Introduction to Software Design

Lab1 참고자료: Vim 확장 사용법

Yoonsang Lee
Spring 2020

Vim 확장 사용법

편리한 사용을 위한 shell 설정

- set completion-ignore-case on
  - bash(shell)은 tab을 눌러 자동완성을 하는 기능이 있는데, 이 때 대소문자 구분 없이 tab 자동완성이 되도록 하면 편리하다.
  - .inputrc 파일에 'set completion-ignore-case on'라고 적어주면 되는데, 가장 편한 방법은

(Shell – home directory)

```
echo 'set completion-ignore-case On' >> ~/.inputrc
```

  - Terminal을 닫았다가 다시 열면 적용될 것을 알 수 있다.

편리한 사용을 위한 shell 설정 2

- stty -ixon
  - Ctrl와 함께 사용하는 단축키가 사용가능 하도록 하는 설정이다.
  - .bashrc 파일에 'stty -ixon'을 추가해 주면 된다. 또는, 다음과 같은 script를 terminal에 타이핑한다.

(Shell – home directory)

```
echo  ' stty -ixon' >> ~/.bashrc
```

  - Terminal을 닫았다가 다시 열면 적용될 것을 알 수 있다.

vim-gtk 설치

- Ubuntu에 기본으로 포함되어 있는 vim은 많은 유용한 기능이 포함되지 않은 vim-tiny 버전임

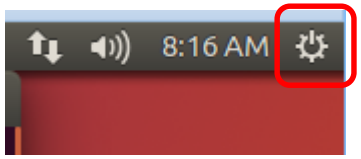
- 아래의 명령어를 실행하여 대부분의 기능을 포함하고 있는 vim-gtk를 설치하자.

(Shell)

```
sudo apt-get install vim-gtk
```

  - password를 요구하면 입력 후 엔터

  - 다운로드 속도가 느릴 경우 오른쪽 상단 버튼 – System Settings – Software & Updates – Downlaod from – Other… - Select Best Server을 눌러서 새로운 서버를 선택하면 많이 빨라질 수 있다.



Vim 설정 방법

- Home directory의 **.vimrc** 파일을 통해 Vim의 각종 설정을 변경할 수 있다.

- .vimrc는 vim script라는 Vim 자체 script language로 작성된다.
  - 주석은 “로 시작
  - 자세한 내용을 알고 싶으면 http://learnvimscriptthehardway.stevelosh.com/

- 처음에는 .vimrc 파일이 없는 상태이므로, home directory에서 아래의 명령을 통해 .vimrc 파일의 편집을 시작해보자.

(Shell)

```
vi .vimrc
```

Vim 추천 설정

.vimrc

```
syntax on   "use syntax highlighting
filetype plugin indent on   "use auto-indentation

set expandtab    "use spaces instead of a tab
set tabstop=4    "number of spaces for a tab
set shiftwidth=4 "number of spaces for each step of indent (e.g. when using '>' or '<')

set nowrap  "stop line breaking
set clipboard=unnamedplus "use system clipboard (e.g. when using 'yy')
set ignorecase  "case-insensitive search
set incsearch   "use incremental search

"disable automatic comment insertion
autocmd FileType * setlocal formatoptions-=c formatoptions-=r formatoptions-=o
```

- .vimrc을 수정한 후 :so% 를 입력하면 수정된 설정이 적용된다. 위 파일을 수정해보며 테스트해보자.
- 위 파일은 최소한의 항목만 담고 있으므로, 자유롭게 설정을 수정 및 추가/제거하면서 사용해보자.

Vim 편리한 기능 | Visual Mode

텍스트 선택 (비주얼 모드)

https://vim.rtorr.com/lang/ko/

`v` - 선택 모드 시작. 텍스트 선택해서 명령 수행 (가령 y로 복사)

`V` - 행 단위 선택 모드 시작

`o` - 선택 영역의 반대쪽 끝으로 이동

`Ctrl` + `v` - 블록 선택 모드 시작

- .vimrc 파일에서 블록 지정 후 복사(y), 붙여넣기(p) 등을 해보자.

Vim 편리한 기능 | Windows

여러 파일 작업

https://vim.rtorr.com/lang/ko/

`Ctrl` + `ws` - 상하로 창 분할

`Ctrl` + `ww` - 창 전환

`Ctrl` + `wq` - 창 닫기

`Ctrl` + `wv` - 좌우로 창 분할

`Ctrl` + `wh` - 오른쪽 창으로 이동 (좌우 분할)

`Ctrl` + `wl` - 왼쪽 창으로 이동 (좌우 분할)

`Ctrl` + `wj` - 아래 창으로 이동 (상하 분할)

`Ctrl` + `wk` - 위 창으로 이동 (상하 분할)

- .vimrc 파일을 창 분할하여 여러 개의 창에 열고, 창 사이를 이동해보자.

Vim Plug-ins

- Vim의 기능을 확장시켜주는 많은 plugin들을 사용할 수 있다.
  - https://vim.sourceforge.io/scripts/script_search_results.php?keywords=&script_type=&order_by=rating&direction=descending&search=search
- 직접 파일을 다운받아 사용할 수도 있지만,
- 요즘은 보통 vim plugin manager들 중 하나를 사용해서 plugin 설치 및 관리를 한다.

vim-plug 설치

- 많이 쓰이는 vim plugin manager 중 하나
- https://github.com/junegunn/vim-plug

- vim-plug를 사용하려면 git을 먼저 설치한 후, 초기 설정을 해야 한다.
  - 아래 회색 부분을 본인의 정보로 바꾸어 입력

(Shell)

```
sudo apt-get install git
git config --global user.name "Your Name"
git config --global user.email "you@example.com"
```

- 위 페이지의 안내를 따라 vim-plug를 설치한다

vim-plug 사용

- 가장 유용한 2개의 plugin만을 설치해보자.
  - 아래 내용을 .vimrc에 추가&저장 후, :so% 와 :PlugInstall

.vimrc

```
call plug#begin()
Plug 'scrooloose/nerdtree'
Plug 'scrooloose/nerdcommenter'
call plug#end()
```

- Plug '<Github계정명>/<Github프로젝트명>' 같은 식으로 다른 plugin을 마음껏 추가해서 사용할 수 있다.
  - 대부분의 Vim plugin들도 Github에서 프로젝트가 진행된다.
  - 예) NERDTree의 Github 주소는 https://github.com/scrooloose/nerdtree 이다.

NERDTree

- Vim상에서 directory browsing을 할 수 있게 해주는 plugin

- 실행방법: Vim에서 :NERDTree를 입력 후 엔터

- NERDTree 창으로 이동한 후 ?를 눌러 나오는 도움말을 보면서 여러 가지 동작을 테스트해본다.

NERD Commenter

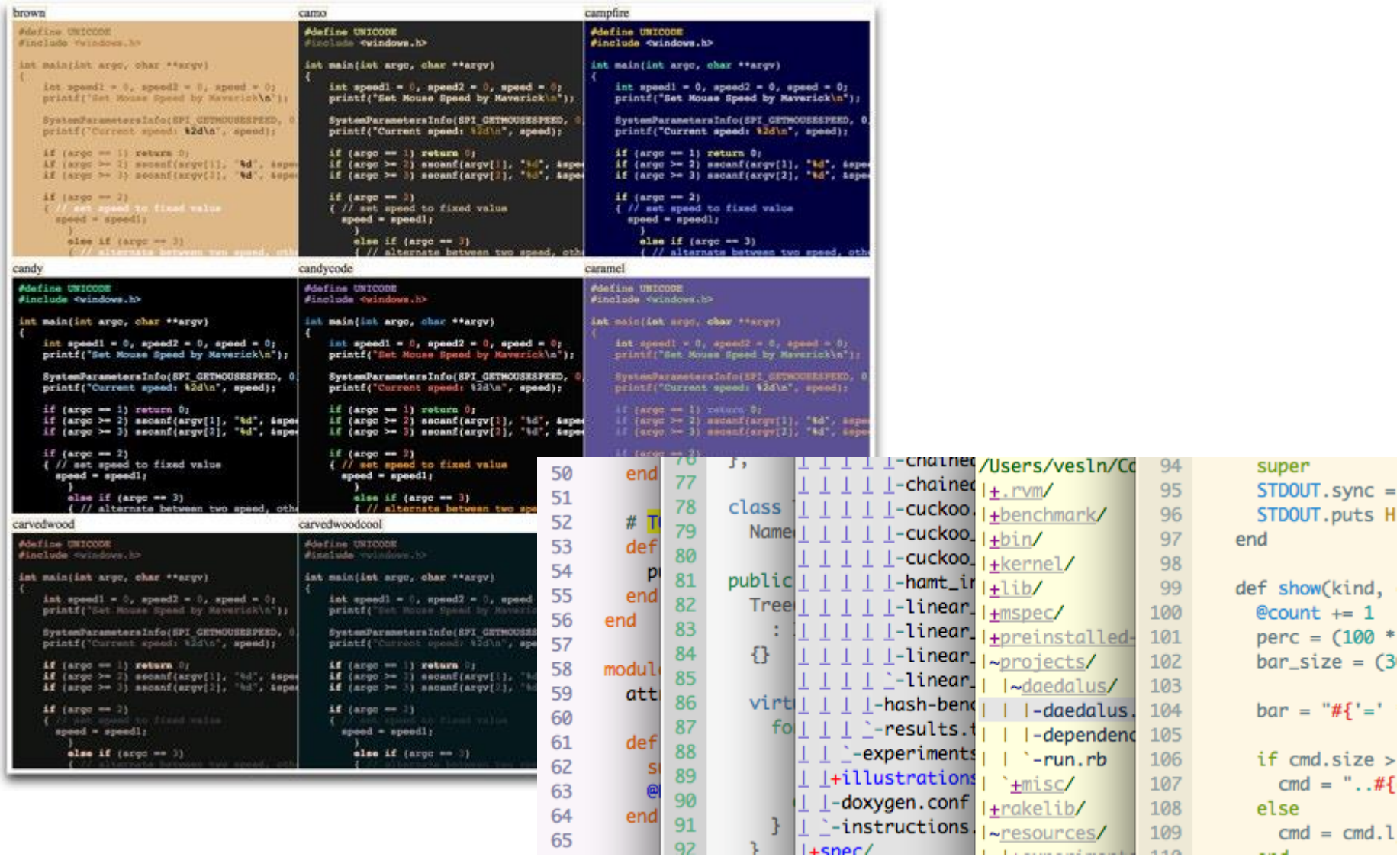
- 여러 종류의 programming language에 대한 comment/uncomment 기능을 제공하는 plugin
- 아래의 내용을 .vimrc에 추가하여 단축키를 새로 설정하자.

.vimrc

```
let mapleader="," " change <leader> key
let NERDCreateDefaultMappings = 0 "disable default mapping
let NERDCommentWholeLinesInVMode = 1 "always comment whole line
map <Leader>c <plug>NERDCommenterComment
map <Leader>x <plug>NERDCommenterUncomment
```

- :so%를 해서 .vimrc를 reload하고 나면,
- ,c : comment
- ,x : uncomment

Vim 꾸미기 - colorscheme



Vim colorscheme 설치

- colorscheme도 plugin처럼 설치 가능

- 많이 사용하는 colorscheme을 2개만 설치해보자
  - 아래 굵은 글씨 두 줄을 .vimrc에 추가&저장 후, :so% 와 :PlugInstall

.vimrc

```
call plug#begin()
Plug 'scrooloose/nerdtree'
Plug 'scrooloose/nerdcommenter‘
Plug 'vim-scripts/xoria256.vim'
Plug 'vim-scripts/peaksea'
call plug#end()
```

Vim colorscheme 변경

- .vimrc에 아래 내용을 추가해서 colorscheme을 바꿔보자.
  - 아래 내용을 .vimrc에 추가&저장 후, :so%
  - 주석 처리 부분을 바꿔서 colorscheme을 바꿔보자.

.vimrc

```
colorscheme xoria256
“colorscheme peaksea
```

Colorscheme을 더 찾으려면?

- http://vimcolors.com/
- https://vim.sourceforge.io/scripts/script_search_results.php?keywords=&script_type=color+scheme&order_by=rating&direction=descending&search=search
- Google에서 ‘vim colorscheme’ 검색

- 위의 방법으로 colorscheme의 Github 프로젝트 주소를 찾은 후, plug#begin()과 plug#end() 사이에 추가하면 됨.

덧붙이자면…

- Linux / Unix에서 개발할 때 꼭 vim을 사용해야 하나요?
  - 그렇지 않다. Emacs, Gedit, Sublime Text, Atom, Qt Creator, Eclipse… 등 다른 text editor를 사용할 수 있다.

- 하지만 vim에 익숙해지면 Linux/Unix 시스템 사용 시 도움이 되는 부분이 많기 때문에, 본 강좌의 실습에서는 vim을 사용하는 것을 권함.