Linked List

• Objectives:
  – Discuss linked lists
    • Review
    • Typical operations
Linked List

Classification:

• Holds the series of objects (also called node)
• Singly linked list: each object, except the last one, has an address (reference) of another object.
• 2-way linked list: each object has an address of another object and in return the referenced object will has the referencing-object address.
• Doubly linked list: each object contains 2 addresses to reference two nodes. One of these two nodes must be of the same type and the other does not.
Linked List

- each Node must have at least 2 fields
  - Info field
    - Contains information,
    - is an object (not a node object) from another well-defined class
  - Link field
    - Contains a reference to another Node
    - Is the address of a Node
    - The address can be its own Node address.
    - It can be null. That mean it has no Node to reference to.
Linked List

• Example of a Node at 2205 in heap
Memory (RAM) structure

- 8-bit data bus
- Address Bus
- One Byte
Simple Node and RAM

Hex: 2205
Bin: 0010 0010 0000 0101

0000 0000
0000 1001
1110 1110
0000 0001

8-bit data bus
Symbolic Node

One Node

Reference line
Symbolic Internal Node

Reference line

infoField
Reference Variable & Node

Variable name: head
Singly linked list

- 2 nodes
circular linked list

• 3 nodes
2-way linked list

• 3 nodes

head

null

null

null
tail
Doubly linked list

- 3 nodes
Insert a new node

• The dotted linked must be established.
• head = new Node(new (infoField), head) ;
Delete a node

- The dotted linked must be established.
Linked List

summary

• store many objects with one or more references
• the objects are created by new and inserted into the list.
• An object is deleted when it is de-linked out of the list. This object will not be discarded in the memory until Virtual Machine reclaims it.