Singly linked list

#include <stdio.h>

#include <stdlib.h>

// Define the structure for a node in the linked list

struct Node {

int data;

struct Node\* next;

};

// Define the structure to encapsulate the linked list

struct LinkedList {

struct Node\* head;

};

// Function to initialize an empty linked list

void initializeList(struct LinkedList\* list) {

list->head = NULL;

}

// Function to insert a new node at the beginning of the linked list

void insertAtBeginning(struct LinkedList\* list, int data) {

struct Node\* newNode = (struct Node\*)malloc(sizeof(struct Node));

if (newNode == NULL) {

printf("Memory allocation failed\n");

exit(EXIT\_FAILURE);

}

newNode->data = data;

newNode->next = list->head;

list->head = newNode;

}

// Function to display the linked list

void displayList(struct LinkedList\* list) {

struct Node\* current = list->head;

printf("Linked List: ");

while (current != NULL) {

printf("%d -> ", current->data);

current = current->next;

}

printf("NULL\n");

}

int main() {

// Create a linked list and containerize it

struct LinkedList myList;

initializeList(&myList);

// Insert elements into the linked list

insertAtBeginning(&myList, 3);

insertAtBeginning(&myList, 7);

insertAtBeginning(&myList, 12);

// Display the linked list

displayList(&myList);

return 0;

}

Dockerfile

# Use the official Ubuntu base image

FROM ubuntu:latest

# Install build-essential to compile C code

RUN apt-get update && apt-get install -y build-essential

# Set the working directory

WORKDIR /app

# Copy the C program files into the container

COPY . .

# Compile the C program

RUN gcc -o myprogram myprogram.c

# Command to run when the container starts

CMD ["./myprogram"]

Commands

1. docker build -t myprogram-image .
2. docker run myprogram-image