# sbrk

**brk** and **sbrk** are basic memory managementsystem calls used in Unix and Unix-like operating systems to control the amount of memory allocated to thedata segment of the process.[1] These functions are typically called from a higher-level memory management library function such as `malloc`. In the original Unix system, `brk` and `sbrk` were the only ways in which applications could acquire additional data space; later versions allowed this to also be done using the`mmap` call.

## Contents



## Description

The brk and sbrk calls dynamically change the amount of space allocated for the data segment of the calling process. The change is made by resetting the program break of the process, which determines the maximum space that can be allocated. The program break is the address of the first location beyond the current end of the data region. The amount of available space increases as the break value increases. The available space is initialized to a value of zero.[2] The break value can be automatically rounded up to a size appropriate for the memory managementarchitecture.

## Function signatures and behavior

```
#include <unistd.h>
int brk(void *end_data_segment );
void *sbrk(intptr_t increment );
```

The `brk` subroutine sets the program break value to the value of the `end_data_segment` parameter and changes the amount of available space accordingly

The `sbrk` subroutine adds to the program break value the number of bytes contained in the `increment` parameter and changes the amount of available space accordingly. The `increment` parameter can be a negative number, in which case the amount of available space is decreased.

Upon successful completion, the`brk` subroutine returns a value of 0, and the `sbrk` subroutine returns the prior value of the program break (if the available space is increased then this prior value also points to the start of the new area). If either subroutine is unsuccessful, a value of−1 is returned and the`errno` global variable is set to indicate the error

The current Mac OS X implementation of `sbrk` is an emulation, and has a maximum allocation of 4 megabytes.[3] When this limit is reached, −1 is returned and the`errno` is set to `ENOMEM`.

## Error codes

The error `ENOMEM` is set and the allocated space remains unchanged if one or more of the following are true:

- The requested change allocates more space than is allowed by a system-imposed maximum.
- The requested change sets the break value to a value greater than or equal to the start address of any attached shared memory segment.

# See also

- Exec (computing)
- Memory address#Address space in application programming

# References


1. "Process Memory Concepts" (https://www.gnu.org/software/libc/manual/html_node/Memory-Concepts.html). Free Software Foundation. Retrieved 9 October 2015.
2. *X/Open CAE Specification, System Interfaces and Headers* (http://pubs.opengroup.org/onlinepubs/9695969499/toc.pdf) (PDF). X/Open Company Ltd., U.K. p. 64. Retrieved 9 October 2015.
3. https://opensource.apple.com/source/Libc/Libc-763.12/emulated/brk.c