Universidad Rey Juan Carlos

Ingeniería de robótica software

Sistemas Operativos

MASH Documentation Guide

Author:
Javier Izquierdo Hernández

May 15, 2023

Contents

Part I

Introduction

Introduction on the mash shell

Part II

Shell Grammar

The following two tables form the syntax in the shell, being the first one the basic and the next one the extended. Next is some important information about it:
Only for ASCII characters, if used for another type it won't work as expected.
—— = Default or copy
Color = Change in syntax table
Subexec = Substitute by output of command inside

Basic Syntax Table			
Character	Standard	File	Substitution
\0	end_line	end_file	end_sub
\t	blank	end_file_started	end_sub
\n	blank	end_file	end_sub
Space	blank	end_file_started	end_sub
"	——	——	end_sub
#	——	end_file	copy_and_end_sub
$	start_sub	start_sub	copy_and_end_sub
&	background	end_file	end_sub
'	——	——	end_sub
(	error	error	end_sub
)	error	error	end_sub
*	do_glob	do_glob	end_sub
-	——	——	copy_and_end_sub
;	——	end_file	end_sub
<	start_file_in	end_file	end_sub
>	start_file_out	end_file	end_sub
?	do_glob	do_glob	copy_and_end_sub
@	——	——	copy_and_end_sub
[	do_glob	do_glob	end_sub
\	——	——	end_sub
_	——	end_file	copy_and_end_sub
{	here_doc	error	end_sub
}	error	error	end_sub
|	pipe_tok	end_file	end_sub
~	——	——	end_sub
Rest	——	——	——

Extended Syntax Table					
Character	Standard	File	Substitution	'Single Quote'	"Double Quote"
\0	end_line	end_file	end_sub	request_new_line	request_new_line
\t	blank	end_file_started	end_sub	—	—
\n	blank	end_file	end_sub	—	request_new_line
Space	blank	end_file_started	end_sub	—	—
"	start_dquote	start_dquote	end_sub	—	end_dquote
#	comment	end_file	copy_and_end_sub	—	—
$	start_sub	start_sub	copy_and_end_sub	—	start_sub
&	background	end_file	end_sub	—	—
'	start_squote	start_squote	end_sub	end_squote	—
(	error	error	end_sub	—	—
)	error	error	end_sub	—	—
*	do_glob	do_glob	end_sub	—	—
-	—	—	copy_and_end_sub	—	—
;	end_pipe	end_file	end_sub	—	—
<	start_file_in	end_file	end_sub	—	—
>	start_file_out	end_file	end_sub	—	—
?	do_glob	do_glob	copy_and_end_sub	—	—
@	—	—	copy_and_end_sub	—	—
[	do_glob	do_glob	end_sub	—	—
\	escape	escape	end_sub	—	esp_escape
_	—	end_file	copy_and_end_sub	—	—
{	here_doc	error	end_sub	—	—
}	error	error	end_sub	—	—
|	pipe_tok	end_file	end_sub	—	—
~	tilde_tok	tilde_tok	end_sub	—	—
Rest	—	—	—	—	—
&&	and	end_file	end_sub	—	—
||	or	end_file	end_sub	—	—
$(	subexec	subexec	—	—	subexec
&>	start_file_out	end_file	end_sub	—	—
2>	start_file_out	end_file	end_sub	—	—
1>	start_file_out	end_file	end_sub	—	—

Part III

Redirections

Redirecting input

Redirection of input causes the file to be opened for reading on the standard input.

The general format for redirecting input is:

<filename

Redirecting Standard Output

Redirection of output causes the file to be opened for reading on the standard output. If the file does not exist it is created; if it does exist it is truncated to zero size.

The general format for redirecting output is:

>filename

Redirecting Standard Error

Redirection of error causes the file to be opened for reading on the standard error. If the file does not exist it is created; if it does exist it is truncated to zero size.

The general format for redirecting error is:

2>filename

Redirecting Standard Output and Standard Error

This construct allows both the standard output (file descriptor 1) and the standard error output (file descriptor 2) to be redirected to the file.

The general format for redirecting output and error is:

&>filename

Here documents

This type of redirection instructs the shell to read input from the current source until a line containing only delimiter '}' (with no trailing characters) is seen. All of the lines read up to that point are then used as the standard input for a command.

The general format for here documents is:

```
HERE{
    here-document
 }
```

Part IV

Prompting

When executing interactively, mash displays the primary prompt PROMPT when it is ready to read a command, and '> ' when it needs more input to complete a command.
Mash allows these prompt strings to be customized by inserting a number of special strings that are decoded as follows:

- @- does nothing
- @ifcustom if file named .mash_prompt is found in directory the enter if statement
- @ifgit if directory is a git repository then enter if statement
- @else if not enter if statement then exec the following
- @endif end of if statement
- @user the username of the current user
- @where the current working directory, with $HOME abbreviated with a tilde
- @host the hostname
- @custom replace with contents of file .mash_prompt. Use with @ifcustom.
- @gitstatuscolor if git is up to date show color green, if not red
- @gitstatus the number of files to add to commit
- @branch the name of the git branch
- @black show color black
- @red show color red
- @green show color green
- @yellow show color yellow
- @blue show color blue
- @pink show color pink
- @cyan show color cyan
- @white show color white
- @nocolor remove all colors

Part V

Builtins

1 Arithmetic evaluation

1.1 Description

Write result of the arithmetic expression to the standard output.

1.2 Usage

$((expression))

1.3 Extended description

Evaluation is done in fixed-width integers with no check for overflow, though division by 0 is considered as an error. The operators and their precedence, associativity, and values are the same as in the C language. Shell variables are allowed as operands; parameter expansion is performed before the expression is evaluated. Within an expression, shell variables may also be referenced by name without using the $ character.

1.4 Operands

- + addition
- - subtraction
- * multiplication
- / division
- ^ exponentiation

1.5 Exit Status

Returns success unless an error in the expression is found.

2 Alias

2.1 Description

Define or display aliases.

2.2 Usage

alias [name=value]

2.3 Extended description

Without arguments, ‘alias’ prints the list of aliases in the reusable form ‘alias NAME=VALUE’ on standard output.

Otherwise, an alias is defined for each NAME whose VALUE is given.

2.4 Exit Status

alias returns 0 unless a VALUE is missing or the is out of memory.

3 Bg

3.1 Description

Move jobs to the background.

3.2 Usage

bg [jobspec]

3.3 Extended description

Place the jobs identified by the JOB_SPEC in the background, as if they had been started with '&'.

If JOB_SPEC is not present, the shell's notion of the current job is used.

3.4 Exit Status

Returns success unless job control is not enabled or an error occurs.

4 Builtin

4.1 Description

Execute shell builtins.

4.2 Usage

builtin shell-builtin [arg ..]

4.3 Extended description

Execute SHELL-BUILTIN with arguments ARGs without performing command lookup.

4.4 Exit Status

Returns the exit status of SHELL-BUILTIN, or 1 if SHELL-BUILTIN not a shell builtin.

5 Cd

5.1 Description

Change the shell working directory.

5.2 Usage

cd [directory]

5.3 Extended description

Change the current directory to DIR. The default DIR is the value of the HOME shell variable.

5.4 Exit Status

Returns 0 if the directory is changed, and non-zero otherwise.

6 Command

6.1 Description

Execute a simple command or display information about commands.

6.2 Usage

command [-Vv] command [arg ..]

6.3 Extended description

Runs COMMAND with ARGS suppressing shell function lookup, or display information about the specified COMMANDs.

6.4 Options

- -v print a description of COMMAND similar to the ‘type’ builtin
- -V print a more verbose description of each COMMAND

6.5 Exit Status

Returns exit status of COMMAND, or failure if COMMAND is not found.

7 Disown

7.1 Description

Remove jobs from current shell.

7.2 Usage

disown [-ar] [jobspec ... | pid ...]

7.3 Extended description

Removes each JOBSPEC argument from the table of active jobs. Without any JOBSPECs, the shell uses its notion of the current job.

7.4 Options

- -a remove all jobs if JOBSPEC is not supplied
- -r remove only running jobs

7.5 Exit Status

Returns success unless an invalid option or JOBSPEC is given, or job control is not enabled.

8 Echo

8.1 Description

Write arguments to the standard output.

8.2 Usage

echo [-n] [arg ...]

8.3 Extended description

Display the ARGs, separated by a single space character and followed by a newline, on the standard output.

8.4 Options

- -n do not append a newline

8.5 Exit Status

Returns success unless a write error occurs.

9 Exit

9.1 Description

Exit the shell.

9.2 Usage

exit [n]

9.3 Extended description

Exits the shell with a status of N. If N is omitted, the exit status is that of the last command executed.

10 Export

10.1 Description

Set export attribute for shell variables.

10.2 Usage

export [name=value]

10.3 Extended description

Marks each NAME for automatic export to the environment of subsequently executed commands. If VALUE is supplied, assign VALUE before exporting.

10.4 Exit Status

Returns success unless an invalid option is given or NAME is invalid.

11 Fg

11.1 Description

Move job to the foreground.

11.2 Usage

fg [jobspec]

11.3 Extended description

Place the job identified by JOB_SPEC in the foreground, making it the current job. If JOB_SPEC is not present, the shell's notion of the current job is used.

11.4 Exit Status

Status of command placed in foreground, or failure if an error occurs or job control is not enabled.

12 Help

12.1 Description

Display information about builtin commands.

12.2 Usage

help [-dms] [pattern ...]

12.3 Extended description

Displays brief summaries of builtin commands. If PATTERN is specified, gives detailed help on all commands matching PATTERN, otherwise the list of help topics is printed.

12.4 Options

- -d output short description for each topic
- -m display usage in pseudo-manpage format
- -s output only a short usage synopsis for each topic matching PATTERN

12.5 Arguments

- PATTERN Pattern specifying a help topic

12.6 Exit Status

Returns success unless PATTERN is not found or an invalid option is given.

13 Ifnot

13.1 Description

Execute the command if previous command failed.

13.2 Usage

ifnot command [arg ..]

13.3 Extended description

Runs COMMAND with ARGS if the last executed command finished with failure.

13.4 Exit Status

Returns exit status of COMMAND, or success if last command ended with success.
If enviroment variable 'result' does not exist returns failure.

14 Ifok

14.1 Description

Execute the command if previous command ended successfully.

14.2 Usage

ifok command [arg ..]

14.3 Extended description

Runs COMMAND with ARGS if the last executed command finished with success.

14.4 Exit Status

Returns exit status of COMMAND, or success if last command ended with failure.
If enviroment variable 'result' does not exist returns failure.

15 Jobs

15.1 Description

Display status of jobs.

15.2 Usage

jobs [-lprs] [jobspec]

15.3 Extended description

Lists the active jobs. JOBSPEC restricts output to that job.
Without options, the status of all active jobs is displayed.

15.4 Options

- -l lists process IDs in addition to the normal information
- -p lists process IDs only
- -r restrict output to running jobs
- -s restrict output to stopped jobs

15.5 Exit Status

Returns success unless an invalid option is given, an error occurs or job control is not enabled.

16 Kill

16.1 Description

Send a signal to a job.

16.2 Usage

kill [-s sigspec] | [-n signum] | [-sigspec] jobspec or pid or kill -l [sigspec]

16.3 Extended description

Send the processes identified by PID or JOBSPEC the signal named by SIGSPEC or SIGNUM. If neither SIGSPEC nor SIGNUM is present, then SIGTERM is assumed.

16.4 Options

- -s sig SIG is a signal name
- -n sig SIG is a signal number
- -l list the signal names; if an argument follows '-l' it is assumed to be a signal name for which number should be listed
- -L synonym for -l

16.5 Exit Status

Returns success unless an invalid option is given or an error occurs or job control is not enabled.

17 Math

17.1 Description

Write result of the arithmetic expression to the standard output.

17.2 Usage

math expression

17.3 Extended description

Display the result of the arithmetic expression, followed by a newline, on the standard output.

17.4 Operands

- + addition
- - subtraction
- * multiplication
- / division
- ^ exponentiation

17.5 Exit Status

Returns success unless an error in the expression is found.

18 Pwd

18.1 Description

Print the name of the current working directory.

18.2 Usage

pwd

18.3 Extended description

Print the name of the current working directory.

18.4 Exit Status

Returns 0 unless an invalid option is given or the current directory cannot be read.

19 Sleep

19.1 Description

Pause for NUMBER seconds.

19.2 Usage

sleep NUMBER[SUFFIX]...

19.3 Extended description

Pause for NUMBER seconds. SUFFIX may be 's' for seconds (default),'m' for minutes, 'h' for hours or 'd' for days.
NUMBER need to be an integer.
Given two or more arguments, pause for the amount of time specified by the sum of their values.

19.4 Exit Status

Returns success unless an invalid option or time is given.

20 Source

20.1 Description

Execute commands from a file in the current shell.

20.2 Usage

source filename

20.3 Extended description

Read and execute commands from FILENAME in the current shell. The entries in $PATH are used to find the directory containing FILENAME.

20.4 Exit Status

Returns success unless FILENAME cannot be read.

21 Wait

21.1 Description

Wait for job completion and return exit status.

21.2 Usage

wait [jobspec or id]

21.3 Extended description

Waits for each process identified by an ID, which may be a process ID or a job specification, and reports its termination status.
If ID is not given, waits for the current active child processes, and the return it's status. If ID is a job specification, waits for all processes in that job's pipeline.

21.4 Exit Status

Returns the status of the last ID; fails if ID is invalid, stopped, an invalid option is given or job control is not enabled.

Bibliography

[1] Brian Fox and Chet Ramey, “Bash man page“.

[2] Byron Rakitzis, “Rc shell“.