



Bash, découverte avancée

- Objet : Comprendre à quoi servent les différentes configurations de bash
- Niveau requis :
[débutant](#), [avisé](#)
- Commentaires : *Bash possède bon nombre d'options et de fichiers de configurations. Comment comprendre tout ça ?*
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi :
[à-compléter](#), [à-tester](#)
 - Création par :  [devpsp](#) le 18/08/2010 15:05
 - Testé par :
- Commentaires sur le forum : [ici](#)¹⁾

Nota : Contributeurs, les  sont là pour vous aider, supprimez-les une fois le problème corrigé ou le champ rempli !

Les scripts de configuration

Bash possède comme la plupart des logiciels des fichiers de configuration globaux à tout le système, rangés dans `/etc` et des fichiers personnels, fichiers cachés²⁾ rangés dans le répertoire utilisateur (« `~` »).

Les scripts concernant tous les utilisateurs du système sont les suivants :

- `/etc/profile`
- `/etc/bash.bashrc`

Les fichiers de configuration personnels sont les suivants

- `.bash_profile` ou `.bash_login` ou `.profile`
- `.bashrc`
- `.bash_logout`
- `.bash_history`

Au lancement d'un shell de login (par exemple via ssh ou dans un tty), **bash** interprète le fichier `/etc/profile` s'il existe, puis le fichier `~/.bash_profile`.

S'il ne trouve pas ce dernier, il essaye d'interpréter le fichier `~/.bash_login`, et à défaut tente le fichier `~/.profile`.

Il est possible de simuler un shell de login en tapant

```
bash -l
```

Au lancement d'un shell non-interactif, c'est le fichier `/etc/bash.bashrc` qui est lu, suivi de `~/.bashrc`.

Configuration globale

Le fichier "/etc/profile"

PATH : Chemins contenant les exécutable

Les chemins contenant les exécutable sont déclarés dans la variable d'environnement PATH via le fichier /etc/profile

- Pour les utilisateurs : /usr/local/bin:/usr/bin:/bin:/usr/games
- Pour l'administrateur : /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

Voici l'extrait du /etc/profile définissant les PATH :

[/etc/profile](#)

```
if [ "`id -u`" -eq 0 ]; then
    PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
else
    PATH="/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games"
fi
export PATH
```

Ici 4 chemins sont indiqués, séparés par le caractère ":".

Vous pouvez les utiliser ainsi :

- /usr/local/bin : Vos premiers script à mettre ici pour votre système local (pleins de petits scripts)
- /usr/bin : ici pour les scripts avancés, pour d'autres utilisateurs (des scripts avec des menus)
- /bin : ici il y en a déjà beaucoup, pour les scripts assurément finis (assez gros, avec des fonctions internes à vos scripts et aux autres)
- /usr/games : il y en a déjà pas mal concernant les jeux...

Chargement des options spécifiques à bash

Le fichier /etc/profile vérifie ensuite d'il s'agit d'un shell bash interactif. Si c'est le cas, il charge les options globales spécifiques à bash (contenues dans /etc/bash.bashrc).

[/etc/profile](#)

```
if [ "$PS1" ]; then
    if [ "$BASH" ] && [ "$BASH" != "/bin/sh" ]; then
        if [ -f /etc/bash.bashrc ]; then
            . /etc/bash.bashrc
        fi
    else
        if [ "`id -u`" -eq 0 ]; then
```

```
    PS1='# '
else
    PS1='$ '
fi
fi
fi
```

Dans le cas contraire, il définit les prompts minimaux pour root et pour les utilisateurs.

Chargement des configurations externes

Enfin, `/etc/profile` charge les fichiers de configuration externes (potentiellement installés par l'utilisateur ou par d'autres paquets Debian), situés dans `/etc/profile.d/`:

[/etc/profile](#)

```
# The default umask is now handled by pam_umask.
# See pam_umask(8) and /etc/login.defs.

if [ -d /etc/profile.d ]; then
  for i in /etc/profile.d/*.sh; do
    if [ -r $i ]; then
      . $i
    fi
  done
  unset i
fi
```

Le fichier `"/etc/bash.bashrc"`



Configuration utilisateur

Ces scripts ne concernent que l'utilisateur dans le dossier personnel duquel ils se trouvent. Ce sont des fichiers cachés, leur nom débute avec un point.

Mais si vous utilisez simplement [la commande "ls"](#) ainsi :

```
ls
```

Vous obtenez :

[retour de la commande](#)

```

2010-02-26--10.13.22      fstab          phpmyadmin.png
caisse_060810.ods        gparted1.png    resilier-sfr
commande_sed1.txt        liste-mail      RMLL_2.m4v
commande_sed2.txt        konversation    scripts_user
debian-504-i386-netinst.iso listecamera_site site_agen_100806.ods
debian-facile-le-magazine.pdf liste_site.ods  site_agen_100809.ods
default
.../...

```

Ce sont vos fichiers simples présents dans votre *home*.
Mais avec son option `-l` ainsi :

```
ls -a
```

vous verrez en plus tous les fichiers cachés :

[retour de la commande](#)

```

.          .gnome2_private    nfs-df.odt
..         .gnupg             .openoffice.org
2010-02-26--10.13.22  gparted1.png    .openoffice.org2
.apptitude .gstreamer-0.10  out.ogv
.../...

```

~/.bash_profile, ~/.bash_login et ~/.profile

On voit que le fichier `~/.bash_profile` par défaut ne fait rien d'autre que charger le `~/.bashrc`.

[~/.bash_profile](#)

```

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

```

De même, le `~/.profile` par défaut vérifie que le shell est bien bash, puis interprète le `~/.bashrc`.

[~/.profile](#)

```

if [ -n "$BASH_VERSION" ]; then
    # include .bashrc if it exists
    if [ -f "$HOME/.bashrc" ]; then
        . "$HOME/.bashrc"
    fi
fi

```

Il vérifie en plus si le dossier `~/bin` existe, et, le cas échéant, l'ajoute au PATH.

~/.profile

```
if [ -d "$HOME/bin" ] ; then
    PATH="$HOME/bin:$PATH"
fi
```

~/.bashrc

Le fichier ~/.bashrc est celui contenant la réelle configuration utilisateur de **bash**.

Uniquement pour le mode interactif

Il commence par vérifier qu'il est bien lancé par un utilisateur et non interpréteur d'un script³⁾. Il s'arrête si ça n'est pas le cas.

~/.bashrc

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

Historique

Les lignes suivantes concernent la configuration de l'historique de bash.

~/.bashrc

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

Ici, l'historique ne stocke pas les doublons, conserve le précédent historique, mais le limite à 1000 lignes et 2ko.

Taille du terminal

Il met à jour les variables d'environnement de taille du terminal:

~/.bashrc

```
shopt -s checkwinsize
```

Chroot

Il définit la variable `debian_chroot` s'il trouve un fichier `/etc/debian_chroot`. Cette variable sera ensuite utilisée dans le prompt.

~/.bashrc

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

Terminal en couleur

Il tente de déterminer par plusieurs moyens s'il s'agit d'un terminal couleur ou non :

~/.bashrc

```
case "$TERM" in
    xterm-color) color_prompt=yes;;
esac

#force_color_prompt=yes

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
```

Vous pouvez forcer la détection en décommentant `#force_color_prompt=yes`.

Prompt

Il définit le prompt suivant que le terminal soit en couleur ou non :

~/.bashrc

```
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
```

Il est possible de configurer son prompt avec des éléments et des couleurs de votre choix. Il suffit de configurer la variable \$PS1 vous même dans votre fichier de configuration .bashrc. Exemple :

nom



```
PS1=' \[ \e ] 0 ; \u@\h :
\w\a\] ${debian_chroot:+($debian_chroot)} \[\033[01;32m\] \u@
h\[\033[00m\]: \[\033[01;34m\] \w\[\033[00m\] \$ '
```

 [Article détaillant les codes couleurs et éléments possibles](#) , ou un générateur de prompt en ligne, pratique : <https://ezprompt.net/>

Titre du terminal

Si le terminal est compatible xterm ou rxvt, il change le prompt de manière à changer le titre du terminal.

~/ .bashrc

```
# 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
```

Alias

Les alias sont des outils très pratiques, voir [alias](#) pour une description plus approfondie.

Le fichier ~/ .bashrc commence par définir des alias colorés pour ls dans le cas d'un terminal coloré :

~/bashrc

```
# 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
```

Le fichier propose ensuite de définir quelques alias bien pratiques :

~/bashrc

```
# some more ls aliases
#alias ll='ls -l'
#alias la='ls -A'
#alias l='ls -CF'
```

Enfin, si le fichier ~/.bash_aliases existe, il l'interprète :

~/bashrc

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

Aussi, si vous voulez définir des alias personnels, la méthode la plus propre et portable est de les définir dans un fichier ~/.bash_aliases, à créer.

Autocomplétion

Bash tente enfin de charger l'autocomplétion si elle est disponible et souhaitable :

~/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
```



```
fi
fi
```

"~/bash_logout"

Le fichier ~/bash_logout s'occupe d'effacer le terminal à la déconnexion :

~/bash_logout

```
if [ "$SHLVL" = 1 ]; then
    [ -x /usr/bin/clear_console ] && /usr/bin/clear_console -q
fi
```

"~/bash_history"

Contient tout [l'historique des commandes](#) que vous avez saisies.

Voir aussi

- [Apprendre à rédiger des scripts sous bash](#)

¹⁾
N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !
²⁾
cachés: le nom du fichier commence par un point : « . », ils apparaissent avec un `ls -a`.
³⁾
via `#!/bin/bash`

From:
<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:
<http://debian-facile.org/doc:programmation:bash:config>

Last update: **16/08/2023 01:05**

