

NAME

`pcat` -- concatenate arbitrary strings

SYNOPSIS

```
pcat(s1,n1,s2[,...],0)
char *s1,*s2;
int n1;
```

DESCRIPTION

`Pcat` returns a pointer indicating the address of the terminating null character for the target string s1. The address returned is the same as that returned by the `plen` function. The `pcat` function concatenates the strings s2, s3, s4, etc into the target string s1 which has a maximum size indicated by n1. `Pcat` accepts a variable number of arguments.

s1 buffer area for the target string.

n1 integer which specifies the maximum number of characters which can be stored into s1 including the terminating null character.

s2 source string which is copied into s1

... secondary source strings are concatenated with s1.

0 a null pointer terminates the argument list.

If the address pointed to by s1 or s2 is zero or if the value of n1 is zero or negative, `pcat` will immediately terminate and return the address zero. If the target string is filled to maximum, `pcat` will return the address of the last position in s1. If the number of characters requested to be stored in the target string including the terminating null character is larger than n1, `pcat` will return the address zero but a properly terminated string will remain in s1. In this case, the function `plen` will return the address of the last character in s1. The `sizeof` function can be used for n1. It should be noted that `pcat` becomes a copy string function when only one source string argument, s2, is supplied.

The strings s1, s2, and etc. are each defined as a null terminated array of characters. The returned pointer is the address of the terminating null character.

If s2 and all subsequent arguments point to empty strings, the target string s1 will be set empty and the returned address will be the address of the first character in s1. If one of the source strings s2, s3, etc is empty, the remaining strings will be concatenated as if the empty string did not exist.

PCAT(3L)

SCCS

Aug 20, 1979

PCAT(3L)

LIBRARY

/lib/lib3.a

SEE ALSO

icat(3)