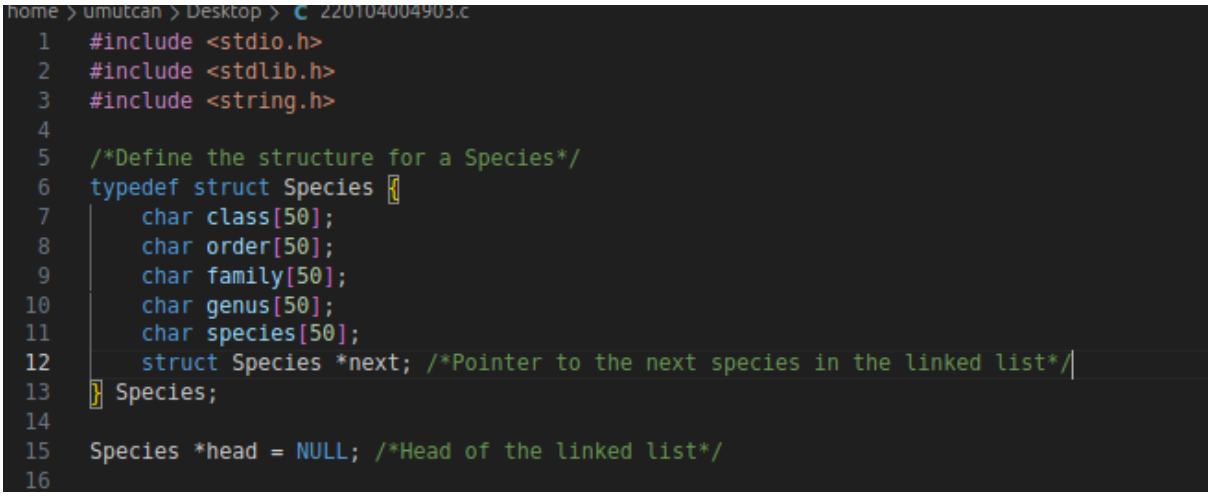
CSE 102 Spring 2024 – Computer Programming Assignment 13

Umutcan Ocak // 220104004903

Youtube : https://youtu.be/EpZhVJ7pRuo

This C program manages a linked list of species, allowing users to add, modify, delete, and sort species based on various criteria. It provides a menu-driven interface for user interaction, ensuring efficient species management through dynamic memory allocation and sorting algorithms.




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

/*Define the structure for a Species*/
typedef struct Species {
    char class[50];
    char order[50];
    char family[50];
    char genus[50];
    char species[50];
    struct Species *next; /*Pointer to the next species in the linked list*/
} Species;

Species *head = NULL; /*Head of the linked list*/

```

This code defines a structure for a species and initializes a pointer to the head of a linked list for storing species data.

```c
/*Function to add a new species to the linked list*/
void addSpecies() {
    Species *newSpecies = (Species *)malloc(sizeof(Species));
    if (newSpecies == NULL) {
        printf("Memory allocation failed!\n");
        return;
    }

    /*Get species information from the user*/
    printf("Enter class: ");
    fgets(newSpecies->class, sizeof(newSpecies->class), stdin);
    newSpecies->class[strcspn(newSpecies->class, "\n")] = 0;

    printf("Enter order: ");
    fgets(newSpecies->order, sizeof(newSpecies->order), stdin);
    newSpecies->order[strcspn(newSpecies->order, "\n")] = 0;

    printf("Enter family: ");
    fgets(newSpecies->family, sizeof(newSpecies->family), stdin);
    newSpecies->family[strcspn(newSpecies->family, "\n")] = 0;

    printf("Enter genus: ");
    fgets(newSpecies->genus, sizeof(newSpecies->genus), stdin);
    newSpecies->genus[strcspn(newSpecies->genus, "\n")] = 0;

    printf("Enter species: ");
    fgets(newSpecies->species, sizeof(newSpecies->species), stdin);
    newSpecies->species[strcspn(newSpecies->species, "\n")] = 0;

    newSpecies->next = head;
    head = newSpecies;
}

```

This function allocates memory for a new species, collects its data from the user, and adds it to the head of the linked list.

Inputs:

```
Menu:
1. Add Species
2. Modify Species
3. Delete Species
4. Sort and Display by Criteria
5. Exit
Enter your choice: 1
Enter class: Mammalia
Enter order: Carnivora
Enter family: Felidae
Enter genus: Panthera
Enter species: Panthera leo

Menu:
1. Add Species
2. Modify Species
3. Delete Species
4. Sort and Display by Criteria
5. Exit
Enter your choice: 1
Enter class: Mammalia
Enter order: Carnivora
Enter family: Canidae
Enter genus: Canis
Enter species: Canis lupus

Menu:
1. Add Species
2. Modify Species
3. Delete Species
4. Sort and Display by Criteria
5. Exit
Enter your choice: 1
Enter class: Magnoliopsida
Enter order: Rosales
Enter family: Rosaceae
Enter genus: Rosa
Enter species: Rosa canina

Menu:
```

```c
}

/*Function to print a species' details*/
void printSpecies(Species *sp) {
    printf("Class: %s\nOrder: %s\nFamily: %s\nGenus: %s\nSpecies: %s\n\n", sp->class, sp->order, sp->family, sp->genus, sp->species);
}
```

```c
void modifySpecies() {
    char targetSpecies[50];
    printf("Enter the species name to modify: ");
    fgets(targetSpecies, sizeof(targetSpecies), stdin);
    targetSpecies[strcspn(targetSpecies, "\n")] = 0;

    Species *current = head;
    while (current != NULL) {
        if (strcmp(current->species, targetSpecies) == 0) {
            printf("Current information:\n");
            printSpecies(current);

            printf("Enter new class (or press Enter to keep current): ");
            char newClass[50];
            fgets(newClass, sizeof(newClass), stdin);
            if (newClass[0] != '\n') {
                newClass[strcspn(newClass, "\n")] = 0;
                strcpy(current->class, newClass);
            }

            printf("Enter new order (or press Enter to keep current): ");
            char newOrder[50];
            fgets(newOrder, sizeof(newOrder), stdin);
            if (newOrder[0] != '\n') {
                newOrder[strcspn(newOrder, "\n")] = 0;
                strcpy(current->order, newOrder);
            }

            printf("Enter new family (or press Enter to keep current): ");
            char newFamily[50];
            fgets(newFamily, sizeof(newFamily), stdin);
            if (newFamily[0] != '\n') {
                newFamily[strcspn(newFamily, "\n")] = 0;
                strcpy(current->family, newFamily);
            }

            printf("Enter new genus (or press Enter to keep current): ");
            char newGenus[50];
            fgets(newGenus, sizeof(newGenus), stdin);
            if (newGenus[0] != '\n') {
                newGenus[strcspn(newGenus, "\n")] = 0;
                strcpy(current->genus, newGenus);
            }

            printf("Enter new species (or press Enter to keep current): ");
            char newSpecies[50];
            fgets(newSpecies, sizeof(newSpecies), stdin);
            if (newSpecies[0] != '\n') {
                newSpecies[strcspn(newSpecies, "\n")] = 0;
                strcpy(current->species, newSpecies);
            }

            printf("Species modified successfully!\n");
            return;
        }
        current = current->next;
    }
```

This code provides functions to print a species' details and to modify an existing species' details in a linked list by updating its attributes based on user input.

OUTPUT :

```
Menu:
1. Add Species
2. Modify Species
3. Delete Species
4. Sort and Display by Criteria
5. Exit
Enter your choice: 2
Enter the species name to modify: Rosa canina
Current information:
Class: Magnoliopsida
Order: Rosales
Family: Rosaceae
Genus: Rosa
Species: Rosa canina

Enter new class (or press Enter to keep current):
Enter new order (or press Enter to keep current):
Enter new family (or press Enter to keep current): Rosas
Enter new genus (or press Enter to keep current):
Enter new species (or press Enter to keep current):
Species modified successfully!

Menu:
1. Add Species
2. Modify Species
3. Delete Species
4. Sort and Display by Criteria
5. Exit
Enter your choice: 4
Sort by (class, order, family, genus, species): class
Class: Magnoliopsida
Order: Rosales
Family: Rosas
Genus: Rosa
Species: Rosa canina

Class: Mammalia
Order: Carnivora
Family: Felidae
Genus: Panthera
Species: Panthera leo
```

```c
/*Function to delete a species from the linked list*/
void deleteSpecies() {
    char targetSpecies[50];
    printf("Enter the species name to delete: ");
    fgets(targetSpecies, sizeof(targetSpecies), stdin);
    targetSpecies[strcspn(targetSpecies, "\n")] = 0;

    Species *current = head;
    Species *previous = NULL;
    while (current != NULL) {
        if (strcmp(current->species, targetSpecies) == 0) {
            if (previous == NULL) {
                head = current->next;
            } else {
                previous->next = current->next;
            }
            free(current);/*Free the memory*/
            printf("Species deleted successfully!\n");
            return;
        }
        previous = current;
        current = current->next;
    }
    printf("Species not found!\n");
}
```

This function deletes a species from the linked list by matching the species name provided by the user, updating the links, and freeing the allocated memory if a match is found.

```
Menu:
1. Add Species
2. Modify Species
3. Delete Species
4. Sort and Display by Criteria
5. Exit
Enter your choice: 3
Enter the species name to delete: Canis lupus
Species deleted successfully!

Menu:
1. Add Species
2. Modify Species
3. Delete Species
4. Sort and Display by Criteria
5. Exit
Enter your choice: 4
Sort by (class, order, family, genus, species): genus
Class: Mammalia
Order: Carnivora
Family: Felidae
Genus: Panthera
Species: Panthera leo

Class: Magnoliopsida
Order: Rosales
Family: Rosaceae
Genus: Rosa
Species: Rosa canina


Menu:
```

```c
/*Function to sort the linked list by a given criteria*/
void sortLinkedList(int (*compare)(Species *, Species *)) {
    if (head == NULL) return;

    int swapped;
    Species *ptr1;
    Species *lptr = NULL;

    do {
        swapped = 0;
        ptr1 = head;

        /* Traverse the linked list */
        while (ptr1->next != lptr) {
            if (compare(ptr1, ptr1->next) > 0) {
                /* Swap the data */
                char tempClass[50], tempOrder[50], tempFamily[50], tempGenus[50], tempSpecies[50];

                strcpy(tempClass, ptr1->class);
                strcpy(tempOrder, ptr1->order);
                strcpy(tempFamily, ptr1->family);
                strcpy(tempGenus, ptr1->genus);
                strcpy(tempSpecies, ptr1->species);

                strcpy(ptr1->class, ptr1->next->class);
                strcpy(ptr1->order, ptr1->next->order);
                strcpy(ptr1->family, ptr1->next->family);
                strcpy(ptr1->genus, ptr1->next->genus);
                strcpy(ptr1->species, ptr1->next->species);

                strcpy(ptr1->next->class, tempClass);
                strcpy(ptr1->next->order, tempOrder);
                strcpy(ptr1->next->family, tempFamily);
                strcpy(ptr1->next->genus, tempGenus);
                strcpy(ptr1->next->species, tempSpecies);

                /* Set swapped */
                swapped = 1;
            }
            ptr1 = ptr1->next; /* Move to the next node */
        }
        lptr = ptr1;
        /* Update the last node */
    } while (swapped);
}
```

This function sorts the linked list of species based on a given comparison function, using the bubble sort algorithm to swap species data when necessary.

OUTPUT :

```
Menu:
1. Add Species
2. Modify Species
3. Delete Species
4. Sort and Display by Criteria
5. Exit
Enter your choice: 4
Sort by (class, order, family, genus, species): family
Class: Mammalia
Order: Carnivora
Family: Canidae
Genus: Canis
Species: Canis lupus

Class: Mammalia
Order: Carnivora
Family: Felidae
Genus: Panthera
Species: Panthera leo

Class: Magnoliopsida
Order: Rosales
Family: Rosaceae
Genus: Rosa
Species: Rosa canina
```

```
Menu:
1. Add Species
2. Modify Species
3. Delete Species
4. Sort and Display by Criteria
5. Exit
Enter your choice: 4
Sort by (class, order, family, genus, species): species
Class: Mammalia
Order: Carnivora
Family: Canidae
Genus: Canis
Species: Canis lupus

Class: Mammalia
Order: Carnivora
Family: Felidae
Genus: Panthera
Species: Panthera leo

Class: Magnoliopsida
Order: Rosales
Family: Rosaceae
Genus: Rosa
Species: Rosa canina
```

```c
/*Functions to compare species by different criteria*/
int compareByClass(Species *a, Species *b) {
    return strcmp(a->class, b->class);
}

int compareByOrder(Species *a, Species *b) {
    return strcmp(a->order, b->order);
}

int compareByFamily(Species *a, Species *b) {
    return strcmp(a->family, b->family);
}

int compareByGenus(Species *a, Species *b) {
    return strcmp(a->genus, b->genus);
}

int compareBySpecies(Species *a, Species *b) {
    return strcmp(a->species, b->species);
}
```

These functions compare two species based on different criteria (class, order, family, genus, and species) by using the `'strcmp'` function to return the result of the comparison.

```c
/*Function to sort and display species by a given criteria*/
void sortAndDisplaySpecies(int (*compareFunc)(Species *, Species *)) {
    sortLinkedList(compareFunc);
    Species *current = head;
    while (current != NULL) {
        printSpecies(current);
        current = current->next;
    }
}
```

This function sorts the species in the linked list based on a given comparison function and then displays the sorted species.

```c
void menu() {
    int choice;
    do {
        printf("\nMenu:\n");
        printf("1. Add Species\n");
        printf("2. Modify Species\n");
        printf("3. Delete Species\n");
        printf("4. Sort and Display by Criteria\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        if (scanf("%d", &choice) != 1) {
            while (getchar() != '\n');
            printf("Invalid input! Please enter a number.\n");
            continue;
        }
        /*Clear the input buffer*/
        while (getchar() != '\n');

        switch (choice) {
            case 1:
                addSpecies();
                break;
            case 2:
                modifySpecies();
                break;
            case 3:
                deleteSpecies();
                break;
            case 4: {
                char criteria[50];
                /*Get the criteria from the user*/
                printf("Sort by (class, order, family, genus, species): ");
                fgets(criteria, sizeof(criteria), stdin);
                criteria[strcspn(criteria, "\n")] = 0;

                if (strcmp(criteria, "class") == 0) {
                    sortAndDisplaySpecies(compareByClass);
                } else if (strcmp(criteria, "order") == 0) {
                    sortAndDisplaySpecies(compareByOrder);
                } else if (strcmp(criteria, "family") == 0) {
                    sortAndDisplaySpecies(compareByFamily);
                } else if (strcmp(criteria, "genus") == 0) {
                    sortAndDisplaySpecies(compareByGenus);
                } else if (strcmp(criteria, "species") == 0) {
                    sortAndDisplaySpecies(compareBySpecies);
                } else {
                    printf("Invalid criteria choice!\n");
                }
                break;
            }
            case 5:
                printf("Exiting...\n");
                /*Free the memory*/
                break;
            default:
                printf("Invalid choice! Please try again.\n");
        }
    } while (choice != 5);
```

This function displays a menu that allows the user to add, modify, delete, sort, and display species based on various criteria, or exit the program.

```c

int main() {
    /*Display the menu*/
    menu();
    return 0; /*Return 0 to indicate successful completion*/
}

```