

NAME

end, etext, edata – last locations in program

SYNOPSIS

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

DESCRIPTION

These names refer neither to routines nor to locations with interesting contents. Instead, their addresses coincide with the first address above the program text region (*etext*), above the initialized data region (*edata*), or uninitialized data region (*end*). The last is the same as the program break. Values are given to these symbols by the link editor *ld(1)* when, and only when, they are referred to but not defined in the set of programs loaded.

The usage of these symbols is rather specialized, but one plausible possibility is

```
extern end;  
...  
... = brk(&end+...);
```

The problem with this is that it ignores any other subroutines which may want to extend core for their purposes; these include *sbrk*, *alloc(III)*, and also secret subroutines invoked by the profile (*-p*) option of *cc*. Of course it was for the benefit of such systems that the symbols were invented, and user programs, unless they are in firm control of their environment, are wise not to refer to the absolute symbols directly.

One technique sometimes useful is to call *sbrk(0)*, which returns the value of the current program break, instead of referring to *&end*, which yields the program break at the instant execution started.

These symbols are accessible from assembly language if it is remembered that they should be prefixed by “_”.

SEE ALSO

break(II), alloc(III)