

Linux Bashrc 設定

```
case $- in
    *i*) ;;
    *) return;;
esac

HISTCONTROL=ignoreboth
shopt -s histappend
HISTSIZE=1000
HISTFILESIZE=2000
shopt -s checkwinsize

[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"

if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then
    debian_chroot=$(cat /etc/debian_chroot)
fi

case "$TERM" in
    xterm-color|*-256color) color_prompt=yes;;
esac

if [ -n "$force_color_prompt" ]; then
    if [ -x /usr/bin/tput ] && tput setaf 1 >&/dev/null; then
        color_prompt=yes
    else
        color_prompt=
    fi
fi

if [ "$color_prompt" = yes ]; then
    PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '
else
    PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '
fi
unset color_prompt force_color_prompt

case "$TERM" in
    xterm*|rxvt*)
```

```

PS1="\[\e]0;${debian_chroot:+($debian_chroot)}\u@h: \w\a]$PS1"

;;

*)

;;

esac

if [ -x /usr/bin/dircolors ]; then
    test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval "$(dircolors -b)"
    alias ls='ls --color=auto'

    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'
fi

alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'

alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error)" "$(history|tail -n1|sed -e s/^//)"'

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

export PATH="/usr/local/cuda-12.2/bin:/home/lzw/.local/share/gem/ruby/3.0.0/bin:/home/lzw/.local/bin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/root/.cargo/bin:/root/.local/bin:/root/.local/share/gem/ruby/3.0.0/bin:/root/.nvm/versions/node/v20.11.0/bin"

export GLOBAL_PROXY='127.0.0.1:7890'

function start_proxy {

```

```

export HTTP_PROXY="http://$GLOBAL_PROXY"
export HTTPS_PROXY="http://$GLOBAL_PROXY"
export http_proxy="http://$GLOBAL_PROXY"
export https_proxy="http://$GLOBAL_PROXY"
export HTTP_PROXY_REQUEST_FULLURI=false
export HTTPS_PROXY_REQUEST_FULLURI=false
export ALL_PROXY=$http_proxy
}

function start_proxy_without_prefix {
    export http_proxy=$GLOBAL_PROXY
        export HTTP_PROXY=$GLOBAL_PROXY
        export https_proxy=$GLOBAL_PROXY
    export HTTPS_PROXY=$GLOBAL_PROXY
    export HTTP_PROXY_REQUEST_FULLURI=false
    export HTTPS_PROXY_REQUEST_FULLURI=false
        export ALL_PROXY=$http_proxy
}

function stop_proxy {
    export http_proxy=
        export HTTP_PROXY=
        export https_proxy=
    export HTTPS_PROXY=
    export HTTP_PROXY_REQUEST_FULLURI=true
    export HTTPS_PROXY_REQUEST_FULLURI=true
        export ALL_PROXY=
}

export NO_PROXY="localhost,127.0.0.1,.example.com,::1"

function start_git_proxy {
    git config --global http.proxy $GLOBAL_PROXY
    git config --global https.proxy $GLOBAL_PROXY
}

function stop_git_proxy {
    git config --global --unset http.proxy
    git config --global --unset https.proxy
}

```

```

}

start_proxy
start_git_proxy

alias python=python3
alias pip=pip3

function gpa {
    python ~/bin/gitmessageai.py --api mistral --allow-pull-push
}

function gca {
    python ~/bin/gitmessageai.py --no-push
}

function gm {
    python ~/bin/gitmessageai.py --only-message
}

function gpp {
    git push || {
        echo "Push failed, attempting pull and merge"
        git pull --rebase || {
            echo "Pull failed, please resolve conflicts manually"
            return 1
        }
    }
    git push || {
        echo "Push failed after pull, please resolve conflicts manually"
        return 1
    }
}

preexec() {
    local network_commands=(
        "gpa"
        "git"
        "ssh"
        "scp"

```

```
"sftp"
"rsync"
"curl"
"wget"
"apt"
"yum"
"dnf"
"npm"
"yarn"
"pip"
"pip3"
"gem"
"cargo"
"docker"
"kubect1"
"ping"
"traceroute"
"netstat"
"ss"
"ip"
"ifconfig"
"dig"
"nslookup"
"nmap"
"telnet"
"ftp"
"nc"
"tcpdump"
"adb"
"bundle"
"brew"
"cpanm"
"bundle exec jekyll"
"make"
"python"
"glcloud"
)
```

```
local cmd
```

```
cmd=$(echo "$1" | awk '{print $1}')
```

```

display_proxy() {
    echo -e "  **Proxy Settings Detected:**"

    [ -n "$HTTP_PROXY" ] && echo "    - HTTP_PROXY: $HTTP_PROXY"
    [ -n "$HTTPS_PROXY" ] && echo "    - HTTPS_PROXY: $HTTPS_PROXY"

    echo ""
}

for network_cmd in "${network_commands[@]}"; do
    if [[ "$1" == "$network_cmd*" ]]; then
        if [ -n "$HTTP_PROXY" ] || [ -n "$http_proxy" ] || \
            [ -n "$HTTPS_PROXY" ] || [ -n "$https_proxy" ] || \
            [ -n "$ALL_PROXY" ] || [ -n "$all_proxy" ]; then

            display_proxy
        fi
        break
    fi
done
}

function checkproxy {
    echo "HTTP_PROXY: $HTTP_PROXY"
    echo "HTTPS_PROXY: $HTTPS_PROXY"
    echo "Git HTTP Proxy:"
    git config --get http.proxy
    echo "Git HTTPS Proxy:"
    git config --get https.proxy
}

```