**System Programming**

**#2**

1. **.bashrc 파일해석**
2. **Shell prompt변경**

**1분반**

**장경식 교수님**

**2012136020 김동우**

**2016.10.07**

==.bashrc 파일 해석

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

// ~/.bashrc : 비로그인쉘들을 위해 bash(1)이 실행된다.

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

// /usr/share/doc/bash/examples/startup-files(bash-doc 패키지 안에있는)을 보아라.

# for examples

// 예를 들면

# If not running interactively, don't do anything

// 상호작용적으로 실행되지 않으면, 아무것도 하지 마라.

case $- in

\*i\*) ;;

\*) return;;

esac

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

// 히스토리 안에 중복라인 혹은 공백으로 시작하는 라인을 넣지 마라.

# See bash(1) for more options

// 더 많은 옵션을 위해 bash(1)을 보아라.

HISTCONTROL=ignoreboth

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

// 히스토리 파일을 덮어쓰지말고, 추가해라.

shopt -s histappend

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

// 히스토리 길이설정을 위해, bash(1)안의 HISTSIZE 와 HISTFILESIZE를 보아라.

HISTSIZE=1000

HISTFILESIZE=2000

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

// 각 명령어 다음 윈도우 사이즈를 확인하고, 필요하면,

# update the values of LINES and COLUMNS.

// 행과 열의 값을 업데이트 하라.

shopt -s checkwinsize

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

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

// 설정되었다면, 경로 이름 확장 문맥에 사용되는 "\*\*" 패턴은

// 모든파일, 0개 혹은 그 이상의 디렉토리들과 서브디렉토리가 일치할 것이다.

#shopt -s globstar

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

// 비-텍스트 입력 파일들을 위해 less와 좀더 친해져라, lesspile(1)을 보아라.

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

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

// (프롬프트 아래에서) 작업하고 있는 chroot를 식별하는 변수를 설정하라

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

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

fi

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

// 상상하고 있는 프롬프트를 정하라. (non-color, 우리가 원하는 색을 알지 못한다면)

case "$TERM" in

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

esac

# 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

// 터미널을 사용할 능력이 된다면, 색상 프롬프트에 대한 코멘트를 하지마라.

// 유저에게 혼란을 주지 않기 위해 기본 값으로 해라. :

// 터미널 윈도우에서의 초점은 프롬프트가 아닌 결과물에 있어야 한다.

#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

// 색상을 지원한다. Ecma-48를 따른다고 가정하라.

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

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

// 그러한 지원의 부족은 극도로 드물고,

// 여러 경우는 setf보다 setaf를 지원하는 경향이 있다.

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

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

// xterm 설정 인 경우 user@host:dir로 제목을 설정한다.

case "$TERM" in

xterm\*|rxvt\*)

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

;;

\*)

;;

esac

# enable color support of ls and also add handy aliases

// ls의 색 지원을 가능하게 하고, 또한 유용한 가명(명령)도 더하라

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

# colored GCC warnings and errors

// 색이 있는 GCC 경고 그리고 에러

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

// GCC\_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'를 표현하라.

# some more ls aliases

alias ll='ls -alF'

alias la='ls -A'

alias l='ls -CF'

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

// 장기 실행 명령을 위해 "ALERT" 가명(명령)을 추가하라. 이렇게

# 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$//'\'')"'

# Alias definitions.

// 가명(명령) 정의

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

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

// 당신은 이 곳에 직접적으로 첨가하는 대신에 ~/.bash\_aliases 같이 분리된 파일에

// 너의 추가사항을 넣고 싶어 할 지도 모른다.

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

// bash-doc 패키지 안의 /usr/share/doc/bash-doc/examples을 보아라

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

. ~/.bash\_aliases

fi

# 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).

// 프로그램 완성 기능을 사용(만일 /etc/bash.bashrc and /etc/profile

// sources /etc/bash.bashrc을 이미 사용할 수 있다면)

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

//프롬프트 수정 : 사용자 이름과 현재 디렉토리를 나타낸다.

PS1="[\U@ \W]\\$"

쉘 변수 기호

\t 24시간의 단위로 현재시각을 HH:MM:SS 로 표시

\T 12시간의 단위로 현재시각을 HH:MM:SS 로 표시

\@ 12시간의 단위로 현재시각을 오전/오후 로 표시

\d 현재 날짜를 나타냄. 요일, 월, 일 형식으로

\s 현재 사용중인 쉘의 이름을 나타냄 (C쉘이면 /bin/csh, bash쉘이면 /bin/bash)

\w 현재 디렉토리의 전체 절대경로를 모두 표시

\W 현재 디렉토리의 전체 절대경로명 중 마지막 디렉토리명만을 표시. 즉 현재디렉토리만 표시.

\u 사용자명을 표시

\h 서버의 호스트명을 표시함 (www.uzuro.com에서 www 부분)

\H 서버의 도메인명을 표시함 (www.uzuro.com에서 uzuro.com 부분)

\# 접속한 순간부터 사용한 명령어의 번호를 1번부터 차례대로 표시

\! 사용한 명령어의 history 번호를 표시

\\$ 현재 사용자가 root(uid 가 0 )이면 # 을 표시하고 아니면 $ 를 표시

\\ '\' 문자 자체를 표시

\a ASCII 종소리 문자 (07)

\e ASCII 의 escape 문자 (033)

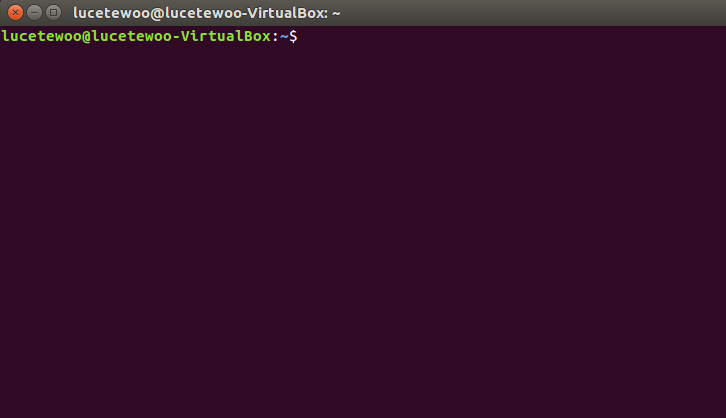
\n 개행문자 (줄바꿈)

\v 사용중인 bash 의 버전

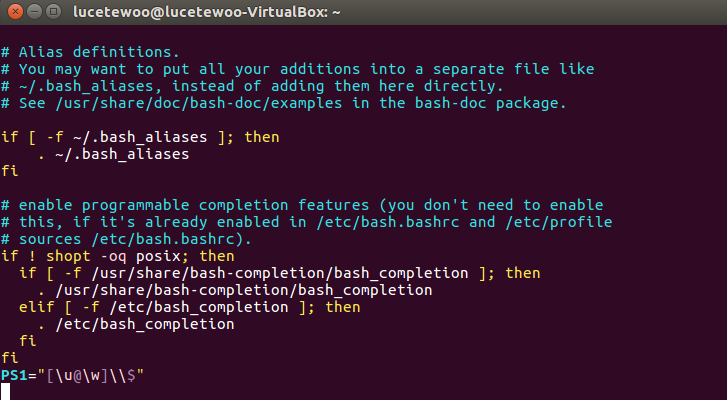
\V 사용중인 bash 의 배포, 버전+패치수준으로 버전을 상세히 표시

\r Carrage retrun

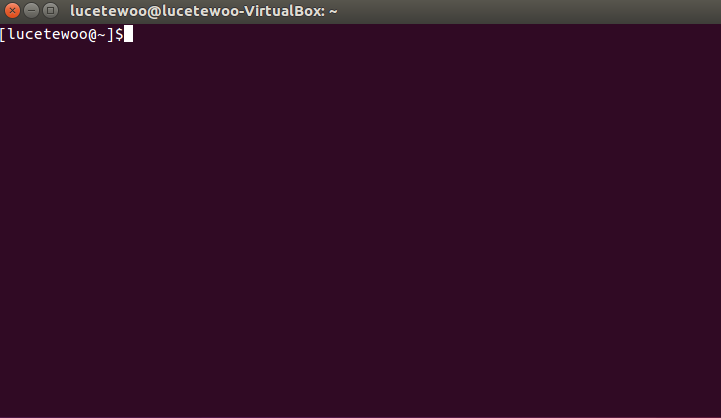
\nnn 8진수 nnn 에 해당하는 문자

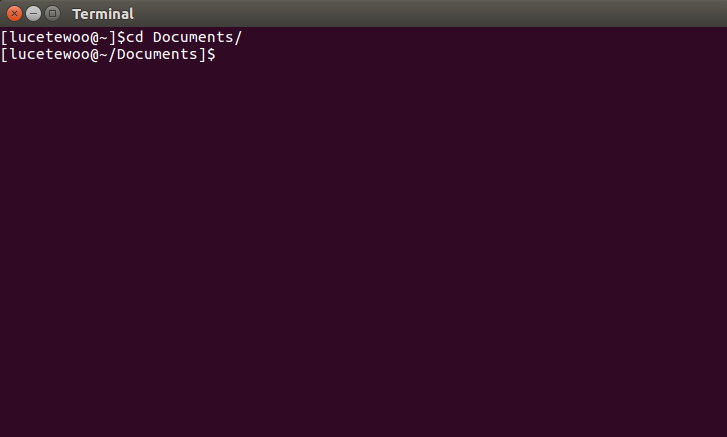
****

< 초기 터미널 >

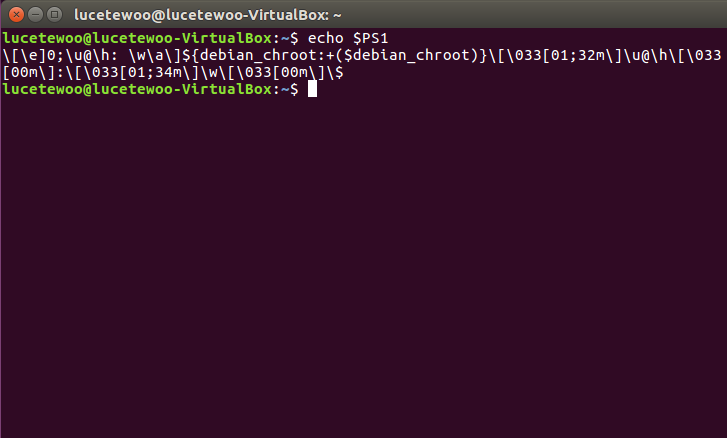


< vi .bashrc 편집 >

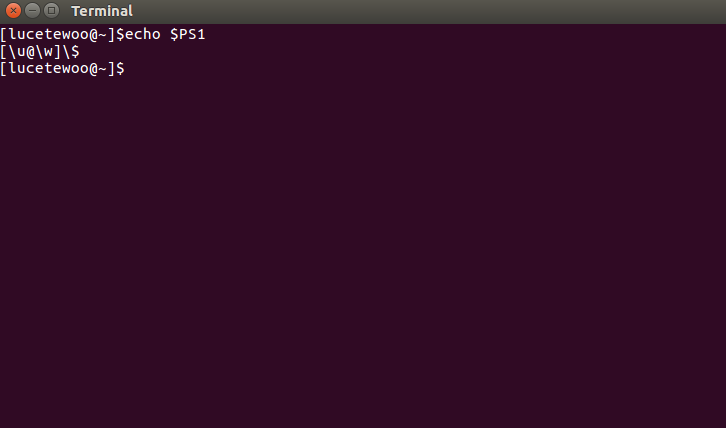


< PS1=”[\u@\w][\\s](file:///\\s)” vi 편집 후 >

< 프롬프트 변경 후 >



<프롬프트 변경 전 $PS1의 쉘 변수>



< 프롬프트 변경 후 $PS1의 쉘 변수>