NAME

proc_rwmem, **proc_readmem**, **proc_writemem** - read from or write to a process address space

SYNOPSIS

#include <sys/types.h>
#include <sys/ptrace.h>

int
proc_rwmem(*struct proc *p, struct uio *uio*);

ssize_t
proc_readmem(*struct thread *td, struct proc *p, vm_offset_t va, void *buf, size_t len*);

ssize_t
proc_writemem(*struct thread *td, struct proc *p, vm_offset_t va, void *buf, size_t len*);

DESCRIPTION

These functions are used to read to or write from the address space of the process *p*. The **proc_rwmem**() function requires the caller to specify the I/O parameters using a *struct uio*, described in uio(9). The **proc_readmem**() and **proc_writemem**() functions provide a simpler, less general interface which allows the caller to read into or write the kernel buffer *buf* of size *len* from or to the memory at offset *va* in the address space of *p*. The operation is performed on behalf of thread *td*, which will most often be the current thread.

These functions may sleep and thus may not be called with any non-sleepable locks held. The process *p* must be held by the caller using PHOLD(9).

RETURN VALUES

The **proc_rwmem**() function returns 0 on success. EFAULT is returned if the specified user address is invalid, and ENOMEM is returned if the target pages could not be faulted in due to a resource shortage.

The **proc_readmem**() and **proc_writemem**() functions return the number of bytes read or written, respectively. This may be smaller than the number of bytes requested, for example if the request spans multiple pages in the process address space and one of them after the first is not mapped. Otherwise, -1 is returned.

SEE ALSO

copyin(9), locking(9), PHOLD(9), uio(9)

AUTHORS

This manual page was written by Mark Johnston <*markj@FreeBSD.org*>.