Macros and Registers in Vim

Jan Bundesmann

CLT 2023

Register

Terminology

Clipboard	`CTRL + (SHIFT +)C` / Copy & Paste
Primary selection	Current selection
	Paste with the middle mouse button
Registers	Internal clipboards in Vim

Vim modes

… at least the interesting ones

Normal	default at startup
Insert	enter by pressing `[iIaAsoO...]`
	exit using `<ESC>`
Visual	enter by pressing `[vV^v]`

Copy and Paste in Vim

{ycd}<MV>	yank / change / delete until end of movement MV
yy|cc|dd	yank / change / delete complete line
p|P	paste after / before current position
<V>{ycdp}	perform yank / change / delete on visual selection
{ycdv}ap	yank / change / delete / visually mark all paragraph
	variants: in paragraph; words or delimiters (`bBt[<`) instead of paragraphs

Copy and Paste in Vim

{ycd}<MV>	yank / change / delete until end of movement MV
yy|cc|dd	yank / change / delete complete line
p|P	paste after / before current position
<V>{ycdp}	perform yank / change / delete on visual selection
{ycdv}ap	yank / change / delete / visually mark all paragraph
	variants: in paragraph; words or delimiters (`bBt[<`) instead of paragraphs

Copy and Paste in Vim

{ycd}<MV>	yank / change / delete until end of movement MV
yy|cc|dd	yank / change / delete complete line
p|P	paste after / before current position
<V>{ycdp}	perform yank / change / delete on visual selection
{ycdv}ap	yank / change / delete / visually mark all paragraph
	variants: in paragraph; words or delimiters (`bBt[<`) instead of paragraphs

Copy and Paste in Vim

{ycd}<MV>	yank / change / delete until end of movement MV
yy|cc|dd	yank / change / delete complete line
p|P	paste after / before current position
<V>{ycdp}	perform yank / change / delete on visual selection
{ycdv}ap	yank / change / delete / visually mark all paragraph
	variants: in paragraph; words or delimiters (`bBt[<`) instead of paragraphs

Copy and Paste in Vim

{ycd}<MV>	yank / change / delete until end of movement MV
yy|cc|dd	yank / change / delete complete line
p|P	paste after / before current position
<V>{ycdp}	perform yank / change / delete on visual selection
{ycdv}ap	yank / change / delete / visually mark all paragraph
	variants: in paragraph; words or delimiters (`bBt[<`) instead of paragraphs

Registers

- Store and retrieve data from registers
- Named registers

"x{ycdp}	access register `x` for the next action
""{ycdp}	access default or unnamed register for the next action

48 available registers

see `:help registers` for reference

unnamed	`"`	read-only	`:`, `.`, `%`
named	`a` to `z`	alternate file	`#`
small delete	`-`	expression	`=`
numbered	`0` to `9`	last search	`/`
selection	`*`, `+`, `~`	black hole	`_`

Registers

- Store and retrieve data from registers
- Named registers

"x{ycdp}	access register `x` for the next action
""{ycdp}	access default or unnamed register for the next action

48 available registers

see `:help registers` for reference

unnamed	`"`	read-only	`:`, `.`, `%`
named	`a` to `z`	alternate file	`#`
small delete	`-`	expression	`=`
numbered	`0` to `9`	last search	`/`
selection	`*`, `+`, `~`	black hole	`_`

More on registers

query registers	`:registers`
`with-x`	+ contains clipboard
	* contains primary selection
filename	%

- `with-x` is a compiler option
  - required for clipboard interaction
  - neovim has this by default
  - `vim-gtk` for Debianoids
- Access register x in insert mode by pressing `^R+x`

Typical scenario

- Work on remote server
- Copy or move content between files

```
vim /path/to/file1 /path/to/file2
```

- Set vim behaviour to just hide buffers instead of closing

```
:set hidden
```

- Switch active files with the buffer command

```
:b1
```

- Copy or move content between files and eventually save and close all files

```
:wqall
```

Execute current file

Add to your settings:

```
map <F2> <Esc>:w<CR>:!%:p<CR>
```

Macros

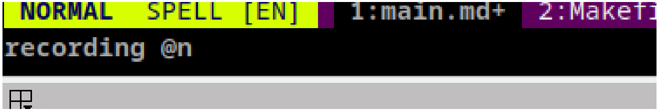
Principle

- Every action in vim is a set of key strokes.

Now there are two options:

- Write down your keystrokes to make them repeatable!
- Store them as macros!

Did you already use? Did you accidentally stumble upon it?



```
NORMAL  SPELL [EN]  1:main.md+  2:Makefi
recording @n
```

Using macros

Recording

1. Start recording macro `x` by pressing `qx`
2. Enter your commands
3. Stop recording by pressing `q` again

Replay

1. Press `@x` to replay macro x
2. If desired place a multiplier in front: `5@x`

One-time-macros

Sometimes repetition is required only once. Just call a complex set of keys multiple times:

```
<N>@='<COMMAND>'
```

Modifying macros

`:registers` lists defined macros

```
"xp
# modify macro and visually mark it
"xy
```

or create a new similar macro

```
"xp
# modify macro and visually mark it
"zy
```

Recursive macros

What happens when a macro `x` calls itself?

1. Ensure macro is empty

```
qxq
```

2. Record macro, don’t forget to call it at the end

```
qx
<DO SOMETHING>
@x
q
```

3. Call macro: It will run until it fails

Outro

Summary

- Registers in Vim are named clipboards
- Access to system clipboard through registers + and *
- Macros allow recording of repetitive work
- Macros are stored as registers

Thank you for your attention!

- Ottavia Balducci: Konfigurationsmanagement über verschiedene Netze mit AWX (So, 10 Uhr, V1)
- Pascal Fries: WebAssembly auf der Serverseite: Was ist WASI? (So, 16 Uhr, V2)
- Tobias Manske, Lukas Paluch: Autoscaling in Kubernetes – From Zero to Hero (So, 16:30 Uhr, V4)
- Sascha Rausch, Vincent Welker: Das Chaos überblicken – Monitoring & Tracing in Kubernetes (So, 17 Uhr, V6)

https://atix.de/chemnitzer-linux-tage-2023/

Summary

- Registers in Vim are named clipboards
- Access to system clipboard through registers + and *
- Macros allow recording of repetitive work
- Macros are stored as registers

Thank you for your attention!

- Ottavia Balducci: Konfigurationsmanagement über verschiedene Netze mit AWX (So, 10 Uhr, V1)
- Pascal Fries: WebAssembly auf der Serverseite: Was ist WASI? (So, 16 Uhr, V2)
- Tobias Manske, Lukas Paluch: Autoscaling in Kubernetes – From Zero to Hero (So, 16:30 Uhr, V4)
- Sascha Rausch, Vincent Welker: Das Chaos überblicken – Monitoring & Tracing in Kubernetes (So, 17 Uhr, V6)

Summary

- Registers in Vim are named clipboards
- Access to system clipboard through registers + and *
- Macros allow recording of repetitive work
- Macros are stored as registers

Thank you for your attention!

- Ottavia Balducci: Konfigurationsmanagement über verschiedene Netze mit AWX (So, 10 Uhr, V1)
- Pascal Fries: WebAssembly auf der Serverseite: Was ist WASI? (So, 16 Uhr, V2)
- Tobias Manske, Lukas Paluch: Autoscaling in Kubernetes – From Zero to Hero (So, 16:30 Uhr, V4)
- Sascha Rausch, Vincent Welker: Das Chaos überblicken – Monitoring & Tracing in Kubernetes (So, 17 Uhr, V6)

https://atix.de/chemnitzer-linux-tage-2023/