**NAME**

arch_prctl − set architecture-specific thread state

**SYNOPSIS**

**#include <asm/prctl.h>**
**#include <sys/prctl.h>**

**int arch_prctl(int** *code***, unsigned long** *addr***);**
**int arch_prctl(int** *code***, unsigned long \****addr***);**

**DESCRIPTION**

**arch_prctl**() sets architecture-specific process or thread state. *code* selects a subfunction and passes argument *addr* to it; *addr* is interpreted as either an *unsigned long* for the "set" operations, or as an *unsigned long \**, for the "get" operations.

Subfunctions for x86-64 are:

**ARCH_SET_FS**
Set the 64-bit base for the *FS* register to *addr*.

**ARCH_GET_FS**
Return the 64-bit base value for the *FS* register of the current thread in the *unsigned long* pointed to by *addr*.

**ARCH_SET_GS**
Set the 64-bit base for the *GS* register to *addr*.

**ARCH_GET_GS**
Return the 64-bit base value for the *GS* register of the current thread in the *unsigned long* pointed to by *addr*.

**RETURN VALUE**

On success, **arch_prctl**() returns 0; on error, −1 is returned, and *errno* is set to indicate the error.

**ERRORS**

**EFAULT**
*addr* points to an unmapped address or is outside the process address space.

**EINVAL**
*code* is not a valid subcommand.

**EPERM**
*addr* is outside the process address space.

**CONFORMING TO**

**arch_prctl**() is a Linux/x86-64 extension and should not be used in programs intended to be portable.

**NOTES**

**arch_prctl**() is supported only on Linux/x86-64 for 64-bit programs currently.

The 64-bit base changes when a new 32-bit segment selector is loaded.

**ARCH_SET_GS** is disabled in some kernels.

Context switches for 64-bit segment bases are rather expensive. As an optimization, if a 32-bit TLS base address is used, **arch_prctl**() may use a real TLS entry as if **set_thread_area**(2) had been called, instead of manipulating the segment base register directly. Memory in the first 2 GB of address space can be allocated by using **mmap**(2) with the **MAP_32BIT** flag.

Because of the aforementioned optimization, using **arch_prctl**() and **set_thread_area**(2) in the same thread is dangerous, as they may overwrite each other's TLS entries.

As of version 2.7, glibc provides no prototype for **arch_prctl**(). You have to declare it yourself for now. This may be fixed in future glibc versions.

*FS* may be already used by the threading library. Programs that use **ARCH_SET_FS** directly are very

likely to crash.

**SEE ALSO**

**mmap**(2), **modify_ldt**(2), **prctl**(2), **set_thread_area**(2)

AMD X86-64 Programmer's manual

**COLOPHON**

This page is part of release 5.02 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/.