NAME

restart_syscall − restart a system call after interruption by a stop signal

SYNOPSIS

long restart_syscall(void);

Note: There is no glibc wrapper for this system call; see NOTES.

DESCRIPTION

The **restart_syscall**() system call is used to restart certain system calls after a process that was stopped by a signal (e.g., **SIGSTOP** or **SIGTSTP**) is later resumed after receiving a **SIGCONT** signal. This system call is designed only for internal use by the kernel.

restart_syscall() is used for restarting only those system calls that, when restarted, should adjust their time-related parameters—namely **poll**(2) (since Linux 2.6.24), **nanosleep**(2) (since Linux 2.6), **clock_nanosleep**(2) (since Linux 2.6), and **futex**(2), when employed with the **FUTEX_WAIT** (since Linux 2.6.22) and **FUTEX_WAIT_BITSET** (since Linux 2.6.31) operations. **restart_syscall**() restarts the interrupted system call with a time argument that is suitably adjusted to account for the time that has already elapsed (including the time where the process was stopped by a signal). Without the **restart_syscall**() mechanism, restarting these system calls would not correctly deduct the already elapsed time when the process continued execution.

RETURN VALUE

The return value of **restart_syscall**() is the return value of whatever system call is being restarted.

ERRORS

errno is set as per the errors for whatever system call is being restarted by **restart_syscall**().

VERSIONS

The **restart_syscall**() system call is present since Linux 2.6.

STANDARDS

This system call is Linux-specific.

NOTES

There is no glibc wrapper for this system call, because it is intended for use only by the kernel and should never be called by applications.

The kernel uses **restart_syscall**() to ensure that when a system call is restarted after a process has been stopped by a signal and then resumed by **SIGCONT**, then the time that the process spent in the stopped state is counted against the timeout interval specified in the original system call. In the case of system calls that take a timeout argument and automatically restart after a stop signal plus **SIGCONT**, but which do not have the **restart_syscall**() mechanism built in, then, after the process resumes execution, the time that the process spent in the stop state is *not* counted against the timeout value. Notable examples of system calls that suffer this problem are **ppoll**(2), **select**(2), and **pselect**(2).

From user space, the operation of **restart_syscall**() is largely invisible: to the process that made the system call that is restarted, it appears as though that system call executed and returned in the usual fashion.

SEE ALSO

sigaction(2), **sigreturn**(2), **signal**(7)