

| | | |
|------------------|---|-----------|
| wcslen | <i>Wide-String Length (C99)</i> | <wchar.h> |
| | <code>size_t wcslen(const wchar_t *s);</code> | |
| | Wide-character version of strlen. | 25.5 |
| wcsncat | <i>Bounded Wide-String Concatenation (C99)</i> | <wchar.h> |
| | <code>wchar_t *wcsncat(wchar_t * restrict s1, const wchar_t * restrict s2, size_t n);</code> | |
| | Wide-character version of strncat. | 25.5 |
| wcsncmp | <i>Bounded Wide-String Comparison (C99)</i> | <wchar.h> |
| | <code>int wcsncmp(const wchar_t *s1, const wchar_t *s2, size_t n);</code> | |
| | Wide-character version of strncmp. | 25.5 |
| wcsncpy | <i>Bounded Wide-String Copy (C99)</i> | <wchar.h> |
| | <code>wchar_t *wcsncpy(wchar_t * restrict s1, const wchar_t * restrict s2, size_t n);</code> | |
| | Wide-character version of strncpy. | 25.5 |
| wcspbrk | <i>Search Wide String for One of a Set of Characters (C99)</i> | <wchar.h> |
| | <code>wchar_t *wcspbrk(const wchar_t *s1, const wchar_t *s2);</code> | |
| | Wide-character version of strpbrk. | 25.5 |
| wcsrchr | <i>Search Wide String in Reverse for Character (C99)</i> | <wchar.h> |
| | <code>wchar_t *wcsrchr(const wchar_t *s, wchar_t c);</code> | |
| | Wide-character version of strrchr. | 25.5 |
| wcsrtombs | <i>Convert Wide String to Multibyte String – Restartable (C99)</i> | <wchar.h> |
| | <code>size_t wcsrtombs(char * restrict dst, const wchar_t ** restrict src, size_t len, mbstate_t * restrict ps);</code> | |

Converts a sequence of wide characters from the array indirectly pointed to by `src` into a sequence of corresponding multibyte characters that begins in the conversion state described by the object pointed to by `ps`. If `ps` is a null pointer, `wcsrtombs` uses an internal object to store the conversion state. If `dst` isn't a null pointer, the converted characters are then stored in the array pointed to by `dst`. Conversion continues up to and including a terminating null wide character, which is also stored. Conversion stops earlier if a wide character is reached that doesn't correspond to a valid multibyte character or—if `dst` isn't a null pointer—