

Environment & Customization (Plus)

Linux Commands Course · Section 17

Goal

Understand how to **customize your shell environment** – what runs when you log in, how variables and aliases work, and how to personalize command history.

What Is the Shell Environment?

When you start a shell session, it loads **configuration files** that define your environment.

They control:

- Which variables are set (e.g., PATH)
 - Which aliases and functions are available
 - How your prompt looks
 - Which scripts run at login or for new terminals
-

Profile and RC Files Overview

File	Loaded When	Purpose
<code>~/.bash_profile</code>	login shells	Personal startup settings
<code>~/.bashrc</code>	interactive shells	Aliases, functions, variables
<code>~/.profile</code>	login shells (if no <code>.bash_profile</code>)	Environment setup
<code>/etc/profile</code>	system-wide login setup	Default for all users
<code>/etc/profile.d/*.sh</code>	system-wide scripts	Extend <code>/etc/profile</code>

Login vs Non-Login Shells

- **Login shell** → first shell after login (e.g., via console or SSH).
 - Reads `/etc/profile` → `~/.bash_profile` → optionally `~/.bashrc`.
- **Non-login shell** → when opening a new terminal window or running a script.
 - Reads `~/.bashrc` only.

You can make `.bash_profile` load `.bashrc` manually:

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

Editing .bashrc

Add custom commands, aliases, and variables.

Example entries:

```
# Custom aliases
alias ll='ls -lh --color=auto'
alias grep='grep --color=auto'

# Custom PATH
export PATH="$PATH:$HOME/scripts"

# Custom prompt
export PS1="\u@\h:\w$ "
```

After editing, reload it:

```
source ~/.bashrc
```

Environment Variables – export

List environment variables:

```
printenv
```

Set a variable for current session:

```
MYVAR="hello"  
echo $MYVAR
```

Make it available to child processes:

```
export MYVAR="hello"
```

Unset a variable:

```
unset MYVAR
```

Permanent exports belong in `.bashrc` or `.profile`.

The PATH Variable

`PATH` defines where the shell looks for executables.

View it:

```
echo $PATH
```

Add a new directory to `PATH` (for current session):

```
export PATH="$PATH:$HOME/bin"
```

To make it permanent, add the export line to your `.bashrc`.

Check where a command is found:

```
which python
```

Aliases – Shortcuts for Commands

Create simple command shortcuts.

```
alias cls='clear'  
alias update='sudo apt update && sudo apt upgrade -y'
```

View all aliases:

```
alias
```

Remove an alias:

```
unalias cls
```

For persistence, define them in ~/.bashrc.

Command Completion – bash-completion

`bash-completion` provides smart tab-completion for many commands.

Check if it's installed:

```
type _init_completion
```

Install if missing (Debian/Ubuntu):

```
sudo apt install bash-completion
```

Then source it in your `.bashrc` (if not already):

```
[[ $PS1 && -f /usr/share/bash-completion/bash_completion ]] && . /usr/share/bash-completion/bash_completion
```

Now commands like `git`, `docker`, and `ssh` autocomplete intelligently.

History Behavior – Environment Variables

Customize how Bash records your command history.

HISTCONTROL

Defines how duplicates and leading spaces are handled.

```
export HISTCONTROL=ignoredups:ignorespace
```

Options:

- `ignoredups` – skip duplicate commands
- `ignorespace` – don't save commands starting with a space

HISTTIMEFORMAT

Adds timestamps to history entries.

```
export HISTTIMEFORMAT="%F %T "
```

View history with time:

```
history
```

Other Useful History Variables

Variable	Description
<code>HISTSIZE</code>	number of commands kept in memory
<code>HISTFILESIZE</code>	number of lines kept in history file
<code>HISTFILE</code>	path to history file (usually <code>~/.bash_history</code>)
<code>HISTIGNORE</code>	pattern list to skip saving certain commands

Example:

```
export HISTSIZE=5000
export HISTIGNORE="ls:cd:exit"
```

Making Persistent Customizations

When satisfied with your customizations:

```
source ~/.bashrc
```

To apply system-wide for all users, use `/etc/profile.d/custom.sh`:

```
sudo nano /etc/profile.d/custom.sh
```

Example:

```
export PATH="$PATH:/opt/tools"  
alias ll='ls -lh --color=auto'
```

This will auto-load for all users.

Recap

- **Startup files:** `.bashrc`, `.bash_profile`, `.profile`, `/etc/profile`
- **Variables:** use `export` to persist and `PATH` to find commands
- **Aliases:** make common tasks faster
- **bash-completion:** improves workflow
- **History tuning:** timestamped and filtered history improves traceability

Customizing your shell makes Linux truly *yours*.

Practice

1. Add a custom alias `update` to run system updates.
 2. Add `$HOME/scripts` to your PATH.
 3. Enable timestamps in your history.
 4. Install and test `bash-completion` for `git`.
 5. Create a `/etc/profile.d/myenv.sh` that defines a global variable.
-

Next Up

System Information & Troubleshooting (Plus) – checking health, logs, and performance stats.