

NAME

`break`, `brk`, `sbrk` — change memory allocation

SYNOPSIS

```
char *sbrk (incr)
char *brk (addr)
```

DESCRIPTION

Brk sets the system's idea of the lowest location not used by the program to *addr* (rounded up to the next multiple of 64 bytes). Locations not less than *addr* and below the stack pointer are not in the address space and will thus cause a memory violation if accessed.

The call to *sbrk* should normally be used; *incr* more bytes are added to the program's data space and a pointer to the start of the new area is returned.

The assembler call *break* call will act exactly like the *brk* call described above. A zero is return upon success.

When a program begins execution via *exec* the *break* is set at the highest location defined by the program's data area (for programs running without separated I&D space the highest location is the sum of the text and data space sizes). The amount of space available for *break* to grab is thus the difference between the highest program location and the bottom of the stack. By definition, the amount of space reserved for the stack (but not necessarily allocated) is 4096 words.

Also, be careful when using *maus* together with *break*. *Maus* has the effect of reducing the amount of space available for allocation with *break*.

SEE ALSO

`maus(2)`, `exec(2)`, `malloc(3)`

DIAGNOSTICS

The *c*-bit is set whenever it is impossible to grant the memory request. From C, `-1` is returned for these errors.

BUGS

It is possible to reference memory past the end of the *break* without incurring an error. In fact, it is possible to overflow normal references (for instance an array subscript gone wild) into the stack without a memory fault occurring.

ASSEMBLER

```
(break = 17.)
sys break; addr
```