NAME

brk, sbrk – change the amount of space allocated for the calling process's data segment

SYNOPSIS

#include <unistd.h>

int brk(void ∗*endds***);**

void ∗**sbrk(int** *incr***);**

DESCRIPTION

brk( ) and **sbrk( )** are used to change dynamically the amount of space allocated for the calling process's data segment (see **exec**(2)). The change is made by resetting the process's break value and allocating the appropriate amount of space. The break value is the address of the first location beyond the end of the data segment. The amount of allocated space increases as the break value increases. Newly allocated space is set to zero. If, however, the same memory space is reallocated to the same process its contents are undefined.

When a program begins execution using **execve( )** the break is set at the highest location defined by the program and data storage areas.

The **getrlimit**(2) function may be used to determine the maximum permissible size of the *data* segment; it will not be possible to set the break beyond the **rlim_max** value returned from a call to **getrlimit( )**, that is to say, "**etext + rlim.rlim_max**." (See **end**(3C) for the definition of **etext( )**.)

brk( ) sets the break value to *endds* and changes the allocated space accordingly.

sbrk( ) adds *incr* bytes to the break value and changes the allocated space accordingly. *incr* can be negative, in which case the amount of allocated space is decreased.

RETURN VALUES

Upon successful completion, **brk( )** returns a value of 0 and **sbrk( )** returns the old break value. Otherwise, a value of −1 is returned and **errno** is set to indicate the error.

ERRORS

brk( ) and **sbrk( )** will fail and no additional memory will be allocated if one of the following occurs:

ENOMEM The data segment size limit, as set by **setrlimit( )** (see **getrlimit**(2)), would be exceeded.

ENOMEM The maximum possible size of a data segment (compiled into the system) would be exceeded.

ENOMEM Insufficient space exists in the swap area to support the expansion.

ENOMEM Out of address space; the new break value would extend into an area of the address space defined by some previously established mapping (see **mmap**(2)).

EAGAIN Total amount of system memory available for private pages is temporarily insufficient. This may occur even though the space requested was less than the maximum data segment size (see **ulimit**(2)).

SEE ALSO **exec**(2), **getrlimit**(2), **mmap**(2), **shmop**(2), **ulimit**(2), **end**(3C), **malloc**(3C)

WARNINGS Programs combining the **brk( )** and **sbrk( )** functions and **malloc( )** will not work. Many library routines use **malloc( )** internally, so use **brk( )** and **sbrk( )** only when you know that **malloc( )** definitely will not be used by any library routine.

NOTES The value of *incr* may be adjusted by the system before setting the new break value. Upon successful completion, the implementation guarantees a minimum of *incr* bytes will be added to the data segment if *incr* is a positive value. If *incr* is a negative value, a maximum of *incr* bytes will be removed from the data segment. This adjustment may not be necessary for all machine architectures.

The value of the arguments to both **brk( )** and **sbrk( )** are rounded up for alignment with eight-byte boundries.

BUGS 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**( )**.**