Nxxxx: Proposal to Add streq and strne to <string.h>

n3455 - Additional String Comparison Functions to Complement strcmp

Author: Yair Lenga

Contact: Yair.lenga@gmail.com

Document Number: n3455

Date: 2025-03-22

Target: C2x

Status: Proposal

Abstract

This paper proposes the addition of two simple and safe string comparison functions to the C standard library: streq and strne. These functions provide direct, boolean results for the most common use cases of strcmp: testing for equality and inequality.

Motivation

The standard function strcmp is widely used in C for comparing strings. However, its interface and semantics are often unnecessarily verbose and error-prone when used to test equality.

Common idioms such as:

```
if (strcmp(a, b) == 0) { ... }
if (!strcmp(a, b)) { ... }
```

are harder to read, easy to misuse (e.g. accidentally writing strcmp(a, b)), and impose unnecessary cognitive load.

A scan of popular open-source C codebases shows that the vast majority of strcmp calls are used to test for equality, with occasional use for inequality, and only a small fraction for ordering comparisons (<, >, etc.).

In addition, many codebases attempt to optimize comparisons like:

```
if (!(strcmp(a, "foo") && strcmp(a, "bar")))
```

which is better written as:

```
if (streq(a, "foo") || streq(a, "bar"))
```

Nxxxx: Proposal to Add streq and strne to <string.h>

Proposed Interfaces

In <string.h>, the following two functions are added:

```
bool streq(const char *a, const char *b);
bool strne(const char *a, const char *b);
```

These functions behave as follows:

- streq(a, b) returns true if and only if both a and b are non-null and contain equal strings.
- strne(a, b) returns true if and only if both a and b are non-null and contain unequal strings.
- If either argument is NULL, both functions return false.

Semantics

Expression	Result
streq("foo", "foo")	true
streq("foo", "bar")	false
streq(NULL, "bar")	false
streq(NULL, NULL)	false
strne("foo", "bar")	true
strne("foo", "foo")	false
strne(NULL, "bar")	false

Rationale

- Clarity: streq(a, b) expresses the programmer's intent more clearly than strcmp(a, b) == 0.
- Safety: Built-in null-pointer checks reduce the risk of undefined behavior.
- Performance: These functions return boolean directly, simplifying branching.
- Maintainability: Multi-string comparisons and negated logic become simpler.
- Portability: Efficient and portable on all platforms.

Implementation

Example portable implementation:

```
#include <stdbool.h>
#include <string.h>

bool streq(const char *a, const char *b) {
    return a && b && strcmp(a, b) == 0;
}

bool strne(const char *a, const char *b) {
    return a && b && strcmp(a, b) != 0;
}
```

Alternatives Considered

- Macros: Lack type safety, may evaluate arguments multiple times.
- Inline user functions: Duplicated across projects, lack standardization.

Summary

This proposal introduces two minimal, safe, and expressive functions for the most common use cases of string comparison in C. It aligns with modern programming practices while preserving C's philosophy of minimalism and efficiency.

Compatibility Note

The proposed names streq and strne fall within the reserved namespace for string-related functions (prefix str*), which aligns with standard naming conventions in <string.h>. This deliberate choice ensures minimal conflict with existing symbols and allows projects that have already defined similar utilities to adopt the standard versions quickly.