenu line height to height
- mw, --min-width width
Set minimum width to width
- g, --columns grid
Set the number of grids to grid
- gc, --generate-cache
Generate image cache
- ngc, --no-generate-cache
Don't generate image cache
- mc, --max-cache size
Set max image cache size to size
- rw, --relative-width
Enable relative input width
- nrw, --no-relative-width
Disable relative input width
- f, --fast
Grabs keyboard before reading stdin
- F, --fuzzy
Enable fuzzy matching
- NF, --no-fuzzy
Disable fuzzy matching
- P, --password
Hide characters
- nP, --no-password
Don't hide characters
- p, --prompt text
Set spmenu prompt text to text
- It, --input text
Set initial input text to text
- ip, --indent
Indent items to prompt width
- nip, --no-indent
Don't indent items to prompt width
- ci, --color-items
Color items
- nci, --no-color-items
Don't color items
- sgr, --sgr

Interpret SGR sequences

- nsg, --no-sgr
Display SGR sequences as text
- a, --alpha
Enable alpha
- na, --no-alpha
Disable alpha
- tp, --allow-typing
Allow the user to type
- nt, --no-allow-typing
Don't allow typing, the user must select an option
- x, --x-position x offset
Offset spmenu x position by x offset
- y, --y-position y offset
Offset spmenu y position by y offset
- n, --preselect line
Preselect line line in the list of items
- z, --width width
Width of the spmenu window
- nmt, --normal-mode-text text
Set normal mode text to text
- imt, --insert-mode-text text
Set insert mode text to text
- clon, --caps-lock-on-text text
Set caps lock on text to text
- clof, --caps-lock-off-text text
Set caps lock off text to text
- bw, --border-width width
Set width of the border to width . 0 will disable the border
- so, --sort
Sort matches
- nso, --no-sort
Don't sort matches
- pri, --priority pri1,pri2,pri3
Specify a list of items that take priority
- s, --case-sensitive
Use case-sensitive matching
- ns, --case-insensitive
Use case-insensitive matching
- nm, --normal
Start spmenu in normal mode
- im, --insert
Start spmenu in insert mode
- t, --top

Position spmenu at the top of the screen

-b, --bottom
Position spmenu at the bottom of the screen

-c, --center
Position spmenu at the center of the screen

-hm, --hide-mode
Hide mode indicator

-hit, --hide-item
Hide items

-hmc, --hide-match-count
Hide match count

-hla, --hide-left-arrow
Hide left arrow

-hra, --hide-right-arrow
Hide right arrow

-hpr, --hide-prompt
Hide prompt

-hip, --hide-input
Hide input

-hpl, --hide-powerline
Hide powerline

-hc, --hide-caret, --hide-cursor
Hide caret

-hhl, --hide-highlighting
Hide highlight

-hi, --hide-image
Hide image

-hcl, --hide-caps
Hide caps lock indicator

-sm, --show-mode
Show mode indicator

-sit, --show-item
Show items

-smc, --show-match-count
Show match count

-sla, --show-left-arrow
Show left arrow

-sra, --show-right-arrow
Show right arrow

-spr, --show-prompt
Show prompt

-sin, --show-input
Show input

-spl, --show-powerline

Show powerline
-sc, --show-caret, --show-cursor
 Show caret
-shl, --show-highlighting
 Show highlight
-si, --show-image
 Show image
-scl, --show-caps
 Show caps lock indicator
-xrdb, --xrdb
 Load .Xresources on runtime
-nxrdb, --no-xrdb
 Don't load .Xresources on runtime
-gbc, --global-colors
 Recognize global colors (such as *.color1) on runtime
-ngbc, --no-global-colors
 Don't recognize global colors (such as *.color1) on runtime
-m, --monitor monitor
 Specify a monitor to run spterm on
-w, --embed window id
 Embed spterm inside window id
-H, --hist-file hist file
 Specify a path to save the history to
-ig, --image-gaps gaps
 Set image gaps to gaps
-txp, --text-padding padding
 Set text padding to padding
-vem, --vertical-margin margin
 Set the vertical margin
-hem, --horizontal-margin margin
 Set the horizontal margin
-lp, --vertical-padding padding
 Set the vertical padding
-hp, --horizontal-padding padding
 Set the horizontal padding
-la, --left-arrow-symbol symbol
 Set the left arrow to symbol
-ra, --right-arrow-symbol symbol
 Set the right arrow to symbol
-is, --image-size size
 Image size
-it, --image-top
 Position the image at the top
-ib, --image-bottom

Position the image at the bottom

-ic, --image-center
Position the image in the center

-itc, --image-topcenter
Position the image in the top center

-wm, --managed, --x11-client
Spawn spmenu as a window manager controlled client/window.
Useful for testing

-nwm, --unmanaged
Don't spawn spmenu as a window manager controlled client/
window. Useful for testing

-cf, --config-file file
Set config file to load to file

-lcfg, --load-config
Load spmenu configuration (~/.config/spmenu/spmenu.conf,
~/spmenurc or ~/.config/spmenu/spmenurc)

-ncfg, --no-load-config
Don't load spmenu configuration (~/.config/spmenu/spmenu.conf,
~/spmenurc or ~/.config/spmenu/spmenurc)

-v, --version
Print spmenu version to stdout

-fn, --font font
Set the spmenu font to font

-nif, --normal-item-foreground color
Set the normal item foreground color

-nib, --normal-item-background color
Set the normal item background color

-sif, --selected-item-foreground color
Set the selected item foreground color

-sib, --selected-item-background color
Set the selected item background color

-npf, --normal-item-priority-foreground color
Set the normal item (high priority) foreground color

-npb, --normal-item-priority-background color
Set the normal item (high priority) background color

-spf, --selected-item-priority-foreground color
Set the selected item (high priority) foreground color

-spb, --selected-item-priority-background color
Set the selected item (high priority) background color

-pfg, --prompt-foreground color
Set the prompt foreground color

-pbg, --prompt-background color
Set the prompt background color

-ifg, --input-foreground color

Set input foreground color
-ibg, --input-background color
Set input background color
-mnbg, --menu-background color
Set the menu background color
-nhf, --normal-highlight-foreground color
Set the normal highlight foreground color
-nhb, --normal-highlight-background color
Set the normal highlight background color
-shf, --selected-highlight-foreground color
Set the selected highlight foreground color
-shb, --selected-highlight-background color
Set the selected highlight background color
-nfg, --number-foreground color
Set the foreground color for the match count
-nbg, --number-background color
Set the background color for the match count
-mfg, --mode-foreground color
Set the foreground color for the mode indicator
-mbg, --mode-background color
Set the background color for the mode indicator
-laf, --left-arrow-foreground color
Set the left arrow foreground color
-raf, --right-arrow-foreground color
Set the right arrow foreground color
-lab, --left-arrow-background color
Set the left arrow background color
-rab, --right-arrow-background color
Set the right arrow background color
-cfc, --caret-foreground color
Set the caret foreground color
-cbc, --caret-background color
Set the caret background color
-bc, --border-background color
Set the border color
-sgr0, --sgr0 color
Set the SGR 0 color
-sgr1, --sgr1 color
Set the SGR 1 color
-sgr2, --sgr2 color
Set the SGR 2 color
-sgr3, --sgr3 color
Set the SGR 3 color
-sgr4, --sgr4 color

Set the SGR 4 color
-sgr5, --sgr5 color
Set the SGR 5 color
-sgr6, --sgr6 color
Set the SGR 6 color
-sgr7, --sgr7 color
Set the SGR 7 color
-sgr8, --sgr8 color
Set the SGR 8 color
-sgr9, --sgr9 color
Set the SGR 9 color
-sgr10, --sgr10 color
Set the SGR 10 color
-sgr11, --sgr11 color
Set the SGR 11 color
-sgr12, --sgr12 color
Set the SGR 12 color
-sgr13, --sgr13 color
Set the SGR 13 color
-sgr14, --sgr14 color
Set the SGR 14 color
-sgr15, --sgr15 color
Set the SGR 15 color

dmenu compatibility can be achieved using these arguments:

-S
Don't sort matches
-i
Use case-insensitive matching
-nb color
Set the normal background color
-nf color
Set the normal foreground color
-sb color
Set the selected background color
-sf color
Set the selected foreground color

Keybinds

See `keybinds.h` for a list.

Modes

One of the features that separate `spmenu` from `dmenu` is `spmenu`'s different modes. As of version 0.2, there are two modes. Normal mode and Insert mode. These modes are of course similar to Vim.

Normal mode is the mode `spmenu` starts in unless a mode argument is specified. In normal mode, all keys perform some action, but you cannot type any actual text to filter items. This mode is used for navigation, as well as quickly selecting an item.

Insert mode is entered through (by default) pressing `i` in normal mode. In this mode, most keybinds do nothing. When you are in insert mode, you filter items by typing text into the field. Once you're done with insert mode, you can press Escape to enter normal mode again.

-p option

`spmenu` has a `-p` option, which stands for prompt. It allows you to specify text to display next to the item list. It is displayed on the left side of the `spmenu` window. It should be noted that the prompt is purely visual though.

Images

`spmenu` supports drawing images. This image is placed on the left side of the menu window. To use an image, pipe `IMG:/path/to/image` to `spmenu`. If you want you can specify arguments like usual. Note that you should add a Tab (`\t`) character after the path to the image file. Otherwise the text after will be interpreted as part of the filename and the image will not be drawn.

Any text after the Tab character will be interpreted as a regular item. In practice, drawing an image might look like this:

```
printf "IMG:/path/to/image\tLook at that image, isn't it awesome?  
\n" | spmenu
```

There are also a few image related arguments, such as:

`-is`, `-ig`, `-it`, `-ib`, `-ic`, `-itc` and `-gc`.

NOTE: Vector images (such as .svg) can be displayed too.

Colored text

spmenu supports colored text through SGR sequences. This is the same colors that you might already be using in your shell scripts. This means you can pipe practically any colored shell script straight into spmenu, no need to filter the output or anything.

Not only does it support colored text, but it also supports colored backgrounds. This allows something similar to the emoji highlight patch, except even more useful. Example:

```
printf "\033[0;44m😊\033[0m Emoji highlighting\n" | spmenu --  
columns 1
```

For 256 color support to work, you must add to the array. See `libs/color.h` if you want this.

See ‘SGR sequences’ for more information.

SGR sequences

A basic supported SGR sequence looks like this: `\033[X;YZm`

Here, X specifies if you want normal or bright colors. Y specifies if you want background or foreground. Z specifies the color number.

Foreground colors: 30 through 37 Background colors: 40 through 47
Reset: 0

NOTE: `;` is a separator, and in this example it separates the color number and normal/bright. `\033` may also be written as `^]` or simply ESC. The separator may be omitted for some sequences, such as `\033[0m` which resets the colorscheme.

spmenu supports most color sequences, although not true color by default (unless `-sgr` arguments are used).

There are a few arguments, you can override SGR colors on-the-fly using the `-sgrX` arguments. See ‘Arguments’ for more information.

Just as a tip, you can pipe your colored spmenu output to `sed -e 's/\x1b\[[0-9;]*m//g'`. This will clear the SGR sequences from the output. This is useful when you want to check what the output actually is.

Pango markup

If spmenu was compiled with Pango enabled (default), you should be able to utilize Pango markup in every part of spmenu. That is, the mode indicator, items, input, prompt, etc.

Pango markup allows you to style text similar to an HTML document. It also provides the `` tag, which can be used to do surprisingly complex things.

There are many convenient tags as well which can be used to avoid using a `` tag, such as:

```
<b>
    Bold text (<b>Bold</b>)
<i>
    Italic text (<i>Italic</i>)
<s>
    Strikethrough text (<s>Strikethrough</s>)
<u>
    Underline text (<u>Underline</u>)
<sub>
    Subscript (<sub>Subscript</sub>)
<sup>
    Superscript (<sup>Superscript</sup>)
<tt>
    Monospace font (<tt>Monospaced font is used here</tt>)
<small>
    Small text (<small>text is so small here</small>)
<big>
    Big text (<big>text is so big here</big>)
```

Note that Pango markup is NOT escaped, and is piped to stdout. Therefore you need to parse it manually. Doing so with `sed` is very easy. For example: `... | spmenu ... | sed 's/<big>//g; s/</big>//g'`

See [this page](#) for more information.

Configuration

Unlike dmenu, spmenu has a configuration file which can be edited by hand. It is located in `~/.config/spmenu/spmenu.conf`, but you can override this by exporting `$XDG_CONFIG_HOME`.

When spmenu is installed, it copies a sample configuration to `/usr/share/spmenu/spmenu.conf`. You can copy this to your `.config/spmenu` directory. This configuration file is loaded on startup.

You can also include other configuration files in the configuration file using `@include "path/to/config"`.

.Xresources

spmenu also has `.Xresources` (xrdb) support built in. It reads the xrdb (`.Xresources` database) on runtime. You may disable it by passing `-nxrdb`, or enable it by padding `-xrdb`. You can also set this in the regular config file.

spmenu loads `~/.config/spmenu/spmenurc` or alternatively if you're old fashioned, `~/.spmenurc` on startup. This requires that xrdb is available on your operating system. Of course, you don't NEED to use them, as you can just `xrdb -override` any `.Xresources` file you want.

You can also use wildcards (such as `*`) to achieve a global colorscheme. Programs like `pywal` do this to apply universal colorschemes.

Profiles

spmenu supports profiles. Profiles are like configuration files (See Configuration) that can be switched between quickly using a keybind.

Pressing (by default) `Ctrl+Shift+p` will list out profiles, and also allow you to add/remove existing profiles. Selecting a profile will switch to that profile. The selected profile will now be loaded on startup just like the `spmenurc` until another profile is selected.

Selecting 'Add' allows you to create a new profile. When a new profile is created it is going to use the spmenu defaults (copied from `/usr/share/spmenu/example.Xresources`). Profiles are going to be in

~/ .config/spmenu/profiles/ and the current profile is in ~/ .config/spmenu/ .profile. The profile can simply be edited using any text editor and be configured in .Xresources syntax.

Selecting `Remove` will allow you to pick a profile which will be permanently removed. Selecting `Default` will simply load `spmenurc` and nothing else on startup, as if this feature did not exist.

Not only can profiles be used to configure color schemes and fonts, allowing you to swap color schemes quickly, but also any other options you may want.

Note that any profiles are applied ON TOP of the current loaded profile, meaning if any options are missing from the selected profile, the setting in the profile used before will be used in place.

There are a few color schemes for `spmenu` included in the repository, see the `themes/` directory. Feel free to copy those to your profile directory.

Run launcher

`spmenu` includes a shell script called `spmenu_run`. It lists executable programs in `$PATH` and displays them to the user in a list. Not only that but it shows recently run programs first in the list.

`spmenu_run` will interpret any arguments as `spmenu` arguments and pass them to `spmenu`. Therefore `spmenu_run --prompt 'Run: '` will set the prompt to `Run: .` While it is similar to `dmenu_run` (and achieves the same goal), this version has some extra features.

The selected option is piped to `/bin/sh` (by default). Unlike `dmenu_run`, `spmenu_run` has some cool features. For example:

- Prepending `#` will spawn it in a terminal instead of just a shell.
- Prepending `?` will run the command in a function, most of the time used to display the man page.
- Prepending `magnet` will open a magnet link in `$TORRENT`
- Prepending `www` will open a page in `$BROWSER`

Most of the time you don't need to prepend `www` though, for example typing in `https://gnu.org` will open `gnu.org` in `$BROWSER` even without the prefix. Same goes for magnet links.

You can also configure the run launcher through editing `~/.config/spmenu/run/config` which is configured in shell syntax.

spmenu_desktop

In addition to the aforementioned `spmenu_run`, the `spmenu` package also provides `spmenu_desktop` which instead of reading `$PATH` only lists out `.desktop` entries.

Unlike the regular run launcher though, `spmenu_desktop` supports displaying an icon for entries that use one.

It can be configured through editing `~/.config/spmenu/desktop/config`. The configuration file can also be moved by setting `${XDG_CONFIG_HOME}`.

spmenu commands

`spmenu` has a few special commands. These work similar to the images. For example to list the version, in addition to the `--version` argument you can also simply run `printf 'spmenu:version' | spmenu`. There are a few of these.

`spmenu:version`

Print the `spmenu` version

`spmenu:license`

Print the `spmenu` license

`spmenu:test`

Print a test script for `spmenu` which tests various features. Report any weird issues you may have.

License

`spmenu` is licensed under the MIT license because that's the original suckless license. See the included `LICENSE` file for more information.

Reporting issues

Please report issues on the [Codeberg repository](#) or alternatively email me.