comp1511 week 09

Catherine Cheng

admin

- how are we going with **assignment 2?**
- **myexperience** surveys are out: https://myexperience.unsw.edu.au/

agenda for today

- free
- linked list exercises

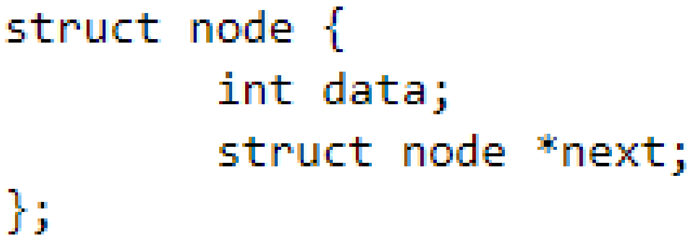
free (see free.c)

what is a memory leak?

how do we check for memory leaks?

how do we free a list?

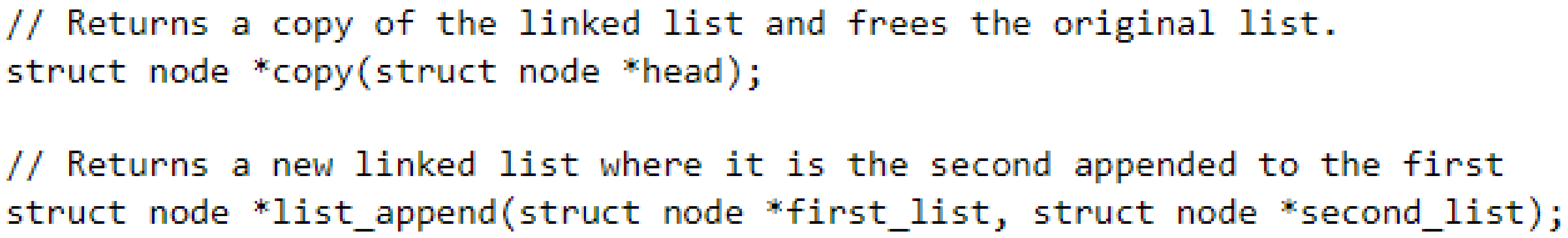
what is a use after free error?



```
struct node {
        int data;
        struct node *next;
};
```

linked list exercises

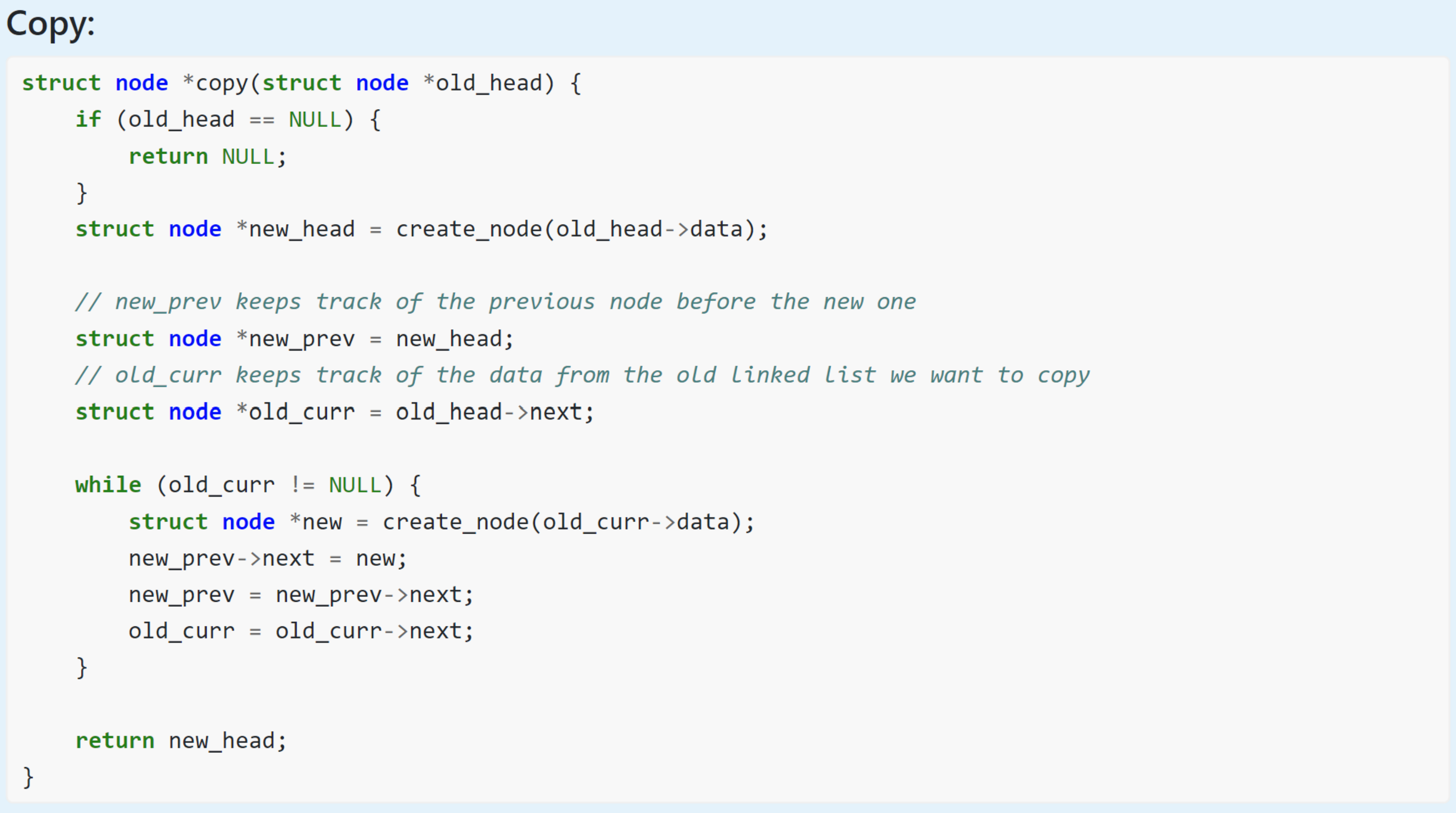
- draw a diagram of what things will look like, before and after
- identify how many things will need to be malloced or freed
- plan out steps for what will need to happen in the function
- list any special cases



```
// Returns a copy of the linked list and frees the original list.
struct node *copy(struct node *head);

// Returns a new linked list where it is the second appended to the first
struct node *list_append(struct node *first_list, struct node *second_list);
```

solutions



Copy:

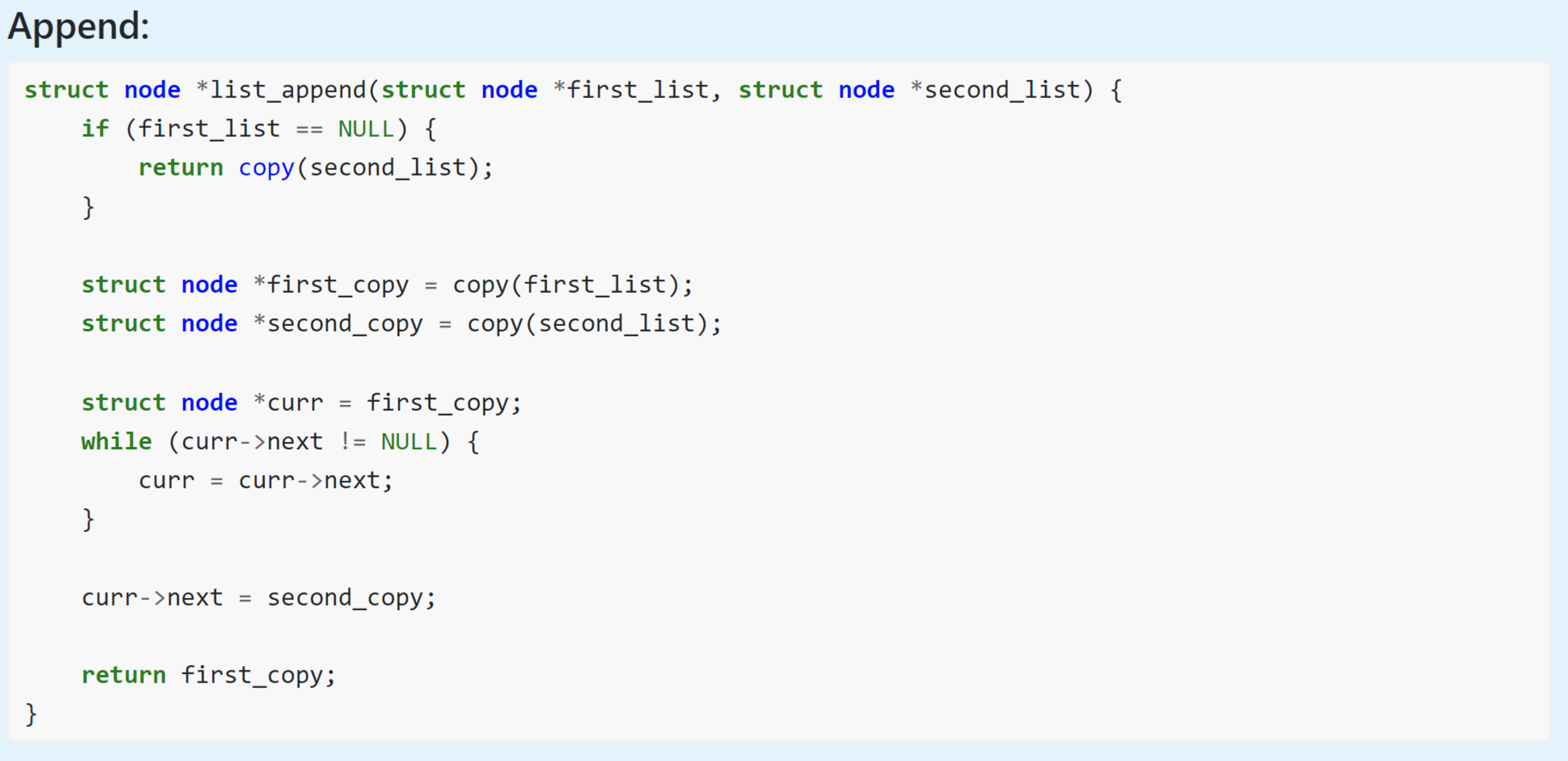
```
struct node *copy(struct node *old_head) {
    if (old_head == NULL) {
        return NULL;
    }
    struct node *new_head = create_node(old_head->data);

    // new_prev keeps track of the previous node before the new one
    struct node *new_prev = new_head;
    // old_curr keeps track of the data from the old linked list we want to copy
    struct node *old_curr = old_head->next;

    while (old_curr != NULL) {
        struct node *new = create_node(old_curr->data);
        new_prev->next = new;
        new_prev = new_prev->next;
        old_curr = old_curr->next;
    }

    return new_head;
}
```

solutions



Append:

```
struct node *list_append(struct node *first_list, struct node *second_list) {
    if (first_list == NULL) {
        return copy(second_list);
    }

    struct node *first_copy = copy(first_list);
    struct node *second_copy = copy(second_list);

    struct node *curr = first_copy;
    while (curr->next != NULL) {
        curr = curr->next;
    }

    curr->next = second_copy;

    return first_copy;
}
```