NAME

strlcpy, **strlcat** – size-bounded string copying and concatenation

SYNOPSIS

#include <string.h>

size_t
strlcpy(*char *dst*, *const char *src*, *size_t size*);

size_t
strlcat(*char *dst*, *const char *src*, *size_t size*);

DESCRIPTION

The **strlcpy**() and **strlcat**() functions copy and concatenate strings respectively. They are designed to be safer, more consistent, and less error prone replacements for strncpy(3) and strncat(3). Unlike those functions, **strlcpy**() and **strlcat**() take the full size of the buffer (not just the length) and guarantee to NUL-terminate the result (as long as *size* is larger than 0). Note that you should include a byte for the NUL in *size*.

The **strlcpy**() function copies up to *size* - 1 characters from the NUL-terminated string *src* to *dst*, NUL-terminating the result.

The **strlcat**() function appends the NUL-terminated string *src* to the end of *dst*. It will append at most *size* - strlen(dst) - 1 bytes, NUL-terminating the result.

RETURN VALUES

The **strlcpy**() and **strlcat**() functions return the total length of the string they tried to create. For **strlcpy**() that means the length of *src*. For **strlcat**() that means the initial length of *dst* plus the length of *src*. While this may seem somewhat confusing it was done to make truncation detection simple.

EXAMPLES

The following code fragment illustrates the simple case:

```
char *s, *p, buf[BUFSIZ];

...

(void)strlcpy(buf, s, sizeof(buf));
(void)strlcat(buf, p, sizeof(buf));
```

To detect truncation, perhaps while building a pathname, something like the following might be used:

```
char *dir, *file, pname[MAXPATHNAMELEN];

...

if (strlcpy(pname, dir, sizeof(pname)) >= sizeof(pname))
        goto toolong;
if (strlcat(pname, file, sizeof(pname)) >= sizeof(pname))
        goto toolong;
```

Since we know how many characters we copied the first time, we can speed things up a bit by using a copy instead on an append:

```
char *dir, *file, pname[MAXPATHNAMELEN];
size_t n;

...

n = strlcpy(pname, dir, sizeof(pname));
if (n >= sizeof(pname))
        goto toolong;
if (strlcpy(pname + n, file, sizeof(pname) - n) >= sizeof(pname) - n)
        goto toolong;
```

However, one may question the validity of such optimizations, as they defeat the whole purpose of **strlcpy**() and **strlcat**(). As a matter of fact, the first version of this manual page got it wrong.

SEE ALSO

snprintf(3), strncpy(3), strncat(3)