

Computer Science 1510

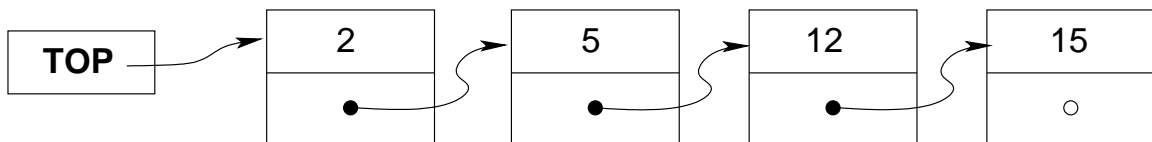
Lecture 20

Lecture Outline

- Linked lists

Linked List

- A linked list is a data structure consisting of a list of elements arranged one after another, with each element connected to the next via a *link*.



- The link is a pointer that points to the memory location of the next element.
- An additional pointer TOP points to the top of the list (the first node).

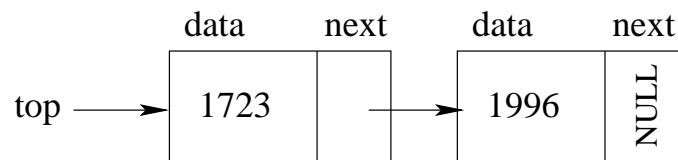
Example: Linked List

- In the following example we will create a linked list consisting of values (integers) entered by the user.
- The list will grow by one node each time that we read in a value.
- We will read values until the user stops entering them.
- Nodes of the linked list can be defined as follows:

```
TYPE List_Node
    INTEGER :: data
    TYPE(List_Node), POINTER :: next
END TYPE List_Node
```

Constructing a Linked List

- As an illustration of the basic steps in the construction of a linked list, assume that we have already stored two integers (1996 and 1723) in a linked list:



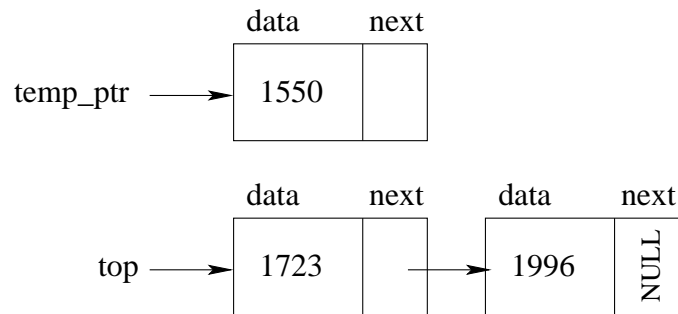
- Suppose that we want to add a new element, say 1550, to the list. To do this we would use two pointers, `top` to point to the first node in the list, and `temp_ptr` as a temporary pointer. We declare these pointers by
`TYPE(List_Node), POINTER :: top, temp_ptr`
- We first acquire a new node temporarily pointed to by `temp_ptr`:

`ALLOCATE(temp_ptr)`

and store the data there

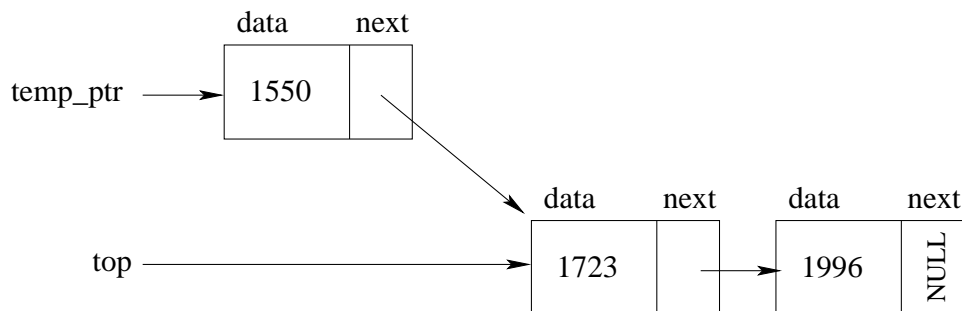
`temp_ptr%data = 1550`

- So now we have:



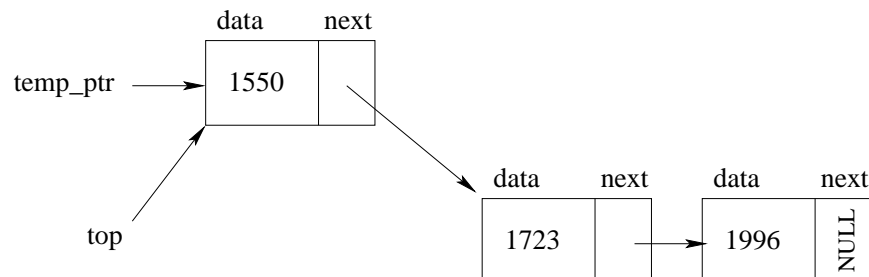
- This node can then be joined to the list by setting its link component to point to the first node:

`temp_ptr%next => top`



- The pointer `top` is then updated to point to this new node:

`top => temp_ptr`



Example: Linked List

```
PROGRAM Linkedlist
  IMPLICIT NONE
  TYPE Node
    INTEGER::value
    TYPE(Node),POINTER::next
  END TYPE
  INTEGER::i,InputStatus,AllocateStatus
  TYPE(Node),POINTER::top,cur

  NULLIFY(top) ! Initialize top to point to no target
! Read in node values
  WRITE(*,*) 'Please enter a value (CTRL^D to stop)'
  READ(*,*,IOSTAT=InputStatus) i
  DO WHILE(InputStatus==0) ! Read from the user until "END-OF-FILE"
    IF (.NOT.ASSOCIATED(top)) THEN ! The list has not been started
      ALLOCATE(top,STAT=AllocateStatus) ! Allocate first node
      IF (AllocateStatus /= 0) STOP
      NULLIFY(top%next) ! next does not yet point to anything
      cur=>top ! Let cur point to the start of the list
      cur%value=i ! set the first element's value to i
    ELSE ! Append the value to those already in the list
      ALLOCATE(cur%next,STAT=AllocateStatus) ! Allocate a new node
      IF (AllocateStatus /= 0) STOP
      NULLIFY(cur%next%next) ! Nullify the new node's next member
      cur=>cur%next ! Make cur point to the next node in the list.
      cur%value=i ! Set the value of this node to i
    END IF
    WRITE(*,*) 'Please enter a value (CTRL^D to stop)'
    READ(*,*,IOSTAT=InputStatus) i
  END DO

  IF (.NOT.ASSOCIATED(top)) THEN
    WRITE(*,*) 'No data read'
    STOP
  END IF
```

```
WRITE(*,*) 'The values are:'  
cur=>top ! Start at the first node in the list  
DO WHILE (ASSOCIATED(cur))  
    WRITE(*,*) cur%value  
    cur=>cur%next ! Move to the next node in the list  
END DO  
END PROGRAM Linkedlist
```