

1. String Manipulation Functions for the GBVM Library

File: `univ_strings.h`

Author: Soham Metha

Date: January 2025

The `univ_strings.h` file provides declarations for various string manipulation functions used in the GBVM library. These functions allow operations like trimming whitespace, splitting strings, converting strings to integers, and more.

1.1. Table of Contents

- [Structures](#)
 - [Function Documentation](#)
 - [ltrim](#)
 - [rtrim](#)
 - [trim](#)
 - [splitStr](#)
 - [strToInt](#)
 - [printString](#)
 - [Example Usage](#)
-

1.2. Structures

1.2.1. `String`

Description:

The `String` structure represents a string with its length and data. It provides a convenient way to manage strings while maintaining metadata about their size.

Members:

Member	Type	Description
<code>length</code>	<code>size_t</code>	The length of the string.
<code>data</code>	<code>const char*</code>	A pointer to the character data.

1.3. Function Documentation

1.3.1. `String ltrim(String s)`

Description:

Removes any leading whitespace characters from the input string and returns the modified string.

Parameters:

Parameter	Type	Description
<code>s</code>	<code>String</code>	The input string.

Returns:

A `String` object with leading whitespace removed.

1.3.2. `String rtrim(String s)`

Description:

Removes any trailing whitespace characters from the input string and returns the modified string.

Parameters:

Parameter	Type	Description
<code>s</code>	<code>String</code>	The input string.

Returns:

A `String` object with trailing whitespace removed.

1.3.3. `String trim(String s)`

Description:

Removes both leading and trailing whitespace characters from the input string and returns the modified string.

Parameters:

Parameter	Type	Description
<code>s</code>	<code>String</code>	The input string.

Returns:

A `String` object with both leading and trailing whitespace removed.

1.3.4. `String splitStr(String* s, char c)`

Description:

Splits the input string into two parts at the first occurrence of the specified character. The first part is returned, and the original string is modified to contain only the second part.

Parameters:

Parameter	Type	Description
-----------	------	-------------

Parameter	Type	Description
s	String*	A pointer to the input string to be split.
c	char	The character at which to split the string.

Returns:

The first part of the string before the specified character as a `String`.

1.3.5. `int strToInt(String s)`

Description:

Converts the input string to an integer.

Parameters:

Parameter	Type	Description
s	String	The input string to convert.

Returns:

The integer value represented by the string.

1.3.6. `void printString(String s)`

Description:

Prints the input string to the standard output.

Parameters:

Parameter	Type	Description
s	String	The string to be printed.

Returns:

None.

1.4. Example Usage

1.4.1. Trimming Whitespace

```
#include "univ_strings.h"

int main() {
    String s = {"14", "    Hello World    "};
    String trimmed = trim(s);
    printString(trimmed); // Output: "Hello World"
```

```
    return 0;
}
```

1.4.2. Splitting a String

```
#include "univ_strings.h"

int main() {
    String s = {"11", "key=value"};
    String key = splitStr(&s, '=');

    printString(key); // Output: "key"
    printString(s);   // Output: "value"
    return 0;
}
```

1.4.3. Converting a String to Integer

```
#include "univ_strings.h"

int main() {
    String s = {"3", "123"};
    int number = strToInt(s);
    printf("Number: %d\n", number); // Output: Number: 123
    return 0;
}
```