NAME

proc_service, ps_pstop, ps_pcontinue, ps_lstop, ps_lcontinue, ps_pglobal_lookup, ps_pdread, ps_pdwrite, ps_ptread, ps_ptwrite, ps_lgetregs, ps_lsetregs, ps_plog, ps_lgetxregsize, ps_lgetxregs, ps_lsetxregs, ps_lgetfpregs, ps_lsetfpregs – process service interface

SYNOPSIS

#include <proc_service.h>

ps_err_e ps_pstop(const struct ps_prochandle **ph*);
ps_err_e ps_pcontinue(const struct ps_prochandle **ph*);
ps_err_e ps_lstop(const struct ps_prochandle **ph*, **lwpid_t** *lwpid*);
ps_err_e ps_lcontinue(const struct ps_prochandle **ph*, **lwpid_t** *lwpid*);

ps_err_e ps_pglobal_lookup(const struct ps_prochandle **ph*,
　　const char **ld_object_name*, **const char** **ld_symbol_name*,
　　paddr_t **ld_symbol_addr*);

ps_err_e ps_pdread(const struct ps_prochandle **ph*, **paddr_t** *addr*,
　　char **buf*, **int** *size*);

ps_err_e ps_pdwrite(const struct ps_prochandle **ph*, **paddr_t** *addr*,
　　char **buf*, **int** *size*);

ps_err_e ps_ptread(const struct ps_prochandle **ph*, **paddr_t** *addr*,
　　char **buf*, **int** *size*);

ps_err_e ps_ptwrite(const struct ps_prochandle **ph*, **paddr_t** *addr*,
　　char **buf*, **int** *size*);

ps_err_e ps_lgetregs(const struct ps_prochandle **ph*, **lwpid_t** *lwpid*,
　　prgregset_t *gregset*);

ps_err_e ps_lsetregs(const struct ps_prochandle **ph*, **lwpid_t** *lwpid*,
　　const prgregset_t *gregset*);

void ps_plog(const char **fmt*, **...);**

ps_err_e ps_lgetxregsize(const struct ps_prochandle **ph*, **lwpid_t** *lwpid*,
　　int **xregsize*);

ps_err_e ps_lgetxregs(const struct ps_prochandle **ph*, **lwpid_t** *lwpid*,
　　prxregset_t **xregset*);

ps_err_e ps_lsetxregs(const struct ps_prochandle **ph*, **lwpid_t** *lwpid*,
　　prxregset_t **xregset*);

ps_err_e ps_lgetfpregs(const struct ps_prochandle **ph*, **lwpid_t** *lwpid*,
　　prfpregset_t **fpregset*);

ps_err_e ps_lsetfpregs(const struct ps_prochandle **ph*, **lwpid_t** *lwpid*,
　　const prfpregset_t **fpregset*);

x86 Only

ps_err_e ps_lgetLDT(const struct ps_prochandle *∗ph*, **lwpid_t** *lwpid*,
struct ssd *∗ldt***);**

DESCRIPTION

proc_service is a set of interfaces that provide operations on a process. These services are utilized by libraries such as libthread_db to examine a process under inspection (PUI). These functions are provided by the user of such libraries.

FUNCTIONS

ps_pstop( ) stops the process with the given process handle and returns after the process has stopped.

ps_continue( ) continues a process with the given process handle that was stopped by **ps_pstop( )**.

ps_lstop( ) stops an LWP on the process with the given lwp identifier and process handle and returns after the LWP has stopped.

ps_lcontinue( ) continues an LWP on the process with the given lwp identifier and process handle that was stopped by **ps_lstop( )**.

ps_pglobal_lookup( ) finds the absolute address of a symbol given a process handle, symbol name and load object name (e.g., "libthread.so"). Only the global name space is searched.

ps_pdread( ) reads from a process's data segment at the given address, addr, for size bytes. Contents are returned in buf.

ps_pdwrite( ) writes contents of buf for size bytes to a process's data segment at the given address, addr.

ps_ptread( ) reads from a process's text segment at the given address, addr, for the size bytes. Contents are returned in buf.

ps_ptwrite( ) writes contents of buf for size bytes to a process's text segment at the address, addr.

ps_lgetregs( ) reads an LWP's (process with process handle ∗ph and lwp with identifier *lwpid*) general register set into gregset.

ps_lsetregs( ) writes an LWP's (process with process handle ∗ph and lwp with identifier *lwpid*) general register set from gregset.

ps_plog( ) logs a message. ps_plog( ) may be used to write diagnostic messages. Input is in printf style format.

ps_lgetxregsize( ) reads an LWP's (process with process handle ∗ph and lwp with identifier *lwpid*) extra register set size.

ps_lgetxregs( ) reads an LWP's (process with process handle ∗ph and lwp with identifier *lwpid*) extra register set into xregset.

ps_lsetxregs( ) writes an LWP's (process with process handle ∗ph and lwp with identifier *lwpid*) extra register set from xregset.

ps_lgetfpregs( ) reads an LWP's (process with process handle ∗ph and lwp with identifier *lwpid*) floating point register set into fpregset.

ps_lsetfpregs( ) writes an LWP's (process with process handle *ph and lwp with identifier *lwpid*) floating point register set from fpregset.

x86 Only **ps_lgetLDT( )** reads an LWP's (process with process handle *ph and lwp with identifier *lwpid*) Local Descriptor Table.

RETURN VALUES

PS_OK	The call succeeded.
PS_ERR	The call failed without a specific reason.
PS_BADPID	This return value indicates that a bad process handled was passed to the function.
PS_BADLID	This return value indicates that a bad lwp identifier was passed to the function.
PS_BADADDR	This return value indicates that a bad lwp identifier was passed to the function.
PS_NOSYM	The symbol could not be found.
PS_NOFREGS	This return value indicates that an FPU register set was not available for the lwp.

SEE ALSO

libthread_db(3T)