

Stack ADT

CS2263 – Systems Software Development

In Christ Alone, Celtic Worship (Keith Getty & Stuart Townend) <https://www.youtube.com/watch?v=8kvFtXphmMU>

1

Learning Outcomes

At the conclusion of this lecture students should be able to:

- itemize the subset of the list ADT functionality required for a queue ADT
- Implement that functionality as a linked-list to create a queue ADT

2

References

- Lu, Yung-Hsiang. 2015. Intermediate C Programming. CRC Press. New York. (Chapter 18)
- Remember – the textbook code is available from github:
 - <https://github.com/yunghsianglu/IntermediateCProgramming>

3

Linked-List Difficulties

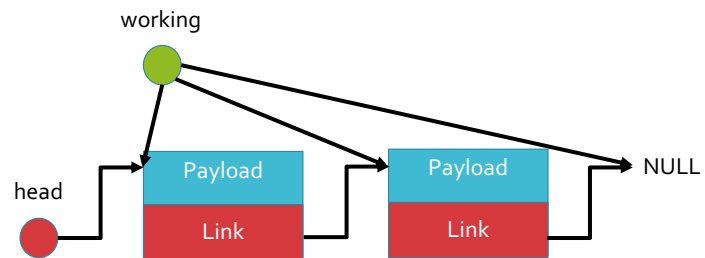
The hard tasks involved in implementing linked-lists

- Reading
 - Traversing the list (fairly easy)
- Updating
 - Adding links (nodes) (hard)
 - Removing links (nodes) (harder)

4

Linked-List Modelling I

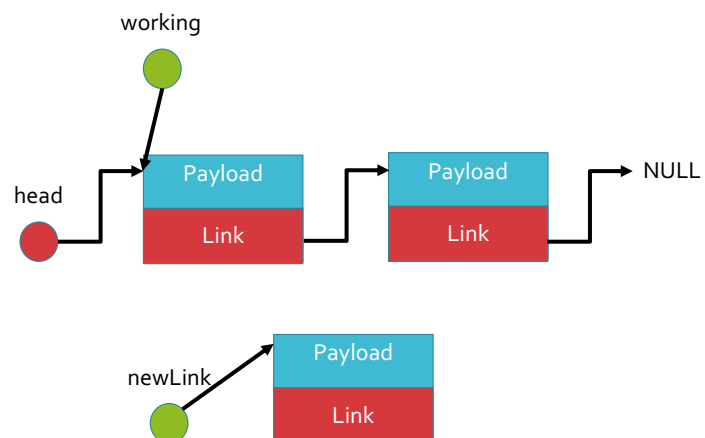
Traversing



5

Linked-List Modelling II

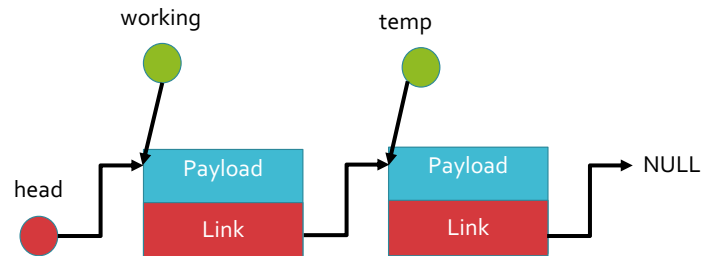
Inserting: (i) before head; (ii) after tail; (iii) in between



6

Linked-List Modelling III

Deleting: (i) head; (ii) tail; (iii) in between



7

Queue ADT

Subsets of List ADT

8

Queue ADT: Public Interface

Declared in the .h file, but in general functions can be

- Creation
- Read
 - ~~Search~~
 - Report
- Update
 - ~~Add an object at some location~~
 - Add an object (ENQUEUE)
 - ~~Remove an object from some location~~
 - Remove an object (DEQUEUE)
- Deletion
 - Including all objects

9

Queue ADT: Private Implementation

Differences from a list ADT

- Data structure
- Creation
- Adding links
- Removing links
- Deletion

10

Queue ADT: Data Structure

```
typedef struct charqueue {
    pCharLink head;
    pCharLink tail;
}CharQ, *pCharQ;
```

11

Queue ADT: Creation

- Need to declare at least a head pointer to the first link
- Handy to have a tail pointer to the last link (avoids traversing the list)

```
/*
 * Create an empty Queue
 */
pCharQ mallocCharQ() {
    pCharQ pCQThis = (pCharQ)malloc(sizeof(CharQ));
    if(pCQThis == (pCharQ)NULL )return pCQThis;
    pCQThis->head = (pCharQ)NULL;
    pCQThis->tail = (pCharQ)NULL;

    return pCQThis;
}
```

12

Queue ADT: Updating

- We will add to the tail; remove from the head (easiest)
- Three cases for addition/removal
 - First position (need to involve head pointer)
 - Last position (the next pointer should be NULL)
 - ~~In between (the general case)~~

```

/*
 * Enqueue an object to a list: add to tail
 */
int enqueueToCharQ(pCharList pCLThis, char c){
    pCharLink newLink;
    newLink = createCharLink(c);
    if(pCQThis->tail == pCQThis->head) pCQThis->head = newLink;
    if(pCQThis->tail == (pCharLink)NULL) pCQThis->tail = newLink;
    else pCQThis->tail->next = newLink;
    pCQThis->tail = newLink;

    return 0;
}

```

0 or 1
item in list

Empty list