

AWK
AWK



Canine-Table



POSIX Nexus serves as a comprehensive cross-language reference hub that explores the implementation and behavior of POSIX-compliant functionality across a diverse set of programming environments. Built atop the foundational IEEE Portable Operating System Interface (POSIX) standards, this project emphasizes compatibility, portability, and interoperability between operating systems.

Abstract

Contents

I	Shell Modules	II
I	The Argument Parser	II
I	Option Separators	II
I	Stride proof and runtime mapping	IV



I Shell Modules

I The Argument Parser

I Option Separators

Variable Positions in trk

Index	Default	Variable	Meaning
—	‘;’	<i>ds</i>	Default delimiter string
1	‘.’	<i>ks</i>	Key separator
2	‘&’	<i>als</i>	Alias / altname separator
3	‘@’	<i>fas</i>	Appendable flag array separator
4	‘#’	<i>kas</i>	Appendable keyword array separator
5	‘<’	<i>go</i>	Begin group marker
6	‘>’	<i>gc</i>	End group marker
7	‘ ’	<i>lo</i>	Begin or continue long option mode
8	‘;’	<i>lc</i>	End long option mode
9	‘=’	<i>fs</i>	Optional set flag separator
10	‘+’	<i>fsa</i>	Optional push flag separator
11	‘-’	<i>fss</i>	Optional pop flag separator

Short vs Long Option Group Leaders

- ➔ **Short form** ⇒ Input: ‘alpha<beta gamma>’ → Flags = a, l, p, h; Group leader = a; Members = b, e, t, a, gamma
- ➔ **Long form** ⇒ Input: ‘ alpha<beta gamma>’ → Group leader = alpha; Members = beta, gamma
- ➔ **Key distinction** ⇒ Leading space (‘lo’) preserves token as long option; without space, token explodes into flags and first flag becomes leader



Defaults for Option Groups

- ➔ Group 1 ⇒ '#' keyword arrays
- ➔ Group 2 ⇒ '@' flag arrays
- ➔ Group 3 ⇒ ':' keywords
- ➔ Group 4 ⇒ ' ' flags
- ➔ Alias Requirement ⇒ Each group requires one alias slot
- ➔ Stride Variable ⇒ Denoted by S
- ➔ Group Count ⇒ Denoted by G
- ➔ Alias Slots ⇒ Denoted by A

Group-of-Groups Behavior

- ⌘ Input ~> ' alpha<beta gamma> delta<epsilon zeta>'
- ⌘ Expansion ~> --beta --gamma --alpha --gamma --beta --epsilon
--gamma --zeta --delta --zeta
- ⌘ NEX_gf_alpha ~> Group 1 index → beta<nx:null/>gamma
- ⌘ Group 1 index ~> Group 2 index → gamma<nx:null/>beta<nx:null/>gamma
- ⌘ NEX_gf_delta ~> Group 1 index → epsilon<nx:null/>zeta
- ⌘ NEX_gf_delta ~> Group 2 index → zeta

```

1 debug_level = 2
2 string_value = " alpha<beta gamma> delta<epsilon zeta>"
3 nx_sh_opts(string_value, debug_level, arr)

```



I Stride proof and runtime mapping

Variables

- ➔ **D1** \Rightarrow Options string
- ➔ **go,gc** \Rightarrow Group open/close markers
- ➔ **als** \Rightarrow Alias symbol
- ➔ **D2** \Rightarrow Runtime group stride contribution (S_{groups})
- ➔ **strde** \Rightarrow Final array stride (S_{total})

$$\begin{aligned}
 A_g &= \text{leader aliases} + \text{member aliases} \\
 S_g &= 1 + A_g \quad (\text{canonical leader slot} + \text{alias slots}) \\
 S_{\text{groups}} &= \sum_g S_g = \sum_g (1 + A_g) \\
 S_{\text{total}} &= 12 + S_{\text{groups}}
 \end{aligned}$$

Runtime correspondence

Symbol	Runtime	Meaning
S_{groups}	$D2$	nx_sh_stride count (groups + aliases)
S_{total}	$strde$	12 + D2 (final stride for nx_parr_stk)

Anchors spaced by S_{total}

Category	Base index	Walk by
Keywords	1	$+ S_{\text{total}}$
Flags	4	$+ S_{\text{total}}$
Flag arrays	7	$+ S_{\text{total}}$
Keyword arrays	10	$+ S_{\text{total}}$
Groups (triplets)	13	$+ 3$ per group header, values spaced by S_{total}

