Week-5 UE20CS207 DSLAB

- Name : P K Navin Shrinivas
- SRN : PES2UG20CS237
- Section : D
- Batch : 2

Lab assignments 1 : Lawyer problem

Code :

main.c

```
#include "2_1.h"
#include <stdio.h>
/*
 *
 *    int slot_no;
 *    int case_no;
 *    char client_nane [1000];
 *    char case_name [1000];
 *    char timec [1000];
 *    ll time; //HHMM
 *
 */
int main()
{
    //considering head of linked list to be front
    struct head* queue=(struct head*)malloc(sizeof(struct head));
    queue->size=-1;
    while(true)
    {
        printf("1.Add appointment\n");
        printf("2.Delete appointment\n");
        printf("3.Display appointments\n");
        printf("4.Exit\n");
        printf("Enter choice : ");
        int choice;
        scanf("%d",&choice);
        if(choice == 5)
            return 0;
        else if(choice == 1)
        {
            struct data_carry input;
            printf("1 : 8-10 | 2: 10-12 | 3 : 1-3 | 4 : 3-5 | 5 : 5-7 | 6 : 7:9 \n");
            printf("Enter slot to appoint :  ");
            scanf("%d",&(input.slot_no));
            printf("Enter case number : ");
            scanf("%d",&(input.case_no));
            printf("Enter client name : ");
            scanf("%s",(input).client_name);
            printf("Enter case name :");
            scanf("\t%[^\n]%*c",(input).case_name);
            printf("Enter time [HH:MM] : ");
            scanf("%[^\n]%*c",input.timec);
            input.time = *((input.timec)+0)-'0'+*((input.timec)+1)-'0'+*
((input.timec)+3)-'0'+*((input.timec)+4)-'0';
            if(input.slot_no > 6){
                printf("Inavlid slot , please enter a valid slot \n");
            }
            int ret = queuepush(queue ,&input);
            if(ret == -1)
                printf("Oops!Appointment already exists on the requested time :/ \n");
            else if(ret == 1)
                printf("Appointment created :) \n");
        }
        /*
         *else if(choice == 2)
         *{
         *    queuepop(queue);
         *}
         *else if(choice == 3)
         *    queuepeek(queue);
         *else if(choice == 4)
         *    queuedisplay(queue);
         */
    }

}
```

1.h

```

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<math.h>
#include<stdbool.h>
typedef long long int ll;



struct node{
    int slot_no;
    int case_no;
    char* client_name;
    char* case_name;
    char* timec;
    ll time; //HHMM , lol easy method to cmpr times
    struct node* link;
};
struct head{
    struct node* link;
    int size;
};
struct data_carry{
    int slot_no;
    int case_no;
    char* client_name;
    char* case_name;
    char* timec;
    ll time;
};
int queuepush(struct head* queue,struct data_carry* data);
void queuedisplay(struct head* queue);
```

1.c

```
int queuepush(struct head* queue,struct data_carry* data)
{
    if(queue->size >= 8){
        printf("No more Appointsment slots left :(.\n");
        return 0;
    }
    if(queue->size==-1)
    {
        struct node* temp = (struct node*)malloc(sizeof(struct node));
        temp->time=data->time;
        temp->link=NULL;
        temp->slot_no=data->slot_no;
        temp->case_no=data->case_no;
        temp->case_name=data->case_name;
        temp->client_name=data->client_name;
        queue->link=temp;
        (queue->size)++;
        return 1;

    }
    else{
        struct node* prev = queue->link;
        struct node* curr = prev->link;
        while(curr->time <= data->time || curr!=NULL){
            if(curr->slot_no == data->slot_no){
                return -1;
            }
            prev=curr;
            curr = curr->link;
        }
        /*
         *int slot_no;
         *int case_no;
         *char client_name [1000];
         *char case_name [1000];
         *char time [1000];
         *ll time; //HHMM , lol easy method to cmpr times
         *struct node* link;
         */
        struct node* temp2 = prev->link;
        struct node* temp = (struct node*)malloc(sizeof(struct node));
        temp->time=data->time;
        temp->link=temp2;
        prev->link = temp;
        temp->slot_no=data->slot_no;
        temp->case_no=data->case_no;
        temp->case_name=data->case_name;
        temp->client_name=data->client_name;
        (queue->size)++;
        return 1;
    }
}


 void queuedisplay(struct head* queue)
 {
     printf("Start of queue->");
     struct node* temp = queue->link;
     while(temp->link!=NULL)
     {
         printf("%d \n %d \n %s \n %s \n %s ->",temp->slot_no , temp->case_no , temp->client_name , temp->case_name , temp->timec);
         temp=temp->link;
     }
    printf("%d \n %d \n %s \n %s \n %s ->",temp->slot_no , temp->case_no , temp->client_name , temp->case_name , temp->timec);
     printf("End of queue\n");
 }```
```