OS CA-2

(Roll nos. 41 to 50)

Name - **SHREYAS NAIK**
Roll no - **41**
Division - D10B

Module 4 - Write a C program to implement LFU page replacement algorithm.

→

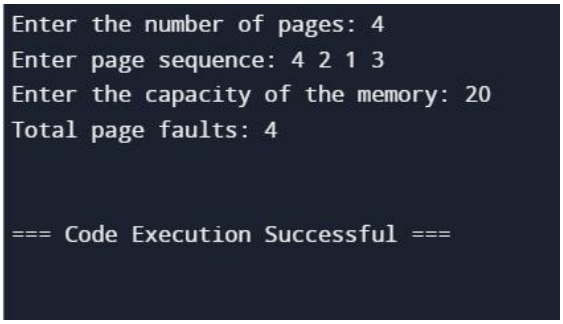
Code -

```
#include <stdio.h>
#include <limits.h>

#define MAX_PAGES 10

// Function to find the index of the least frequently used
page int findLFU(int pages[], int frequency[], int n) {
   int min_freq = INT_MAX;
   int idx = -1;
   for (int i = 0; i < n; ++i) {
      if (frequency[i] < min_freq)
         { min_freq = frequency[i];
         idx = i;
      }
   }
   return idx;
}

// Function to implement LFU page replacement
algorithm void lfu(int pages[], int n, int capacity) {
   int frame[capacity], frequency[capacity], pageFaults = 0;
```

```
for (int i = 0; i < capacity; ++i) {
   frame[i] = -1;
   frequency[i] = 0;
}

for (int i = 0; i < n; ++i) {
   int found = 0;
   for (int j = 0; j < capacity; ++j) {
      if (frame[j] == pages[i]) {
         found = 1;
         ++frequency[j];
         break;
      }
   }

   if (!found) {
      int empty = 0;
      for (int j = 0; j < capacity; ++j) {
         if (frame[j] == -1) {
            frame[j] = pages[i];
            frequency[j] = 1;
            empty = 1;
            break;
         }
      }

      if (!empty) {
         int idx = findLFU(pages, frequency, capacity);
         frame[idx] = pages[i];
         frequency[idx] = 1;
      }
      ++pageFaults;
   }
}
```

```
    printf("Total page faults: %d\n", pageFaults);
}

int main() {
    int n, pages[MAX_PAGES], capacity;

    printf("Enter the number of pages: ");
    scanf("%d", &n);

    printf("Enter page sequence: ");
    for (int i = 0; i < n; ++i)
        scanf("%d", &pages[i]);

    printf("Enter the capacity of the memory: ");
    scanf("%d", &capacity);

    lfu(pages, n, capacity);

    return 0;
}
```

Output -



Module 5 - Implement various disk scheduling algorithms like LOOK, C-LOOK in C/Python/Java.

→

Code -

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

// Function to implement LOOK disk scheduling
algorithm void look(int requests[], int head, int size) {
   int totalSeekTime = 0; int
   cur_track = head;
   printf("Seek Sequence: ");
   for (int i = 0; i < size; ++i) {
      printf("%d ", requests[i]);
      totalSeekTime += abs(cur_track - requests[i]);
      cur_track = requests[i];
   }
   printf("\nTotal Seek Time: %d\n", totalSeekTime);
}

// Function to implement C-LOOK disk scheduling
algorithm void c_look(int requests[], int head, int size) {
   int totalSeekTime = 0; int
   cur_track = head;
   printf("Seek Sequence: ");
   for (int i = 0; i < size; ++i) {
      printf("%d ", requests[i]);
      totalSeekTime += abs(cur_track - requests[i]);
      cur_track = requests[i];
   }
   printf("\nTotal Seek Time: %d\n", totalSeekTime);
}

int main() {
   int size, head;
```

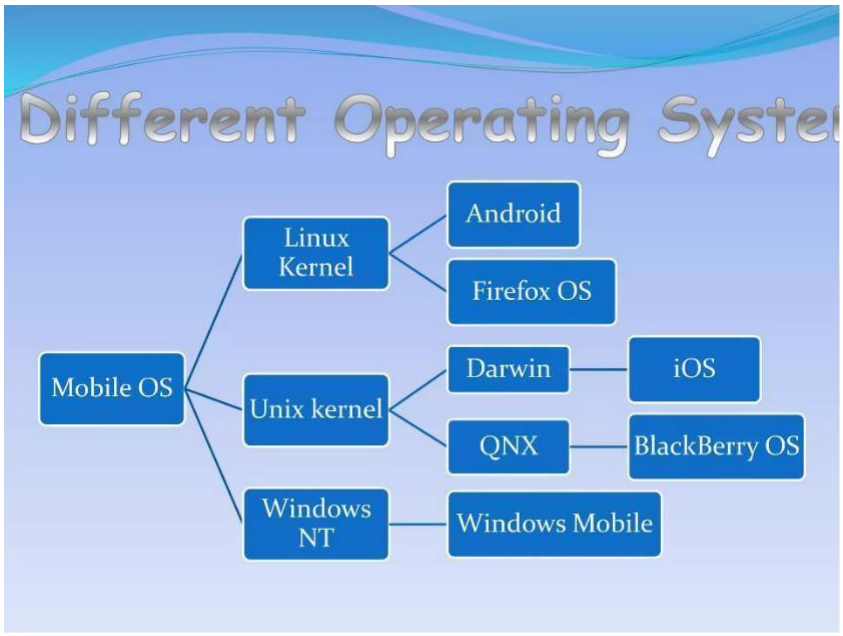
```
    printf("Enter the number of disk requests: ");
    scanf("%d", &size);
    int *requests = (int *)malloc(size * sizeof(int));
    if (requests == NULL) {
        printf("Memory allocation failed.\n");
        return 1;
    }
    printf("Enter the disk requests: ");
    for (int i = 0; i < size; i++) {
        scanf("%d", &requests[i]);
    }
    printf("Enter initial position of head: ");
    scanf("%d", &head);

    // Look Algorithm
    look(requests, head, size);
    // C-Look Algorithm
    c_look(requests, head, size);
    free(requests);

    return 0;
}
```

Output -

```
Enter the number of disk requests: 4
Enter the disk requests: 1 2 5 6
Enter initial position of head: 3
Seek Sequence: 1 2 5 6
Total Seek Time: 7
Seek Sequence: 1 2 5 6
Total Seek Time: 7


=== Code Execution Successful ===
```

Module 6 - Case Study on Mobile Operating System.

→



Introduction:
Mercedes-Benz, a renowned luxury automobile manufacturer, has been at the forefront of innovation, not only in vehicle engineering but also in the integration of advanced technology

Problem Statement:
Mercedes-Benz faced challenges in integrating various vehicle systems and providing a seamless user experience across its range of vehicles.

Solution:

1. Mercedes-Benz User Experience (MBUX):
Mercedes-Benz introduced the MBUX, an advanced infotainment system powered by a sophisticated operating system. MBUX integrates various vehicle functions, including navigation, entertainment, communication, and vehicle settings, into an intuitive and user-friendly interface.

2. Features and Functionality:

Voice Recognition: MBUX features natural language processing for voice commands, allowing drivers to control various functions without taking their hands off the wheel.

Artificial Intelligence: The system employs artificial intelligence to learn driver preferences and habits, providing personalized recommendations and assistance.

Integration with Smart Devices: MBUX seamlessly integrates with smartphones and other smart devices, allowing drivers to access their digital ecosystem from within the vehicle.

Over-the-Air Updates: Mercedes-Benz offers over-the-air updates for MBUX, ensuring that vehicles receive the latest features, improvements, and security patches without requiring a visit to the dealership.

3. Security:

Mercedes-Benz prioritizes the security of its operating system, implementing robust encryption protocols and security measures to protect user data and prevent unauthorized access.
The company regularly conducts security audits and collaborates with cybersecurity experts to identify and address potential vulnerabilities

4. Integration with Vehicle Systems:

MBUX seamlessly integrates with various vehicle systems, including drive modes, climate control, and driver assistance features, providing a cohesive and unified driving experience.
The operating system serves as the backbone for Mercedes-Benz's vision of connected and autonomous driving, enabling advanced features such as semi-autonomous driving and predictive navigation.

Conclusion:
Mercedes-Benz's operating system represents a paradigm shift in automotive technology, providing a foundation for connected, intelligent, and user-centric vehicles. By prioritizing user experience, performance, reliability, and security, Mercedes-Benz has positioned itself as a leader in the digital transformation of the automotive industry.