POSIX-2017

Shell & Utilities

Shell Command Language

1. Shell Introduction

2. Quoting
  | & ; < > ( ) $ ` \ " ' <space> <tab> <newline>
  *** ? [# ~ = %**
    1. Escape Character (Backslash)
    2. Single-Quotes
    3. Double-Quotes

3. Token Recognition
    1. Alias Substitution

4. Reserved Words

!	**do**	**esac**	**in**
{	**done**	**fi**	**then**
}	**elif**	**for**	**until**
case	**else**	**if**	**while**

5. Parameters and Variables
    1. Positional Parameters
    2. Special Parameters
    3. Shell Variables

XBD ***Environment Variables***

ARFLAGS	**IFS**	MAILPATH	**PS1**
CC	**LANG**	MAILRC	**PS2**
CDPATH	**LC_ALL**	MAKEFLAGS	PS3
CFLAGS	**LC_COLLATE**	MAKESHELL	**PS4**
CHARSET	**LC_CTYPE**	MANPATH	**PWD**
COLUMNS	**LC_MESSAGES**	MBOX	RANDOM
DATEMSK	LC_MONETARY	MORE	SECONDS
DEAD	LC_NUMERIC	MSGVERB	SHELL
EDITOR	LC_TIME	**NLSPATH**	TERM
ENV	LDFLAGS	NPROC	TERMCAP
EXINIT	LEX	OLDPWD	TERMINFO
FC	LFLAGS	OPTARG	TMPDIR
FCEDIT	**LINENO**	OPTERR	TZ
FFLAGS	LINES	OPTIND	USER
GET	LISTER	PAGER	VISUAL
GFLAGS	LOGNAME	**PATH**	YACC
HISTFILE	LPDEST	**PPID**	YFLAGS
HISTORY	MAIL	PRINTER	
HISTSIZE	MAILCHECK	PROCLANG	
HOME	MAILER	PROJECTDIR	

6. Word Expansions
    1. Tilde Expansion
    2. Parameter Expansion
      ${*parameter***}**
      ${*parameter*:-[*word*]**}**
      ${*parameter*:=[*word*]**}**
      ${*parameter*:?[*word*]**}**
      ${*parameter*:+[*word*]**}**
      ${#*parameter***}**
      ${*parameter*%[*word*]**}**
      ${*parameter*%%[*word*]**}**
      ${*parameter*#[*word*]**}**
      ${*parameter*##[*word*]**}**
    3. Command Substitution
      $(*command***)**
    4. Arithmetic Expansion
      $((*expression***))**
    5. Field Splitting
    6. Pathname Expansion
    7. Quote Removal

7. Redirection
  [*n***]***redir-op word*
    1. Redirecting Input
    2. Redirecting Output
    3. Appending Redirected Output
    4. Here-Document
    5. Duplicating an Input File Descriptor

6. Duplicating an Output File Descriptor
7. Open File Descriptors for Reading and Writing

8. Exit Status and Errors
   1. Consequences of Shell Errors
   2. Exit Status for Commands

9. Shell Commands
   1. Simple Commands
   2. Pipelines

      [!] *command1* [**|** *command2* ...]

   3. Lists

      command1 **&** [*command2* **&** ...]
      command1 [**;** *command2*] ...
      command1 [**&&** *command2*] ...
      command1 [**||** *command2*] ...

   4. Compound Commands

      (*compound-list* **)**
      { *compound-list* ; **}**

      for *name* [**in** [*word* ...]]
      do
            compound-list
      done

      case *word* **in**
            [(] *pattern1*) *compound-list* ;;
            [[(] *pattern*[| *pattern*] ...) *compound-list* ;;]
            [[(] *pattern*[| *pattern*] ...) *compound-list*]
      esac

      if *compound-list*
      then
            compound-list
      [**elif** *compound-list*
      then
            compound-list] ...
      [**else**
            compound-list]
      fi

      while *compound-list-1*
      do
            compound-list-2
      done

      until *compound-list-1*
      do
            compound-list-2
      done

   5. Function Definition Command

      fname ( ) *compound-command* **[***io-redirect* ...**]**

10. Shell Grammar
    1. Shell Grammar Lexical Conventions
    2. Shell Grammar Rules

11. Signals and Error Handling

12. Shell Execution Environment

13. Pattern Matching Notation
    1. Patterns Matching a Single Character
    2. Patterns Matching Multiple Characters
    3. Patterns Used for Filename Expansion

14. Special Built-In Utilities
    - **break** - exit from for, while, or until loop
    - **colon** - null utility
    - **continue** - continue for, while, or until loop
    - **dot** - execute commands in the current environment
    - **eval** - construct command by concatenating arguments
    - **exec** - execute commands and open, close, or copy file descriptors
    - **exit** - cause the shell to exit
    - **export** - set the export attribute for variables
    - **readonly** - set the readonly attribute for variables
    - **return** - return from a function or dot script
    - **set** - set or unset options and positional parameters
    - **shift** - shift positional parameters
    - **times** - write process times
    - **trap** - trap signals
    - **unset** - unset values and attributes of variables and functions