k.vamshi ,

Roll No:422175;

Unix assignment->03

Create shell scripts for generating static and dynamic libraries. Utilize an example of your preference. Ensure that the program is not a simple calculator task and also from the provided tutorial material. Each program should incorporate a minimum of two functions based on the selected task.

```
//create_node.c: filename

#include <stdio.h>
#include <stdlib.h>
#include "unique_functions.h"

Node* createNode(int data) {
   Node* newNode = (Node*)malloc(sizeof(Node));
   if (newNode == NULL) {
      printf("Memory allocation failed\n");
      exit(EXIT_FAILURE);
   }
   newNode->data = data;
   newNode->next = NULL;
   return newNode;
}
```

```
//delete_node.c:filename:
#include <stdio.h>
#include <stdlib.h>
#include "unique_functions.h"

void deleteNode(Node** headRef, int key) {
    Node* temp = *headRef;
    Node* prev = NULL;

    // If head node itself holds the key to be deleted
    if (temp != NULL && temp->data == key) {
        *headRef = temp->next;
        free(temp);
        return;
    }

    // Search for the key to be deleted, keep track of the previous node as we need to change
'prev->next'
    while (temp != NULL && temp->data != key) {
        prev = temp;
        temp = temp->next;
    }

    // If key was not present in linked list
    if (temp == NULL) {
        printf("Key not found in the linked list\n");
        return;
}

    // Unlink the node from linked list
    prev->next = temp->next;
    free(temp);
}
```

```
//main.c:

#include <stdio.h>
#include "unique_functions.h"

int main() {
    Node* head = NULL;

    printf("Appending elements to the linked list...\n");
    head = createNode(40);
    head->next = createNode(20);
    head->next->next = createNode(80);
    head->next->next->next = createNode(90);

    printf("Printing the linked list...\n");
    printList(head);

    printf("Deleting node with data 20...\n");
    deleteNode(&head, 20);
    printf("Printing the updated linked list...\n");
    printList(head);

    return 0;




}
```

```
//print_list.c:

#include <stdio.h>
#include "unique_functions.h"

void printList(Node* head) {
    printf("Linked list: ");
    while (head != NULL) {
        printf("%d ", head->data);
        head = head->next;
    }
    printf("\n");
}
```

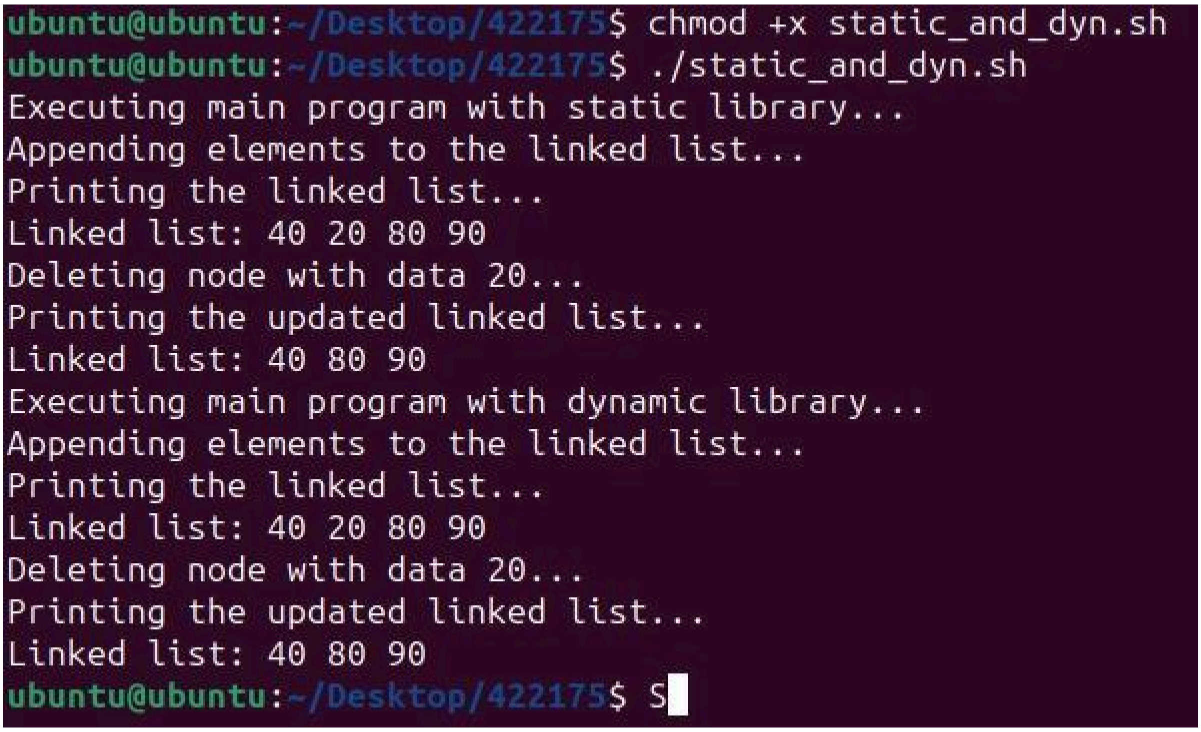
```
# Compile the source files into object files
gcc -c -fPIC create_node.c delete_node.c print_list.c

# Create static library
ar rcs libunique_functions.a create_node.o delete_node.o print_list.o

# Create dynamic library
gcc -shared -o libunique_functions.so create_node.o delete_node.o print_list.o

# Check if the dynamic library was created successfully
if [ ! -f "libunique_functions.so" ]; then
   echo "Error: Unable to create the dynamic library"
   exit 1
fi

# Set library path
export LD_LIBRARY_PATH=$(pwd):$LD_LIBRARY_PATH

# Compile main program with static library
gcc main.c -L. -lunique_functions -o main_static
```

```
# Check if the main program was compiled successfully
if [ ! -f "main_static" ]; then
   echo "Error: Unable to compile the main program with static library"
   exit 1
fi
# Compile main program with dynamic library
gcc main.c -L. -lunique_functions -o main_dynamic

# Check if the main program was compiled successfully
if [ ! -f "main_dynamic" ]; then
   echo "Error: Unable to compile the main program with dynamic library"
   exit 1
fi

# Execute main programs
echo "Executing main program with static library..."
./main_static

echo "Executing main program with dynamic library..."
./main_dynamic

# Clean up object files and executables
rm *.o libunique_functions.a libunique_functions.so main_static main_dynamic
```

```
//unique_functions.h:

#ifndef UNIQUE_FUNCTIONS_H
#define UNIQUE_FUNCTIONS_H
```

```
typedef struct Node {
    int data;
    struct Node* next;
} Node;

Node* createNode(int data);
void deleteNode(Node** headRef, int key);
void printList(Node* head);

#endif /* UNIQUE_FUNCTIONS_H */
```

Output:



```
ubuntu@ubuntu:~/Desktop/422175$ chmod +x static_and_dyn.sh
ubuntu@ubuntu:~/Desktop/422175$ ./static_and_dyn.sh
Executing main program with static library...
Appending elements to the linked list...
Printing the linked list...
Linked list: 40 20 80 90
Deleting node with data 20...
Printing the updated linked list...
Linked list: 40 80 90
Executing main program with dynamic library...
Appending elements to the linked list...
Printing the linked list...
Linked list: 40 20 80 90
Deleting node with data 20...
Printing the updated linked list...
Linked list: 40 80 90
ubuntu@ubuntu:~/Desktop/422175$ S
```