Contents

1 emacs-organizer

Some code should be executing before tangling and evaluating this file. So, look at init.el

1.1 layout

I use colemak keyboard layout see layout.org file – link

1.2 general emacs

1.2.1 package management

Add 4 main package archives

```
(require 'package)
(add-to-list 'package-archives '("gnu" . "https://elpa.gnu.org/packages/") t)
(add-to-list 'package-archives '("melpa" . "https://melpa.org/packages/") t)
(add-to-list 'package-archives '("melpa-stable" . "https://stable.melpa.org/packages/") t
(add-to-list 'package-archives '("nongnu" . "https://elpa.nongnu.org/nongnu/") t)
```

Set priorites for package-archives

```
(setq package-archive-priorities
  '(("gnu" . 40)
    ("nongnu" . 30)
    ("melpa-stable" . 20)
    ("melpa" . 10)))
```

1.2.2 window management

If frame is devided by top and bottom, change it to left and right. Transpose operation like with matrices and tables.

```
(use-package transpose-frame :ensure t)
```

Enable tab-bar-mode

```
(tab-bar-mode)
```

1.2.3 autocomplete

Press C-M-i to activate emacs autocomplete.

Frankly speaking, I copy this snippet from the official vertico docs:
`https://github.com/minad/vertico?tab=readme-ov-file#completion-at-point-and-completion`

This configuration give my ability to perform fuzzy search. Like with dmenu, rofi or fzf, but in emacs.

```
(use-package vertico
  :init (vertico-mode)
   (setq completion-in-region-function
            (lambda (&rest args)
              (apply (if vertico-mode
                         #'consult-completion-in-region
                       #'completion--in-region)
                     args)))
   :ensure t)
```

```
(use-package consult :ensure t)

(use-package consult-eglot :ensure t)
```

1.2.4 Force UTF-8 and LF line endings

This should be executed before loading this file, so this forms also present in init.el.

```
(defvar *fs-encoding* 'utf-8)
(prefer-coding-system 'utf-8-unix)
```

1.2.5 tool-bar

Show both icons and caption

```
(setq tool-bar-style 'both)
```

use emacs icons instead of gtk ones

```
(advice-add #'x-gtk-map-stock :override #'ignore)
```

1.2.6 y-or-n

```
(defalias 'yes-or-no #'y-or-n-p)
```

1.2.7 modus-themes

Enable only in graphical mode.

```
(use-package modus-themes :ensure t)

(when (display-graphic-p)
  (load-theme 'modus-operandi t)
  (enable-theme 'modus-operandi))
```

1.3 config by use cases

I structured my config by use cases I apply emacs in.

1.3.1 file manager

I use build in dired for now.

Copy, move, rename files across panes, like two-panel file manager

```
(setq dired-dwim-target t)
```

1.3.2 git client

```
(use-package magit :ensure t)

(use-package git-modes :ensure t)
```

1.3.3 organizer

```
(use-package howm :config
  (setq howm-view-use-grep t)
  :ensure t)
```

Function to add prop-line, so I can use howm with any other major mode, with org-mode for examlpe

```
    (defun howm-insert-prop-line ()
"Activate major mode and modify the file so that this mode is activated
automatically the next time it is opened"
    (interactive)
    (howm-mode)
    (let*
        ((modes (mapcar #'cdr auto-mode-alist))
         (mode-name (completing-read "Choose major mode: " modes))
         (mode (intern-soft mode-name)))
      (unless (or (null mode)
          (eq mode major-mode))
        (funcall mode)
        (howm-mode)
        (add-file-local-variable-prop-line
         'mode (intern (string-trim-right mode-name "-mode\\'"))))))
```

1.3.4 flascards

```
(use-package anki-editor :ensure t)
```

1.3.5 todo list and pomodoro

```
(setq org-todo-keywords
    '((sequence "TODO" "|" "DONE" "FAIL" "NGMI" )))
```

1. Work arounds Use C locale for time on windows for org-pomodoro

   ```
   (when (eq system-type 'windows-nt)
   (setq system-time-locale "C"))
   ```

1.3.6 xelatex editor

```
(use-package auctex :ensure t)
```

I write my coursework in xelatex.

```
(setq-default TeX-engine 'xetex)
```

From auctex info:

```
(setq TeX-auto-save t)
(setq TeX-parse-self t)
(setq-default TeX-master nil)
```

1.3.7 code editor

1. python

   ```
   (use-package pyvenv :ensure t)

   (use-package elpy :ensure t)
   ```

2. common lisp

   ```
   (use-package slime :ensure t)

   (setq inferior-lisp-program "sbcl")
   ```

3. EditorConfig

   ```
   (use-package editorconfig :ensure t)
   ```

4. assembly and compiler exploration Compiler explorer

   ```
   (use-package rmsbolt :ensure t)
   ```

   Assembly

   ```
   (use-package nasm-mode :ensure t)
   ```

5. data and config files Systemd units

   ```
   (use-package systemd :ensure t)
   ```

   Comma separated values

   ```
   (use-package csv :ensure t)
   ```

1.3.8 terminal

```
(use-package eat
  :config
  (setq eat-kill-buffer-on-exit t)
  (setq eat-enable-mouse t)
  :ensure t)
```

1.3.9 rss reader

Elfeed in my config is interconneted with howm.

1. elfeed use-package:

   ```
   (use-package elfeed
   :ensure t
   :config
   (setq elfeed-db-directory "~/howm/.elfeed")
     (setq elfeed-curl-program-name "curl"))
   (use-package elfeed-protocol)
   ```

2. elfeed-org use-package

```
(use-package elfeed-org
  :ensure t
  :config
  (elfeed-org)
  :after howm)
```

3. functions for interconnecting with howm

```
(defun my-elfeed-file-names-in-howm ()
  "Return list of absolute filenames of org-elfeed files in howm"
  (delete-dups
   (mapcar #'car (howm-grep "\:elfeed\:"
                      (howm-files-in-directory howm-directory)))))
```

4. advices for executing functions

```
(define-advice elfeed (:before (&rest _args))
  (setq rmh-elfeed-org-files (my-elfeed-file-names-in-howm)))


(define-advice elfeed-update (:before (&rest _args))
  (setq rmh-elfeed-org-files (my-elfeed-file-names-in-howm)))
```

1.3.10 email client

```
(setq
   user-full-name "Корякин Артём"
   user-mail-address "karakin2000@gmail.com"
   send-mail-function 'smtpmail-send-it
   smtpmail-smtp-server "smtp.gmail.com"
   smtpmail-stream-type 'starttls ;; was nil (upgrade with STARTTLS if possible)
   smtpmail-smtp-service 587
   smtpmail-servers-requiring-authorization "*"
   gnus-save-score t
   gnus-startup-file "~/howm/.newsrc"
   gnus-backup-startup-file 'never
   gnus-select-method
   '(nnimap "gmail"
            (nnimap-address "imap.gmail.com")
```

```
        (nnmail-expiry-target "nnimap+gmail:[Gmail]/Корзина")
        (nnimap-server-port 993)
        (nnimap-stream ssl)
        (gnus-search-engine gnus-search-imap)
        (nnmail-expiry-wait 5)))
```

1.3.11 epub reader

```
(use-package nov :ensure t)
```