Rocky Enterprise Linux 9.2 Manual Pages on command 'uname.2'

$ man uname.2

```
UNAME(2)                 Linux Programmer's Manual                 UNAME(2)

NAME

       uname - get name and information about current kernel

SYNOPSIS

       #include <sys/utsname.h>

       int uname(struct utsname *buf);

DESCRIPTION

       uname() returns system information in the structure pointed to by buf.  The utsname struct

       is defined in <sys/utsname.h>:

           struct utsname {

               char sysname[];    /* Operating system name (e.g., "Linux") */

               char nodename[];   /* Name within "some implementation-defined

                                     network" */

               char release[];    /* Operating system release

                                     (e.g., "2.6.28") */

               char version[];    /* Operating system version */

               char machine[];    /* Hardware identifier */

           #ifdef _GNU_SOURCE

               char domainname[]; /* NIS or YP domain name */

           #endif

           };

       The length of the arrays in a struct utsname is unspecified (see NOTES);  the  fields  are

       terminated by a null byte ('\0').
```

RETURN VALUE

On success, zero is returned. On error, -1 is returned, and errno is set appropriately.

ERRORS

EFAULT buf is not valid.

CONFORMING TO

POSIX.1-2001, POSIX.1-2008, SVr4. There is no uname() call in 4.3BSD.

The domainname member (the NIS or YP domain name) is a GNU extension.

NOTES

This is a system call, and the operating system presumably knows its name, release and version. It also knows what hardware it runs on. So, four of the fields of the struct are meaningful. On the other hand, the field nodename is meaningless: it gives the name of the present machine in some undefined network, but typically machines are in more than one network and have several names. Moreover, the kernel has no way of knowing about such things, so it has to be told what to answer here. The same holds for the additional do? mainname field.

To this end, Linux uses the system calls sethostname(2) and setdomainname(2). Note that there is no standard that says that the hostname set by sethostname(2) is the same string as the nodename field of the struct returned by uname() (indeed, some systems allow a 256-byte hostname and an 8-byte nodename), but this is true on Linux. The same holds for setdomainname(2) and the domainname field.

The length of the fields in the struct varies. Some operating systems or libraries use a hardcoded 9 or 33 or 65 or 257. Other systems use SYS_NMLN or _SYS_NMLN or UTSLEN or _UT? SNAME_LENGTH. Clearly, it is a bad idea to use any of these constants; just use sizeof(...). Often 257 is chosen in order to have room for an internet hostname.

Part of the utsname information is also accessible via /proc/sys/kernel/{ostype, hostname, osrelease, version, domainname}.

C library/kernel differences

Over time, increases in the size of the utsname structure have led to three successive versions of uname(): sys_olduname() (slot __NR_oldolduname), sys_uname() (slot __NR_olduname), and sys_newuname() (slot __NR_uname). The first one used length 9 for all fields; the second used 65; the third also uses 65 but adds the domainname field. The glibc uname() wrapper function hides these details from applications, invoking the most recent version of the system call provided by the kernel.

SEE ALSO

uname(1), getdomainname(2), gethostname(2), uts_namespaces(7)

COLOPHON

This page is part of release 5.10 of the Linux man-pages project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at https://www.kernel.org/doc/man-pages/.