NAME

pkey_alloc, pkey_free − allocate or free a protection key

LIBRARY

Standard C library (*libc*, *−lc*)

SYNOPSIS

```
#define _GNU_SOURCE         /* See feature_test_macros(7) */
#include <sys/mman.h>

int pkey_alloc(unsigned int fla gs, unsigned int access_rights);
int pkey_free(int pkey);
```

DESCRIPTION

pkey_alloc() allocates a protection key (pkey) and allows it to be passed to **pkey_mprotect**(2).

The **pkey_alloc**() *flags* is reserved for future use and currently must always be specified as 0.

The **pkey_alloc**() *access_rights* argument may contain zero or more disable operations:

PKEY_DISABLE_ACCESS
Disable all data access to memory covered by the returned protection key.

PKEY_DISABLE_WRITE
Disable write access to memory covered by the returned protection key.

pkey_free() frees a protection key and makes it available for later allocations. After a protection key has been freed, it may no longer be used in any protection-key-related operations.

An application should not call **pkey_free**() on any protection key which has been assigned to an address range by **pkey_mprotect**(2) and which is still in use. The behavior in this case is undefined and may result in an error.

RETURN VALUE

On success, **pkey_alloc**() returns a positive protection key value. On success,**pk ey_free**() returns zero. On error, −1 is returned, and *errno* is set to indicate the error.

ERRORS

EINVAL
pkey, *flags*, or *access_rights* is invalid.

ENOSPC
(**pkey_alloc**()) All protection keys available for the current process have been allocated. The number of keys available is architecture-specific and implementation-specific and may be reduced by kernel-internal use of certain keys. There are currently 15 keys available to user programs on x86.

This error will also be returned if the processor or operating system does not support protection keys. Applications should always be prepared to handle this error, since factors outside of the application's control can reduce the number of available pkeys.

VERSIONS

pkey_alloc() and **pkey_free**() were added in Linux 4.9; library support was added in glibc 2.27.

STANDARDS

The **pkey_alloc**() and **pkey_free**() system calls are Linux-specific.

NOTES

pkey_alloc() is always safe to call regardless of whether or not the operating system supports protection keys. It can be used in lieu of any other mechanism for detecting pkey support and will simply fail with the error **ENOSPC** if the operating system has no pkey support.

The kernel guarantees that the contents of the hardware rights register (PKRU) will be preserved only for allocated protection keys. Any time a key is unallocated (either before the first call returning that key from **pkey_alloc**() or after it is freed via **pkey_free**()), the kernel may make arbitrary changes to the parts of the rights register affecting access to that key.

EXAMPLES

See **pkeys**(7).

SEE ALSO

pkey_mprotect(2), **pkeys**(7)