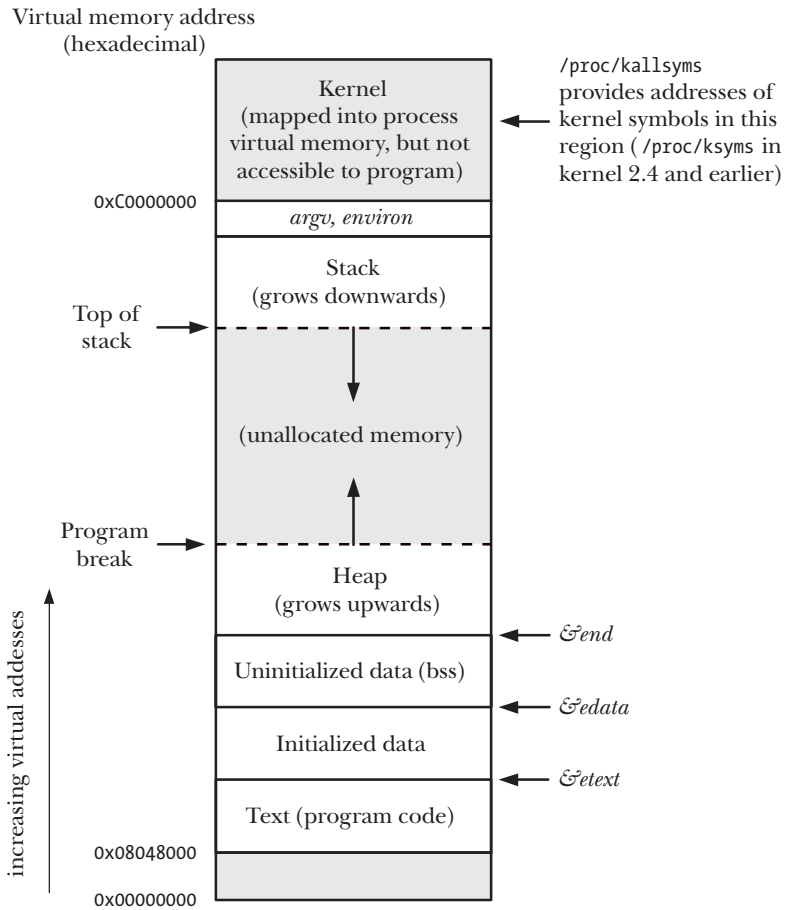

Figure 6-1: Typical memory layout of a process on Linux/x86-32

The upshot of locality of reference is that it is possible to execute a program while maintaining only part of its address space in RAM.

A virtual memory scheme splits the memory used by each program into small, fixed-size units called *pages*. Correspondingly, RAM is divided into a series of *page frames* of the same size. At any one time, only some of the pages of a program need to be resident in physical memory page frames; these pages form the so-called *resident set*. Copies of the unused pages of a program are maintained in the *swap area*—a reserved area of disk space used to supplement the computer's RAM—and loaded into physical memory only as required. When a process references a page that is not currently resident in physical memory, a *page fault* occurs, at which point the kernel suspends execution of the process while the page is loaded from disk into memory.

> On x86-32, pages are 4096 bytes in size. Some other Linux implementations use larger page sizes. For example, Alpha uses a page size of 8192 bytes, and IA-64 has a variable page size, with the usual default being 16,384 bytes. A program can determine the system virtual memory page size using the call *sysconf(_SC_PAGESIZE)*, as described in Section 11.2.