1.109.1
Customize and use the shell environment
Weight 5

Linux Professional Institute Certification — 102

Geoffrey Robertson ge@ffrey.com
Nick Urbanik nicku@nicku.org

This document Licensed under GPL—see section 5

2005 July
Outline

Context
Objective
Bash Configuration Files

Aliases
Bash functions
License Of This Document
1.109.1
Customize and use the shell environment
Weight 5

Geoff Robertson

Context
Objective
Bash Configuration Files
Aliases
Bash functions
License Of This Document
1.109.1 **Customise and use the shell environment** [5]
1.109.2 Customise or Write Simple Scripts [3]
Candidate should be able to customize shell environments to meet users’ needs. This objective includes setting environment variables (e.g. `PATH`) at login or when spawning a new shell. It also includes writing `bash` functions for frequently used sequences of commands.
Key files, terms, and utilities include:

- `~/.bash_profile` — sourced when a person logs in
- `~/.bash_login` — sourced when a person logs in
  - if no `~/.bash_profile`
- `~/.profile`  — sourced when a person logs in if no
  - `~/.bash_profile`
  - `~/.bash_login`
- `~/.bashrc` — sourced when a non-login interactive shell starts
- `~/.bash_logout` — sourced when a person logs out
- `~/.inputrc` — allows a user to specify keystrokes for commands
- `function` — (Bash built-in command)
- `export` — make environment variables available to sub processes
When a user logs in to a bash shell the following configuration files are usually executed:

- `/etc/profile` System wide profile, common to all users and shells
- `~/.bash_profile` sourced after `/etc/profile` at login
- `~/.bashrc` sourced after `~/.bash_profile` at login

Note `~/.bashrc` is executed when any new bash shell is spawned.
Bash Configuration Files

- When a user logs in to a `bash` shell the following configuration files are usually executed:
  
  /etc/profile System wide profile, common to all users and shells
  
  ~/.bash_profile sourced after /etc/profile at login
  
  ~/.bashrc sourced after ~/.bash_profile at login

- Note ~/.bashrc is executed when any new bash shell is spawned
Bash Configuration Files

- When a user logs in to a bash shell the following configuration files are usually executed:
  
  /etc/profile System wide profile, common to all users and shells
  
  ~/.bash_profile sourced after /etc/profile at login
  
  ~/.bashrc sourced after ~/.bash_profile at login

- Note ~/.bashrc is executed when any new bash shell is spawned
Bash Configuration Files

► When a user logs in to a `bash` shell the following configuration files are usually executed:

/`etc/profile` System wide profile, common to all users and shells

~/.`bash_profile` sourced after /`etc/profile` at login

~/.`bashrc` sourced after ~/.`bash_profile` at login

► Note ~/.`bashrc` is executed when any new `bash` shell is spawned
Bash Configuration Files

- When a user logs in to a bash shell the following configuration files are usually executed:
  - /etc/profile System wide profile, common to all users and shells
  - ~/.bash_profile sourced after /etc/profile at login
  - ~/.bashrc sourced after ~/.bash_profile at login

- Note ~/.bashrc is executed when any new bash shell is spawned
Bash Aliases
Bash Functions

- Functions work similarly to aliases but allow more complex constructions.
- They have the following syntax:
  
  ```bash
  $ [ function ] NAME() { COMMAND_LIST; }
  ```

- Where
  
  - `function`: Optional tag
  - `NAME()`: The name of the function
  - `COMMAND_LIST`: The body of the function

- Functions may be stored in `~/.bashrc`
Bash Functions

Functions work similarly to aliases but allow more complex constructions.

They have the following syntax:

```bash
$ [ function ] NAME() { COMMAND_LIST; }
```

Where

- `function` Optional tag
- `NAME()` The name of the function
- `COMMAND_LIST` The body of the function

Functions may be stored in `~/.bashrc`
Bash Functions

- Functions work similarly to aliases but allow more complex constructions.

- They have the following syntax:
  
  ```bash
  $ [ function ] NAME() { COMMAND_LIST; }
  ```

- Where

  - `function` Optional tag
  - `NAME()` The name of the function
  - `COMMAND_LIST` The body of the function

- Functions may be stored in `~/.bashrc`
Bash Functions

- Functions work similarly to aliases but allow more complex constructions.
- They have the following syntax:
  
  ```bash
  $ [ function ] NAME() { COMMAND_LIST;}
  ```

- Where
  
  - `function` Optional tag
  - `NAME()` The name of the function
  - `COMMAND_LIST` The body of the function

- Functions may be stored in `~/.bashrc`
Bash Functions

- Functions work similarly to aliases but allow more complex constructions.
- They have the following syntax:

  ```bash
  $ [ function ] NAME() { COMMAND_LIST; }
  ```

- Where

  - `function` Optional tag
  - `NAME()` The name of the function
  - `COMMAND_LIST` The body of the function

- Functions may be stored in `~/.bashrc`
This simple function prints the current working directory and the list of files in it:

```
$ function look() { pwd; ls; }
```

This function would be used like this:

```
$ look
```

```
/home/geoffrey/lpic/general-linux-2/notes
CVS  _whizzy_gl2.notes.fmt
    _whizzy_gl2.notes.pag
```
Bash Functions

Function Example

- This simple function prints the current working directory and the list of files in it:
  ```
  $ function look() { pwd; ls;}
  ```

- This function would be used like this:
  ```
  $ look
  /home/geoffrey/lpic/general-linux-2/notes
  CVS  _whizzy_gl2.notes.fmt
  _whizzy_gl2.notes.pag
  ```
This simple function prints the current working directory and the list of files in it:

```
$ function look() { pwd; ls; }
```

This function would be used like this:

```
$ look
/home/geoffrey/lpic/general-linux-2/notes
CVS _whizzy_gl2.notes.fmt
_whizzy_gl2.notes.pag
```
Bash Functions
Valid Function Definitions

$ function look() { pwd; ls; }
$ function look { pwd; ls; }
$ look() { pwd; ls; }
$ look()
  > {
  >    pwd;
  >    ls;
  >  }

Objective
Bash Configuration
Files
Aliases
Bash functions
License Of This Document
Bash Functions
Valid Function Definitions

$ function look() { pwd; ls; }
$ function look { pwd; ls; }
$ look() { pwd; ls; }
$ look()
  { 
    pwd;
    ls;
  }
Bash Functions
Valid Function Definitions

$ function look() { pwd; ls; }
$ function look { pwd; ls; }
$ look() { pwd; ls; }
$ look()
> {
>   pwd;
>   ls;
> }

$ function look() { pwd; ls; }
$ function look { pwd; ls; }
$ look() { pwd; ls; }
$ look()
Bash Functions
Valid Function Definitions

$ function look() { pwd; ls; }
$ function look { pwd; ls; }
$ look() { pwd; ls; }
$ look()
   { pwd;
   ls;
   }
Bash Functions
Valid Function Definitions

▶ $ function look() { pwd; ls; }
▶ $ function look { pwd; ls; }
▶ $ look() { pwd; ls; }
▶ $ look()
  > {
  >   pwd;
  >   ls;
  > }

Objective
Bash Configuration
Files
Aliases
Bash functions
License Of This
Document
Bash Functions

Invalid Function Definitions

$ function look() pwd; ls;
$ look() { pwd; ls }
$ function look() {pwd; ls;}

Objective

Bash Configuration
Files
Aliases

Bash functions

License Of This Document
Bash Functions

Invalid Function Definitions

► $ function look() pwd; ls;
► $ look() { pwd; ls }
► $ function look() {pwd; ls;}
Bash Functions
Invalid Function Definitions

$ function look() pwd; ls;
$ look() { pwd; ls }
$ function look() {pwd; ls;}

1.109.1
Customize and use the shell environment
Weight 5

Geoff Robertson

Objective
Bash Configuration
Files
Aliases
Bash functions
License Of This Document
Bash Functions
Invalid Function Definitions

$ function look() { pwd; ls; }
$ look() { pwd; ls }
$ function look() { pwd; ls; }

$ function look() { pwd; ls; }
$ look() { pwd; ls }
$ function look() { pwd; ls; }
Bash Functions
Example from Jeffrey Dean’s Nutshell Book

- A function that uses a command line argument:

  ```bash
  $ laps () {
      > ls -l $1
      > ps aux | grep '/usr/bin/basename $1'
  > }
  ```
  
- Use the `laps()` function:

  ```bash
  $ laps /usr/sbin/sshd
  -rwxr-xr-x  1 root  root  276200 Jun 29 01:28 /usr/sbin/sshd
  root  255 0.0 0.3 2792 1216 ?   S   Aug31 0:00 /usr/sbin/sshd
  geoffrey 1187 0.0 0.1 1332 424 pts/1   R   14:39 0:00 grep sshd
  ```
A function that uses a command line argument:

```bash
laps () {
    > ls -l $1
    > ps aux | grep '/usr/bin/basename $1'
    > }
```

Use the `laps()` function:

```bash
laps /usr/sbin/sshd
```

```
-rwxr-xr-x 1 root root 276200 Jun 29 01:28 /usr/sbin/sshd
root 255 0.0 0.3 2792 1216 ? S Aug31 0:00 /usr/sbin/sshd
geoffrey 1187 0.0 0.1 1332 424 pts/1 R 14:39 0:00 grep sshd
```
Bash Functions
Example from Jeffrey Dean’s Nutshell Book

- A function that uses a command line argument:

```bash
$ laps () {
    > ls -l $1
    > ps aux | grep '/usr/bin/basename $1'
    > }
```

- Use the `laps()` function:

```bash
$ laps /usr/sbin/sshd
```

```
-rwxr-xr-x 1 root root 276200 Jun 29 01:28 /usr/sbin/sshd
root  255  0.0  0.3  2792 1216 ?    S   Aug31  0:00 /usr/sbin/sshd
geoffrey 1187  0.0  0.1  1332  424 pts/1    R   14:39  0:00 grep sshd
```
License Of This Document

Copyright © 2005, 2003 Geoffrey Robertson <ge@ffrey.com> and Nick Urbanik <nicku@nicku.org>. Permission is granted to make and distribute verbatim copies or modified versions of this document provided that this copyright notice and this permission notice are preserved on all copies under the terms of the GNU General Public License as published by the Free Software Foundation—either version 2 of the License or (at your option) any later version.