NAME

kvm_getu, kvm_getcmd – get the u-area or invocation arguments for a process

SYNOPSIS

```
#include <kvm.h>
#include <sys/param.h>
#include <sys/user.h>
#include <sys/proc.h>

struct user *kvm_getu(kvm_t *kd, struct proc *proc);

int kvm_getcmd(kvm_t *kd, struct proc *proc, struct user *u,
        char ***arg, char ***env);
```

MT-LEVEL

Unsafe

DESCRIPTION

kvm_getu( ) reads the u-area of the process specified by *proc* to an area of static storage associated with *kd* and returns a pointer to it. Subsequent calls to **kvm_getu( )** will overwrite this static area.

kd is a pointer to a kernel descriptor returned by **kvm_open**(3K). *proc* is a pointer to a copy (in the current process' address space) of a *proc* structure (obtained, for instance, by a prior **kvm_nextproc**(3K) call).

kvm_getcmd( ) constructs a list of string pointers that represent the command arguments and environment that were used to initiate the process specified by *proc*.

kd is a pointer to a kernel descriptor returned by **kvm_open**(3K). *u* is a pointer to a copy (in the current process' address space) of a *user* structure (obtained, for instance, by a prior **kvm_getu( )** call). If *arg* is not NULL, then the command line arguments are formed into a null-terminated array of string pointers. The address of the first such pointer is returned in *arg*. If *env* is not NULL, then the environment is formed into a null-terminated array of string pointers. The address of the first of these is returned in *env*.

The pointers returned in *arg* and *env* refer to data allocated by **malloc**(3C) and should be freed (by a call to **free( )** (see **malloc**(3C)) when no longer needed. Both the string pointers and the strings themselves are deallocated when freed.

Since the environment and command line arguments may have been modified by the user process, there is no guarantee that it will be possible to reconstruct the original command at all. Thus, **kvm_getcmd( )** will make the best attempt possible, returning –1 if the user process data is unrecognizable.

RETURN VALUES

On success, **kvm_getu( )** returns a pointer to a copy of the u-area of the process specified by *proc*. On failure, it returns NULL.

kvm_getcmd( ) returns:

0 on success.

–1 on failure.

SEE ALSO

kvm_nextproc(3K), **kvm_open**(3K), **kvm_read**(3K), **malloc**(3C)

NOTES

If **kvm_getcmd( )** returns −1, the caller still has the option of using the command line fragment that is stored in the u-area.