服务器配置 VIM 自动生成预定义内容

Vim script

```
"请注意，VIM脚本用双引号做注释符号"
"在home目录新建（首次配置）.vimrc文件"
vim ~/.vimrc
"写入以下内容后保存"

" 基础配置
set number                      " 显示行号
syntax on                       " 语法高亮（修正 syntax=on 为正确语法）
set confirm                     " 处理未保存或只读文件时要求确认
set smartindent                 " 智能对齐
filetype plugin indent on       " 启用文件类型检测
set showmatch                   " 显示括号匹配
set nocompatible                " 禁用 Vi 兼容模式

" Vundle 插件管理配置
set rtp+=~/.vim/bundle/Vundle.vim
call vundle#begin('~/app/software/vundle_')

Plugin 'altercation/vim-colors-solarized'   " 主题
Plugin 'scrooloose/nerdtree'               " 目录树
Plugin 'vim-scripts/taglist.vim'           " 标签列表
Plugin 'Valloric/YouCompleteMe'            " 自动补全
Plugin 'godlygeek/tabular'                 " 代码对齐

call vundle#end()

" 文件类型专属配置
autocmd FileType python set tabstop=4 | set expandtab | set autoindent  " Python缩进规
则

" 自动生成文件头
autocmd BufNewFile *.cpp,*.[ch],*.sh,*.py,*.java exec ":call SetTitle()"

func SetTitle()
    " 基础头信息生成
    if &filetype == 'sh'
        " Shell 脚本
        call setline(1, "#################################################################
##############")
        call append(1,  "# File Name: ".expand("%"))
        call append(2,  "# Author: zxzong")
        call append(3,  "# Created Time: ".strftime("%c"))
        call append(4,  "#################################################################
#############")
        call append(5,  "#!/bin/sh")
        let base_line = 6
```

```
    elseif &filetype == 'python'
        " Python 脚本
        call setline(1,  "# ************************************************************
***************")
        call append(1,  "# File Name: ".expand("%"))
        call append(2,  "# Author: zxzong")
        call append(3,  "# Created Time: ".strftime("%c"))
        call append(4,  "# ************************************************************
*************")
        let base_line = 5
    elseif &filetype == 'cpp' || &filetype == 'c'
        " C/C++ 文件
        call setline(1,  "/* ************************************************************
****************")
        call append(1,   " * File Name: ".expand("%"))
        call append(2,   " * Author: zxzong")
        call append(3,   " * Created Time: ".strftime("%c"))
        call append(4,   " ************************************************************
**************/")
        let base_line = 5
    else
        " 其他文件类型
        call setline(1,  "<!-- ************************************************************
*****************")
        call append(1,   "    File Name: ".expand("%"))
        call append(2,   "    Author: zxzong")
        call append(3,   "    Created Time: ".strftime("%c"))
        call append(4,   " ************************************************************
************* -->")
        let base_line = 5
    endif

    " 类型专属内容追加
    if &filetype == 'cpp'
        call append(base_line,   "#include<iostream>")
        call append(base_line+1, "using namespace std;")
        call append(base_line+2, "")
    elseif &filetype == 'c'
        call append(base_line,   "#include<stdio.h>")
        call append(base_line+1, "")
    elseif &filetype == 'python'
        call append(base_line,   "import warnings")
        call append(base_line+1, "import numpy as np")
        call append(base_line+2, "import pandas as pd")
        call append(base_line+3, "import pretty_errors")
        call append(base_line+4, "import seaborn as sns")
        call append(base_line+5, "import matplotlib.pyplot as plt")
        call append(base_line+6, "")
        call append(base_line+7, "plt.style.use('science')")
        call append(base_line+8, "warnings.filterwarnings('ignore')")
        call append(base_line+9, "")
    endif
```

```
endfunc

" 新建文件后自动定位到文件末尾（移出函数外避免重复定义）
autocmd BufNewFile * normal G
```

此时，新建文件会自动写入预先定义的内容，

例如新建 test.sh 文件，会自动生成：

```Bash
################################################################################
# File Name: test.sh
# Author: zxzong
# Created Time: Sun 02 Mar 2025 7:01:41 AM CST
################################################################################
#!/bin/sh
```

新建 test.py 文件，会自动生成：

```Python
# ************************************************************************
# File Name: test.py
# Author: zxzong
# Created Time: Sun 02 Mar 2025 7:00:09 AM CST
# ************************************************************************
import warnings
import numpy as np
import pandas as pd
import pretty_errors
import seaborn as sns
import matplotlib.pyplot as plt

plt.style.use('science')
warnings.filterwarnings('ignore')
```