GNU/Linux - An Introduction
Part 8-10
⟶ BASH Programming II⟵

Prasad H. L.

March 14, 2008; March 19, 2008; March 21, 2008

Review

- Bash Concepts/Definitions
- Bash Grammer - Simple Commands, Pipelines, Lists, Expressions

1. Bash Grammer
   - Return Status
   - Compound Commands
   - Quoting
   - Parameters/Variables
   - Expansions
2. Summary

Return Status I

Interpretation

- 0 $\Rightarrow$ Success/True
- (! 0) $\Rightarrow$ Failure/False
- Note: Arithmetic expressions still follow C-style.

Return Status I

Interpretation

- 0 $\Rightarrow$ Success/True
- (! 0) $\Rightarrow$ Failure/False
- Note: Arithmetic expressions still follow C-style.

Return Status I

Interpretation

- 0 $\Rightarrow$ Success/True
- (! 0) $\Rightarrow$ Failure/False
- Note: Arithmetic expressions still follow C-style.

Return Status I

Interpretation

- 0 $\Rightarrow$ Success/True
- (! 0) $\Rightarrow$ Failure/False
- Note: Arithmetic expressions still follow C-style.

Return Status II

Command Types

- Simple Command
  - Return status of the command
  - & ⇒ return status is 0.
- Pipelines - return status of last command
- Lists - return status of last pipeline
- Expressions
  - Arithmetic -
    - 0 ⇒ successful and non-zero result
    - (! 0) ⇒ unsuccessful or zero result
  - Conditional -
    - 0 ⇒ successful and true condition
    - (! 0) ⇒ unsuccessful or false condition

Return Status II

Command Types

- Simple Command
  - Return status of the command
  - & $\Rightarrow$ return status is 0.
- Pipelines - return status of last command
- Lists - return status of last pipeline
- Expressions
  - Arithmetic -
    - 0 $\Rightarrow$ successful and non-zero result
    - (! 0) $\Rightarrow$ unsuccessful or zero result
  - Conditional -
    - 0 $\Rightarrow$ successful and true condition
    - (! 0) $\Rightarrow$ unsuccessful or false condition

Return Status II

Command Types

- Simple Command
  - Return status of the command
  - & $\Rightarrow$ return status is 0.
- Pipelines - return status of last command
- Lists - return status of last pipeline
- Expressions
  - Arithmetic -
    - 0 $\Rightarrow$ successful and non-zero result
    - (! 0) $\Rightarrow$ unsuccessful or zero result
  - Conditional -
    - 0 $\Rightarrow$ successful and true condition
    - (! 0) $\Rightarrow$ unsuccessful or false condition

Return Status II

Command Types

- Simple Command
  - Return status of the command
  - & $\Rightarrow$ return status is 0.
- Pipelines - return status of last command
- Lists - return status of last pipeline
- Expressions
  - Arithmetic -
    - 0 $\Rightarrow$ successful and non-zero result
    - (! 0) $\Rightarrow$ unsuccessful or zero result
  - Conditional -
    - 0 $\Rightarrow$ successful and true condition
    - (! 0) $\Rightarrow$ unsuccessful or false condition

Return Status II

Command Types

- Simple Command
  - Return status of the command
  - & $\Rightarrow$ return status is 0.
- Pipelines - return status of last command
- Lists - return status of last pipeline
- Expressions
  - Arithmetic -
    - 0 $\Rightarrow$ successful and non-zero result
    - (! 0) $\Rightarrow$ unsuccessful or zero result
  - Conditional -
    - 0 $\Rightarrow$ successful and true condition
    - (! 0) $\Rightarrow$ unsuccessful or false condition

Return Status II

Command Types

- Simple Command
  - Return status of the command
  - & $\Rightarrow$ return status is 0.
- Pipelines - return status of last command
- Lists - return status of last pipeline
- Expressions
  - Arithmetic -
    - 0 $\Rightarrow$ successful and non-zero result
    - (! 0) $\Rightarrow$ unsuccessful or zero result
  - Conditional -
    - 0 $\Rightarrow$ successful and true condition
    - (! 0) $\Rightarrow$ unsuccessful or false condition

Return Status II

Command Types

- Simple Command
  - Return status of the command
  - & $\Rightarrow$ return status is 0.
- Pipelines - return status of last command
- Lists - return status of last pipeline
- Expressions
  - Arithmetic -
    - 0 $\Rightarrow$ successful and non-zero result
    - (! 0) $\Rightarrow$ unsuccessful or zero result
  - Conditional -
    - 0 $\Rightarrow$ successful and true condition
    - (! 0) $\Rightarrow$ unsuccessful or false condition

Return Status II

Command Types

- Simple Command
  - Return status of the command
  - & $\Rightarrow$ return status is 0.
- Pipelines - return status of last command
- Lists - return status of last pipeline
- Expressions
  - Arithmetic -
    - 0 $\Rightarrow$ successful and non-zero result
    - (! 0) $\Rightarrow$ unsuccessful or zero result
  - Conditional -
    - 0 $\Rightarrow$ successful and true condition
    - (! 0) $\Rightarrow$ unsuccessful or false condition

Return Status II

Command Types

- Simple Command
  - Return status of the command
  - & $\Rightarrow$ return status is 0.
- Pipelines - return status of last command
- Lists - return status of last pipeline
- Expressions
  - Arithmetic -
    - 0 $\Rightarrow$ successful and non-zero result
    - (! 0) $\Rightarrow$ unsuccessful or zero result
  - Conditional -
    - 0 $\Rightarrow$ successful and true condition
    - (! 0) $\Rightarrow$ unsuccessful or false condition

Return Status II

Command Types

- Simple Command
  - Return status of the command
  - & ⇒ return status is 0.
- Pipelines - return status of last command
- Lists - return status of last pipeline
- Expressions
  - Arithmetic -
    - 0 ⇒ successful and non-zero result
    - (! 0) ⇒ unsuccessful or zero result
  - Conditional -
    - 0 ⇒ successful and true condition
    - (! 0) ⇒ unsuccessful or false condition

Return Status II

Command Types

- Simple Command
  - Return status of the command
  - & $\Rightarrow$ return status is 0.
- Pipelines - return status of last command
- Lists - return status of last pipeline
- Expressions
  - Arithmetic -
    - 0 $\Rightarrow$ successful and non-zero result
    - (! 0) $\Rightarrow$ unsuccessful or zero result
  - Conditional -
    - 0 $\Rightarrow$ successful and true condition
    - (! 0) $\Rightarrow$ unsuccessful or false condition

Return Status II

Command Types

- Simple Command
  - Return status of the command
  - & $\Rightarrow$ return status is 0.
- Pipelines - return status of last command
- Lists - return status of last pipeline
- Expressions
  - Arithmetic -
    - 0 $\Rightarrow$ successful and non-zero result
    - (! 0) $\Rightarrow$ unsuccessful or zero result
  - Conditional -
    - 0 $\Rightarrow$ successful and true condition
    - (! 0) $\Rightarrow$ unsuccessful or false condition

Return Status II

Command Types

- Simple Command
  - Return status of the command
  - & $\Rightarrow$ return status is 0.
- Pipelines - return status of last command
- Lists - return status of last pipeline
- Expressions
  - Arithmetic -
    - 0 $\Rightarrow$ successful and non-zero result
    - (! 0) $\Rightarrow$ unsuccessful or zero result
  - Conditional -
    - 0 $\Rightarrow$ successful and true condition
    - (! 0) $\Rightarrow$ unsuccessful or false condition

Return Status II

Command Types

- Simple Command
  - Return status of the command
  - & $\Rightarrow$ return status is 0.
- Pipelines - return status of last command
- Lists - return status of last pipeline
- Expressions
  - Arithmetic -
    - 0 $\Rightarrow$ successful and non-zero result
    - (! 0) $\Rightarrow$ unsuccessful or zero result
  - Conditional -
    - 0 $\Rightarrow$ successful and true condition
    - (! 0) $\Rightarrow$ unsuccessful or false condition

Compound Commands I

- (list)
- { list; }
- ((arithexpr))
- [[conditionexpr]], combination with ( ), !, &&, ||
- [conditionexpr], test conditionexpr

Compound Commands I

- (list)
- { list; }
- ((arithexpr))
- [[conditionexpr]], combination with ( ), !, &&, ||
- [conditionexpr], test conditionexpr

Compound Commands I

- (list)
- { list; }
- ((arithexpr))
- [[conditionexpr]], combination with ( ), !, &&, ||
- [conditionexpr], test conditionexpr

Compound Commands I

- (list)
- { list; }
- ((arithexpr))
- [[conditionexpr]], combination with ( ), !, &&, ||
- [conditionexpr], test conditionexpr

Compound Commands I

- (list)
- { list; }
- ((arithexpr))
- [[conditionexpr]], combination with ( ), !, &&, ||
- [conditionexpr], test conditionexpr

Compound Commands II

- Condition Structure -
  if list; then list; [elif list; then list; [else list;]] fi
- Iteration Structures -
  - For loop - for name [in word] do list done
  - For loop (2) -

Compound Commands II

- Condition Structure -
  if list; then list; [elif list; then list; [else list;]] fi
- Iteration Structures -
  - For loop - for name [in word]; do list; done
  - For loop (2) - for ((expr1; expr2; expr3)); do list; done
  - While loop - while list; do list; done
  - Until loop - until list; do list; done

Compound Commands II

- Condition Structure -
  if list; then list; [elif list; then list; [else list;]] fi
- Iteration Structures -
  - For loop - for name [in word]; do list; done
  - For loop (2) - for ((expr1; expr2; expr3)); do list; done
  - While loop - while list; do list; done
  - Until loop - until list; do list; done

Compound Commands II

- Condition Structure -
  if list; then list; [elif list; then list; [else list;]] fi
- Iteration Structures -
  - For loop - for name [in word]; do list; done
  - For loop (2) - for ((expr1; expr2; expr3)); do list; done
  - While loop - while list; do list; done
  - Until loop - until list; do list; done

Compound Commands II

- Condition Structure -
  if list; then list; [elif list; then list; [else list;]] fi
- Iteration Structures -
  - For loop - for name [in word]; do list; done
  - For loop (2) - for ((expr1; expr2; expr3)); do list; done
  - While loop - while list; do list; done
  - Until loop - until list; do list; done

Compound Commands II

- Condition Structure -
  if list; then list; [elif list; then list; [else list;]] fi
- Iteration Structures -
  - For loop - for name [in word]; do list; done
  - For loop (2) - for ((expr1; expr2; expr3)); do list; done
  - While loop - while list; do list; done
  - Until loop - until list; do list; done

Compound Commands III

- Case Structure -
  case word in [[(] pattern [|pattern] ...) list ;;] ... esac
- Select Structure -
  select name [in word] ; do list ; done
- Functions -
  [function] name () compound-command [redirection]

Compound Commands III

- Case Structure -
  case word in [[(] pattern [|pattern] ...) list ;;] ... esac
- Select Structure -
  select name [in word] ; do list ; done
- Functions -
  [function] name () compound-command [redirection]

Compound Commands III

- Case Structure -
  case word in [[(] pattern [|pattern] ...) list ;;] ... esac
- Select Structure -
  select name [in word] ; do list ; done
- Functions -
  [function] name () compound-command [redirection]

Quoting

- Backslash
- Double Quotes
- Single Quotes
- $' '
- Quote Removal

Quoting

- Backslash
- Double Quotes
- Single Quotes
- $' '
- Quote Removal

Quoting

- Backslash
- Double Quotes
- Single Quotes
- $' '
- Quote Removal

Quoting

- Backslash
- Double Quotes
- Single Quotes
- $' '
- Quote Removal

Quoting

- Backslash
- Double Quotes
- Single Quotes
- $' '
- Quote Removal

Parameters/Variables

Gross Definition

- (name)
- Value
- attributes

Types

- Positional
- Special ($* $@ $# $? $- $$ $! $0)
- Shell Variables
- Arrays ([], * @)

Parameters/Variables

Gross Definition

- (name)
- Value
- attributes

Types

- Positional
- Special (* @ # ? - $! 0)
- Shell Variables
- Arrays ([], * @)

Parameters/Variables

Gross Definition

- (name)
- Value
- attributes

Types

- Positional
- Special (* @ # ? - $ | 0)
- Shell Variables
- Arrays ([], * @)

Parameters/Variables

Gross Definition

- (name)
- Value
- attributes

Types

- Positional
- Special (* @ # ? - $ | 0)
- Shell Variables
- Arrays ([ ], * @)

Parameters/Variables

Gross Definition

- (name)
- Value
- attributes

Types

- Positional
- Special (* @ # ? - $ | 0)
- Shell Variables
- Arrays ([ ], * @)

Parameters/Variables

Gross Definition

- (name)
- Value
- attributes

Types

- Positional
- Special (* @ # ? - $ | 0)
- Shell Variables
- Arrays ([ ], * @)

Parameters/Variables

Gross Definition

- (name)
- Value
- attributes

Types

- Positional
- Special (* @ # ? - $ | 0)
- Shell Variables
- Arrays ([ ], * @)

Expansions I

Brace Expansions

- { }
- Types - , and ..

Tilde Expansions

Home directory

Parameter Expansions

Expansions I

Brace Expansions

- { }
- Types - , and ..

Tilde Expansions

Home directory

Parameter Expansions

Expansions I

Brace Expansions

- { }
- Types - , and ..

Tilde Expansions

Home directory

Parameter Expansions

Expansions I

Brace Expansions

- { }
- Types - , and ..

Tilde Expansions

Home directory

Parameter Expansions

- ${param}
- ${!prefix*}, ${!prefix@}

Expansions I

Brace Expansions

- { }
- Types - , and ..

Tilde Expansions

Home directory

Parameter Expansions

- ${param}
- ${!prefix*}, ${!prefix@}
- ${!name[*]}, ${!name[@]}
- ${#param}
- ${param:offset:length}, etc.

Expansions I

Brace Expansions

- { }
- Types - , and ..

Tilde Expansions

Home directory

Parameter Expansions

- ${param}
- ${!prefix*}, ${!prefix@}
- ${!name[*]}, ${!name[@]}
- ${#param}
- ${param:offset:length}, etc.

Expansions I

Brace Expansions

- { }
- Types - , and ..

Tilde Expansions

Home directory

Parameter Expansions

- ${param}
- ${!prefix*}, ${!prefix@}
- ${!name[*]}, ${!name[@]}
- ${#param}
- ${param:offset:length}, etc.

Expansions I

Brace Expansions

- { }
- Types - , and ..

Tilde Expansions

Home directory

Parameter Expansions

- ${param}
- ${!prefix*}, ${!prefix@}
- ${!name[*]}, ${!name[@]}
- ${#param}
- ${param:offset:length}, etc.

Expansions I

Brace Expansions

- { }
- Types - , and ..

Tilde Expansions

Home directory

Parameter Expansions

- ${param}
- ${!prefix*}, ${!prefix@}
- ${!name[*]}, ${!name[@]}
- ${#param}
- ${param:offset:length}, etc.

Expansions II

Command Substitution

- Back Quotes - ‘command‘
- $(command)

Arithmetic Expansions

- $((arithexpr))
- $[arithexpr]

Expansions II

Command Substitution

- Back Quotes - ‘command‘
- $(command)

Arithmetic Expansions

- $((arithexpr))
- $[arithexpr]

Expansions II

Command Substitution

- Back Quotes - ‘command‘
- $(command)

Arithmetic Expansions

- $((arithexpr))
- $[arithexpr]

Expansions II

Command Substitution

- Back Quotes - ‘command‘
- $(command)

Arithmetic Expansions

- $((arithexpr))
- $[arithexpr]

Expansions III

Process Substitution

- <(list)
- >(list)

Pathname Expansions

- *
- ?
- []

Expansions III

Process Substitution

- <(list)
- >(list)

Pathname Expansions

- *
- ?
- [ ]

Expansions III

Process Substitution

- <(list)
- >(list)

Pathname Expansions

- *
- ?
- [ ]

Expansions III

Process Substitution

- <(list)
- >(list)

Pathname Expansions

- *
- ?
- [ ]

Expansions III

Process Substitution

- <(list)
- >(list)

Pathname Expansions

- *
- ?
- [ ]

Summary

- BASH Compound Commands
- BASH Quoting
- BASH Parameters
- BASH Expansions

References

- Machtelt Garrels. “Bash Guide for Beginners”. Version 1.8. WWW link - `http://tldp.org/LDP/Bash-Beginners-Guide/Bash-Beginners-Guide.pdf`.
- man bash, man:/bash
- info bash, info:/bash
- Mendel Cooper. “Advanced Bash-Scripting Guide”. Version 5.1. WWW Link - `http://tldp.org/LDP/abs/abs-guide.pdf`.