|  |
| --- |
|  |
| 시스템 프로그래밍 |
|  |
| .bashrc 분석 , prompt 변경 |

**1분반**

2010136017

김덕균

시스템 프로그래밍

.bashrc 분석 , prompt 변경

# .bashrc 분석

# ~/.bashrc: executed by bash(1) for non-login shells.

# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)

# for examples

# If not running interactively, don't do anything

case $- in

\*i\*) ;;

\*) return;;

esac

# -변수에 들어있는 값을 참조하여 케이스문

# don't put duplicate lines or lines starting with space in the history.

# See bash(1) for more options

HISTCONTROL=ignoreboth

# 변수에 값 대입

# append to the history file, don't overwrite it

shopt -s histappend

# shopt란 프로그램에 -s histappend 옵션을 주고 실행

# shell options 란 프로그램으로 쉘에 histappend 옵션을 적용하는구문

# histappend 는 command history 적용하는 옵션

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)

HISTSIZE=1000

HISTFILESIZE=2000

# 변수에 값 대입

# check the window size after each command and, if necessary,

# update the values of LINES and COLUMNS.

shopt -s checkwinsize

# shopt 프로그램에 -s checkwinsize 옵션을 주고 실행

# 해당 옵션은 윈도우사이즈를 확인하고 개행시켜주는 옵션

# If set, the pattern "\*\*" used in a pathname expansion context will

# match all files and zero or more directories and subdirectories.

#shopt -s globstar

# make less more friendly for non-text input files, see lesspipe(1)

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

# /usr/bin/lesspipe 파일이 존재하고 실행 가능하다면

# SHELL=/bin/sh lesspipe 명령어 실행

# set variable identifying the chroot you work in (used in the prompt below)

if [ -z "${debian\_chroot:-}" ] && [ -r /etc/debian\_chroot ]; then

debian\_chroot=$(cat /etc/debian\_chroot)

fi

# debian\_chroot 문자가 비어있고 /etc/debian\_chroot 파일이 존재하고 읽을수 있으면

# debian\_chroot 에 /etc/debian\_chroot 파일 내용을 대입

# set a fancy prompt (non-color, unless we know we "want" color)

case "$TERM" in

xterm-color|\*-256color) color\_prompt=yes;;

esac

# TERM변수가 케이스문을 만족하면 color\_prompt에 yes를 대입

# uncomment for a colored prompt, if the terminal has the capability; turned

# off by default to not distract the user: the focus in a terminal window

# should be on the output of commands, not on the prompt

#force\_color\_prompt=yes

if [ -n "$force\_color\_prompt" ]; then

if [ -x /usr/bin/tput ] && tput setaf 1 >&/dev/null; then

# We have color support; assume it's compliant with Ecma-48

# (ISO/IEC-6429). (Lack of such support is extremely rare, and such

# a case would tend to support setf rather than setaf.)

color\_prompt=yes

else

color\_prompt=

fi

fi

# force\_color\_prompt 문자열이 존재하고

# /usr/bin/tput 파일이 존재하면서 실행가능하며 tput setaf 1 을 실행한 결과를

# /dev/null에 리다이렉션하는데 성공하면 color\_prompt에 yes대입

# 아니면 color\_prompt에 널 대입(비우기)

# tput 은 색을 넣어서 출력해주는 프로그램

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

# color\_prompt가 yes면

# PS1(쉘 프롬프트)에 색상을 입혀서 적용

# 아니라면 흰색으로 적용

# If this is an xterm set the title to user@host:dir

case "$TERM" in

xterm\*|rxvt\*)

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

;;

\*)

;;

esac

# TERM 변수(터미널 이름)를 보고 터미널이 xterm이나 rxvt면 쉘 프롬프트를 해당 모양으로 변경

#

# enable color support of ls and also add handy aliases

if [ -x /usr/bin/dircolors ]; then

test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval "$(dircolors -b)"

alias ls='ls --color=auto'

#alias dir='dir --color=auto'

#alias vdir='vdir --color=auto'

alias grep='grep --color=auto'

alias fgrep='fgrep --color=auto'

alias egrep='egrep --color=auto'

fi

# /usr/bin/dircolors 파일이 있고 실행가능하고

# 홈파일에 ./dircolors가 존재하면서 읽을수도 있다면

# dircolors -b ~/.dircolors 명령어 실행 해당 명령어가 실패하면

# dircolors -b 명령어 실행

# ls, grep, fgrep, egrep에 alias를 걸어 해당 명령어 입력시 우측으로 치환# ls, grep, fgrep, egrep에 alias를 걸어 해당 명령어 입력시 우측으로 치환# ls, grep, fgrep, egrep에 alias를 걸어 해당 명령어 입력시 우측으로 치환

# colored GCC warnings and errors

#export GCC\_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'

# some more ls aliases

alias ll='ls -alF'

alias la='ls -A'

alias l='ls -CF'

#ll, la , l 에 alias를 걸어 우측으로 치환할 수 있게 하여 명령어를 쉽게 사용하게 만듦

# Add an "alert" alias for long running commands. Use like so:

# sleep 10; alert

alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error)" "$(history|tail -n1|sed -e '\''s/^\s\*[0-9]\+\s\*//;s/[;&|]\s\*alert$//'\'')"'

# 위와 동일하게 alias 적용

# Alias definitions.

# You may want to put all your additions into a separate file like

# ~/.bash\_aliases, instead of adding them here directly.

# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash\_aliases ]; then

. ~/.bash\_aliases

fi

# 홈디렉토리의 .bash\_aliases가 존재하고 정규 파일이면 if문 실행

# enable programmable completion features (you don't need to enable

# this, if it's already enabled in /etc/bash.bashrc and /etc/profile

# sources /etc/bash.bashrc).

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

# shopt -op posix 가 거짓이면서

# /usr/share/bash-completion/bash\_complition 파일이 존재하고 정규파일이면

# 아래 명령어 실행

# 위의 값이 거짓이고 /etc/bash\_completion 파일이 존재하며 정규파일이면

# 아래 명령어 실행

# 프롬프트 변경

파일 생성

Touch prompt.sh

파일 수정

Vi prompt.sh

내용 작성 후 저장

Export PS1=”[\u@\w]$”

바로 적용

Source prompt.sh

스크린 샷

