Neovim (nvim)

Preparing

- 설치

```
# Ubuntu (debian 계열)
$ sudo apt-get install software-properties-common    # apt repository 관리 툴 제공하는 software-properties-common 설치

$ sudo add-apt-repository ppa:neovim-ppa/stable  # repository 추가
$ sudo apt-get update  # update
$ sudo apt-get install neovim  # neovim 설치

# macos
$ brew install nvim

# windows
> choco install neovim
```

- 설정파일; ~/.config/nvim/init.vim @linux, ~\AppData\Local\nvim\init.vim @Windows
  - init.vim에서 .vimrc 파일 참조

```
set runtimepath^=~/.vim runtimepath+=~/.vim/after
let &packpath = &runtimepath
source ~/.vimrc
```

- .bashrc, .zshrc alias vim="nvim"

Plugins

- vim-plug; plugin 사용

```
# linux
$ sh -c 'curl -fLo "${XDG_DATA_HOME:-$HOME/.local/share}"/nvim/site/autoload/plug.vim --create-dirs \
https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim'

# 또는
$ curl -fLo ~/.local/share/nvim/site/autoload/plug.vim --create-dirs \
    https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim
# windows
> iwr -useb https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim |`
```

```
    ni "$(@($env:XDG_DATA_HOME, $env:LOCALAPPDATA)[$null -eq
$env:XDG_DATA_HOME])/nvim-data/site/autoload/plug.vim" -Force
```

- NERDTree; tree 구조 `Plug 'preservim/nerdtree'`
- Tagbar; 현재 버퍼의 class, struct, prototype, typedef, macro 등 요약. `Plug 'preservim/tagbar'`
- vim-airline; 버퍼 정보 상세 표시 `Plug 'vim-airline/vim-airline'`
- ctrlp; 파일 탐색기
- CoC(Conquer of Completion); 자동완성(auto completion) intellisense, LSP(Language Server Protocol) 지원.
    1. node.js 설치

```
$ curl -sL install-node.now.sh/lts | sudo $SHELL

# yarn 설치
$ curl -sS https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key
add -
$ echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo tee
/etc/apt/sources.list.d/yarn.list
$ sudo apt-get update && sudo apt-get install yarn
```

    2. vim-plug 등록 `Plug 'neoclide/coc.nvim', {'branch': 'release'}`
- LSP(Language Server Protocol); `:CocIntsall <LSP 서버명>` Conquer of Completion @github.com CoC Language Servers @github.com
- nvim-treesitter; syntax highlight `Plug 'nvim-treesitter/nvim-treesitter', {'do': ':TSUpdate'}`
    - `:TSInstall <사용하는 언어>` nvim-treesitter#supported-languages @github.com

References

example) init.vim

```
" ================================================================================
"= 플러그인 설정                          =
" ================================================================================
call plug#begin('~/.vim/plugged') " 플러그인 시작

" Conquer Of Completion 자동완성 플러그인
Plug 'neoclide/coc.nvim', {'branch': 'release'}

" nvim-treesitter 구문 파싱 하이라이팅
Plug 'nvim-treesitter/nvim-treesitter', {'do': ':TSUpdate'}

" Tagbar 코드 뷰어 창
" Plug 'majutsushi/tagbar'
Plug 'preservim/tagbar'
```

```
" NERDTree 코드 뷰어 창
Plug 'preservim/nerdtree'

" 컬러스킴(색상표) jellybeans, gruvbox
Plug 'nanotech/jellybeans.vim'
" Plug 'morhetz/gruvbox'

" 하단에 다양한 상태(몇 번째 줄, 인코딩, etc.)를
" 표시하는 상태바 추가
Plug 'vim-airline/vim-airline'
Plug 'vim-airline/vim-airline-themes'

" CScope 플러그인
Plug 'ronakg/quickr-cscope.vim'

" CtrlP 파일 탐색 플러그인
Plug 'ctrlpvim/ctrlp.vim'

" 비활성 윈도우 강조
" Plug 'blueyed/vim-diminactive'

" vim cutlass 잘라내기 명령어가 yank 에 영향을 주지 않음
Plug 'svermeulen/vim-cutlass'

" VIM GAS(GNU ASsembler) Highlighting
Plug 'Shirk/vim-gas'

call plug#end()
" ================================================================================
"= 단축키 지정                          =
" =  n(normal mode) 명령 모드                  =
" =  v(visual, select mode) 비주얼 모드             =
" =  x(visual mode only) 비주얼 모드               =
" =  s(select mode only) 선택 모드                =
" =  i(insert mode) 편집 모드                  =
" =  t(terminal mode) 편집 모드                  =
" =  c(commnad-line) 모드                    =
" =  re(recursive) 맵핑                     =
" =  nore(no recursive) 맵핑                   =
" ================================================================================
" ------------------------------------
" 편집 모드
" ------------------------------------
" jk 와 kj 를 <ESC> 키로 맵핑
inoremap jk <ESC>
inoremap kj <ESC>
" ------------------------------------
" 명령 모드
" ------------------------------------
" <F1> 을 통해 NERDTree 와 Tagbar 열기
```

```
nnoremap <silent><F1> :NERDTreeToggle<CR><bar>:TagbarToggle <CR>

" <Ctrl + h, l> 를 눌러서 이전, 다음 탭으로 이동
nnoremap <silent><C-j> :tabprevious<CR>
nnoremap <silent><C-k> :tabnext<CR>

" <Ctrl + j, k> 를 눌러서 이전, 다음 버퍼로 전환
nnoremap <silent><C-h> :bp<CR>
nnoremap <silent><C-l> :bn<CR>

" <Shift + h, l> 를 눌러서 현재 버퍼 삭제
nnoremap <silent><S-h> :bp<bar>sp<bar>bn<bar>bd<CR>
nnoremap <silent><S-l> :bp<bar>sp<bar>bn<bar>bd<CR>

" <Ctrl + w> t 를 눌러서 커서를 NERDTree 로 옮기기
nnoremap <silent><C-w>t :NERDTreeFocus<CR>

" 우측 하단(botright)에 창 생성(new), 해당 창을 terminal 로 변경
" 크기를 10 으로 재설정(resize) 후 창 높이를 고정(winfixheight)시킴
" 줄번호는 삭제하고, 터미널 디렉터리 글자색을 변경
nnoremap <silent><F2>
    \:botright new<CR><bar>
    \:terminal<CR><bar><ESC>
    \:resize 10<CR><bar>
    \:set winfixheight<CR><bar>
    \:set nonu<CR><bar>
    \iLS_COLORS=$LS_COLORS:'di=1;33:ln=36'<CR>
" ------------------------------------
" 터미널 모드
" ------------------------------------
" 터미널 모드에서 <Ctrl + w> 누르면 명령 모드로 전환하고 <Ctrl + w> 입력
tmap <silent><C-w> <ESC><C-w>

" jk 혹은 kj 를 누르면 <ESC> 를 실행
tmap <silent>jk <ESC>
tmap <silent>kj <ESC>

" <ESC> 입력 시 <C-\><C-n> 실행 => 터미널 모드에서 기본 모드로 전환
tnoremap <silent><ESC> <C-\><C-n>

" ------------------------------------
" 명령, 비주얼 모드
" ------------------------------------
" iamroot 자동 주석
map <F9> <ESC>o/*<CR> * IAMROOT, <C-R>=strftime("%Y.%m.%d")<CR>
    \: <CR>*/<CR><ESC><UP><UP><END>
" ============================================================================
" =  vim 설정                          =
" ============================================================================
" 탭 정지 = 8 칸마다
```

```
set tabstop=8
" 쉬프트 (<< 혹은 >>) 이동거리 8 칸
set shiftwidth=8

" 줄 번호를 표시한다.
set number

" 괄호 짝을 강조한다.
set showmatch

" 하위 디렉터리를 모두 path 에 추가한다.
" gf 명령어 사용 시 파일을 인식 가능
set path+=**

" 탐색 문자열 강조
set hlsearch

" 항상 상단에 탭 라인을 출력한다.
set showtabline=2

" 행 표시선 출력
set colorcolumn=80

if has('nvim')             " nvim 을 사용 중이라면
    set inccommand=nosplit    " nvim live %s substitute (실시간 강조)
endif

" vim 과 OS 의 클립보드 동기화
set clipboard=unnamedplus

" GUI-Color 를 사용 가능하도록 설정 (TrueColor)
" cterm 혹은 term 대신 gui 를 통해 색상을 설정할 수 있고
" 16,777,216 종류의 색상 표현 가능(기존 256)
set termguicolors

" 모든 마우스 기능을 사용
set mouse=a

" mkview 명령어가 저장하는 요소 중
" 하나인 `options` 를 제거
set viewoptions-=options

" 문법이 존재하면
if has("syntax")
  " 문법 강조를 수행
    syntax on
endif

" 컬러스킴(문법 강조 색상) - 현재 jellybeans
colorschem jellybeans
```

```
" colorschem gruvbox

" ============================================================================
"= 하이라이트 정의                          =
" ============================================================================
" 버퍼(창)과 버퍼의 끝(창의 끝)을 투명하게
highlight Normal guibg=NONE
highlight EndOfBuffer guibg=NONE

" 줄번호 배경색은 투명(NULL)하게,
" 글자는 굵게(bold), 글자색은 하얗게(White)
highlight LineNr guibg=NONE gui=bold guifg=white

" 행 표시선 색상
highlight ColorColumn guibg=White
" ============================================================================
"= 함수 정의                              =
" ============================================================================
" tabsize 를 size 로 변경
function SetTab(size)
    execute "set shiftwidth=".a:size
    execute "set tabstop=".a:size
    execute "set softtabstop=".a:size
endfunction
" ============================================================================
"= 자동 실행 (autocmd)                                                      =
" ============================================================================
" terminal buffer 에 진입했을 때 mode 를 normal 에서 terminal 모드로 변경
" 또한 줄번호를 없앤다.
autocmd BufEnter term://* start " do nothing
autocmd TermOpen term://* execute ":set nonu"

" 파일 명이 *.S 로 시작하면 GAS 문법 강조 사용
autocmd BufRead,BufNew *.S execute ":set ft=gas"

" 버퍼를 저장할때 파일 이름이 .c, .h 와 같다면 ctags 명령어를 실행
" autocmd BufWritePost *.c,*.h silent! !ctags -R &

" 윈도우를 나갈 때 뷰를 저장하고,
autocmd BufWinLeave *.c,*.h mkview

" 윈도우에 들어갈 땐 뷰를 로드한다. (커서위치 저장)
" silent! 는 loadview 중 발생하는 에러를 억압(suppress) 한다.
autocmd BufWinEnter *.c,*.h silent! loadview

" 활성화된 버퍼만 라인 번호 표시 (단, 확장자는 .c 혹은 .h 일때만 동작)
autocmd BufEnter * if (&filetype == 'c' || &filetype == 'cpp')
    \| set number
\| endif
```

```
" 버퍼에서 나갈 땐 줄 번호를 지운다.
autocmd BufLeave * if (&filetype == 'c' || &filetype == 'cpp')
    \| set nonumber
\| endif
" ============================================================================
" = 플러그인 설정                          =
" ============================================================================
" ----------------------------------
" coc 설정
" ----------------------------------
"  nvim 버전이 0.5.0 이상이며, 패치가 8.1.1564 이상이라면
if has("nvim-0.5.0") || has("patch-8.1.1564")
 " 사인(sign column) 열을 숫자 열과 합침
  set signcolumn=number
endif

" <Tab> 을 눌러서 현재 지시자를 옮김.
" inoremap <silent><expr> <TAB>
      "\ pumvisible() ? "\<C-n>" :
      "\ <SID>check_back_space() ? "\<TAB>" :
      "\ coc#refresh()
"inoremap <expr><S-TAB> pumvisible() ? "\<C-p>" : "\<C-h>"

" <Backspace> 키가 지시자 제거, 기존 자동완성 양식 폐기
function! s:check_back_space() abort
  let col = col('.') - 1
  return !col || getline('.')[col - 1]  =~# '\s'
endfunction

" <Ctrl + Space> 를 눌러서 자동완성 적용
if has('nvim')
  inoremap <silent><expr> <c-space> coc#refresh()
else
  inoremap <silent><expr> <c-@> coc#refresh()
endif

" 코드 탐색 단축키
nmap <silent> gr <Plug>(coc-references)

" 커서 아래의 토큰을 강조
autocmd CursorHold * silent call CocActionAsync('highlight')
" ----------------------------------
" nvim-treesitter 설정
" ----------------------------------
lua <<EOF
require'nvim-treesitter.configs'.setup {
  ensure_installed = "maintained",
  ignore_install = { "" },
  highlight = {
    enable = true,
```

```
    disable = { "" },
    additional_vim_regex_highlighting = true,
  },
}
EOF
" ------------------------------------
" tagbar 설정
" ------------------------------------
" tagbar 생성 시 우측 하단에 위치하게끔 생성
let g:tagbar_position = 'rightbelow'
" ------------------------------------
" ConqueTerm 설정
" 창 전환 시 ConqueTerm 에 Insert 상태로 활성화
" let g:ConqueTerm_InsertOnEnter = 1
" ConqueTerm 이 Insert 모드인 상태에서도 <Ctrl>+w, W 를 사용 가능하게
" let g:ConqueTerm_CWInsert = 1
" ------------------------------------
" vim-airline 설정
" ------------------------------------
" powerline-font 활성화
let g:airline_powerline_fonts = 1
" luna 테마 사용
let g:airline_theme = 'luna'
" tabline 에 파일명만 출력 되도록 설정
let g:airline#extensions#tabline#formatter = 'unique_tail'
" 창의 상단에 표시되도록 설정
" let g:airline_statusline_ontop = 1
" 탭라인 허용
let g:airline#extensions#tabline#enabled = 1
" 항상 tabline 을 표시
let g:airline#extensions#tabline#show_tabs = 1
" ------------------------------------
" NERDTree 설정
" ------------------------------------
" 창 크기(가로)를 20 으로 설정
let g:NERDTreeWinSize=30
" ------------------------------------
" vim-cutlass 설정
" ------------------------------------
" c, C 명령어는 yank 에 영향을 주도록 변경
nnoremap c d
xnoremap c d

nnoremap cc dd
nnoremap C Dv
```

- vim-plug @github.com
- NeoVim 기반 개발환경 설정
- [Linux] neovim 설정 (CoC, Vim-Plug, treesitter, NERDTree)
- Neovim, vim-plug 설정

- https://tyanjournal.com/tips/neovim-c-ide/[Vim] Neovim을 C++ IDE로 사용하기
- My neovim settings: 나의 neovim 세팅을 공유합니다.
- Awesome Neovim Setup From Scratch - Full Guide

From:
http://theta5912.net/ - **reth**

Permanent link:
http://theta5912.net/doku.php?id=public:computer:neovim

Last update: **2023/01/05 17:38**