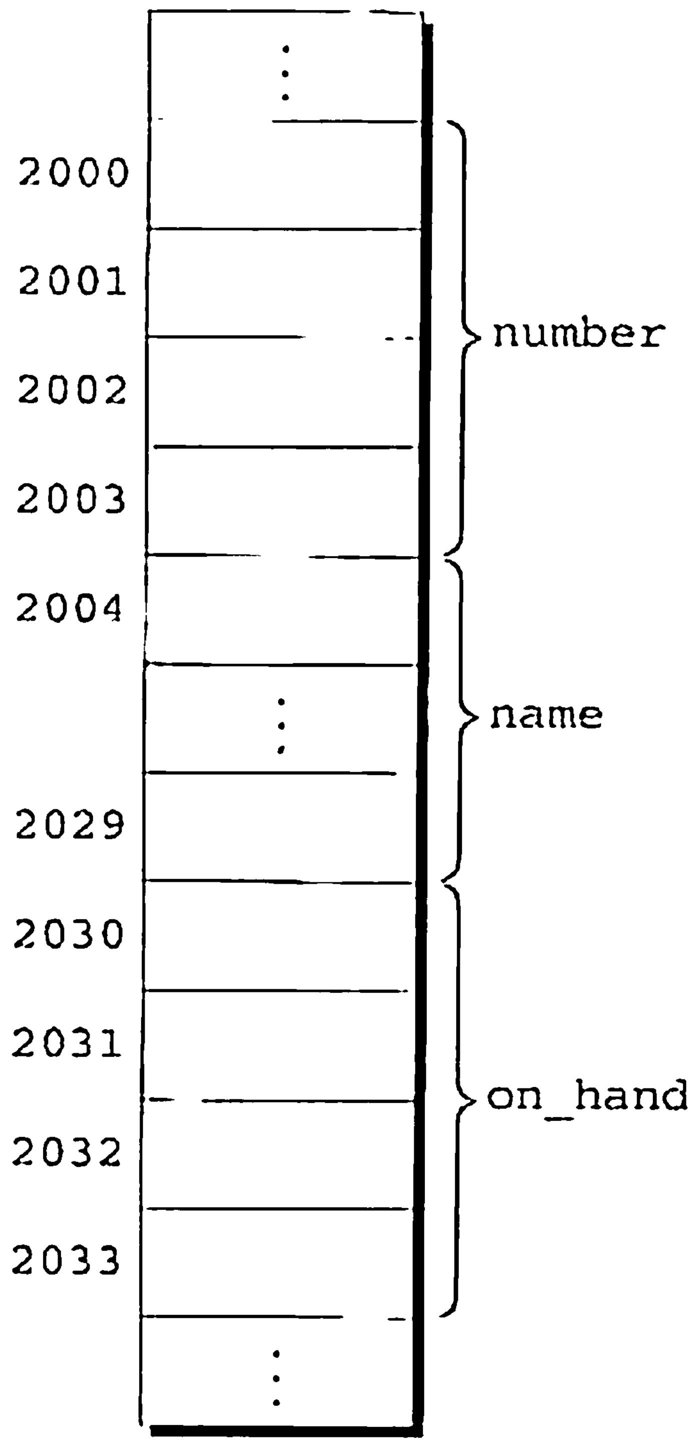
Declaring Structure Variables

When we need to store a collection of related data items, a structure is a logical choice. For example, suppose that we need to keep track of parts in a warehouse. The information that we'll need to store for each part might include a part number (an integer), a part name (a string of characters), and the number of parts on hand (an integer). To create variables that can store all three items of data, we might use a declaration such as the following:

```
struct {
  int number;
  char name[NAME_LEN+1];
  int on_hand;
} part1, part2;
```

Each structure variable has three members: number (the part number), name (the name of the part), and on_hand (the quantity on hand). Notice that this declaration has the same form as other variable declarations in C: struct { ... } specifies a type, while part1 and part2 are variables of that type.

The members of a structure are stored in memory in the order in which they're declared. In order to show what the part1 variable looks like in memory, let's assume that (1) part1 is located at address 2000, (2) integers occupy four bytes, (3) NAME_LEN has the value 25, and (4) there are no gaps between the members. With these assumptions, part1 will have the following appearance:


2000
2001
2002
2003
2004
2029
2030
2031
2032
2033
number
name
on_hand