

**NAME**

get\_nprocs, get\_nprocs\_conf – get number of processors

**LIBRARY**

Standard C library (*libc*, *-lc*)

**SYNOPSIS**

```
#include <sys/sysinfo.h>
```

```
int get_nprocs(void);
```

```
int get_nprocs_conf(void);
```

**DESCRIPTION**

The function **get\_nprocs\_conf()** returns the number of processors configured by the operating system.

The function **get\_nprocs()** returns the number of processors currently available in the system. This may be less than the number returned by **get\_nprocs\_conf()** because processors may be offline (e.g., on hotpluggable systems).

**RETURN VALUE**

As given in DESCRIPTION.

**ATTRIBUTES**

For an explanation of the terms used in this section, see **attributes(7)**.

Interface	Attribute	Value
get_nprocs(), get_nprocs_conf()	Thread safety	MT-Safe

**STANDARDS**

These functions are GNU extensions.

**NOTES**

The current implementation of these functions is rather expensive, since they open and parse files in the */sys* filesystem each time they are called.

The following **sysconf(3)** calls make use of the functions documented on this page to return the same information.

```
np = sysconf(_SC_NPROCESSORS_CONF);    /* processors configured */
np = sysconf(_SC_NPROCESSORS_ONLN);    /* processors available */
```

**EXAMPLES**

The following example shows how **get\_nprocs()** and **get\_nprocs\_conf()** can be used.

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/sysinfo.h>

int
main(void)
{
    printf("This system has %d processors configured and "
           "%d processors available.\n",
           get_nprocs_conf(), get_nprocs());
    exit(EXIT_SUCCESS);
}
```

**SEE ALSO**

**nproc(1)**