**mystrtok.c**[[plain text](http://www.opensource.apple.com/source/postfix/postfix-229/postfix/src/util/mystrtok.c?txt)]

*/\*++*

*/\* NAME*

*/\* mystrtok 3*

*/\* SUMMARY*

*/\* safe tokenizer*

*/\* SYNOPSIS*

*/\* #include <stringops.h>*

*/\**

*/\* char \*mystrtok(bufp, delimiters)*

*/\* char \*\*bufp;*

*/\* const char \*delimiters;*

*/\* DESCRIPTION*

*/\* mystrtok() splits a buffer on the specified \fI delimiters\fR.*

*/\* Tokens are delimited by runs of delimiters, so this routine*

*/\* cannot return zero-length tokens.*

*/\**

*/\* The \fI bufp \fR argument specifies the start of the search; it*

*/\* is updated with each call. The input is destroyed.*

*/\**

*/\* The result value is the next token, or a null pointer when the*

*/\* end of the buffer was reached.*

*/\* LICENSE*

*/\* .ad*

*/\* .fi*

*/\* The Secure Mailer license must be distributed with this software.*

*/\* AUTHOR(S)*

*/\* Wietse Venema*

*/\* IBM T.J. Watson Research*

*/\* P.O. Box 704*

*/\* Yorktown Heights, NY 10598, USA*

*/\*--\*/*

*/\* System library. \*/*

#**include** **"sys\_defs.h"**

#**include** **<string.h>**

*/\* Utility library. \*/*

#**include** **"stringops.h"**

*/\* mystrtok - safe tokenizer \*/*

**char** \***mystrtok**(**char** \*\*src, **const** **char** \*sep)

{

**char** \*start = \*src;

**char** \*end;

*/\**

*\* Skip over leading delimiters.*

*\*/*

start += strspn(start, sep);

**if** (\*start == 0) {

\*src = start;

**return** (0);

}

*/\**

*\* Separate off one token.*

*\*/*

end = start + strcspn(start, sep);

**if** (\*end != 0)

\*end++ = 0;

\*src = end;

**return** (start);

}

#**ifdef** **TEST**

*/\**

*\* Test program: read lines from stdin, split on whitespace.*

*\*/*

#**include** **"vstring.h"**

#**include** **"vstream.h"**

#**include** **"vstring\_vstream.h"**

**int** **main**(**void**)

{

VSTRING \*vp = vstring\_alloc(100);

**char** \*start;

**char** \*str;

**while** (vstring\_fgets(vp, VSTREAM\_IN)) {

start = vstring\_str(vp);

**while** ((str = mystrtok(&start, **" \t\r\n"**)) != 0)

vstream\_printf(**">%s<\n"**, str);

vstream\_fflush(VSTREAM\_OUT);

}

vstring\_free(vp);

**return** (0);

}

#**endif**