SCHEDULE

MANAGER

UTILITY

**PROBLEM STATEMENT**

Schedule Manager Utility is aimed to be a utility that can automate personal daily schedule management.This utility shall allows users to see their daily schedule.Once schedules have been set up by user on daily basis,the utility shall functions as a reminder to user daily routines .

Everyday , we have many chores to do ,doing homework,bussiness work,meetings,birthday,parties,phone calls,billing,TV programs and so on.It is not easy to memorize all and handle all well,in order not to forget or miss any important things we can ask help from Schedule Manager Utility.It is simple software that can schedule your daily plans ,reminder with beep sound to ensure you can’t miss an important appointment or deadline

Additionally the utility can also be made to function as a alarm clock under desired situations

**SUBMODULES**

**1.Enter Schedule**

In this module user set their daily schedule in 24 hour format.

**2.View Schedule**

In this module user can view their schedule which they had already entered.

**3.Scheduler**

It is reminder with beep sound to ensure you can’t miss an important appointment or deadline.

It function as a alarm clock under desired situations.

Extra functionalities like Dismiss alarm and Snooze alarm is also present.

**CODE**

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<time.h>

#include<windows.h>

#include<dos.h>

struct alarm

{

int n;

int hour[50];

int min[50];

char task\_name[51][50];

}A;

time\_t now;

struct tm \*right\_now;

int main ()

{

int ch,no;

time\_t s;

struct tm\* current\_time;

// time in seconds

s = time(NULL);

current\_time = localtime(&s);

printf("%s\n",asctime(current\_time));

printf(" SCHEDULE MANAGEMENT UTILITY\n\n");

do

{

printf("\n1.ENTER YOUR TODAY'S SCHEDULE\n");

printf("2.VIEW YOUR SCHEDULE\n");

printf("3.ALARM\n");

printf("4.EXIT\n\n");

printf("Enter your choice(1-4)");

scanf("%d",&ch);

switch(ch)

{

case 1:no=schedule();

break;

case 2:view\_schedule();

break;

case 3:alarm();

break;

case 4:exit(1);

break;

default:printf("WRONG CHOICE!!\nPlease enter valid choice(1-4)");

}

}while(ch!=4);

return 0;

}

int schedule()

{ system("cls");

int i,n;

FILE \*fout;

fout = fopen("schedule.txt","wb+");

if(!fout)

{

printf("File not found\n");

exit(1);

}

printf(" WELCOME!!Enter your schedule here...\n\n\n");

printf("Enter number of tasks:\n");

fflush(stdin);

scanf("%d",&A.n);

for(i=0;i<A.n;i++)

{

printf("\n\nTASK%d\n\n",(i+1));

printf("Enter time in sorted way\n");

printf("Enter Hour(IN 24 hour FORMAT):\n");

fflush(stdin);

scanf("%d",&A.hour[i]);

printf("Enter min:\n");

fflush(stdin);

scanf("%d",&A.min[i]);

printf("Enter the task name:\n");

fflush(stdin);

scanf("%s",&A.task\_name[i][0]);

}

fwrite(&A,sizeof(struct alarm),A.n,fout);

printf("\nData is stored successfully...\n");

fclose(fout);

return A.n;

}

int view\_schedule()

{ system("cls");

int i;

FILE \*fin;

fin = fopen("schedule.txt","rb");

if(!fin)

{

fprintf(stderr, "\nError opening file\n");

exit (1);

}

// read file contents till end of file

fread(&A, sizeof(struct alarm), 1, fin);

printf(" TODAY'S SCHEDULE IS:\n\n");

for(i=0;i<A.n;i++)

{

printf("Hour:%d\nMin:%d\nTask\_Name:%s\n\n",A.hour[i],A.min[i],A.task\_name[i]);

}

fclose(fin);

return 0;

}

int alarm()

{ system("cls");

int i,time1,time2,calc;

time\_t t,s,val = 1;

int c,count=0;

FILE \*fin;

fin = fopen("schedule.txt","rb");

if(!fin)

{

fprintf(stderr, "\nError opening file\n");

exit (1);

}

// read file contents till end of file

fread(&A, sizeof(struct alarm), 1, fin);

while(A.n)

{

time(&now);

right\_now=localtime(&now);

for(i=0;i<A.n;i++)

{

time1=((A.hour[i]\*3600)+(A.min[i]\*60));

time2=((A.hour[i+1]\*3600)+(A.min[i+1]\*60));

calc=(time2-time1);

if(A.hour[i]==right\_now->tm\_hour & A.min[i]==right\_now->tm\_min)

{

s=time(NULL);

right\_now=localtime(&s);

printf("%02d:%02d",right\_now->tm\_hour,right\_now->tm\_min);

printf("\t%s\n",A.task\_name[i]);

Beep(37500,2000);

printf("\nEnter 1 to Dismiss\tEnter 2 to snooze\tEnter 3 Stop all alarms\n");

scanf("%d",&c);

switch(c)

{

case 1:

if(i==(A.n-1))

{

printf("\nYOUR SCHEDULE ENDS!! \n ThankYou\n\n");

//fclose(fin);

exit(0);

}

printf("\nUPCOMING ALARM %d:%d\n",A.hour[i+1],A.min[i+1]);

printf("Waiting for other Alarm\n");

sleep(calc-5);

//Beep(37500,2000);

break;

case 2:

printf("\nAlarm Snoozed\nAlarm will sound after 5 seconds\n");

sleep(5);

Beep(37500,2000);

if(i==(A.n-1))

{

printf("YOUR SCHEDULE ENDS!! \n ThankYou\n\n");

//fclose(fin);

exit(0);

}

printf("\nUPCOMING ALARM %d:%d\n",A.hour[i+1],A.min[i+1]);

printf("Waiting for other Alarm\n");

sleep(calc-5-5);

Beep(37500,2000);

break;

case 3:return;

default:printf("Select Valid Option\n");

break;

}

}

else

{

free(right\_now);

}

}

}

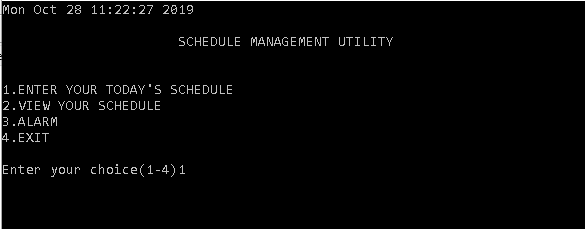
//fclose(fin);

//exit(0);

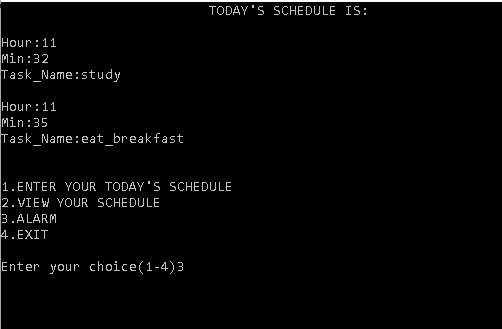
return 0;

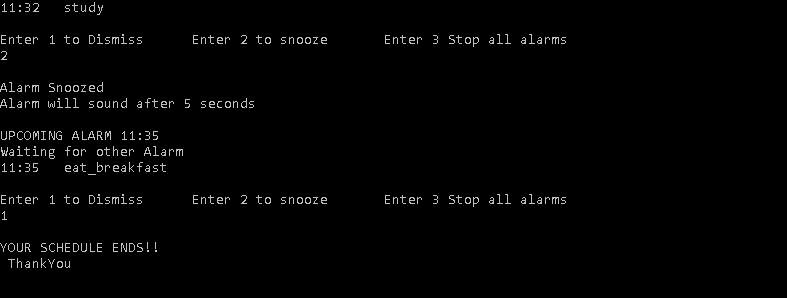
}

**TESTING SCREENSHOTS**









**CONCLUSION/ FUTURE SCOPE**

Schedule Manager Utility function as a reminder to user daily schedule. Additionally the utility can be made to function as an alarm clock under desirable situation.

**Future Scope** is that it would automate the recurring events and tasks like Schedule a phone call,text message,email,social media update for future date/time and simply rid of organizers and diaries

Moreover,all activities would synced with your calender to avoid redundant activities .It would be like having your own private secretary to schedule future updates in a simple yet efficient manner