NAME

freeenv, **kern_getenv**, **getenv_int**, **getenv_long**, **getenv_string**, **getenv_quad**, **getenv_uint**, **getenv_ulong**, **getenv_bool**, **getenv_is_true**, **getenv_is_false**, **kern_setenv**, **testenv**, **kern_unsetenv** - kernel environment variable functions

SYNOPSIS

#include <sys/param.h>
#include <sys/systm.h>

void
freeenv(*char *env*);

*char **
kern_getenv(*const char *name*);

int
getenv_int(*const char *name*, *int *data*);

int
getenv_long(*const char *name*, *long *data*);

int
getenv_string(*const char *name*, *char *data*, *int size*);

int
getenv_quad(*const char *name*, *quad_t *data*);

int
getenv_uint(*const char *name*, *unsigned int *data*);

int
getenv_ulong(*const char *name*, *unsigned long *data*);

int
getenv_bool(*const char *name*, *bool *data*);

bool
getenv_is_true(*const char *name*);

bool

getenv_is_false(*const char *name*);

int
kern_setenv(*const char *name*, *const char *value*);

int
testenv(*const char *name*);

int
kern_unsetenv(*const char *name*);

DESCRIPTION

These functions set, unset, fetch, and parse variables from the kernel's environment.

The **kern_getenv**() function obtains the current value of the kernel environment variable *name* and returns a pointer to the string value. The caller should not modify the string pointed to by the return value. The **kern_getenv**() function may allocate temporary storage, so the **freeenv**() function must be called to release any allocated resources when the value returned by **kern_getenv**() is no longer needed.

The **freeenv**() function is used to release the resources allocated by a previous call to **kern_getenv**(). The *env* argument passed to **freeenv**() is the pointer returned by the earlier call to **kern_getenv**(). Like free(3), the *env* argument can be *NULL*, in which case no action occurs.

The **kern_setenv**() function inserts or resets the kernel environment variable *name* to *value*. If the variable *name* already exists, its value is replaced. This function can fail if an internal limit on the number of environment variables is exceeded.

The **kern_unsetenv**() function deletes the kernel environment variable *name*.

The **testenv**() function is used to determine if a kernel environment variable exists. It returns a non-zero value if the variable *name* exists and zero if it does not.

The **getenv_int**(), **getenv_long**(), **getenv_quad**(), **getenv_uint**(), and **getenv_ulong**() functions look for a kernel environment variable *name* and parse it as a signed integer, long integer, signed 64-bit integer, unsigned integer, or an unsigned long integer, respectively. These functions fail and return zero if *name* does not exist or if any invalid characters are present in its value. On success, these function store the parsed value in the integer variable pointed to by *data*. If the parsed value overflows the integer type, a truncated value is stored in *data* and zero is returned. If the value begins with a prefix of "0x" it is interpreted as hexadecimal. If it begins with a prefix of "0" it is interpreted as octal. Otherwise, the value is interpreted as decimal. The value may contain a single character suffix specifying a unit for the value. The interpreted value is multiplied by the unit's magnitude before being returned. The following unit suffixes are supported:

Unit	**Magnitude**
k	2^10
m	2^20
g	2^30
t	2^40

The **getenv_string**() function stores a copy of the kernel environment variable *name* in the buffer described by *data* and *size*. If the variable does not exist, zero is returned. If the variable exists, up to *size - 1* characters of its value are copied to the buffer pointed to by *data* followed by a null character and a non-zero value is returned.

The **getenv_bool**() function interprets the value of the kernel environment variable *name* as a boolean value by performing a case-insensitive comparison against the strings "1", "0", "true", and "false". If the environment variable exists and has a valid boolean value, then that value will be copied to the variable pointed to by *data*. If the environment variable exists but is not a boolean value, then a warning will be printed to the kernel message buffer. The **getenv_is_true**() and **getenv_is_false**() functions are wrappers around **getenv_bool**() that simplify testing for a desired boolean value.

RETURN VALUES

The **kern_getenv**() function returns a pointer to an environment variable's value on success or NULL if the variable does not exist.

The **kern_setenv**() and **kern_unsetenv**() functions return zero on success and -1 on failure.

The **testenv**() function returns zero if the specified environment variable does not exist and a non-zero value if it does exist.

The **getenv_int**(), **getenv_long**(), **getenv_string**(), **getenv_quad**(), **getenv_uint**(), **getenv_ulong**(), and **getenv_bool**() functions return a non-zero value on success and zero on failure.

The **getenv_is_true**() and **getenv_is_false**() functions return true if the specified environment variable exists and its value matches the desired boolean condition, and false otherwise.