

Министерство цифрового развития, связи
и массовых коммуникаций Российской Федерации

Сибирский государственный университет
телекоммуникаций и информатики

Кафедра прикладной математики и кибернетики

ЛАБОРАТОРНАЯ РАБОТА №2

По дисциплине: «Операционные системы»

Выполнили:

Студенты 3 курса группы ИП-111
Корнилов А.А.,
Попов М.И.,
Толкач И.А.

Проверил:

Профессор кафедры ПМиК
Малков Е.А.

Новосибирск, 2023

Цель работы:

Знакомство с инструментарием для разработки программ на платформе Linux.

Задание:

1. Написать программу для манипуляции данными на основе рекурсивных структур, реализовать функции вставки, удаления и навигации для списка, реализовать сериализацию списка;
2. Провести трассировку программы с помощью GDB.

Инструментарий:

Редактор кода VIM, компилятор GCC, отладчик GDB.

Выполнение работы:

1. В качестве программы в редакторе VIM была написана программа чтения и записи в файл базы данных (Рис. 1.1), сортировки БД, построение АВЛ дерева и поиска по ключу.

```
202 int menu() {
203     int choice;
204     cout << "1) Drop source database\n";
205     cout << "2) Drop sorted database\n";
206     cout << "3) Binary search in database\n";
207     cout << "4) Make AVL-Tree\n";
208     cout << "5) Binary search in tree\n";
209     cout << "6) Encoding using the Gilbert-Moore method \n";
210     cout << "0) Exit\n";
211     cin >> choice;
212     system("cls");
213     return choice;
214 }
215 int main(){
216     FILE* fin = fopen("testBase3.dat", "rb");
217     List* head = new List;
218     List* p = head;
219     Elem* ind_arr[N];
220     unsigned int size = 0;
221     int choice = menu(), z = 0, L = 4;
222     short int count = 1;
223     bool b;
224     for (int i = 0; i < N; i++)
225     {
226         p->data = new Elem;
227         z = fread((Elem*)p->data, sizeof(Elem), 1, fin);
228         if (i != N - 1)
229         {
230             p->next = new List;
231             p = p->next;
232         }
233         else p->next = NULL;
234     }
235     p = head;
236
237     switch (choice){
238     case (1): {
239         system("clear");
240         for (int i = 0; i < N / 20 + 1; i++) {
241             p = printList(head, count, 1);
242             cout << "Next 20? 1/0: ";
243             cin >> b;
244             if (b == false) {
245                 break;
246             }
247             choice = 1;
248         }
249     }
250     case (2): {
251         system("clear");
```

Рисунок 1.1 Фрагмент программы, открытой в редакторе VIM

2. Компилируем программу с параметром -g (Рис. 2.1), для последующего включения дебаггинга GDB и запускаем ее.

```

miron@DESKTOP-UMC1Q46:/mnt/u/Documents/B ВУЗ/САОД/Курсовая/Курсач/Курсач/1$ g++ -g -o curs curs.cpp
curs.cpp: In function 'int PrintSort(List*, short int&)':
curs.cpp:129:1: warning: control reaches end of non-void function [-Wreturn-type]
  129 | }
      | ^
curs.cpp: In function 'int B2Search(Elem**, queue*)':
curs.cpp:182:1: warning: control reaches end of non-void function [-Wreturn-type]
  182 | }
      | ^
miron@DESKTOP-UMC1Q46:/mnt/u/Documents/B ВУЗ/САОД/Курсовая/Курсач/Курсач/1$ ls -l
total 1020
-rwxrwxrwx 1 miron miron 683256 Sep 17 21:29 LR2.docx
-rwxrwxrwx 1 miron miron 92544 Sep 17 21:58 curs
-rwxrwxrwx 1 miron miron 6474 Sep 17 21:55 curs.cpp
-rwxrwxrwx 1 miron miron 256000 Mar 17 2010 testBase3.dat
-rwxrwxrwx 1 miron miron 162 Sep 17 21:30 '~$LR2.docx'

```

Рисунок 2.1 – Компиляция программы компилятором g++

```

miron@DESKTOP-UMC1Q46:/mnt/u/Documents/B ВУЗ/САОД/Курсовая/Курсач/Курсач/1$ ./curs
1) Drop source database
2) Drop sorted database
3) Binary search in database
4) Make AVL-Tree
5) Binary search in tree
6) Encoding using the Gilbert-Moore method
0) Exit

```

1) Власов Патрик Ромуальдович	15000	23-02-96	Патриков Ж В
2) Хасанов Муамар Ахиллесович	15000	18-07-96	Янов О С
3) Власов Поликарп Батырович	40000	21-10-93	Ахиллесов Н О
4) Поликарпов Герасим Власович	30000	08-01-94	Глебов Н Е
5) Ахмедов Влас Никодимович	50000	17-03-95	Патриков Ж К
6) Зосимов Ахмед Евграфович	10000	08-09-95	Батырова О Й
7) Герасимов Герасим Янович	10000	22-07-93	Глебов Н Е
8) Евграфов Остап Феофанович	30000	14-01-96	Батырова О Й
9) Ахиллесов Филимон Демьянович	35000	21-10-94	Батырова О Й
10) Янов Ахиллес Герасимович	25000	17-09-94	Патриков Ж К
11) Патриков Клим Янович	15000	08-10-97	Глебов У Л
12) Власов Влас Архипович	25000	23-02-96	Глебов Н Е
13) Феофанова Изольда Евграфовна	35000	23-02-93	Янов И Ж
14) Тихонов Филимон Демьянович	50000	26-02-94	Архипов П В
15) Герасимов Остап Власович	40000	26-12-94	Патриков Ж К
16) Глебов Жак Остапович	40000	24-11-95	Архипов П В
17) Никодимо Поликарп Никодимович	45000	17-09-93	Глебов У Л
18) Герасимова Степанида Яновна	25000	07-04-94	Янов О С
19) Пантелемоно Евграф Гедеонович	25000	25-03-95	Патриков Ж В
20) Евграфо Пантелемон Патрикович	45000	22-07-96	Янов И Ж

```

Next 20? 1/0: |

```

Рисунки 2.2 и 2.3 – Запуск программы и вывод данных из файла

3. Используя GDB, проведем трассировку программы

```
miro@DESKTOP-UMC1Q46:/mnt/u/Documents/B ВУЗ/CAОД/Курсовая/Курсач/Курсач/1$ gdb curs
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
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 curs...
(gdb) run
Starting program: /mnt/u/Documents/B ВУЗ/CAОД/Курсовая/Курсач/Курсач/1/curs
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
1) Drop source database
2) Drop sorted database
3) Binary search in database
4) Make AVL-Tree
5) Binary search in tree
6) Encoding using the Gilbert-Moore method
0) Exit
|
```

Рисунок 3.1 – Запуск трассировки через GDB

```
(gdb) break 223
Breakpoint 1 at 0x36d8: file curs.cpp, line 224.
(gdb) run
Starting program: /mnt/u/Documents/B ВУЗ/CAОД/Курсовая/Курсач/Курсач/1/curs
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
1) Drop source database
2) Drop sorted database
3) Binary search in database
4) Make AVL-Tree
5) Binary search in tree
6) Encoding using the Gilbert-Moore method
0) Exit
1
[Detaching after vfork from child process 946]
sh: 1: cls: not found

Breakpoint 1, main () at curs.cpp:224
224      for (int i = 0; i < N; i++)
```

Рисунок 3.2 – Создание брейк-поинта для последующей трассировки

```

(gdb) i frame
Stack level 0, frame at 0x7fffffffdded0:
rip = 0x555555576d8 in main (kurs.cpp:224); saved rip = 0x7ffff7b6ad90
source language c++.
Arglist at 0x7fffffffdec0, args:
Locals at 0x7fffffffdec0, Previous frame's sp is 0x7fffffffdded0
Saved registers:
rbx at 0x7fffffffdeb8, rbp at 0x7fffffffdec0, rip at 0x7fffffffdec8
(gdb) i locals
i = 0
fin = 0x55555556ceb0
head = 0x55555556d090
p = 0x55555556d090
ind_arr = {0x0 <repeats 3311 times>, 0x7ffff7fccda5, 0x0, 0x28000, 0x27fe0, 0x1000, 0x0,
0x1, 0x28000, 0x1bd000, 0x1bc4c1, 0x1bc4c1, 0x1000, 0x28000, 0x5, 0x1bd000, 0x215000, 0x2148cc,
0x2148cc, 0x1000, 0x1bd000, 0x1, 0x215000, 0x21b000, 0x21a888, 0x227e50, 0x1000, 0x214000, 0x3,
0x0 <repeats 65 times>, 0x7ffff7fd090e, 0x7ffff7fd0940 <_rtld_global>, 0x7ffff7fd0bb0,
0x7ffff7b41040, 0x7ffff7fccb0e, 0x4000000006, 0x40, 0x40, 0x40, 0x310, 0x310, 0x8, 0x4000000003,
0x1e3e30, 0x1e3e30, 0x1e3e30, 0x1c, 0x1c, 0x10, 0x4000000001, 0x0, 0x0, 0x7ffff7fcc02c, 0x0,
0xe000, 0xd2e0, 0xd2e0, 0x1000, 0x0, 0x1, 0xe000, 0x8a000, 0x898f5, 0x898f5, 0x1000, 0xe000, 0x5,
0x8a000, 0xe5000, 0xe47f4, 0xe47f4, 0x1000, 0x8a000, 0x1, 0xe5000, 0xe7000, 0xe60fc, 0xe6108,
0x1000, 0xe4000, 0x3, 0x1d0, 0x1d0, 0x8, 0x4000000004, 0x350, 0x350, 0x350, 0x30, 0x30, 0x8,
0x4000000004, 0x380, 0x380, 0x380, 0x44, 0x44, 0x4, 0x4000000007, 0x2148f0, 0x2158f0, 0x2158f0,
0x10, 0x90, 0x8, 0x46474e553, 0x350, 0x350, 0x350, 0x30, 0x30, 0x8, 0x46474e550, 0x1e3e4c, 0x0,
0x7ffff7fbc080, 0x7ffff7fd0c1e, 0x70cc, 0x1f, 0x20, 0xa, 0x7ffff7fbc528, 0x0, 0x0, 0x0,
0x7ffff7fd090e, 0x7ffff7fd090e, 0x7ffff7fd0940 <_rtld_global>, 0x7ffff7fd090e, 0x7ffff7a5a040,
0x7ffff7fccb0e, 0x3710, 0x7ffff7fd178, 0x1000, 0x7ffff7fc0003, 0xffffffffff998, 0xffffffffffcf30,
0x3, 0x0, 0x0, 0x7ffff7ffe118 <r_debug>, 0x7fff00000004, 0x7fff00000001, 0xe6108, 0x7ffff7dfffce3,
0x7ffff7fbc080, 0x3, 0x7ffff7a5a040, 0x3000000000, 0x820, 0x1a1b, 0x820, 0x1a1b, 0x1, 0x81a4, 0x0,
0x0, 0xe5a10, 0x1000, 0x730, 0x65069516, 0x1cc25e4a, 0x62c6196b, 0x0, 0x649d10c6, 0x24b328d9, 0x0,
0x0, 0x7ffff7fd06d4, 0x106f2bda, 0x0, 0x0, 0x7ffff7fcd4a3, 0xa, 0x1000000000, 0x0, 0x7fff00000000,
0x1, 0x1000000000, 0x7ffff7fd120, 0x0...}
size = 0
choice = 1
z = 0
L = 4
count = 1
b = false
(gdb) backtrace
#0 main () at kurs.cpp:224

```

Рисунок 3.3 – Вывод фрейма, переменных в памяти и трассировка выполняемой функции

```

(gdb) ptype ind_arr
type = struct Elem {
    char full_name[30];
    unsigned short number;
    char date[10];
    char full_lawyer_name[22];
} *[4000]
(gdb) print ind_arr
$1 = {0x0 <repeats 3311 times>, 0x7ffff7fccda5, 0x0, 0x28000, 0x27fe0, 0x2
0x1bd000, 0x1, 0x215000, 0x21b000, 0x21a888, 0x227e50, 0x1000, 0x214000,
0x4000000006, 0x40, 0x40, 0x40, 0x310, 0x310, 0x8, 0x4000000003, 0x1e3e30,
0x8a000, 0x898f5, 0x898f5, 0x1000, 0xe000, 0x5, 0x8a000, 0xe5000, 0xe47f
0x350, 0x350, 0x30, 0x30, 0x8, 0x4000000004, 0x380, 0x380, 0x380, 0x44, 0
0x1e3e4c, 0x0, 0x7ffff7fbc080, 0x7ffff7fd0c1e, 0x70cc, 0x1f, 0x20, 0xa,
0x7ffff7fccb0e, 0x3710, 0x7ffff7fd178, 0x1000, 0x7ffff7fc0003, 0xfffffff
0x7ffff7fbc080, 0x3, 0x7ffff7a5a040, 0x3000000000, 0x820, 0x1a1b, 0x820,
0x7ffff7fbc080, 0x0, 0x0, 0x7ffff7fd06d4, 0x106f2bda, 0x0, 0x0, 0x7ffff7fcd4a3, 0xa, 0x1000000000,
(gdb) |

```

Рисунок 3.4 – Просмотр типа переменной и её наполнения


```

(gdb) list 10
5      const int N = 4000;
6
7      struct Elem {
8          char full_name[30];
9          unsigned short int number;
10         char date[10];
11         char full_lawyer_name[22];
12     };
13     struct List{
14         Elem elem;
(gdb) list 150
145         q->tail = q2;
146         q->top = q->tail;
147     }
148     else
149     {
150         List* q3;
151         q3 = q->tail->next;
152         q->tail->next = q2;
153         q2->data = R;
154         q2->next = q3;
(gdb) list 120
115     int PrintSort(List* p, short int& count)
116     {
117         int b = 0;
118         for (int i = 0; i < N / 20 + 1; i++)
119         {
120             p = printList(p, count, 1);
121             std::cout << "Next 20? 1/0: ";
122             cin >> b;
123             if (b == false)
124             {
(gdb) b 117
Breakpoint 2 at 0x555555556e39: file curs.cpp, line 117.
(gdb) run
Starting program: /mnt/u/Documents/B ВУЗ/CAOD/Курсовая/Курсач/Курсач/1/curs

```

```

(gdb) next
226         p->data = new Elem;
(gdb) next
227         z = fread((Elem*)p->data, sizeof(Elem), 1, fin);
(gdb) i locals
i = 0
fin = 0x555555556ceb0
head = 0x555555556d090
p = 0x555555556d090
ind_arr = {0x0 <repeats 3311 times>, 0x7ffff7fccda5, 0x0, 0x28000, 0x27fe0, 0x27fe0, 0x1000, 0x0,
0x1, 0x28000, 0x1bd000, 0x1bc4c1, 0x1bc4c1, 0x1000, 0x28000, 0x5, 0x1bd000, 0x215000, 0x2148cc,
0x2148cc, 0x1000, 0x1bd000, 0x1, 0x215000, 0x21b000, 0x21a888, 0x227e50, 0x1000, 0x214000, 0x3,
0x0 <repeats 65 times>, 0x7ffff7fd090e, 0x7ffff7ffd040 <_rtld_global>, 0x7ffff7fd0b0,
0x7ffff7b41040, 0x7ffff7fccb0e, 0x4000000006, 0x40, 0x40, 0x40, 0x310, 0x310, 0x8, 0x4000000003,
0x1e3e30, 0x1e3e30, 0x1e3e30, 0x1c, 0x1c, 0x10, 0x4000000001, 0x0, 0x0, 0x7ffff7fcc02c, 0x0,
0xe000, 0xd2e0, 0xd2e0, 0x1000, 0x0, 0x1, 0xe000, 0x8a000, 0x898f5, 0x898f5, 0x1000, 0xe000, 0x5,
0x8a000, 0xe5000, 0xe47f4, 0xe47f4, 0x1000, 0x8a000, 0x1, 0xe5000, 0xe7000, 0xe60fc, 0xe6108,
0x1000, 0xe4000, 0x3, 0x1d0, 0x1d0, 0x8, 0x4000000004, 0x350, 0x350, 0x350, 0x30, 0x30, 0x8,
0x4000000004, 0x380, 0x380, 0x380, 0x44, 0x44, 0x4, 0x4000000007, 0x2148f0, 0x2158f0, 0x2158f0,

```

Рисунки 3.5 и 3.6 – Просмотр и создание следующего
брейк-поинта, и выполнение программы по шагам