

**NAME**

*etext*, *edata*, *end* – end of program segments

**SYNOPSIS**

```
extern etext;
extern edata;
extern end;
```

**DESCRIPTION**

The addresses of these symbols indicate the end of various program segments:

*etext* This is the first address past the end of the text segment (the program code).

*edata* This is the first address past the end of the initialized data segment.

*end* This is the first address past the end of the uninitialized data segment (also known as the BSS segment).

**STANDARDS**

Although these symbols have long been provided on most UNIX systems, they are not standardized; use with caution.

**NOTES**

The program must explicitly declare these symbols; they are not defined in any header file.

On some systems the names of these symbols are preceded by underscores, thus: *\_etext*, *\_edata*, and *\_end*. These symbols are also defined for programs compiled on Linux.

At the start of program execution, the program break will be somewhere near *&end* (perhaps at the start of the following page). However, the break will change as memory is allocated via **brk**(2) or **malloc**(3). Use **sbrk**(2) with an argument of zero to find the current value of the program break.

**EXAMPLES**

When run, the program below produces output such as the following:

```
$ ./a.out
First address past:
    program text (etext)      0x8048568
    initialized data (edata)  0x804a01c
    uninitialized data (end)  0x804a024
```

**Program source**

```
#include <stdio.h>
#include <stdlib.h>

extern char etext, edata, end; /* The symbols must have some type,
                                or "gcc -Wall" complains */

int
main(void)
{
    printf("First address past:\n");
    printf("    program text (etext)      %10p\n", &etext);
    printf("    initialized data (edata)  %10p\n", &edata);
    printf("    uninitialized data (end)  %10p\n", &end);

    exit(EXIT_SUCCESS);
}
```

**SEE ALSO****objdump(1), readelf(1), sbrk(2), elf(5)**