# /etc/profile and /etc/profile.d/*.sh

- runs only when you login

```
/home/cis90/simben $ cat /etc/profile
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# It's NOT a good idea to change this file unless you know what you
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as this
# will prevent the need for merging in future updates.

pathmunge () {
    case ":${PATH}:" in
        *:"$1":*)
            ;;
        *)
            if [ "$2" = "after" ] ; then
                PATH=$PATH:$1
            else
                PATH=$1:$PATH
            fi
    esac
}


if [ -x /usr/bin/id ]; then
    if [ -z "$EUID" ]; then
        # ksh workaround
        EUID=`/usr/bin/id -u`
        UID=`/usr/bin/id -ru`
    fi
    USER="`/usr/bin/id -un`"
    LOGNAME=$USER
    MAIL="/var/spool/mail/$USER"
fi

# Path manipulation
if [ "$EUID" = "0" ]; then
    pathmunge /usr/sbin
    pathmunge /usr/local/sbin
else
    pathmunge /usr/local/sbin after
    pathmunge /usr/sbin after
fi

HOSTNAME=`/usr/bin/hostname 2>/dev/null`
HISTSIZE=1000
if [ "$HISTCONTROL" = "ignorespace" ] ; then
    export HISTCONTROL=ignoreboth
else
    export HISTCONTROL=ignoredups
fi

export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL

# By default, we want umask to get set. This sets it for login shell
# Current threshold for system reserved uid/gids is 200
# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file
if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
    umask 002
else
    umask 022
fi
```

```
for i in /etc/profile.d/*.sh ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge
/home/cis90/simben $
```

```
/home/cis90/simben $ head -100 /etc/profile.d/*.sh
==> /etc/profile.d/256term.sh <==
# Enable 256 color capabilities for appropriate terminals

# Set this variable in your local shell config (such as ~/.bashrc)
# if you want remote xterms connecting to this system, to be sent 256 colors.
# This must be set before reading global initialization such as /etc/bashrc.
#   SEND_256_COLORS_TO_REMOTE=1

# Terminals with any of the following set, support 256 colors (and are local)
local256="$COLORTERM$XTERM_VERSION$ROXTERM_ID$KONSOLE_DBUS_SESSION"

if [ -n "$local256" ] || [ -n "$SEND_256_COLORS_TO_REMOTE" ]; then

  case "$TERM" in
    'xterm') TERM=xterm-256color;;
    'screen') TERM=screen-256color;;
    'Eterm') TERM=Eterm-256color;;
  esac
  export TERM

  if [ -n "$TERMCAP" ] && [ "$TERM" = "screen-256color" ]; then
    TERMCAP=$(echo "$TERMCAP" | sed -e 's/Co#8/Co#256/g')
    export TERMCAP
  fi
fi

unset local256

==> /etc/profile.d/colorgrep.sh <==
# color-grep initialization

/usr/libexec/grepconf.sh -c || return

alias grep='grep --color=auto' 2>/dev/null
alias egrep='egrep --color=auto' 2>/dev/null
alias fgrep='fgrep --color=auto' 2>/dev/null



==> /etc/profile.d/colorls.sh <==
# color-ls initialization

# Skip all for noninteractive shells.
[ ! -t 0 ] && return

#when USER_LS_COLORS defined do not override user LS_COLORS, but use them.
if [ -z "$USER_LS_COLORS" ]; then

  alias ll='ls -l' 2>/dev/null
  alias l.='ls -d .*' 2>/dev/null

  INCLUDE=
  COLORS=
  for colors in "$HOME/.dir_colors.$TERM" "$HOME/.dircolors.$TERM" \
      "$HOME/.dir_colors" "$HOME/.dircolors"; do
    [ -e "$colors" ] && COLORS="$colors" && \
    INCLUDE="`/usr/bin/cat "$COLORS" | /usr/bin/grep '^INCLUDE' | /usr/bin/cut -d ' ' -f2-`" && \
    break
  done

  [ -z "$COLORS" ] && [ -e "/etc/DIR_COLORS.$TERM" ] && \
  COLORS="/etc/DIR_COLORS.$TERM"

  [ -z "$COLORS" ] && [ -e "/etc/DIR_COLORS.256color" ] && \
  [ "x`/usr/bin/tty -s && /usr/bin/tput colors 2>/dev/null`" = "x256" ] && \
  COLORS="/etc/DIR_COLORS.256color"

  [ -z "$COLORS" ] && [ -e "/etc/DIR_COLORS" ] && \
  COLORS="/etc/DIR_COLORS"

  # Existence of $COLORS already checked above.
  [ -n "$COLORS" ] || return

  if [ -e "$INCLUDE" ];
  then
    TMP="`/usr/bin/mktemp .colorlsXXX -q --tmpdir=/tmp`"
    [ -z "$TMP" ] && return

    /usr/bin/cat "$INCLUDE" >> $TMP
    /usr/bin/grep -v '^INCLUDE' "$COLORS" >> $TMP

    eval "`/usr/bin/dircolors --sh $TMP 2>/dev/null`"
    /usr/bin/rm -f $TMP
  else
    eval "`/usr/bin/dircolors --sh $COLORS 2>/dev/null`"
  fi

  [ -z "$LS_COLORS" ] && return
  /usr/bin/grep -qi "^COLOR.*none" $COLORS >/dev/null 2>/dev/null && return
fi

unset TMP COLORS INCLUDE

alias ll='ls -l --color=auto' 2>/dev/null
alias l.='ls -d .* --color=auto' 2>/dev/null
alias ls='ls --color=auto' 2>/dev/null



==> /etc/profile.d/lang.sh <==
# /etc/profile.d/lang.sh - set i18n stuff

sourced=0

if [ -n "$LANG" ]; then
    saved_lang="$LANG"
    [ -f "$HOME/.i18n" ] && . "$HOME/.i18n" && sourced=1
    LANG="$saved_lang"
    unset saved_lang
else
    for langfile in /etc/locale.conf "$HOME/.i18n" ; do
        [ -f $langfile ] && . $langfile && sourced=1
    done
fi

if [ "$sourced" = 1 ]; then
    [ -n "$LANG" ] && export LANG || unset LANG
    [ -n "$LC_ADDRESS" ] && export LC_ADDRESS || unset LC_ADDRESS
    [ -n "$LC_CTYPE" ] && export LC_CTYPE || unset LC_CTYPE
    [ -n "$LC_COLLATE" ] && export LC_COLLATE || unset LC_COLLATE
    [ -n "$LC_IDENTIFICATION" ] && export LC_IDENTIFICATION || unset LC_IDENTIFICATION
    [ -n "$LC_MEASUREMENT" ] && export LC_MEASUREMENT || unset LC_MEASUREMENT
    [ -n "$LC_MESSAGES" ] && export LC_MESSAGES || unset LC_MESSAGES
    [ -n "$LC_MONETARY" ] && export LC_MONETARY || unset LC_MONETARY
    [ -n "$LC_NAME" ] && export LC_NAME || unset LC_NAME
    [ -n "$LC_NUMERIC" ] && export LC_NUMERIC || unset LC_NUMERIC
    [ -n "$LC_PAPER" ] && export LC_PAPER || unset LC_PAPER
    [ -n "$LC_TELEPHONE" ] && export LC_TELEPHONE || unset LC_TELEPHONE
    [ -n "$LC_TIME" ] && export LC_TIME || unset LC_TIME
    if [ -n "$LC_ALL" ]; then
       if [ "$LC_ALL" != "$LANG" ]; then
         export LC_ALL
       else
         unset LC_ALL
       fi
    else
       unset LC_ALL
    fi
    [ -n "$LANGUAGE" ] && export LANGUAGE || unset LANGUAGE
    [ -n "$LINGUAS" ] && export LINGUAS || unset LINGUAS
    [ -n "$_XKB_CHARSET" ] && export _XKB_CHARSET || unset _XKB_CHARSET

    consoletype=$CONSOLETYPE
    if [ -z "$consoletype" ]; then
      consoletype=$(/sbin/consoletype stdout)
    fi

    if [ -n "$LANG" ]; then
      case $LANG in
        *.utf8*|*.UTF-8*)
        if [ "$TERM" = "linux" ]; then
            if [ "$consoletype" = "vt" ]; then
                case $LANG in
                        ja*) LANG=en_US.UTF-8 ;;
                        ko*) LANG=en_US.UTF-8 ;;
                        si*) LANG=en_US.UTF-8 ;;
                        zh*) LANG=en_US.UTF-8 ;;
                        ar*) LANG=en_US.UTF-8 ;;
                        fa*) LANG=en_US.UTF-8 ;;
                        he*) LANG=en_US.UTF-8 ;;
                        en_IN*) ;;
                        *_IN*) LANG=en_US.UTF-8 ;;
                esac
            fi
        fi
        ;;
        *)
        if [ "$TERM" = "linux" ]; then
            if [ "$consoletype" = "vt" ]; then
                case $LANG in
                        ja*) LANG=en_US ;;
                        ko*) LANG=en_US ;;
                        si*) LANG=en_US ;;
                        zh*) LANG=en_US ;;
                        ar*) LANG=en_US ;;
                        fa*) LANG=en_US ;;
                        he*) LANG=en_US ;;
                        en_IN*) ;;
                        *_IN*) LANG=en_US ;;
                esac
            fi
        fi
        ;;
      esac
    fi

    unset SYSFONTACM SYSFONT consoletype
fi
unset sourced
unset langfile
```

```
==> /etc/profile.d/less.sh <==
# less initialization script (sh)
[ -x /usr/bin/lesspipe.sh ] && export LESSOPEN="${LESSOPEN-||/usr/bin/lesspipe.sh %s}"



==> /etc/profile.d/vim.sh <==
if [ -n "$BASH_VERSION" -o -n "$KSH_VERSION" -o -n "$ZSH_VERSION" ]; then
  [ -x /usr/bin/id ] || return
  ID=`/usr/bin/id -u`
  [ -n "$ID" -a "$ID" -le 200 ] && return
  # for bash and zsh, only if no alias is already set
  alias vi >/dev/null 2>&1 || alias vi=vim
fi



==> /etc/profile.d/which2.sh <==
# Initialization script for bash and sh

# export AFS if you are in AFS environment
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'
/home/cis90/simben $
```

# .bash_profile

- runs only when you login

```
[simmsben@opus ~]$ cat .bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/../bin:$HOME/bin:.
BASH_ENV=$HOME/.bashrc
USERNAME=""
PS1='$PWD $ '
export USERNAME BASH_ENV PATH
umask 002
set -o ignoreeof
stty susp
eval `tset -s -m vt100:vt100 -m :\?${TERM:-ansi} -r -Q `

[simmsben@opus ~]$
```

# .bashrc

```
[simmsben@opus ~]$ cat .bashrc
# .bashrc

# User specific aliases and functions

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi
alias print="echo -e"
[simmsben@opus ~]$
```

# /etc/bashrc

```
/home/cis90/simben $ cat /etc/bashrc
# /etc/bashrc

# System wide functions and aliases
# Environment stuff goes in /etc/profile

# It's NOT a good idea to change this file unless you know what you
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as this
# will prevent the need for merging in future updates.

# are we an interactive shell?
if [ "$PS1" ]; then
  if [ -z "$PROMPT_COMMAND" ]; then
    case $TERM in
    xterm*|vte*)
      if [ -e /etc/sysconfig/bash-prompt-xterm ]; then
          PROMPT_COMMAND=/etc/sysconfig/bash-prompt-xterm
      elif [ "${VTE_VERSION:-0}" -ge 3405 ]; then
          PROMPT_COMMAND="__vte_prompt_command"
      else
          PROMPT_COMMAND='printf "\033]0;%s@%s:%s\007" "${USER}" "${HOSTNAME%%.*}"
"${PWD/#$HOME/~}"'
      fi
      ;;
    screen*)
      if [ -e /etc/sysconfig/bash-prompt-screen ]; then
          PROMPT_COMMAND=/etc/sysconfig/bash-prompt-screen
      else
          PROMPT_COMMAND='printf "\033k%s@%s:%s\033\\" "${USER}" "${HOSTNAME%%.*}"
"${PWD/#$HOME/~}"'
      fi
      ;;
    *)
      [ -e /etc/sysconfig/bash-prompt-default ] && PROMPT_COMMAND=/etc/sysconfig/bash-prompt-
default
      ;;
    esac
  fi
  # Turn on parallel history
  shopt -s histappend
  history -a
  # Turn on checkwinsize
  shopt -s checkwinsize
  [ "$PS1" = "\\s-\\v\\\$ " ] && PS1="[\u@\h \W]\\$ "
  # You might want to have e.g. tty in prompt (e.g. more virtual machines)
  # and console windows
  # If you want to do so, just add e.g.
  # if [ "$PS1" ]; then
  #   PS1="[\u@\h:\l \W]\\$ "
  # fi
  # to your custom modification shell script in /etc/profile.d/ directory
fi

if ! shopt -q login_shell ; then # We're not a login shell
    # Need to redefine pathmunge, it get's undefined at the end of /etc/profile
    pathmunge () {
        case ":${PATH}:" in
            *:"$1":*)
                ;;
            *)
                if [ "$2" = "after" ] ; then
                    PATH=$PATH:$1
                else
                    PATH=$1:$PATH
                fi
        esac
    }

    # By default, we want umask to get set. This sets it for non-login shell.
    # Current threshold for system reserved uid/gids is 200
    # You could check uidgid reservation validity in
    # /usr/share/doc/setup-*/uidgid file
    if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
       umask 002
    else
       umask 022
    fi

    SHELL=/bin/bash
    # Only display echos from profile.d scripts if we are no login shell
    # and interactive - otherwise just process them to set envvars
    for i in /etc/profile.d/*.sh; do
        if [ -r "$i" ]; then
            if [ "$PS1" ]; then
                . "$i"
            else
                . "$i" >/dev/null
            fi
        fi
    done

    unset i
    unset -f pathmunge
fi
# vim:ts=4:sw=4
/home/cis90/simben $
```