Lists

Data structures

Definition

- We define a list to be a finite, ordered sequence of data items known as elements.

Examples

- In a FAT file system, the metadata of a large file is organized as a linked list of FAT entries.
- A list of images that need to be burned to a CD in a medical imaging application
- A list of users of a website that need to be emailed some notification
- A list of objects in a 3D game that need to be rendered to the screen

FAT file system

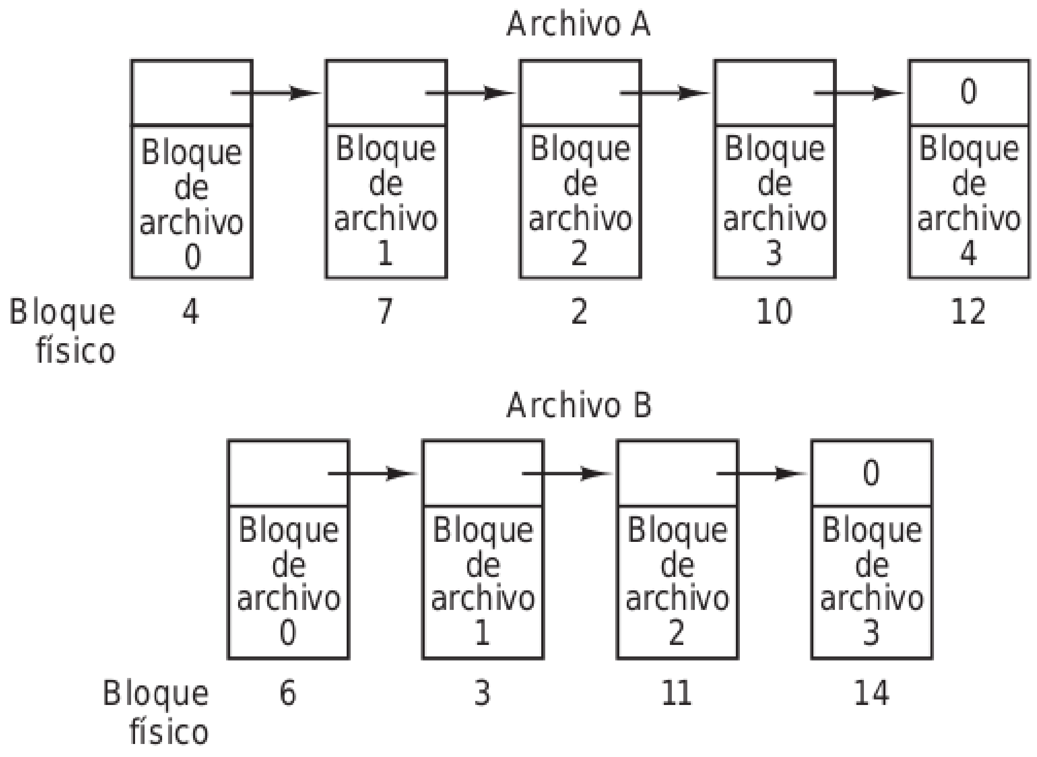


Figura 4-11. Almacenamiento de un archivo como una lista enlazada de bloques de disco.

Why lists over arrays?

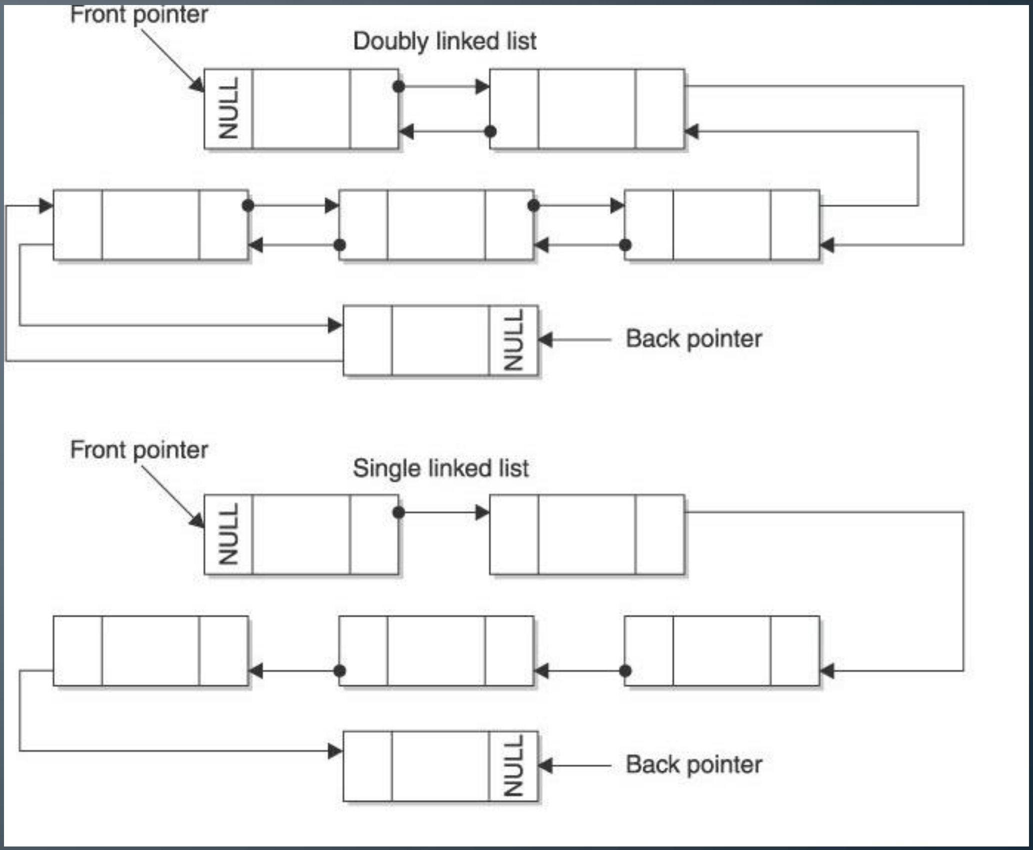
- Programmers choose linked list over an array because linked list can grow or shrink in size during runtime.
- If you change the size of the arrray, the operating system tries to increase the array by using memory alongside the array. If this location is unavailable, then the operating system finds another location large enough to hold elements of the array and new array elements. Elements of the array are then copies to the new location.

List elements

- Each entry in a linked list is called a node.
- A list is said to be empty when it contains no elements. The number of elements currently stored is called the length of the list. The beginning of the list is called the head, the end of the list is called the tail.

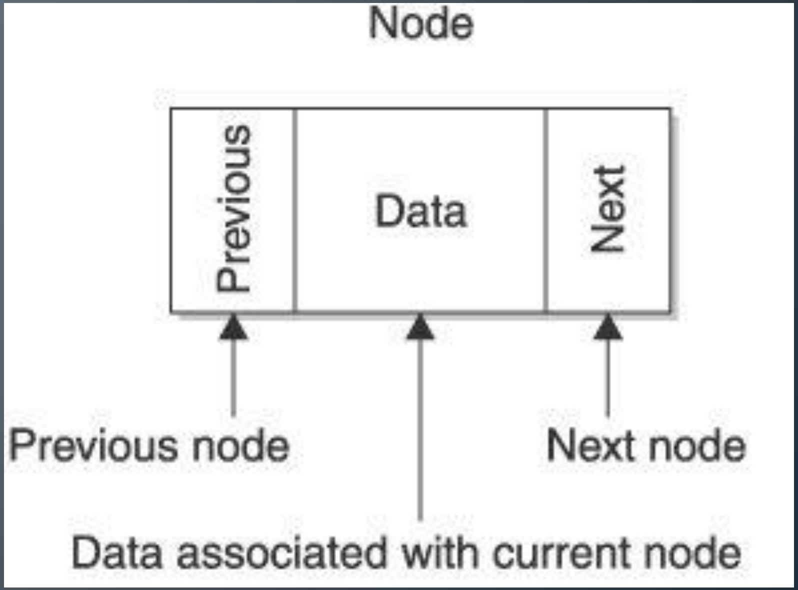
List clasification

- Single linked list
- Double linked list
- Circular list



A node

- A node is an entry that has tree subentries.
- One contains the data.
- Another a pointer to the previous node.
- The last one a pointer to the next node.



List operations

- Insert node
- Delete node
- Search and copy
- Show
- Destroy list