NAME

sprintf – format characters in memory

SYNOPSIS

#include <sys/ddi.h>

char *sprintf(char ****buf*, const char ****fmt***, . . .);**

INTERFACE LEVEL

Solaris DDI specific (Solaris DDI).

ARGUMENTS

buf Pointer to a character string.

fmt Pointer to a character string.

DESCRIPTION

sprintf( ) builds a string in *buf* under the control of the format *fmt*. The format is a character string with either plain characters, which are simply copied into *buf*, or conversion specifications, each of which converts zero or more arguments, again copied into *buf*. The results are unpredictable if there are insufficient arguments for the format; excess arguments are simply ignored. It is the user's responsibility to ensure that enough storage is available for *buf*.

Each conversion specification is introduced by the **%** character, after which the following appear in sequence:

An optional decimal digit specifying a minimum field width for numeric conversion. The converted value will be right-justified and padded with leading zeroes if it has fewer characters than the minimum.

An optional **l** (**ll**) specifying that a following **d**, **D**, **o**, **O**, **x**, **X**, or **u** conversion character applies to a **long** (**long long**) integer argument. An **l** (**ll**) before any other conversion character is ignored.

A character indicating the type of conversion to be applied:

d,**D**,**o**,**O**,**x**,**X**,**u**
The integer argument is converted to signed decimal (**d**, **D**), unsigned octal (**o**, **O**), unsigned hexadecimal (**x**, **X**) or unsigned decimal (**u**), respectively, and copied. The letters **abcdef** are used for **x** and **X** conversion.

c The character value of argument is copied.

b This conversion uses two additional arguments. The first is an integer, and is converted according to the base specified in the second argument. The second argument is a character string in the form **<***base***>[<***arg***> . . .]**. The base supplies the conversion base for the first argument as a binary value; \10 gives octal, \20 gives hexadecimal. Each subsequent <arg> is a sequence of characters, the first of which is the bit number to be tested, and subsequent characters, up to the next bit number or terminating null, supply the name of the bit.

A bit number is a binary-valued character in the range 1-32. For each bit set in the first argument, and named in the second argument, the bit names are copied, separated by commas, and bracketed by **<** and **>**. Thus, the following function call would generate **reg=3<BitTwo,BitOne>\n** in *buf*.

sprintf(buf, "reg=%b\n", 3, "\10\2BitTwo\1BitOne")

s The argument is taken to be a string (character pointer), and characters from the string are copied until a null character is encountered. If the character pointer is **NULL**, the string **<null string>** is used in its place.

% Copy a %; no argument is converted.

RETURN VALUES

sprintf( ) returns its first argument, *buf*.

CONTEXT

sprintf( ) can be called from user or interrupt context.

SEE ALSO

Writing Device Drivers