COSC 301: Operating Systems - Homework 2

Due Monday, September 17, 2012

Create a new function for our linked list library called list_insert_ordered to insert items in case-insensitive alphabetic (or rather, lexicographic) order. You can use the built-in C function strcasecmp(s1, s2) to compare strings in a case-insensitive way.

You should be able to use the list_dump function already defined to print the contents of the list for debugging, or modify main() in other ways to help with testing.

(This code and description are posted in the class git repo.)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct node {
    char name[128];
    struct node *next;
};

void list_insert_ordered(char *name,  struct node **head) {




}


void list_insert(char *name, struct node **head) {
     struct node *newnode = malloc(sizeof(struct node));
     strncpy(newnode->name, name, 127);

     newnode->next = *head;
     *head = newnode;
}

void list_dump(struct node *list) {
    int i = 0;
    printf("\n\nDumping list\n");
    while (list != NULL) {
        printf("%d: %s\n", i+1, list->name);
        list = list->next;
    }
}

void list_clear(struct node *list) {
    while (list != NULL) {
        struct node *tmp = list;
        list = list->next;
```

```
        free(tmp);
    }
}

int main(int argc, char **argv)
{
    char buffer[128];

    struct node *head = NULL;

    printf("Next string to add: ");
    fflush(stdout);
    while (fgets(buffer, 128, stdin) != NULL) {

        // old call to list_insert
        // list_insert(buffer, &head);

        list_insert_ordered(buffer, &head);

        printf("Next string to add: ");
        fflush(stdout);
    }

    list_print(head);
    list_clear(head);

    return 0;
}
```