

---

# Testa Shell Documentation

*Release 1.0.0*

**Harun GÜLEÇ**

**Muhammed Ali BURSALI**

May 08, 2017

**Operating Systems (FCS-00017)**

harunglec35@gmail.com

muhammedalibursali@gmail.com

## Contents

<b>1. Short Explanation .....</b>	<b>3</b>
<b>2. Installation .....</b>	<b>3</b>
<b>3. Specification .....</b>	<b>3</b>
<b>3.1 Built-in Functions .....</b>	<b>3</b>
<b>3.1.1 <i>cd</i>.....</b>	<b>3</b>
<b>3.1.2 <i>author</i> .....</b>	<b>3</b>
<b>3.1.3 <i>exit</i> .....</b>	<b>4</b>
<b>3.2 System Calls .....</b>	<b>4</b>
<b>3.3 Synchronously and Asynchronously Commands .....</b>	<b>4</b>
<b>3.4 Running Script.....</b>	<b>4</b>
<b>3.5 History .....</b>	<b>4</b>

## 1. Short Explanation

Testa is a shell or command language interpreter for the Linux operating system. The name is Latin language meaning of “Shell” and it developed in Akademik Delta Research Center Room 909.

Testa is a simply processor that executes commands. It provides most of GNU Utilities. It interprets the commands and also runs own scripts to automate common tasks. Testa may be used interactively or non-interactively. In interactive mode, they accept input typed from the keyboard. When executing non-interactively, Testa executes commands read from a file.

Testa allows execution of GNU commands, both synchronously and asynchronously. As default, it executes synchronously but when user uses & as a parameter, Testa will execute asynchronously.

Testa has some built-in functions. So basic operations can be handled without system call.

## 2. Installation

Testa Shell can easily and fastly compile and install by user.

```
$ make clean
$ make testa
$ make install
```

## 3. Specifications

### 3.1 Built-in Functions

#### 3.1.1 *cd*

*cd* command changes directory.

Usage: prompt>cd documents

#### 3.1.2 *author*

*author* command displays author of Testa.

Usage: prompt>author

### 3.1.3 *exit*

exit command is quits from Testa.

Usage: prompt>exit

## 3.2 System Calls

System commands can be call directly by Testa.

e.g. prompt>ls -alh

## 3.3 Synchronously and Asynchronously Commands

As default, it executes synchronously but when user uses & as a parameter, Testa will execute asynchronously.

e.g.(synchronously) prompt>eclipse

e.g.(asynchronously) prompt>eclipse &  
prompt>author

## 3.4 Running Script

User can run the script by 2 ways:

1. Execute the script using *Shebang* `#!/usr/local/bin/testa`
2. Execute the script as second argument of Testa.  
`./testa script.ts`

## 3.5 History

User can see last 20 previous commands with SIGQUIT (ctrl+\) and these commands save in home folder as testa\_history.