CC Mode

CC mode is the most common *Emacs* mode for C/C++ programmers.

Google Style

See google-c-style.el

Linux Style

The *Emacs* built-in style “linux” is not strictly conforming to the *Linux* kernel style specified in document kernel-doc-nano-HOWTO.txt. Therefore i created a new style “rho/linux” which is based on style “linux”.[1)]

```
;;;;;;;; Linux kernel style

;; NOTE: This is replacement for the `linux' style, because `linux' style
does
;;       not change `indent-tabs-mode' and does not have the correct comment
;;       style (which is `gtkdoc' in Emacs, as default).
(defconst rho/linux-style
  '("linux"                             ; base style
    (indent-tabs-mode . t)
    (tab-width . 8)
    (glasses-mode . nil)                ; handled in `c-set-style' advice
    (comment-column . 0)                ; left aligned
    (c-comment-only-line-offset . (0 . 0)) ; always indent syntactically
    (c-backslash-column . 0)
    (c-doc-comment-style . rho/kernel) ; see `rho/kernel-font-lock-doc-
comments'
    (c-offsets-alist
     ;; Because Linux style requires to line-up only by using tabs (also for
     ;; argument continuation), function `c-lineup-arglist' in "linux" style
     ;; has to be replaced by `rho/c-lineup-arglist-tabs-only'.
     . ((arglist-cont-nonempty . (c-lineup-gcc-asm-reg rho/c-lineup-arglist-
tabs-only))))))

;; see kernel-doc-nano-HOWTO.txt or the following link:
;; https://www.kernel.org/doc/html/latest/doc-guide/kernel-doc.html
(defconst rho/kernel-font-lock-doc-comments
  (let ((header "^ \\* "))
    `((,(concat header "\\(" rho/c-symbol-regexp "\\):\\s-") ; section:
(description)?
       1 ,c-doc-markup-face-name prepend nil)
      (,(concat rho/c-symbol-regexp "()") ; function
```

```
       0 ,c-doc-markup-face-name prepend nil)
      (,(concat header "\\(" "@" rho/c-symbol-regexp "\\)\\s-*:\\s-") ;
@parameterx
       1 ,c-doc-markup-face-name prepend nil)
      (,(concat "[%@$]" rho/c-symbol-regexp) ; %SYMBOL, @symbol, $SYMBOL
       0 ,c-doc-markup-face-name prepend nil)
      (,(concat "&" rho/c-symbol-regexp "\\s-*\\(\\(\\.\\|->\\)\\s-*" rho/c-
symbol-regexp "\\)*") ; &symbol->a, &symbol.b
       0 ,c-doc-markup-face-name prepend nil)
      (,(concat "&\\(\\(enum\\|struct\\|typedef\\)\\s-\\)*" rho/c-symbol-
regexp) ; &enum symbol, &struct symbol, &typedef symbol
       0 ,c-doc-markup-face-name prepend nil)
    (,rho/prog-mode-comment-codetags-regexp ; FIXME, TODO, ...
       0 ,font-lock-warning-face prepend nil)))
    "Linux kernel style comment markup.")

;; kernel-doc-nano-HOWTO.txt: Inside a struct description, you can use the
;; "private:" and "public:" comment tags.  Structure fields that are inside
;; a "private:" area are not listed in the generated output documentation.
;; The "private:" and "public:" tags must begin immediately following a "/*"
;; comment marker.  They may optionally include comments between the ":" and
;; the ending "*/" marker.
;; Example:
;; /**
;;  * struct my_struct - short description
;;  * @a: first member
;;  * @b: second member
;;  *
;;  * Longer description
;;  */
;; struct my_struct {
;;     int a;
;;     int b;
;; /* private: internal use only */
;;     int c;
;; };
(defconst rho/kernel-font-lock-tags-protection "\\(private\\|public\\):")

(defconst rho/kernel-font-lock-doc-protection
  `((,rho/kernel-font-lock-tags-protection
     1 c-doc-markup-face-name prepend nil)))

(defconst rho/kernel-font-lock-keywords
  `((,(lambda (limit)
        ;; NOTE: (defun c-font-lock-doc-comments (prefix limit keywords)
        ;; Fontify the comments between the point and LIMIT whose start
        ;; matches PREFIX with `c-doc-face-name'.  Assumes comments have
        ;; been fontified with `font-lock-comment-face' already.
        (c-font-lock-doc-comments
            "/\\*\\*$"                  ; /**
            limit
```

```
          rho/kernel-font-lock-doc-comments)
        (c-font-lock-doc-comments
           (concat "/\\* " rho/kernel-font-lock-tags-protection ".*\\*/")
          limit
          rho/kernel-font-lock-doc-protection)
        ))))

(defun rho/c-lineup-arglist-tabs-only (ignored)
  "Line up argument lists by tabs, not spaces."
  (let* ((anchor (c-langelem-pos c-syntactic-element))
         (column (c-langelem-2nd-pos c-syntactic-element))
         (offset (- (1+ column) anchor))
         (steps (floor offset c-basic-offset)))
    (* (max steps 1) c-basic-offset)))


(setq c-doc-comment-style 'set-from-style)
(c-add-style "rho/linux" rho/linux-style)
```

1)

Including a new comment style “rho/kernel” in favor of doc-style “gtk-doc”.

From:
https://wiki.rho62.de/ - **rho62 Wiki**

Permanent link:
https://wiki.rho62.de/doku.php?id=development:emacs:cc-mode

Last update: **2023/05/22 08:23**