DS Assignment

12.11.2020

Ratanshi Puri
RA1911030010025

CALL LOG MANAGEMENT SYSTEM
IMPLEMENTATION USING CIRCULAR QUEUE

Problem Statement:

Simulate the call logs in the mobile using an appropriate data structure. The Call Logs can store the telephone number, name, date and time of the 10 most recent callers. Once the limit of 10 is reached, and another call is made, the least recent number is deleted to make room for the most recent number.

Link :

https://repl.it/join/gbgirwkv-ratanshipuri

Code:

```
#include<iostream>
#include <stdlib.h>
//for printing the time
#include <time.h>
using namespace std;

//definition of queue
struct queue
{
long number;
char name[20];
char date[20];
char tm[30];
struct queue *next;
} *front, *back, *p;

//function to print the time of calling
void printtime()
{
 int hours, minutes, seconds, day, month, year;
  time_t now;
  time(&now);
```

```
  struct tm *local = localtime(&now);
  hours = local->tm_hour;
  minutes = local->tm_min;
  seconds = local->tm_sec;
  day = local->tm_mday;
  month = local->tm_mon + 1;
  year = local->tm_year + 1900;
 printf("\nDate    : %02d/%02d/%d                \n", day, month, year);
  if (hours < 12)
    printf("Time    : %02d:%02d:%02d am               \n", hours, minutes,
seconds);
  else
    printf("Time : %02d:%02d:%02d pm               \n", hours - 12,
minutes, seconds);
}

//function to display call logs
void display(struct queue *t)
{
 printtime();
 cout<<"Contact : "<<t->name<<"\n";
 cout<<"Number  : "<<t->number<<"\n";
}

//function for enqueue operation
void enqueue()
{
printf("Enter contact details to place a call\n(1)First Name:");
```

```
p = (struct queue *)malloc(sizeof(struct queue));
scanf("%s",p->name);
printf("(2)Contact Number:");
scanf("%li",&p->number);
p->next = NULL;
//circular queue implementation
if(front == NULL)
{
  front = p;
  back = p;
  back->next = front;
}
else
{
 p->next = front;
 back->next = p;
 back = p;
}
printf("\n     Dialing...");
display(back);
}

//function for dequeue operation
void dequeue()
{
struct queue *temp = front;
printf("\n      DELETING...");
```

```
display(temp);
if(front == back)
  front = back=NULL;
else
 {
  front = temp->next;
  back->next=front;
 }
}


int main()
{
printf("\n          WELCOME TO CALL LOG MANAGEMENT SYSTEM\n
_____________________________________________________\n\n");
int count = 1;
char ch;
do
{
  if(ch != '0' && count <=10)
  {
    enqueue();
    count++;
    printf("enter any number to make another call and 0 to stop\n");
  }
  else if(ch !=0)
  {
    printf("\nCall log full!! \n Removing first contact\n");
    dequeue();
```

```
    printf("\nYou may place your call now\n");
    enqueue();
    printf("Press any key to make another call and 0 to stop\n");
   }
   cin>>ch;
 }while(ch != '0');
printf("\n***Thank you for using CALL LOG MANAGER!***\n");
return 0;
}
```

Output:

main.cpp

```
#include<iostream>
#include <stdlib.h>
//for printing the time
#include <time.h>
//for formatting the output
#include<iomanip>
using namespace std;

//definition of queue
struct queue
{
long number;
char name[20];
char date[20];
char tm[30];
struct queue *next;
} *front, *back, *p;

//function to print the time of calling
void printtime()
{
  int hours, minutes, seconds, day, month, year;
  time_t now;
  time(&now);
  struct tm *local = localtime(&now);
  hours = local->tm_hour;
  minutes = local->tm_min;
  seconds = local->tm_sec;
  day = local->tm_mday;
  month = local->tm_mon + 1;
  year = local->tm_year + 1900;
  printf("\nDate    : %02d/%02d/%d                \n", day, month,
  year);
  if (hours < 12)
    printf("Time    : %02d:%02d:%02d am              \n", hours,
    minutes, seconds);
```

```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main

          WELCOME TO CALL LOG MANAGEMENT SYSTEM
_____________________________________________________________

Enter contact details to place a call
(1)First Name:Ratanshi
(2)Contact Number:9816583669

     Dialing...
Date    : 12/11/2020
Time    : 08:05:27 am
Contact : Ratanshi
Number  : 9816583669
enter any number to make another call and 0 to stop
1
Enter contact details to place a call
(1)First Name:abcd
(2)Contact Number:123456

     Dialing...
Date    : 12/11/2020
Time    : 08:05:44 am
Contact : abcd
Number  : 123456
enter any number to make another call and 0 to stop
1
Enter contact details to place a call
(1)First Name:defghi
(2)Contact Number:78934

     Dialing...
Date    : 12/11/2020
Time    : 08:06:10 am
Contact : defghi
Number  : 78934
enter any number to make another call and 0 to stop
```

Documentation:

HEADER FILES:

```
#include<iostream>
#include <stdlib.h>
//for printing the time
#include <time.h>
using namespace std;
```

To minimize the potential for **errors**, C++ has adopted the convention of using header files to contain declarations.

- #include<iostream> : Used for basic Input output operations like Cin and Cout.
- #include<stdlib> : Defines several general purpose functions, including dynamic memory management, random number generation, communication with the environment, integer arithmetics, searching, sorting and converting.
- #include<time.h> : The time.h header defines four variable types, two macro and various functions for manipulating date and time.

QUEUE ADT's:

```
struct queue
{
long number;
char name[20];
char date[20];
char tm[30];
struct queue *next;
} *front, *back, *p;
```

This is the definition of the queue structure which forms a common datatype of long number, char name , date and time.

```
void enqueue()
```

Adds the next queue structure to a circular queue whose implementation has been shown in the code below:

```
if(front == NULL)
 {
   front = p;
   back = p;
   back->next = front;
 }
 else
 {
  p->next = front;
  back->next = p;
  back = p;
 }
}
```

```
void dequeue()
{
 struct queue *temp = front;
 if(front == back)
   front = back=NULL;
 else
  {
   front = temp->next;
   back->next=front;
  }
}
```

Dequeue is used to remove the first caller ID in the call log after 10 calls have been made to make the programme more space efficient.

```
void display(struct queue *t)
{
  printtime();
  cout<<"Contact : "<<t->name<<"\n";
  cout<<"Number  : "<<t->number<<"\n";
}
```

Used to display the call logs in terms of date , time, name and contact number.

```
void printtime()
```

Used to display date and time.

```
int main()
```

Includes the function calls and print statements for programme output.

The call log management has hence been implemented to dial a number by taking in input of name and contact number and storing 10 dialled numbers in the call log. On dialing the 11th number, the first number in the queue gets deleted. Similarly for the 12th number the 2nd number gets deleted and this process continues till the user inputs 0 to stop the input .