Race Conditions

David Ferry
CSCI 2510 – Principles of Computing Systems
Saint Louis University
St. Louis, MO 63103

Definition

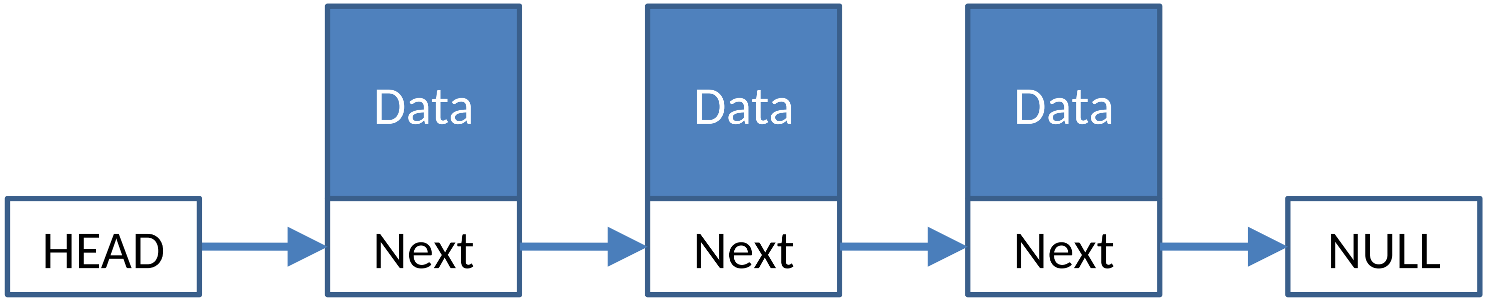
A *race condition* occurs whenever the output of a computation changes depending on the timing of execution.

Suppose x=0 initially:

Thread 1	Thread 2
u = x	v = x
u = u + 1	v = v * 2
x = u	x = v

What are the possible outcome values for x?

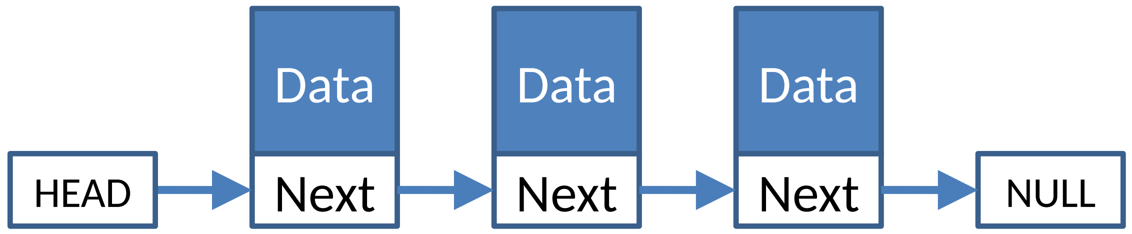
Linked List push() Example



```
push( node* newNode ){
   node* current = HEAD;
   while( current->next != NULL ){
      current = current->next;
   }
   current->next = newNode;
   newNode->next = NULL;
}
```

push() Race

Suppose two threads execute push() simultaneously:



Thread 1:

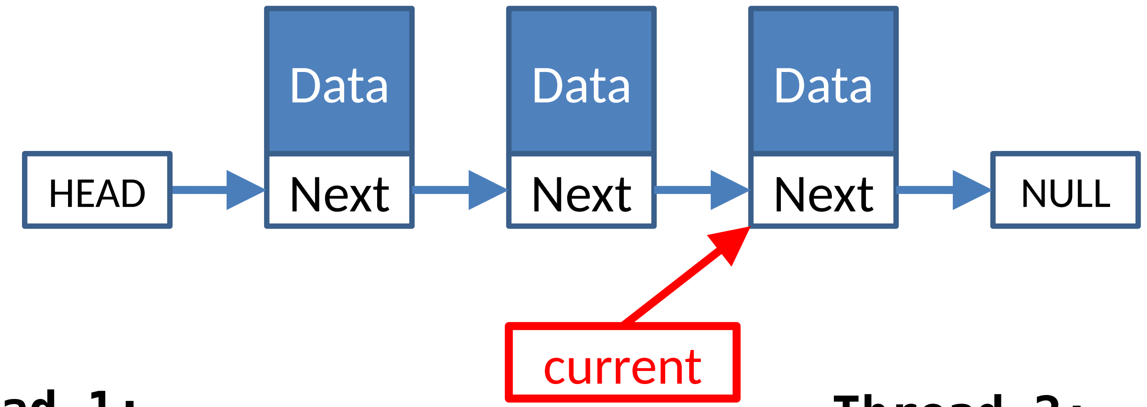
```
push( node* newNode ){
    node* current = HEAD;
    while( current->next != NULL ){
        current = current->next;
    }
    current->next = newNode;
    newNode->next = NULL;
}
```

Thread 2:

```
push( node* newNode ){
    node* current = HEAD;
    while( current->next != NULL )
        current = current->next;
    }
    current->next = newNode;
    newNode->next = NULL;
}
```

push() Race

Suppose two threads execute push() simultaneously:



hread 1:

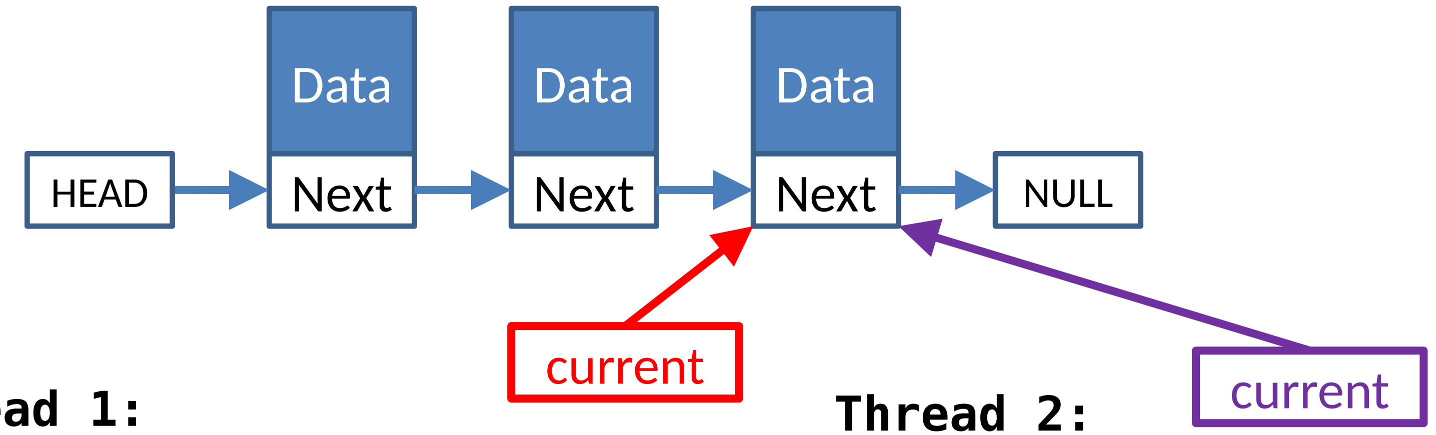
```
ush( node* newNode ){
    node* current = HEAD;
    while( current->next != NULL ){
        current = current->next;
    }
    current->next = newNode;
    newNode->next = NULL;
}
```

Thread 2:

```
push( node* newNode ){
    node* current = HEAD;
    while( current->next != NULL )
        current = current->next;
    }
    current->next = newNode;
    newNode->next = NULL;
}
```

push() Race

Suppose two threads execute push() simultaneously:



Thread 1:

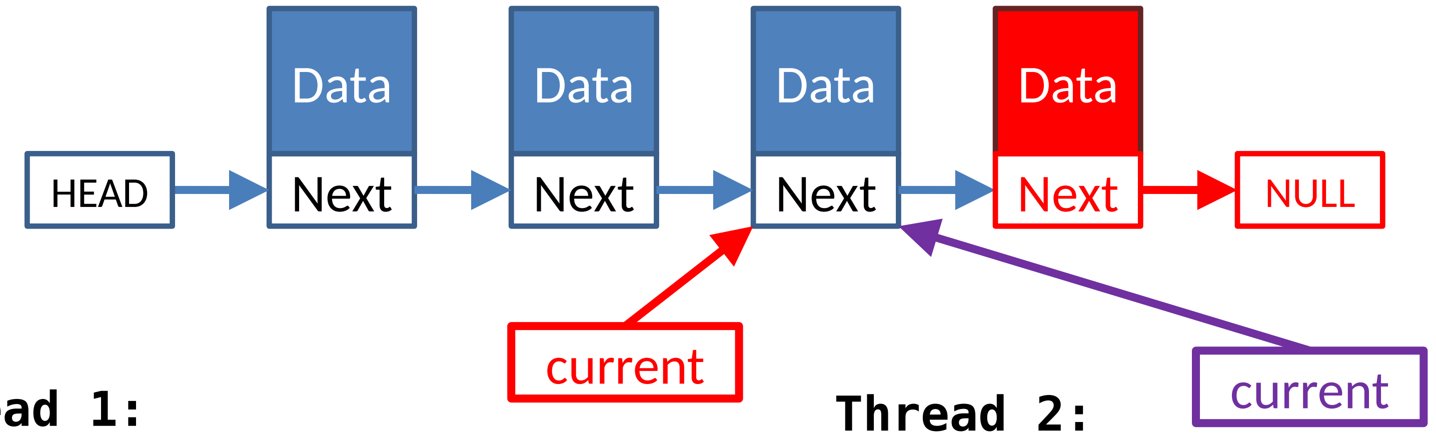
```
push( node* newNode ){
    node* current = HEAD;
    while( current->next != NULL ){
        current = current->next;
    }
    current->next = newNode;
    newNode->next = NULL;
}
```

Thread 2:

```
push( node* newNode ){
    node* current = HEAD;
    while( current->next != NULL )
        current = current->next;
    }
    current->next = newNode;
    newNode->next = NULL;
}
```

push() Race

Suppose two threads execute push() simultaneously:



Thread 1:

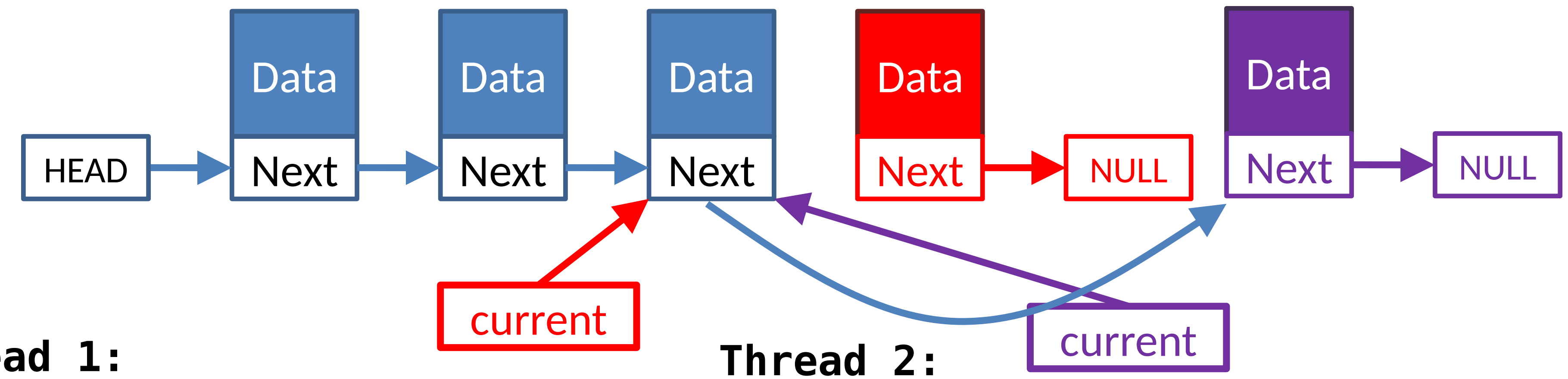
```
push( node* newNode ){
    node* current = HEAD;
    while( current->next != NULL ){
        current = current->next;
    }
    current->next = newNode;
    newNode->next = NULL;
}
```

Thread 2:

```
push( node* newNode ){
    node* current = HEAD;
    while( current->next != NULL )
        current = current->next;
    }
    current->next = newNode;
    newNode->next = NULL;
}
```

push() Race

Suppose two threads execute push() simultaneously:



Thread 1:

```
ush( node* newNode ){
    node* current = HEAD;
    while( current->next != NULL ){
        current = current->next;
    }
    current->next = newNode;
    newNode->next = NULL;
}
```

Thread 2:

```
push( node* newNode ){
    node* current = HEAD;
    while( current->next != NULL )
        current = current->next;
    }
    current->next = newNode;
    newNode->next = NULL;
}
```

At least basic arithmetic is safe, right? What could go wrong?

Thread 1:	Thread 2:
x++	x++

Not even increment is safe…

Suppose x=0 initially:

Thread 1: Thread 2:

X++ x++

Becomes:

Thread 1:

```
load x to register
increment register
store reg. to memory
```

Thread 2:

```
load x to register
increment register
store reg. to memory
```