NAME

syslog, klogctl − read and/or clear kernel message ring buffer; set console_loglevel

SYNOPSIS

int syslog(int *type***, char ****bufp***, int** *len***);**
/* No wrapper provided in glibc */

/* The glibc interface */
#include <sys/klog.h>

int klogctl(int *type***, char ****bufp***, int** *len***);**

DESCRIPTION

If you need the C library function **syslog**() (which talks to **syslogd**(8)), then look at **syslog**(3). The system call of this name is about controlling the kernel *printk*() buffer, and the glibc wrapper function is called **klogctl**().

The kernel log buffer

The kernel has a cyclic buffer of length **LOG_BUF_LEN** in which messages given as arguments to the kernel function **printk**() are stored (regardless of their loglevel). In early kernels, **LOG_BUF_LEN** had the value 4096; from kernel 1.3.54, it was 8192; from kernel 2.1.113 it was 16384; since 2.4.23/2.6 the value is a kernel configuration option (**CONFIG_LOG_BUF_SHIFT**). In recent kernels the size can be queried with command type 10 (see below).

Commands

The *type* argument determines the action taken by this function. The list below specifies the values for *type*. The symbolic names are defined in the kernel source, but are not exported to user space; you will either need to use the numbers, or define the names yourself.

SYSLOG_ACTION_CLOSE (0)
Close the log. Currently a NOP.

SYSLOG_ACTION_OPEN (1)
Open the log. Currently a NOP.

SYSLOG_ACTION_READ (2)
Read from the log. The call waits until the kernel log buffer is nonempty, and then reads at most *len* bytes into the buffer pointed to by *bufp*. The call returns the number of bytes read. Bytes read from the log disappear from the log buffer: the information can be read only once. This is the function executed by the kernel when a user program reads */proc/kmsg*.

SYSLOG_ACTION_READ_ALL (3)
Read all messages remaining in the ring buffer, placing then in the buffer pointed to by *bufp*. The call reads the last *len* bytes from the log buffer (nondestructively), but will not read more than was written into the buffer since the last "clear ring buffer" command (see command 5 below)). The call returns the number of bytes read.

SYSLOG_ACTION_READ_CLEAR (4)
Read and clear all messages remaining in the ring buffer. The call does precisely the same as for a *type* of 3, but also executes the "clear ring buffer" command.

SYSLOG_ACTION_CLEAR (5)
The call executes just the "clear ring buffer" command. The *bufp* and *len* arguments are ignored.

This command does not really clear the ring buffer. Rather, it sets a kernel bookkeeping variable that determines the results returned by commands 3 (**SYSLOG_ACTION_READ_ALL**) and 4 (**SYSLOG_ACTION_READ_CLEAR**). This command has no effect on commands 2 (**SYSLOG_ACTION_READ**) and 9 (**SYSLOG_ACTION_SIZE_UNREAD**).

SYSLOG_ACTION_CONSOLE_OFF (6)
Disable printk to console. The call sets the console log level to the minimum, so that no messages are printed to the console. The *bufp* and *len* arguments are ignored.

SYSLOG_ACTION_CONSOLE_ON (7)
The call sets the console log level to the default, so that messages are printed to the console. The *bufp* and *len* arguments are ignored.

SYSLOG_ACTION_CONSOLE_LEVEL (8)
The call sets the console log level to the value given in *len*, which must be an integer between 1 and 8 (inclusive). See the **loglevel** section for details. The *bufp* argument is ignored.

SYSLOG_ACTION_SIZE_UNREAD (9) (since Linux 2.4.10)
The call returns the number of bytes currently available to be read from the kernel log buffer via command 2 (**SYSLOG_ACTION_READ**). The *bufp* and *len* arguments are ignored.

SYSLOG_ACTION_SIZE_BUFFER (10) (since Linux 2.6.6)
This command returns the total size of the kernel log buffer. The *bufp* and *len* arguments are ignored.

All commands except 3 and 10 require privilege. In Linux kernels before 2.6.37, command types 3 and 10 are allowed to unprivileged processes; since Linux 2.6.37, these commands are allowed to unprivileged processes only if */proc/sys/kernel/dmesg_restrict* has the value 0. Before Linux 2.6.37, "privileged" means that the caller has the **CAP_SYS_ADMIN** capability. Since Linux 2.6.37, "privileged" means that the caller has either the **CAP_SYS_ADMIN** capability (now deprecated for this purpose) or the (new) **CAP_SYSLOG** capability.

The loglevel
The kernel routine **printk**() will only print a message on the console, if it has a loglevel less than the value of the variable *console_loglevel*. This variable initially has the value **DEFAULT_CONSOLE_LOGLEVEL** (7), but is set to 10 if the kernel command line contains the word "debug", and to 15 in case of a kernel fault (the 10 and 15 are just silly, and equivalent to 8). This variable is set (to a value in the range 1-8) by a **syslog**() call with a *type* of 8. Calls to **syslog**() with *type* equal to 6 or 7 set the variable to 1 (kernel panics only) or 7 (all except debugging messages), respectively.

Every text line in a message has its own loglevel. This level is *DEFAULT_MESSAGE_LOGLEVEL − 1* (6) unless the line starts with <d> where *d* is a digit in the range 1-7, in which case the level is *d*. The conventional meaning of the loglevel is defined in <*linux/kernel.h*> as follows:

```
#define KERN_EMERG    "<0>"  /* system is unusable               */
#define KERN_ALERT    "<1>"  /* action must be taken immediately */
#define KERN_CRIT     "<2>"  /* critical conditions              */
#define KERN_ERR      "<3>"  /* error conditions                 */
#define KERN_WARNING  "<4>"  /* warning conditions               */
#define KERN_NOTICE   "<5>"  /* normal but significant condition */
#define KERN_INFO     "<6>"  /* informational                    */
#define KERN_DEBUG    "<7>"  /* debug-level messages             */
```

RETURN VALUE

For *type* equal to 2, 3, or 4, a successful call to **syslog**() returns the number of bytes read. For *type* 9, **syslog**() returns the number of bytes currently available to be read on the kernel log buffer. For *type* 10, **syslog**() returns the total size of the kernel log buffer. For other values of *type*, 0 is returned on success.

In case of error, −1 is returned, and *errno* is set to indicate the error.

ERRORS

EINVAL
Bad arguments (e.g., bad *type*; or for *type* 2, 3, or 4, *buf* is NULL, or *len* is less than zero; or for *type* 8, the *level* is outside the range 1 to 8).

ENOSYS
This **syslog**() system call is not available, because the kernel was compiled with the **CONFIG_PRINTK** kernel-configuration option disabled.

EPERM
An attempt was made to change console_loglevel or clear the kernel message ring buffer by a process without sufficient privilege (more precisely: without the **CAP_SYS_ADMIN** or **CAP_SYSLOG** capability).

ERESTARTSYS
System call was interrupted by a signal; nothing was read. (This can be seen only during a trace.)

CONFORMING TO
This system call is Linux-specific and should not be used in programs intended to be portable.

NOTES
From the very start people noted that it is unfortunate that a system call and a library routine of the same name are entirely different animals.

SEE ALSO
syslog(3), **capabilities**(7)

COLOPHON
This page is part of release 3.53 of the Linux *man-pages* project. A description of the project, and information about reporting bugs, can be found at http://www.kernel.org/doc/man−pages/.