NAME

brk, **sbrk** - change data segment size

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <unistd.h>

int
brk(*const void *addr*);

*void **
sbrk(*intptr_t incr*);

DESCRIPTION

The brk() and sbrk() functions are legacy interfaces from before the advent of modern virtual memory management. They are deprecated and not present on the arm64 or riscv architectures. The mmap(2) interface should be used to allocate pages instead.

The **brk**() and **sbrk**() functions are used to change the amount of memory allocated in a process's data segment. They do this by moving the location of the "break". The break is the first address after the end of the process's uninitialized data segment (also known as the "BSS").

The **brk**() function sets the break to *addr*.

The **sbrk**() function raises the break by *incr* bytes, thus allocating at least *incr* bytes of new memory in the data segment. If *incr* is negative, the break is lowered by *incr* bytes.

NOTES

While the actual process data segment size maintained by the kernel will only grow or shrink in page sizes, these functions allow setting the break to unaligned values (i.e., it may point to any address inside the last page of the data segment).

The current value of the program break may be determined by calling **sbrk**(*0*). See also end(3).

The getrlimit(2) system call may be used to determine the maximum permissible size of the data segment. It will not be possible to set the break beyond "*etext + rlim.rlim_max*" where the *rlim.rlim_max* value is returned from a call to **getrlimit**(*RLIMIT_DATA, &rlim*). (See end(3) for the definition of *etext*).

RETURN VALUES

The **brk**() function returns the value 0 if successful; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

The **sbrk**() function returns the prior break value if successful; otherwise the value (*void* *)-1 is returned and the global variable *errno* is set to indicate the error.

ERRORS

The **brk**() and **sbrk**() functions will fail if:

[EINVAL]	The requested break value was beyond the beginning of the data segment.
[ENOMEM]	The data segment size limit, as set by setrlimit(2), was exceeded.
[ENOMEM]	Insufficient space existed in the swap area to support the expansion of the data segment.

SEE ALSO

execve(2), getrlimit(2), mmap(2), end(3), free(3), malloc(3)

HISTORY

The **brk**() function appeared in Version 7 AT&T UNIX. FreeBSD 11.0 introduced the arm64 and riscv architectures which do not support **brk**() or **sbrk**().

BUGS

Mixing **brk**() or **sbrk**() with malloc(3), free(3), or similar functions will result in non-portable program behavior.

Setting the break may fail due to a temporary lack of swap space. It is not possible to distinguish this from a failure caused by exceeding the maximum size of the data segment without consulting getrlimit(2).

sbrk() is sometimes used to monitor heap use by calling with an argument of 0. The result is unlikely to reflect actual utilization in combination with an mmap(2) based malloc.

brk() and **sbrk**() are not thread-safe.