

Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной
математики

Кафедра вычислительной математики и программирования

Лабораторная работа №3 по курсу «Дискретный анализ»

Студент: П. А. Мохляков
Преподаватель: Н. С. Капралов
Группа: М8О-208Б
Дата:
Оценка:
Подпись:

Москва, 2020

Лабораторная работа №3

Задача: Для реализации словаря из предыдущей лабораторной работы, необходимо провести исследование скорости выполнения и потребления оперативной памяти. В случае выявления ошибок или явных недочётов, требуется их исправить.

Результатом лабораторной работы является отчёт, состоящий из:

- Дневника выполнения работы, в котором отражено что и когда делалось, какие средства использовались и какие результаты были достигнуты на каждом шаге выполнения лабораторной работы.
- Выводов о найденных недочётах.
- Сравнение работы исправленной программы с предыдущей версией.
- Общих выводов о выполнении лабораторной работы, полученном опыте.

Минимальный набор используемых средств должен содержать утилиту `gprof` и библиотеку `dmalloc`, однако их можно заменять на любые другие аналогичные или более развитые утилиты (например, `Valgrind` или `Shark`) или добавлять к ним новые (например, `gcov`)

Используемые утилиты: `Valgrind`, `Callgrind`, `QCachegrind`.

1 Описание

В процессе работы использовались три утилиты, Vlagrind для поиска ошибок работы с памятью, Callgrind для профилирования программы и поиска кода, замедляющего программу и QCachegrind для просмотра отчетов Calgrind. В ходе тестирования было 2 основных теста это ввод пустого теста и ввод теста на 6500 элементов.

2 Дневник работы

Компиляция:

Сборка компилятором прошла успешно, как и запуск программы. Она успешно, как сначала кажется, выполняет заданное условие. Проходя пробный тест из условия задачи.

```
pavel@DESKTOP-VBSMFB3:~/Projects/mai/2_course/DA/LB3/solution_73$ g++ main.cpp
-o solution
pavel@DESKTOP-VBSMFB3:~/Projects/mai/2_course/DA/LB3/solution_73$ cat test0
+ a 1
+ A 2
+ aa 18446744073709551615
aa
A
-A
a
pavel@DESKTOP-VBSMFB3:~/Projects/mai/2_course/DA/LB3/solution_73$ cat test0
| ./solution
OK
Exist
OK
OK: 18446744073709551615
OK: 1
OK
NoSuchWord
```

Ошибка 1: При вводе пустого теста программа падает. Проверяю программу с этим тестом в Valgrind, получаем следующий вывод:

```

==1713== Memcheck, a memory error detector
==1713== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==1713== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==1713== Command: ./solution
==1713==
==1713== Conditional jump or move depends on uninitialised value(s)
==1713==   at 0x4A85143: tolower (ctype.c:46)
==1713==   by 0x10A54C: main (in /home/pavel/Projects/mai/2_course/DA/LB3/73/solution/solution)
==1713==
==1713== Use of uninitialised value of size 8
==1713==   at 0x4A85157: tolower (ctype.c:46)
==1713==   by 0x4A85157: tolower (ctype.c:44)
==1713==   by 0x10A54C: main (in /home/pavel/Projects/mai/2_course/DA/LB3/73/solution/solution)
==1713==
==1713== HEAP SUMMARY:
==1713==   in use at exit: 0 bytes in 0 blocks
==1713==   total heap usage: 5 allocs, 5 frees, 74,298 bytes allocated
==1713==
==1713== All heap blocks were freed -- no leaks are possible
==1713==
==1713== Use --track-origins=yes to see where uninitialised values come from
==1713== For lists of detected and suppressed errors, rerun with: -s
==1713== ERROR SUMMARY: 2 errors from 2 contexts (suppressed: 0 from 0)
pavel@DESKTOP-VBSMFB3:~/Projects/mai/2_course/DA/LB3/73/solution$ valgrind ./solution

```

Источником этой проблемы стала попытка приведения строки к нижнему регистру вне зависимости была ли она введена. То есть мы пытались изменить не инициализированную область памяти.

Листинг неисправного кода:

```

1 | friend std::istream& operator>>(std::istream &in, TString &str)
2 | {
3 |
4 |     in >> str.str;
5 |     str.lower();
6 |
7 |     while(str.str[str.size] != '\0')
8 |         ++str.size;
9 |     return in;
10| }

```

Исправить это можно сделав ввод условием при котором выполняется оставшаяся часть.

Исправленный вариант:

```

1 | friend std::istream& operator>>(std::istream &in, TString &str)
2 | {
3 |     if(in >> str.str)
4 |     {
5 |         while(str.str[str.size] != '\0')
6 |             ++str.size;
7 |     }
8 |     return in;
9 | }

```

Ошибка 2: При вводе большого теста Valgrind выводит большое количество ошибок:

```
==2012==
==2012== 1 errors in context 1 of 18:
==2012== Syscall param write(buf) points to uninitialised byte(s)
==2012== at 0x485F1E7: write (write.c:26)
==2012== by 0x491E75D: std::_basic_filebuf<char>::xspn(char const*, long) (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.28)
==2012== by 0x495E190: std::basic_filebuf<char, std::char_traits<char> >::_M_convert_to_external(char*, long) (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.28)
==2012== by 0x495E56B: std::basic_filebuf<char, std::char_traits<char> >::overflow(int) (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.28)
==2012== by 0x495E293: std::basic_filebuf<char, std::char_traits<char> >::_M_terminate_output() (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.28)
==2012== by 0x49615BE: std::basic_filebuf<char, std::char_traits<char> >::close() (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.28)
==2012== by 0x111BCC: NDict::TDict::parse_comand(NString::TString&) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x10A59B: main (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== Address 0x547d613 is 19 bytes inside a block of size 8,192 alloc'd
==2012== at 0x483C583: operator new[](unsigned long) (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==2012== by 0x490DF63: std::basic_filebuf<char, std::char_traits<char> >::_M_allocate_internal_buffer() (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.28)
==2012== by 0x49621C6: std::basic_filebuf<char, std::char_traits<char> >::open(char const*, std::_ios_Openmode) (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.28)
==2012== by 0x11AFA8: NDict::TDict::parse_comand(NString::TString&) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x10A59B: main (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== Uninitialised value was created by a heap allocation
==2012== at 0x483BE63: operator new(unsigned long) (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==2012== by 0x1112CC: rb::rb_tree<NPair::TPair>::load(char*) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x11235A: NDict::TDict::parse_comand(NString::TString&) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x10A59B: main (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012==
==2012== 9 errors in context 2 of 18:
==2012== Conditional jump or move depends on uninitialised value(s)
==2012== at 0x1121FB: NDict::TDict::parse_comand(NString::TString&) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x10A59B: main (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== Uninitialised value was created by a heap allocation
==2012== at 0x483C583: operator new[](unsigned long) (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==2012== by 0x111209: rb::rb_tree<NPair::TPair>::load(char*) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x11235A: NDict::TDict::parse_comand(NString::TString&) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x10A59B: main (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012==
==2012== 9 errors in context 3 of 18:
==2012== Conditional jump or move depends on uninitialised value(s)
==2012== at 0x1121F2: NDict::TDict::parse_comand(NString::TString&) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x10A59B: main (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== Uninitialised value was created by a heap allocation
==2012== at 0x483C583: operator new[](unsigned long) (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==2012==
==2012== 12 errors in context 14 of 18:
==2012== Conditional jump or move depends on uninitialised value(s)
==2012== at 0x11190F: NDict::TDict::parse_comand(NString::TString&) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x10A59B: main (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== Uninitialised value was created by a heap allocation
==2012== at 0x483C583: operator new[](unsigned long) (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==2012== by 0x111209: rb::rb_tree<NPair::TPair>::load(char*) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x11235A: NDict::TDict::parse_comand(NString::TString&) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x10A59B: main (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012==
==2012== 21 errors in context 15 of 18:
==2012== Conditional jump or move depends on uninitialised value(s)
==2012== at 0x111DA6: NDict::TDict::parse_comand(NString::TString&) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x10A59B: main (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== Uninitialised value was created by a heap allocation
==2012== at 0x483C583: operator new[](unsigned long) (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==2012== by 0x111209: rb::rb_tree<NPair::TPair>::load(char*) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x11235A: NDict::TDict::parse_comand(NString::TString&) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x10A59B: main (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012==
==2012== 21 errors in context 16 of 18:
==2012== Conditional jump or move depends on uninitialised value(s)
==2012== at 0x111D90: NDict::TDict::parse_comand(NString::TString&) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x10A59B: main (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== Uninitialised value was created by a heap allocation
==2012== at 0x483C583: operator new[](unsigned long) (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==2012== by 0x111209: rb::rb_tree<NPair::TPair>::load(char*) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x11235A: NDict::TDict::parse_comand(NString::TString&) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x10A59B: main (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012==
==2012== 21 errors in context 17 of 18:
==2012== Conditional jump or move depends on uninitialised value(s)
==2012== at 0x111D90: NDict::TDict::parse_comand(NString::TString&) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x10A59B: main (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== Uninitialised value was created by a heap allocation
==2012== at 0x483C583: operator new[](unsigned long) (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==2012== by 0x111209: rb::rb_tree<NPair::TPair>::load(char*) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x11235A: NDict::TDict::parse_comand(NString::TString&) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x10A59B: main (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012==
==2012== 21 errors in context 18 of 18:
==2012== Conditional jump or move depends on uninitialised value(s)
==2012== at 0x111D80: NDict::TDict::parse_comand(NString::TString&) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x10A59B: main (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== Uninitialised value was created by a heap allocation
==2012== at 0x483C583: operator new[](unsigned long) (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
```

```

==2012==
==2012==
==2012== 21 errors in context 15 of 18:
==2012== Conditional jump or move depends on uninitialised value(s)
==2012== at 0x111DA6: NDict::TDict::parse_comand(NString::TString&) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x10A59B: main (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== Uninitialised value was created by a heap allocation
==2012== at 0x483C583: operator new[](unsigned long) (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==2012== by 0x1112D9: rb::rb_tree<NPair::TPair>::load(char*) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x11235A: NDict::TDict::parse_comand(NString::TString&) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x10A59B: main (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012==
==2012== 21 errors in context 16 of 18:
==2012== Conditional jump or move depends on uninitialised value(s)
==2012== at 0x111D9D: NDict::TDict::parse_comand(NString::TString&) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x10A59B: main (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== Uninitialised value was created by a heap allocation
==2012== at 0x483C583: operator new[](unsigned long) (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==2012== by 0x1112D9: rb::rb_tree<NPair::TPair>::load(char*) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x11235A: NDict::TDict::parse_comand(NString::TString&) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x10A59B: main (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012==
==2012== 21 errors in context 17 of 18:
==2012== Conditional jump or move depends on uninitialised value(s)
==2012== at 0x111D9D: NDict::TDict::parse_comand(NString::TString&) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x10A59B: main (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== Uninitialised value was created by a heap allocation
==2012== at 0x483C583: operator new[](unsigned long) (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==2012== by 0x1112D9: rb::rb_tree<NPair::TPair>::load(char*) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x11235A: NDict::TDict::parse_comand(NString::TString&) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x10A59B: main (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012==
==2012== 21 errors in context 18 of 18:
==2012== Conditional jump or move depends on uninitialised value(s)
==2012== at 0x111D80: NDict::TDict::parse_comand(NString::TString&) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x10A59B: main (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== Uninitialised value was created by a heap allocation
==2012== at 0x483C583: operator new[](unsigned long) (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==2012== by 0x1112D9: rb::rb_tree<NPair::TPair>::load(char*) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x11235A: NDict::TDict::parse_comand(NString::TString&) (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012== by 0x10A59B: main (in /home/pavel/Projects/mai/2_course/DA/LB3/90/solution/solution)
==2012==
==2012== ERROR SUMMARY: 232 errors from 18 contexts (suppressed: 0 from 0)
pavel@DESKTOP-V85NFB3:~/Projects/mai/2_course/DA/LB3/90/solution$

```

При тестировании я пришел к выводу, что эти ошибки появляются после загрузки пустого файла, поэтому следует искать ошибку там. Ошибка оказалась в условии. При прочтении пустого файла не начиналось чтение из файла, но при этом создавался "мусорный" элемент. Поэтому решением будет поместить код отвечающий за создание элемента под тоже условие.

Листинг неисправного кода:

```

1 | Root = new rb_tree_elem<T>;
2 | Root->Par = Nil;
3 | Root->Left = Nil;
4 | Root->Right = Nil;
5 | if(!isEmpty(ch))
6 | load_tree(Root,rf);

```

Исправленный вариант:

```

1 | Root = Nil;
2 | if(!isEmpty(ch)){
3 |     Root = new rb_tree_elem<T>;
4 |     Root->Par = Nil;
5 |     Root->Left = Nil;
6 |     Root->Right = Nil;
7 |     load_tree(Root,rf);
8 | }

```

```

==20078==
==20078== HEAP SUMMARY:
==20078==    in use at exit: 122,880 bytes in 6 blocks
==20078==    total heap usage: 23,374 allocs, 23,368 frees, 6,999,797 bytes allocated
==20078==
==20078== LEAK SUMMARY:
==20078==    definitely lost: 0 bytes in 0 blocks
==20078==    indirectly lost: 0 bytes in 0 blocks
==20078==    possibly lost: 0 bytes in 0 blocks
==20078==    still reachable: 122,880 bytes in 6 blocks
==20078==           suppressed: 0 bytes in 0 blocks
==20078== Rerun with --leak-check=full to see details of leaked memory
==20078==
==20078== For lists of detected and suppressed errors, rerun with: -s
==20078== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
pavel@DESKTOP-VBSMF33:~/Projects/mai/2_course/DA/LB2/tosol/solution$

```

Callgrind - одна из самых удобных утилит для анализа времени исполнения программы. Callgrind выдает отчет в текстовом формате, который удобно просматривать с помощью утилит kcachegrind и qcachegrind, и их помощью можно легко просмотреть количество вызовов и затратность по времени тех или иных функций.

```
pavel@DESKTOP-VBSMFB3:~/Projects/mai/2_course/DA/LB2$ cat test | valgrind
```

Откроем сгенерированный файл в qcachegrind:

[illegible]

Слева мы видим список функций, который мы можем отсортировать по количеству вызовов и времени исполнения, тем самым определяя проблемные участки. Справа сверху можно просмотреть исходный код функции, а снизу например древо вызовов функций. В моем случае я не могу просмотреть код функций, так как код отчета локально обращается к файлам с исходным кодом, из-за этого после перемещения отчета мы не можем обратиться к ним.

3 Выводы

Выполнив третью лабораторную работу по курсу «Дискретный анализ», я познакомился со утилитами для профилирования и научился ими пользоваться для проверки программ на ошибки в работе с памятью и искать недочеты в коде.