

**NAME**

cpystr – copy multiple strings

**SYNOPSIS**

```
#include "mjsu.h"
```

```
CHAR *cpystr(CHAR *dest, ...);
```

**DESCRIPTION**

**cpystr()** concatenates a series of strings (pointed-to by the arguments ...) into the destination string at *dest*. The list of source strings is terminated by a NULL pointer.

Similar functionality can be obtained by using **strepy()** followed by repeated calls to **strcat()**, but using **cpystr()** is more convenient and often more time-efficient.

**RETURNS**

**cpystr()** returns a pointer to the terminating `'\0'` of the constructed destination string. This return value is extremely handy for subsequent appends to *dest*.

**EXAMPLE**

To build a concatenation of three strings:

```
CHAR line[BUFSIZ], *p;
CHAR head[BUFSIZ], middle[BUFSIZ], tail[BUFSIZ];

/* the one-shot method:
*/
cpystr(line, head, middle, tail, NULL);

/* the gradual method:
*/
p = cpystr(line, head, NULL);
p = cpystr(p, middle, NULL);
cpystr(p, tail, NULL);

/* the nested method:
*/
cpystr(cpystr(cpystr(line, head, NULL), middle, NULL), tail, NULL);
```

**SEE ALSO**

**mjsu(7)**.

**strepy()** and **strcat()**, as defined by ANSI X3.159-1989.

**NOTES**

None of the source strings must overlap with *dest*, ie: you cannot append (part of) *dest* to itself. Attempting to do so could cause arbitrary behaviour.

**AVAILABILITY**

**cpystr()** is written in C, conforming to ANSI X3.159-1989.