Collision Resolution & Implementation

15-123

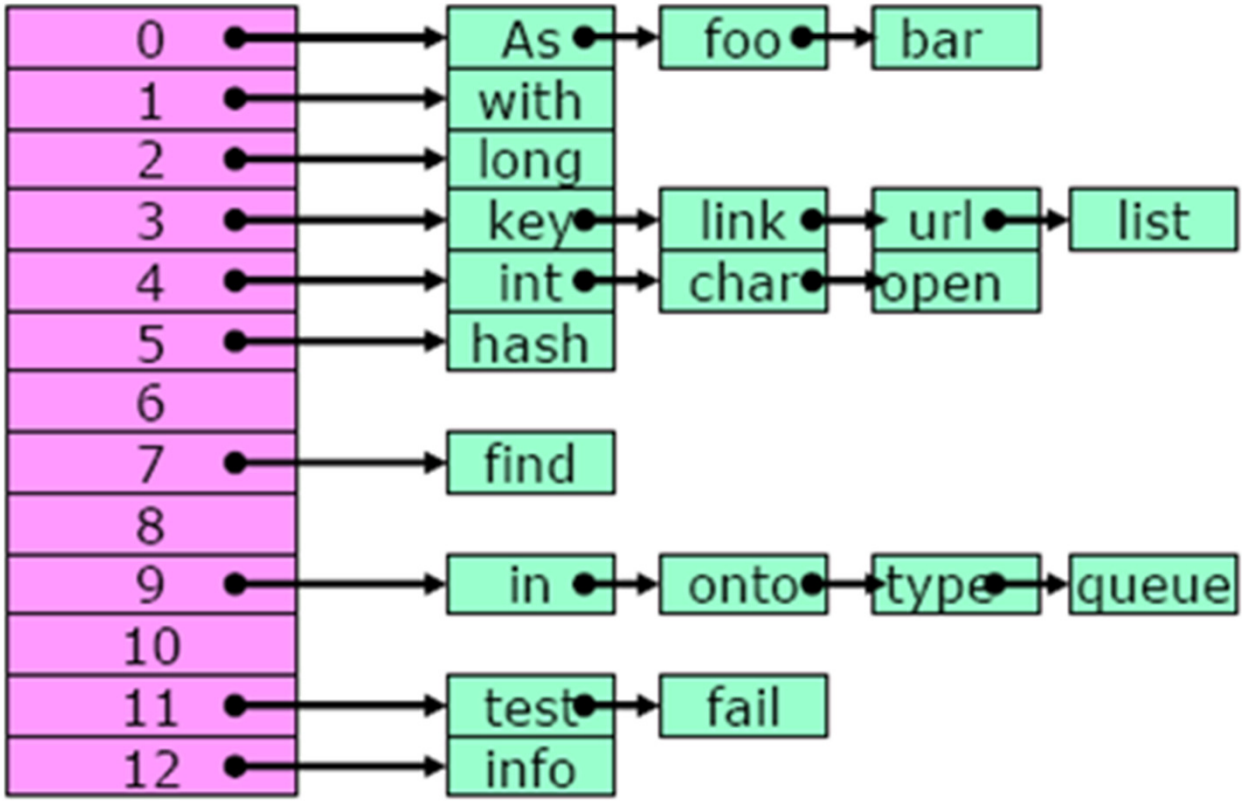
Systems Skills in C and Unix

What is a collision

- A collision occurs when two keys map to the same location
- Why do collisions occur?
  - Mainly due to bad hash functions
  - Eg: imagine hashing 1000 keys, where each key is on average 6 characters long, using a simple function like H(s) = ∑ characters

How to resolve collisions

Separate Chaining



Separate Chaining

- Pros
  - No probing necessary
    - Each node has a place in the same hashcode
  - List gets never full
    - Performance can go down though
- Cons
  - Complicated implementation of array of linked lists
  - Still lots of collisions can create a “bad” hash table

Load factor

- Need to keep the load factor reasonably under control
- If load factor becomes too large, rehash

Rehash

- The process of creating a larger table to distribute the keys better

Implementation

```
struct hashtable {
    void*  list;
    int size;
}
```

Client implementation

```
int hash(void* s, int m) {
    /* this takes a pointer to a key and
       computes the hash code. m is the table
       size
  */
}
```

Code Examples