MINI PROJECT
Conact Managment System

Nikhil Dwivedi
0801CS223D03
B.tech 2nd Year, SGSITS INDORE

December 9, 2022

---

1. **AIM -**
   To create a Simple Contact Management System to Store Contacts , Modify Contacts and Display list of all contacts

2. **FOLLOWING ARE THE FUNCTION -**

2.1. **LIBRARY: stdio.h -**
allows to perform input/output function (ie,printf(); , scanf(); , flush(); , fclose();)

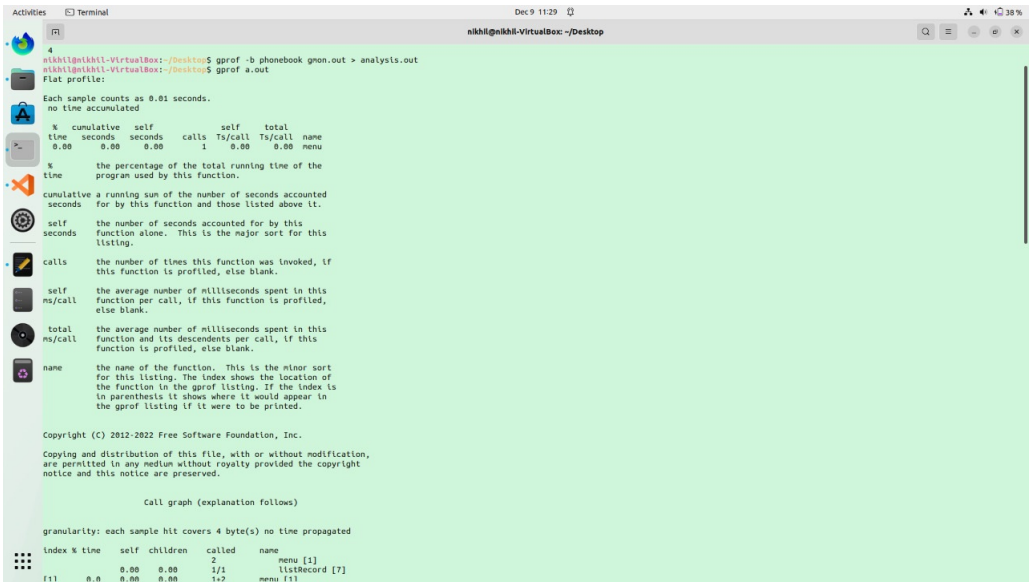
2.2. **LIBRARY: stdlib.h -**
information of memory allocation/freeing functions (ie,exit();)

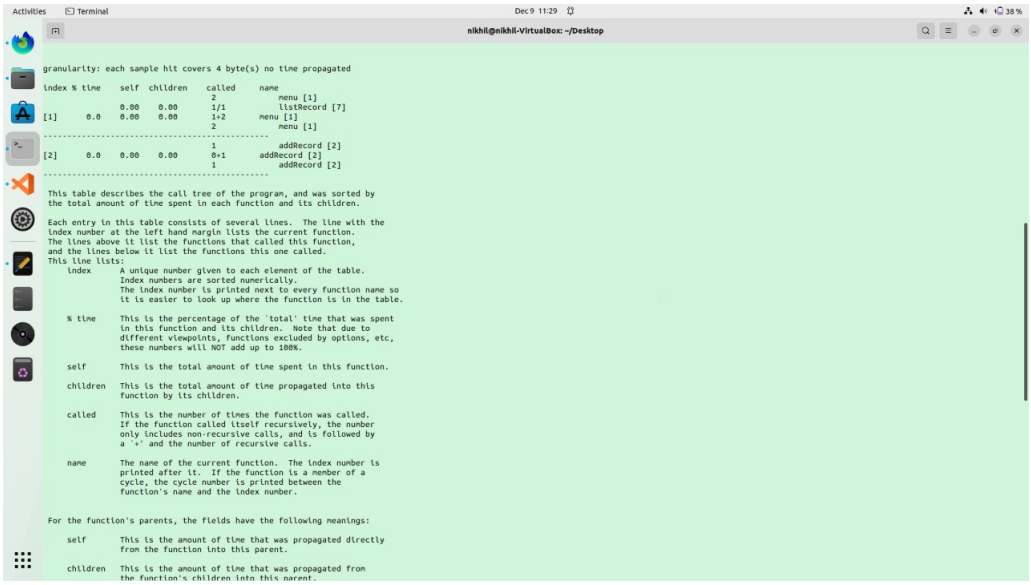
2.3. **LIBRARY: string.h -**
defines one variable type, one macro, and various functions for manipulating arrays of characters.(ie,Null;)

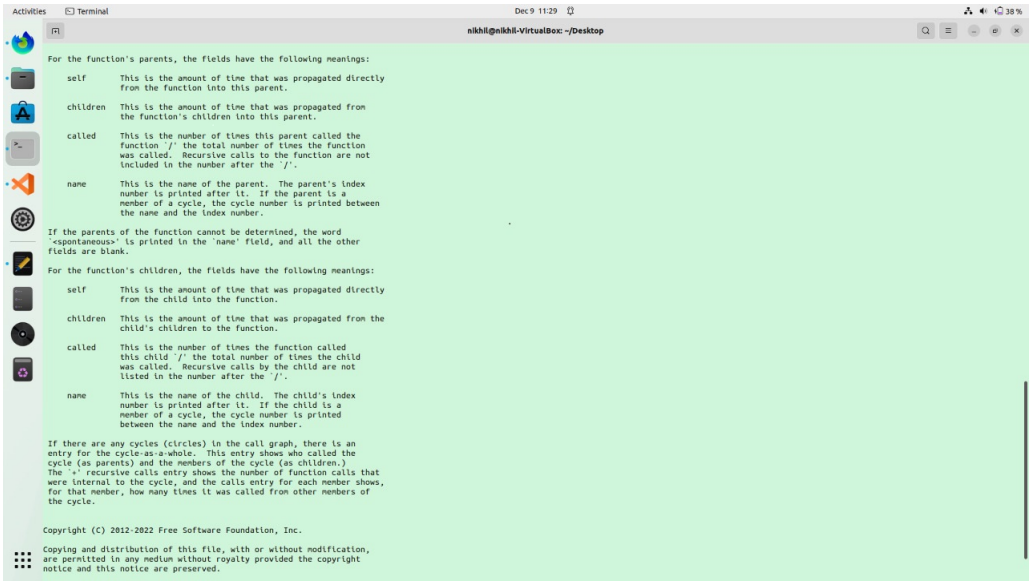
2.3. **USER DEFINED FUNCTIONS -**
void menu(); , void addRecord(); , void listRecord(); , modifyRecord(); void back();

3. **DESCRIPTION: -**
   This mini project prgram is to provide easy to use Contact Management interface Consisting of tasks like add Contact, List Contact,Modify Saved Contacts

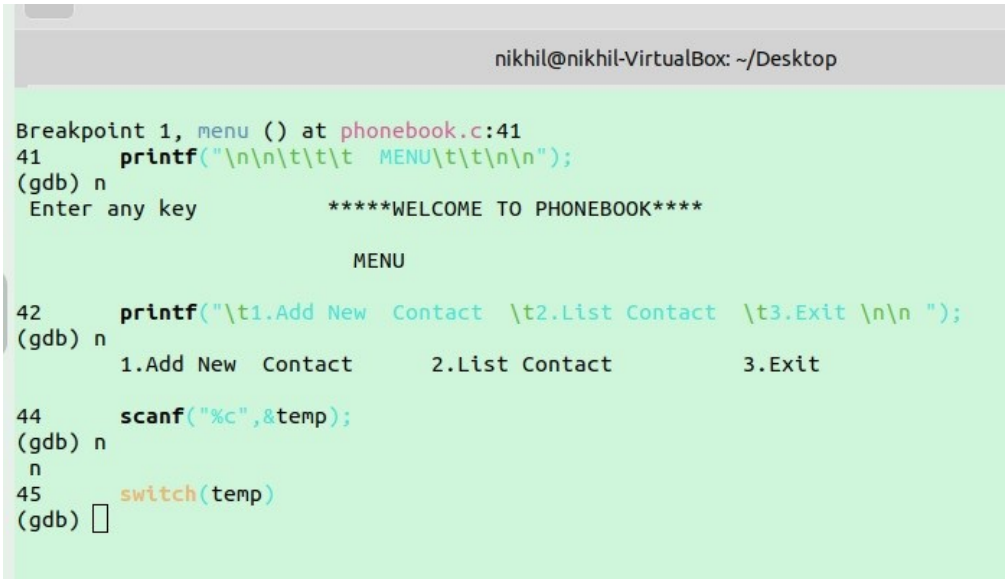
4. **Profiling -**


Activities
Terminal
Dec 9 11:29
nikhil@nikhil-VirtualBox: ~/Desktop
4
nikhil@nikhil-VirtualBox:~/Desktop$ gprof -b phonebook gmon.out > analysis.out
nikhil@nikhil-VirtualBox:~/Desktop$ gprof a.out
Flat profile:
Each sample counts as 0.01 seconds.
no time accumulated
% cumulative self self total
time seconds seconds calls Ts/call Ts/call name
0.00 0.00 0.00 1 0.00 0.00 menu
% the percentage of the total running time of the
time program used by this function.
cumulative a running sum of the number of seconds accounted
seconds for by this function and those listed above it.
self the number of seconds accounted for by this
seconds function alone. This is the major sort for this
listing.
calls the number of times this function was invoked, if
this function is profiled, else blank.
self the average number of milliseconds spent in this
ms/call function per call, if this function is profiled,
else blank.
total the average number of milliseconds spent in this
ms/call function and its descendents per call, if this
function is profiled, else blank.
name the name of the function. This is the minor sort
for this listing. The index shows the location of
the function in the gprof listing. If the index is
in parenthesis it shows where it would appear in
the gprof listing if it were to be printed.
Copyright (C) 2012-2022 Free Software Foundation, Inc.
Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.
Call graph (explanation follows)
granularity: each sample hit covers 4 byte(s) no time propagated
index % time self children called name
menu [1]
0.00 0.00 1/1 llstRecord [7]
[1] 0.0 0.00 0.00 1+2 menu [1]


Activities
Terminal
Dec 9 11:29
nikhil@nikhil-VirtualBox: ~/Desktop
granularity: each sample hit covers 4 byte(s) no time propagated
index % time self children called name
2 menu [1]
0.00 0.00 1/1 llstRecord [7]
[1] 0.0 0.00 0.00 1+2 menu [1]
2 menu [1]
1 addRecord [2]
[2] 0.0 0.00 0.00 0+1 addRecord [2]
1 addRecord [2]
This table describes the call tree of the program, and was sorted by
the total amount of time spent in each function and its children.
Each entry in this table consists of several lines. The line with the
index number at the left hand margin lists the current function.
The lines above it list the functions that called this function,
and the lines below it list the functions this one called.
This line lists:
index A unique number given to each element of the table.
Index numbers are sorted numerically.
The index number is printed next to every function name so
it is easier to look up where the function is in the table.
% time This is the percentage of the `total' time that was spent
in this function and its children. Note that due to
different viewpoints, functions excluded by options, etc,
these numbers will NOT add up to 100%.
self This is the total amount of time spent in this function.
children This is the total amount of time propagated into this
function by its children.
called This is the number of times the function was called.
If the function called itself recursively, the number
only includes non-recursive calls, and is followed by
a `+' and the number of recursive calls.
name The name of the current function. The index number is
printed after it. If the function is a member of a
cycle, the cycle number is printed between the
function's name and the index number.
For the function's parents, the fields have the following meanings:
self This is the amount of time that was propagated directly
from the function into this parent.
children This is the amount of time that was propagated from
the function's children into this parent.



5. GDB DEBUGGING -



```
   84 |         fclose(f");
      |                ^
phonebook.c:84:17: error: missing terminating " character
   84 |         fclose(f");
      |                ^~~
phonebook.c:84:17: error: expected ‘)’ before ‘printf’
   84 |         fclose(f");
      |               ~ ^
      |                 )
   85 |
   86 |         printf("\n\nEnter any key for exit");
      |         ~~~~~~
phonebook.c:91:17: error: expected ‘;’ before ‘}’ token
   91 |          menu();
      |                ^
      |                ;
   92 | }
      | ~
nikhil@nikhil-VirtualBox:~/Desktop$ gdb ./a.out
GNU gdb (Ubuntu 12.0.90-0ubuntu1) 12.0.90
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./a.out...
(gdb) break 14
Breakpoint 1 at 0x1291: file phonebook.c, line 25.
(gdb) run
Starting program: /home/nikhil/Desktop/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at phonebook.c:25
warning: Source file is more recent than executable.
25          start();
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/nikhil/Desktop/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at phonebook.c:25
25          start();
(gdb)
```

Q3. **Source Code -**

```c
//allows input\output function (ie,printf(); , scanf(); , flush(); , fclose();)
#include<stdio.h>
//defines one variable type, one macro, and various functions for manipulating arrays of characters.(
#include<string.h>
//information of memory allocation/freeing functions (ie,exit();)/
#include<stdlib.h>

//creating structure variables called person for general details
struct person
{
    char        name[35];
    char        address[50];
    long int    mobileNo;
    char        mail[100];
};

void menu();
void got();
void start();
void back();
void addRecord();
void listRecord();
void modifyRecord();
int main()        //starts the start function
{
    start();
    return 0;
}
void back()         //back to the menu
{
    start();
}
void start()        // starts the menu function
{
    menu();
}
void menu()          //menu Screen of phonebook
{
```

```
39      system("clear");      // clear the screen
40      printf("\t\t**WELCOME TO CONTACT MANAGEMENT SYSTEM*");
41
42      printf("\n\n\t\t\t  MENU\t\t\n\n");
43      printf("\t1.Add New  Contact  \t2.List Contact \t3.Modify Record \t4.Exit \n\n "
44      char temp;
45      scanf("%c",&temp); //choice for option
46      switch(temp)
47      {
48      case '1':
49      addRecord();
50      break;
51      case '2':
52      listRecord();
53      break;
54      case '3':
55      modifyRecord();
56      break;
57      case '4':
58      exit(0);
59      break;
60      default:
61              system("clear");
62              printf("\nEnter 1 to 4 only");
63              printf("\n Enter any key");
64              scanf("%c",&temp);
65              menu();
66  }
67  }
68  //function to add the records in phonebook
69  void addRecord()
70  {
71      system("clear");
72      FILE *f;        // Declare the file pointer
73      struct person p;        // Create a structure variable of person called p
74      f=fopen("project","ab+");
75      printf("\n\n Enter Details to Save Record \n\n\n ");
76      printf("\n Enter name: ");
```

```
77          scanf("%s",p.name);
78          printf("\nEnter the address: ");
79          scanf("%s",p.address);
80          printf("\nEnter phone no.:");
81          scanf("%ld",&p.mobileNo);
82          printf("\nEnter e-mail:");
83          scanf("%s",p.mail);
84          fwrite(&p,sizeof(p),1,f);
85          fflush(stdin);
86          printf("\nrecord saved");
87          fclose(f);
88          printf("\n\nEnter any key for exit");
89       //getch();
90       char temp;
91       scanf("%c",&temp);
92           system("clear");
93           menu();
94   }
95   //function to list the records
96   void listRecord()
97   {
98       struct person p;
99       FILE *f;     // Declare the file pointer
100      f=fopen("project","rb");        // Open the existing file pho
101      if(f==NULL)
102      {
103      printf("\nfile opening error in listing :");
104
105      exit(1);
106      }
107      char temp;
108      while(fread(&p,sizeof(p),1,f)==1)
109      {
110      printf("\n\n\n YOUR Saved RECORD IS\n\n ");
111      printf("\nName=%s\nAdress=%s\nMobile no=%ld\nE-mail=%s",p.nam
112        scanf("%c",&temp);
113        system("clear");
114      }
```

```
115     fclose(f);      // Closing the file using fclose()
116     printf("\n Enter any key for exit ");
117     scanf("%c",&temp);
118     system("clear");
119     menu();
120 }
121
122 void modifyRecord()
123 {
124     int      c;
125     char     tempr;
126     FILE *f;
127     int      flag=0;
128     struct   person p,s;
129     char     name[50];
130     f=fopen("project","rb+");
131     if(f==NULL)
132         {
133
134             printf("CONTACT'S DATA NOT ADDED YET.");
135             exit(1);
136
137
138         }
139     else
140     {
141         system("clear");
142             printf("\nEnter CONTACT'S NAME TO MODIFY:\n");
143             scanf("%s",name);
144             while(fread(&p,sizeof(p),1,f)==1)
145             {
146                 if(strcmp(name,p.name)==0)
147                 {
148
149                     printf("\n Enter name:");
150                     scanf("%s",s.name);
151                     printf("\nEnter the address:");
152                     scanf("%s",s.address);
```

```
153                     printf("\nEnter phone no:");
154                     scanf("%ld",&s.mobileNo);
155                     printf("\nEnter e-mail:");
156                     scanf("%s",s.mail);
157                     fseek(f,-sizeof(p),SEEK_CUR);
158                     fwrite(&s,sizeof(p),1,f);
159                     flag=1;
160                     break;
161 
162                 }
163                 fflush(stdin);
164 
165 
166             }
167             if(flag==1)
168             {
169                 printf("\n your data id modified");
170 
171             }
172             else
173             {
174                     printf(" \n data is not found");
175 
176             }
177             fclose(f);
178         }
179          printf("\n Enter any key");
180          scanf("%c",&tempr);
181         system("clear");
182         menu();
183 
184 }
```

Q1. **Outputs -**

```
               ***WELCOME TO CONTACT MANAGEMENT SYSTEM**

                        MENU

       1.Add New  Contact      2.List Contact  3.Modify Record        4.Exit
```

```
 YOUR Saved RECORD IS


Name=Rishu
Adress=Bihar
Mobile no=8871436225
E-mail=rishu@123
```

```
 Enter Details to Save Record


 Enter name: Rishu

Enter the address: Bihar

Enter phone no.:8871436225

Enter e-mail:rishu@123
```

```
 YOUR RECORD IS


Name=Rashi
Adress=Panna
Mobile no=7477230781
E-mail=rashi@agn
```