TakoStrLib

Generated by Doxygen 1.8.17

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Class Documentation

3.1 CharBuf Struct Reference

Struct which keeps dynamic char array and it's size.

`#include <CharBuf.hh>`

Public Attributes

- char * buf
- size_t size

3.1.1 Detailed Description

Struct which keeps dynamic char array and it's size.

Definition at line 10 of file CharBuf.hh.

3.1.2 Member Data Documentation

3.1.2.1 buf

`char* CharBuf::buf`

Definition at line 12 of file CharBuf.hh.

3.1.2.2 size

```
size_t CharBuf::size
```

Definition at line 13 of file CharBuf.hh.

The documentation for this struct was generated from the following file:

- CharBuf.hh

3.2 StrArray Struct Reference

Struct to keep all strings.

```
#include <CharBuf.hh>
```

Public Attributes

- String * lines
- size_t size

3.2.1 Detailed Description

Struct to keep all strings.

Definition at line 24 of file CharBuf.hh.

3.2.2 Member Data Documentation

3.2.2.1 lines

```
String* StrArray::lines
```

Definition at line 26 of file CharBuf.hh.

3.2.2.2 size

```
size_t StrArray::size
```

Definition at line 27 of file CharBuf.hh.

The documentation for this struct was generated from the following file:

- CharBuf.hh

Chapter 4

File Documentation

4.1 CharBuf.hh File Reference

```
#include <stdio.h>
#include <stdlib.h>
```

Classes

- struct CharBuf

  Struct which keeps dynamic char array and it's size.
- struct StrArray

  Struct to keep all strings.

Typedefs

- typedef CharBuf String

  Alias for CharBuf struct.

Functions

- CharBuf * cb_init (CharBuf *cb, size_t elem_num)

  CharBuf constructor.
- CharBuf * cb_init (CharBuf *cb_new, const CharBuf *cb_old)

  CharBuf copy constructor.
- CharBuf * cb_destr (CharBuf *cb)

  CharBuf destructor.
- StrArray * sa_init (StrArray *sa, size_t elnum)

  StrArray constructor.
- StrArray * sa_destr (StrArray *sa)

  StrArray destructor.
- void sa_print (StrArray sa, FILE *fp=stdout)

  prints string array

4.1.1 Typedef Documentation

4.1.1.1 String

```
typedef CharBuf String
```

Alias for CharBuf struct.

Definition at line 19 of file CharBuf.hh.

4.1.2 Function Documentation

4.1.2.1 cb_init() [1/2]

```
CharBuf* cb_init (
            CharBuf * cb,
            size_t elem_num )
```

CharBuf constructor.

Parameters

out	*cb*	CharBuf object
in	*elem_num*	number of elements in buffer

Returns

pointer to input CharBuf object

4.1.2.2 cb_init() [2/2]

```
CharBuf* cb_init (
            CharBuf * cb_new,
            const CharBuf * cb_old )
```

CharBuf copy constructor.

Parameters

out	*cb_new*	pointer to uninitialized CharBuf
in	*cb_old*	old CharBuf

Returns

pointer to input CharBuf object

4.1.2.3 cb_destr()

```
CharBuf* cb_destr (
            CharBuf * cb )
```

CharBuf destructor.

Parameters

out	*cb*	CharBuf object

Returns

pointer to input CharBuf object

4.1.2.4 sa_init()

```
StrArray* sa_init (
            StrArray * sa,
            size_t elnum )
```

StrArray constructor.

Parameters

out	*sa*	StrArray object
in	*elnum*	number of lines

Returns

pointer to input object

4.1.2.5 sa_destr()

```
StrArray* sa_destr (
            StrArray * sa )
```

StrArray destructor.

Parameters

out	*sa*	StrArray object

Returns

pointer to input object

4.1.2.6 sa_print()

```
void sa_print (
            StrArray sa,
            FILE * fp = stdout )
```

prints string array

Parameters

in	*sa*	

4.2 filesys.hh File Reference

```
#include <stdio.h>
#include "CharBuf.hh"
```

Functions

- ssize_t tfl_fsize (FILE *fp)
  caclulates file's size
- int file_to_buf (const char *fname, CharBuf *raw_data)
  filling buffer with data from file

4.2.1 Function Documentation

4.2.1.1 tfl_fsize()

```
ssize_t tfl_fsize (
            FILE * fp )
```

caclulates file's size

Parameters

in	*fp*	pointer to file

Returns

size of file in case of success, negative value otherwise.

4.2.1.2 file_to_buf()

```
int file_to_buf (
            const char * fname,
            CharBuf * raw_data )
```

filling buffer with data from file

Parameters

in	*fname*	filename
out	*raw_data*	buffer to fill

Returns

0 on success

4.3 tsl.hh File Reference

```
#include "CharBuf.hh"
```

Functions

- int tsl_fputs (const char ∗str, FILE ∗stream)

  writes the string str to stream, without terminating null byte ('\0').
- int tsl_puts (const char ∗str)

  writes the string str and a trailing newline to stdout.
- char ∗ tsl_strchr (char ∗str, int ch)

  returns a pointer to the first occurrence of the character c in the string str.
- const char ∗ tsl_const_strchr (const char ∗str, int ch)

  returns a pointer to the first occurrence of the character c in the string s.
- size_t tsl_strlen (const char ∗str)

  calculates the length of the string pointed to by str, excluding the terminating null byte ('\0').
- char ∗ tsl_strcpy (char ∗dst, const char ∗src)

  copies the string pointed to by src, including the terminating null byte ('\0'), to the buffer pointed to by dest.
- char ∗ tsl_strncpy (char ∗dst, const char ∗src, size_t n)

similar to tsl_strcpy, except that at most n bytes of src are copied

- char * tsl_strcat (char *dst, const char *src)

  appends the src string to the dest string, overwriting the terminating null byte ('\0') at the end of dest, and then adds a terminating null byte.

- char * tsl_strncat (char *dst, const char *src, size_t n)

  is similar to tsl_strcat, except that it will use at most n bytes from src and src does not need to be null-terminated if it contains n or more bytes.

- char * tsl_fgets (char *str, int size, FILE *stream)

  reads in at most one less than size characters from stream and stores them into the buffer pointed to by str.

- char * tsl_strdup (const char *str)

  returns a pointer to a new string which is a duplicate of the string str.

- int tsl_test ()

  unit test for tsl functions

- StrArray tsl_split_lines (CharBuf raw)

  split input buffer to lines, allocates memory in StrArray::lines.

- int tsl_cb_cmp (const void *lhs, const void *rhs)

  comparator for CharBuf strings

- int tsl_cb_back_cmp (const void *lhs, const void *rhs)

  backward comparator for CharBuf strings

4.3.1 Function Documentation

4.3.1.1 tsl_fputs()

```
int tsl_fputs (
            const char * str,
            FILE * stream )
```

writes the string str to stream, without terminating null byte ('\0').

Parameters

in	*str*	
in	*stream*	

Returns

nonnegative number on success, or EOF on error.

4.3.1.2 tsl_puts()

```
int tsl_puts (
            const char * str )
```

writes the string str and a trailing newline to stdout.

Parameters

in	*str*	

Returns

a nonnegative number on success, or EOF on error.

4.3.1.3 tsl_strchr()

```
char* tsl_strchr (
            char * str,
            int ch )
```

returns a pointer to the first occurrence of the character c in the string str.

Parameters

in	*str*	
in	*ch*	

Returns

pointer to the matched character or NULL if the character is not found.

4.3.1.4 tsl_const_strchr()

```
const char* tsl_const_strchr (
            const char * str,
            int ch )
```

returns a pointer to the first occurrence of the character c in the string s.

Parameters

in	*str*	
in	*ch*	

Returns

pointer to the matched character or NULL if the character is not found.

4.3.1.5 tsl_strlen()

```
size_t tsl_strlen (
            const char * str )
```

calculates the length of the string pointed to by str, excluding the terminating null byte ('\0').

Parameters

in	*str*	

Returns

the number of bytes in the string pointed to by s.

4.3.1.6 tsl_strcpy()

```
char* tsl_strcpy (
            char * dst,
            const char * src )
```

copies the string pointed to by src, including the terminating null byte ('\0'), to the buffer pointed to by dest.

The strings may not overlap, and the destination string dest must be large enough to receive the copy.

Parameters

out	*dst*	
in	*src*	

Returns

pointer to the destination string.

4.3.1.7 tsl_strncpy()

```
char* tsl_strncpy (
            char * dst,
            const char * src,
            size_t n )
```

similar to tsl_strcpy, except that at most n bytes of src are copied

Parameters

out	*dst*	
in	*src*	
in	*n*	

Returns

pointer to the destination string.

4.3.1.8 tsl_strcat()

```
char* tsl_strcat (
            char * dst,
            const char * src )
```

appends the src string to the dest string, overwriting the terminating null byte ('\0') at the end of dest, and then adds a terminating null byte.

Parameters

out	*dst*	
in	*src*	

Returns

pointer to the resulting string.

4.3.1.9 tsl_strncat()

```
char* tsl_strncat (
            char * dst,
            const char * src,
            size_t n )
```

is similar to tsl_strcat, except that it will use at most n bytes from src and src does not need to be null-terminated if it contains n or more bytes.

Parameters

out	*dst*	
in	*src*	
in	*n*	

Returns

pointer to the resulting string.

4.3.1.10 tsl_fgets()

```
char* tsl_fgets (
            char * str,
            int size,
            FILE * stream )
```

reads in at most one less than size characters from stream and stores them into the buffer pointed to by str.

Reading stops after an EOF or a newline. If a newline is read, it is stored into the buffer. A terminating null byte ('\0') is stored after the last character in the buffer.

Parameters

out	*str*	
in	*size*	
in	*stream*	

Returns

str on success, and nullptr on error or when end of file occurs while no characters have been read.

4.3.1.11 tsl_strdup()

```
char* tsl_strdup (
            const char * str )
```

returns a pointer to a new string which is a duplicate of the string str.

Memory for the new string is obtained with malloc, and can be freed with free.

Parameters

in	*str*	

Returns

pointer to the duplicated string or nullptr in case of failure.

4.3.1.12 tsl_test()

```
int tsl_test ( )
```

unit test for tsl functions

Returns

0 on success

4.3.1.13 tsl_split_lines()

```
StrArray tsl_split_lines (
            CharBuf raw )
```

split input buffer to lines, allocates memory in StrArray::lines.

Do not forget to free

Parameters

in	*raw*	input char buffer

Returns

StrArray

4.3.1.14 tsl_cb_cmp()

```
int tsl_cb_cmp (
            const void * lhs,
            const void * rhs )
```

comparator for CharBuf strings

Parameters

in	*lhs*	
in	*rhs*	

Returns

== 0 if elements are equal

> 0 if lhs > rhs

< 0 if lhs < rhs

4.3.1.15 tsl_cb_back_cmp()

```
int tsl_cb_back_cmp (
            const void * lhs,
            const void * rhs )
```

backward comparator for CharBuf strings

Parameters

in	*lhs*	
in	*rhs*	

Returns

== 0 if elements are equal

$>$ 0 if lhs $>$ rhs

$<$ 0 if lhs $<$ rhs

Index