A few of our favourite tools

Marie-Hélène Burle & Alex Razoumov

training@westgrid.ca

March 16, 2022
lazygit

(Terminal UI for Git written in Go)
**lazygit** brings the ease of use of a visual Git tool to the command line.

Unlike slow, buggy, & limited Git GUIs, it is nimble, portable, & super fast.

In the following demo, as an additional visual goody, I will introduce **diff-so-fancy** to display the outputs of **git diff**.
Bat

(“A cat clone with wings”)

```
##
# sequential
#
# run it with:
# julia parallel.jl
##

using BenchmarkTools

function slow(n::Int, digits::String)
    total = 0.
    for i in 1:n
        if !occursin(digits, string(i))
            total += 1. / i
        end
    end
    println("total = ", total)
end

@btime slow(Int64(1e9),"9")
```
**bat** is a great tool to display the content of files in your terminal.

The standard Unix utility for this task is **cat**, but **bat** adds a lot to it.
Syntax highlighting...

...in a huge number of languages. See the impressive list with:

```
bat -L
```

```
cat /home/marie/parvus/prog/slides/static/files/tools/julia.jl
```
Syntax highlighting...

...in a huge number of languages. See the impressive list with:

```
bat -L
```

```
cat /home/marie/parvus/prog/slides/static/files/tools/julia.jl
```

```
# sequential

# run it with:
# julia parallel.jl

using BenchmarkTools

function slow(n::Int, digits::String)
    total = 0.
    for i in 1:n
        if !occursin(digits, string(i))
            total += 1. / i
        end
    end
    println("total = ", total)
end

@btime slow(Int64(1e9),"9")
```
Syntax highlighting...

...in a huge number of languages. See the impressive list with:

```
bat -L
```

```
bat /home/marie/parvus/prog/slides/static/files/tools/julia.jl
```

```
<p>| File: /home/marie/parvus/prog/slides/static/files/tools/julia.jl |</p>
<table>
<thead>
<tr>
<th>Size: 472 B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td># sequential</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td># run it with:</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td># julia parallel.jl</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>#=================================================================</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>using BenchmarkTools</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>function slow(n::Int, digits::String)</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>total = 0.</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>for i in 1:n</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>if !occursin(digits, string(i))</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>total += 1. / i</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>println(&quot;total = &quot;, total)</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>14</td>
</tr>
<tr>
<td>@btime slow(Int64(1e5),&quot;9&quot;)</td>
</tr>
</tbody>
</table>
```
Smart behaviour with pager

Uses `less` by default with options `-RFX`

Very convenient: quits the pager automatically if the entire content fits on one script
Integration with Git

```python
import torch
from torchvision import datasets, transforms
from matplotlib import pyplot as plt

# To run on CPU, replace 'cpu' by 'cuda'
device = torch.device('cpu')
dtype = torch.float
epochs = 30

# Prepare data
train = datasets.MNIST(~
    # '/data',
    # '/home/parus/parus/prog/slides/static/files/tools/two_layer_net_tensor.py'
    # transforms.Compose([
    #     transforms.ToTensor(),
    #     transforms.Normalize((0.1307,), (0.3081,))])

test = datasets.MNIST(~
    # '/data',
    # '/home/parus/parus/prog/slides/static/files/tools/two_layer_net_tensor.py'
    # transforms.Compose([
    #     transforms.ToTensor(),
    #     transforms.Normalize((0.1307,), (0.3081,))])

N, D_in, H, D_out = 60000, 784, 100, 10
x = train.data.view(60000, 784).type(dtype)
y = train.targets.unsqueeze().expand(60000, 10).type(dtype)

# Randomly initialize weights:
w1 = torch.randn(D_in, H, device = device, dtype = dtype)
w2 = torch.randn(H, D_out, device = device, dtype = dtype)

optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)

# Forward pass: compute predicted y by passing x to the model.
+ y_pred = model(x)
```
```
| File: /home/marie/parvus/prog/slides/static/files/tools/two_layer_net_tensor.py |
| Size: 2.3 KB |

```

```python
train = datasets.MNIST(
    # './data',
    '~/projects/def-sponsor00/data',
    train = False,
    download = False,
    transform = transforms.Compose(
        transforms.ToTensor(),
    ),
)
```
```
```
```
Man page syntax highlighting

Even colourizes your man pages if you set the `MANPAGER` environment variable (e.g. in your `.bashrc`):

```bash
export MANPAGER="sh -c 'col -bx | bat -l man -p'
```
Useful flags

bat -n  # Remove frame

bat -p  # Remove frame & line numbers

I personally have an alias for this: alias rat='bat -p'

bat -pp  # Remove frame & line numbers & don't use pager

bat -f  # Keep colour highlighting after piping

I made this the default behaviour for bat with the alias: alias bat='bat -f'
The defaults are great, but if they don’t suit you, you can change tab width, language, wrapping method...

```
bat --config-file                      # Location of config file (if it exists)

bat --generate-config-file            # Create config file template

bat --list-themes                     # List themes with examples
```
ripgrep

(A fast recursive alternative to grep)
ripgrep provides the utility `rg`:
an alternative to grep very fast for recursive searches
Syntax

The syntax is similar to that of `grep`.

Because, by default, `rg` is recursive (`grep` isn’t) & ignores some files, `rg -uuu` is equivalent to `grep -r`.
I will compare `grep` and `rg` to recursively search the word “Unix” in my directory “~/parvus/prog/tcl”

```
grep -r Unix ~/parvus/prog/tcl
rg Unix ~/parvus/prog/tcl
```

I will do a cold test (clearing my computer cache before each run) using `hyperfine`

```
hyperfine -i --prepare 'sync; echo 3 | sudo tee /proc/sys/vm/drop_caches' \
   'grep -r Unix ~/parvus/prog/tcl' \
   'rg Unix ~/parvus/prog/tcl'
```
Results

Benchmark 1: grep -r Unix ~/parvus/prog/tcl
Time (mean ± σ): 1.027 s ± 0.016 s [User: 0.269 s, System: 0.222 s]
Range (min ... max): 1.007 s ... 1.052 s 10 runs

Benchmark 2: rg Unix ~/parvus/prog/tcl
Time (mean ± σ): 76.0 ms ± 5.6 ms [User: 21.5 ms, System: 29.6 ms]
Range (min ... max): 63.0 ms ... 82.4 ms 10 runs

Summary
'rg Unix ~/parvus/prog/tcl' ran 13.51 ± 1.02 times faster than 'grep -r Unix ~/parvus/prog/tcl'
What makes rg fast?

Uses parallelism

Uses Rust regex engine with finite automata, SIMD, & aggressive literal optimizations (regular expressions follow the syntax of `grep -E`; see this & this for details)

By default, ignores files in .gitignore & similar, hidden files/directories, & binary files (this behaviour can be changed with flags)
fd
(Fast & friendly alternative to find)

```
08:38 m ~/parvus/prog/tdoc fd -e jl
dcl/ccl/getting_help/script.jl
dcl/ccl/getting_help/script_fixed.jl
dj1/cjl/covid/covid.jl
dj1/cjl/covid/covid_old.jl
dj1/cjl/dagger/benchmark.jl
dj1/cjl/dagger/dagger_oom.jl
dj1/cjl/dagger/dagger_threads.jl
dj1/cjl/flux/flux.jl
dj1/cjl/intro/distributed.jl
dj1/cjl/intro/intro.jl
dj1/cjl/intro/live_code.jl
dj1/cjl/intro/pi.jl
dj1/cjl/intro/pi_parallel.jl
dj1/cjl/intro/plots.jl
dj1/cjl/intro/plots_bkup.jl
dj1/cjl/intro/psort.jl
dj1/cjl/intro/sort.jl
dj1/cjl/intro/variables_types.jl
dj1/cjl/jupyter/covid_jupyterdays.jl
dj1/cjl/jupyter/hello.jl
dj1/cjl/jupyter/literate_cleaned.jl
dj1/cjl/jupyter/literate_example.jl
dj1/cjl/jupyter/nbinclude.jl
```
fd has an intuitive syntax & is super fast
pass

(A command line password manager for Unix systems)
pass is a command line password manager using GnuPG for encryption

pass has a Git integration & several extensions

pass is easy to use in scripts or combined with other utilities
Main pass commands

- `pass init your-gpg-id`  # Initialize password store
- `pass insert somesite.ca/your@email`  # Enter password for a site
- `pass somesite.ca/your@email`  # Print password
- `pass -c somesite.ca/your@email`  # Copy password to clipboard
- `pass edit somesite.ca/your@email`  # Edit password
- `pass generate othersite.com/userid 12`  # Generate password of size 12
- `pass generate -c othersite.com/userid 12`  # Same, but copy to clipboard
- `pass generate -n othersite.com/userid 12`  # Same, without special characters
- `pass rm othersite.com/userid`  # Delete password
- `pass`  # Print content of password store
TRAMP

(“A remote file editing package for Emacs”)
The TRAMP (Transparent Remote Access, Multiple Protocol) Emacs package allows to edit remote files as if they were local.

You can thus use your init file & usual packages to write files on a remote machine or for another user (e.g. root).

You can also use the Emacs GUI to edit remote files without efficiency loss.

It is a convenient alternative to using Emacs on a remote machine after you have sshed into it or to using sudo emacs to edit root files.
Using TRAMP from within Emacs

Remote file
C-x C-f /ssh:user@server:/home/user/path/to/file RET

Root
C-x C-f /sudo::/path/to/file RET
Using TRAMP from the command line

Remote file

```
emacs /ssh:user@server:/home/user/path/to/file
```

Root

```
emacs /sudo::/path/to/file
```
Helm

(“Incremental completion & selection narrowing framework”)

Helm is an incremental completion & selection narrowing framework.

Helm provides a powerful and flexible completion mechanism that allows for interactive and efficient code editing.

The Helm package provides a wide range of customization options, allowing users to tailor the completion experience to their specific needs.

Helm is widely used in the Emacs environment, where it is a highly regarded tool for code completion and selection narrowing.

With Helm, users can quickly jump to the next keyword in a function, navigate through completions, and select the desired item with ease.

Helm's vast library of packages and integrations makes it a versatile tool for a wide range of programming tasks.

In summary, Helm is a must-have tool for anyone looking to enhance their code editing experience in Emacs.
The Helm package provides an incremental completion & narrowing selection API for search applications in Emacs

Virtually every Emacs search application has built-in integration with Helm
Grep equivalent
Navigate mark ring
Navigate kill ring
Navigate open & recent files
Navigate recent files
Find file
Search objects in working environment
Search in Git repo
Manage major/minor modes

<table>
<thead>
<tr>
<th>Major mode</th>
<th>Minor mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage</td>
<td>Manage</td>
</tr>
<tr>
<td>major</td>
<td>minor</td>
</tr>
<tr>
<td>mode</td>
<td>modes</td>
</tr>
</tbody>
</table>

```
(defun my-major-mode (func arg)
  "Function to manage major mode.
  \_func\_ arg\_ arg\_
"
)
```

- **search email**
  - (global-set-key (kbd "C-h l") "helm-locate")
- **search help**
  - (global-set-key (kbd "C-h m") "helm-help")
- **goto the code of the function at point**
  - (global-set-key (kbd "C-h i") "helm-info")
- **function description**
  - (global-set-key (kbd "C-h a") "helm-apropos")
- **mode**
  - (global-set-key (kbd "C-c f") "describe-mode")
  - (global-set-key (kbd "C-c n") "describe-mode")
  - (global-set-key (kbd "C-c a") "describe-mode")
- **helm mode manager**
  - (global-set-key (kbd "C-c s") "helm-switch-major-mode")
  - (global-set-key (kbd "C-c e") "helm-enable-minor-mode")
  - (global-set-key (kbd "C-c d") "helm-disable-minor-mode")
- **keybinding**
  - (global-set-key (kbd "C-c b") "helm-descbinds")
- **package**
  - (global-set-key (kbd "C-c u") "helm-up")

```
(global-set-key (kbd "C-c u") "helm-up")
```

---

```
(defun my-minor-mode (func arg)
  "Function to manage minor mode.
  \_func\_ arg\_ arg\_
"
)
```
Search help files
## Search key bindings

### Major Mode Bindings:
- **C-x**: Prefix Command
- **ESC**: Prefix Command
- **DEL**: backward-delete-char-untabify
- **C-x C-a**: Prefix Command
- **M-q**: indent-pp-saving
- **C-M-x**: eval-defun
- **C-x C-a C-c**: debug-go-mode
- **C-x C-a C-1**: debug-where
- **C-x C-a RET**: debug-set-initial-mode
- **C-x C-a C-n**: debug-next-mode
- **C-x C-a C-s**: debug-step-mode

### 'tracking-mode' Minor Mode Bindings:
- **C-c**: Prefix Command
- **C-c C-B**: tracking-next-buffer
- **C-c C-SPC**: tracking-next-buffer

### 'flyspell-mode' Minor Mode Bindings:
- **C-c**: Prefix Command
- **C-c**: flyspell-goto-next-error
- **C-c**: flyspell-auto-correct-word
- **<ignore>**: flyspell-auto-correct-previous-word
- **C-c $**: flyspell-correct-word-before-point

### 'outshine-mode' Minor Mode Bindings:
- **TAB**: outshine-kbd-TAB
- **ESC**: Prefix Command
- **<M-$-down>**: outshine-kbd-M-$<down>
- **<M-$-left>**: outshine-kbd-M-$<left>
- **<M-$-right>**: outshine-kbd-M-$<right>
- **<M-$-up>**: outshine-kbd-M-$<up>
- **<M-down>**: outline-next-visible-heading
- **<M-left>**: outline-kbd-M<left>
- **<M-right>**: outline-kbd-M<right>
- **<M-up>**: outline-previous-visible-heading
- **<backtab>**: outline-kbd<backtab>
- **<remap>**: Prefix Command
- **C-M-1**: outshine-cycle-buffer
- **M-RET**: outshine-kbd-M-RET
- **<remap> <self-insert-command>**: outshine-self-insert-command

### 'outline-minor-mode' Minor Mode Bindings:
- **C-c**: Prefix Command
- **ESC**: Prefix Command
- **;**: Prefix Command
- **C-;**: org-cycle
- **M-n**: outline-next-visible-heading
- **M-p**: outline-previous-visible-heading
- **; [**: outline-cycle-buffer
- **; [** Candidate(s): C-h e: Help Tab: Act #12/7173/27/N/A ACT C-1:Log suspend C-h c:Conf pattern:
<table>
<thead>
<tr>
<th>Date</th>
<th>Subject</th>
<th>From</th>
</tr>
</thead>
<tbody>
<tr>
<td>22-02-23</td>
<td>Mountain Conditions Re... <a href="mailto:mhburle@gmail.com">mhburle@gmail.com</a></td>
<td><a href="mailto:mhburle@gmail.com">mhburle@gmail.com</a></td>
</tr>
<tr>
<td>22-02-04</td>
<td>Mountain Conditions Re... <a href="mailto:mhburle@gmail.com">mhburle@gmail.com</a></td>
<td><a href="mailto:mhburle@gmail.com">mhburle@gmail.com</a></td>
</tr>
<tr>
<td>22-04-04</td>
<td>Mountain Conditions Re... <a href="mailto:mhburle@gmail.com">mhburle@gmail.com</a></td>
<td><a href="mailto:mhburle@gmail.com">mhburle@gmail.com</a></td>
</tr>
<tr>
<td>27-05-08</td>
<td>Mountain Conditions Re... <a href="mailto:mhburle@gmail.com">mhburle@gmail.com</a></td>
<td><a href="mailto:mhburle@gmail.com">mhburle@gmail.com</a></td>
</tr>
<tr>
<td>28-11-12</td>
<td>Mountain Conditions Re... <a href="mailto:mhburle@gmail.com">mhburle@gmail.com</a></td>
<td><a href="mailto:mhburle@gmail.com">mhburle@gmail.com</a></td>
</tr>
<tr>
<td>28-11-12</td>
<td>Mountain Conditions Re... <a href="mailto:mhburle@gmail.com">mhburle@gmail.com</a></td>
<td><a href="mailto:mhburle@gmail.com">mhburle@gmail.com</a></td>
</tr>
</tbody>
</table>

Search in emails with `mu` command.
**Select colour**

<table>
<thead>
<tr>
<th>Colour</th>
<th>RGB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black</td>
<td>#000000</td>
</tr>
<tr>
<td>Dim gray</td>
<td>#666666</td>
</tr>
<tr>
<td>Dark gray</td>
<td>#333333</td>
</tr>
<tr>
<td>Gray</td>
<td>#AAAAAA</td>
</tr>
<tr>
<td>Light gray</td>
<td>#BBBBBB</td>
</tr>
<tr>
<td>WhiteSmoke</td>
<td>#DDDDDD</td>
</tr>
<tr>
<td>White</td>
<td>#FFFFFF</td>
</tr>
<tr>
<td>Red</td>
<td>#FF0000</td>
</tr>
<tr>
<td>OrangeRed</td>
<td>#FF8C00</td>
</tr>
<tr>
<td>Orange</td>
<td>#FFA07A</td>
</tr>
<tr>
<td>Olive</td>
<td>#808000</td>
</tr>
<tr>
<td>Yellow</td>
<td>#FFFF00</td>
</tr>
<tr>
<td>Chartreuse</td>
<td>#CCFFCC</td>
</tr>
<tr>
<td>Lawn Green</td>
<td>#ADD8E6</td>
</tr>
<tr>
<td>Green</td>
<td>#00FF00</td>
</tr>
<tr>
<td>Spring Green</td>
<td>#9ACD32</td>
</tr>
<tr>
<td>Cyan</td>
<td>#00FFFF</td>
</tr>
<tr>
<td>Medium Blue</td>
<td>#B0B0B0</td>
</tr>
<tr>
<td>Dark violet</td>
<td>#800080</td>
</tr>
<tr>
<td>Dark magenta</td>
<td>#8B0000</td>
</tr>
<tr>
<td>Magenta</td>
<td>#DC143C</td>
</tr>
<tr>
<td>Dark Red</td>
<td>#8B0000</td>
</tr>
<tr>
<td>Brown</td>
<td>#9932CC</td>
</tr>
<tr>
<td>Firebrick</td>
<td>#A52A2A</td>
</tr>
<tr>
<td>Indian Red</td>
<td>#BC006B</td>
</tr>
<tr>
<td>Light Coral</td>
<td>#FFD700</td>
</tr>
<tr>
<td>Salmon</td>
<td>#FF00FF</td>
</tr>
<tr>
<td>Light Salmon</td>
<td>#B03060</td>
</tr>
<tr>
<td>Tomato</td>
<td>#E9967A</td>
</tr>
<tr>
<td>Coral</td>
<td>#FFF5EE</td>
</tr>
<tr>
<td>Dark Salmon</td>
<td>#8B008B</td>
</tr>
<tr>
<td>Rosy Brown</td>
<td>#E38868</td>
</tr>
<tr>
<td>Siena</td>
<td>#8A2BE2</td>
</tr>
<tr>
<td>Saddle Brown</td>
<td>#8B4513</td>
</tr>
<tr>
<td>Chocolate</td>
<td>#8B4513</td>
</tr>
<tr>
<td>Peru</td>
<td>#E9967A</td>
</tr>
<tr>
<td>Sandy Brown</td>
<td>#E2B88C</td>
</tr>
<tr>
<td>Light Grey</td>
<td>#B8B8B8</td>
</tr>
<tr>
<td>Navajo White</td>
<td>#C71585</td>
</tr>
<tr>
<td>Olive</td>
<td>#808000</td>
</tr>
<tr>
<td>Dark Goldenrod</td>
<td>#B3E561</td>
</tr>
<tr>
<td>Goldenrod</td>
<td>#DA70D6</td>
</tr>
<tr>
<td>Light Goldenrod</td>
<td>#EEDDD2</td>
</tr>
</tbody>
</table>

**Global-key bindings**

- `C-c C-c` : `helm-switch-major-mode`
- `C-c C-o` : `helm-enable-minor-mode`
- `C-c C-d` : `helm-disable-minor-mode`
- `C-c C-b` : `helm-deschinds`
- `C-c C-f` : `helm-system-packages`
- `C-c C-p` : `helm-current-prefix-arg--recursive`
- `C-c C-A` : `helm-system-packages`
- `C-c C-g` : `undo-tree`

**Search web engines with surfraw**

- `C-x C-f` : `surfraw`
- `C-c C-s` : `surfraw`

** contemplated**

- `C-c C-s` : `surfraw`

**Colors**

- Black
- Dim gray
- Dark gray
- Gray
- Light gray
- WhiteSmoke
- White
- Red
- OrangeRed
- Orange
- Olive
- Yellow
- Chartreuse
- Lawn Green
- Green
- Spring Green
- Cyan
- Medium Blue
- Dark Violet
- Dark magenta
- Magenta
- Dark Red
- Brown
- Firebrick
- Indian Red
- Light Coral
- Salmon
- Light Salmon
- Tomato
- Coral
- Dark Salmon
- Rosy Brown
- Siena
- Saddle Brown
- Chocolate
- Peru
- Sandy Brown
- Light Grey
- Navajo White
- Olive
- Dark Goldenrod
- Goldenrod
- Light Goldenrod
Select emoji
Questions?