Linked-Lists

Kernel Linked Lists

- Linux kernel provides both doubly- and singly-linked list data structures
  - /usr/src/test_kernel/include/linux/list.h
- Useful for elevator module
  - List of people per floor
  - List of people in elevator
- Can use other Linux kernel data structures, but
  - I won't help you with debugging/implementation
  - They have more complexity than needed
    - You're just making queues

Kernel Linked Lists

- Pros
  - Safer/quicker than own ad-hoc implementation
  - Comes with several ready functions
  - Relatively easy to set up a FIFO queue
- Cons
  - Pointer manipulation can be tricky

struct list_head

```
struct list_head {
    struct list_head *next;
    struct list_head *prev;
};

/* Declare the start of the list */
struct list_head todo_list

/* Initialize the list to empty */
INIT_LIST_HEAD(&todo_list);
```

Embedding a list_head

```
struct item {
    struct list_head list
    int item_id;
    void *item_data;
};
struct item my_item;
/* fill my_item */

/* Add my_item to end of todo_list*/
list_add_tail(&my_item.list, &todo_list);
```

Other Linked List Functions

- Removing items
  - list_del(struct list_head *entry);
  - list_del_init(struct list_head *entry);
- Get surround struct of list_head node
  - list_entry(struct list_head *ptr, <type_of_struct>, name);
- Iterate through a linked list
  - list_for_each(struct list_head *ptr, struct list_head *list)