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: Ю. Ю. Обыденкова
Преподаватель: А. Н. Ридли
Группа: М8О-308Б-18
Дата:
Оценка:
Подпись:

Москва, 2021

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 исправление.
Минимальный набор используемых средст должен содержать утилиту gprof и библиотеку dmalloc, однако их можно заменять на любые другие аналогичные или более известные утилиты (например, Valgrind или Shark) или добавлять к ним новые (например, gcov).
Структура данных - PATRICIA

1 Описание

Для анализа работы программы будем использовать утилиты valgrind, gprof и perf.

1 Valgrind

Valgrind предназначен для отладки использования памяти, обнаружения утечек памяти, а также профилирования. Valgrind является по сути виртуальной машиной, использующей методы JIT-компиляции, среди которых – динамическая перекомпиляция. Valgrind транслитует программу во временную, более простую форму, называемую промежуточным представлением и работает с этим представлением.

2 Gprof

Gprof – инструмент для анализа производительности UNIX приложений . Gprof вносит в программу дополнительный код на этапе компиляции, для извлечения необходимой информации.

3 Gcov

Gcov — свободно распространяемая утилита для исследования покрытия кода. Gcov генерирует точное количество исполнений для каждого оператора в программе и позволяет добавить аннотации к исходному коду. Gcov поставляется как стандартная утилита в составе пакета GCC.[1]

2 Тестирование программы

Создадим небольшой генератор тестов для нашей программы, который будет генерировать файлы, передаваемые на вход программе с помощью перенаправления ввода вывода.

1 Valgrind

```
#include <ctime>
#include <random>
#include <map>
#include <limits>
#include <tuple>
#include <string>
#include <cstdlib>
#include <iostream>
#include <chrono>
#include <iomanip>

#include "TPatricia.h"
#include "profile.h"

using namespace std;

default_random_engine rng;

uint64_t get_number(uint64_t min = 0,uint64_t max = numeric_limits<unsigned long long
    >::max()) {
    uniform_int_distribution<unsigned long long> dist_ab(min, max);
    return dist_ab(rng);
}

string get_string() {
    size_t string_size = get_number(1,256);
    string string;
    string.resize(string_size);
    for (size_t i = 0; i < string_size; ++i) {
        string[i] = 'a' + get_number(0,25);
    }
    return string;
}

int main(int argc, char** argv) {
    if (argc < 2) {
        return 0;
    }
    size_t count = stoll(argv[1]);
```

```
39      rng.seed(std::chrono::system_clock::now().time_since_epoch().count());
40      vector<pair<string,unsigned long long>> test_data(count);
41
42      {
43          for (size_t i = 0; i < count; ++i) {
44              test_data[i].first = get_string();
45              test_data[i].second = get_number(0, numeric_limits<unsigned long long>::max
                    ());
46          }
47          for (size_t i = 0; i < count; ++i) {
48              cout << "+ " << test_data[i].first << " " << test_data[i].second << "\n";
49          }
50          std::shuffle(test_data.begin(), test_data.end(), rng);
51          for (size_t i = 0; i < count; ++i) {
52              cout << "+ " << test_data[i].first << " " << test_data[i].second << "\n";
53          }
54          std::shuffle(test_data.begin(), test_data.end(), rng);
55          std::cout << "! Save tree_file\n";
56          for (size_t i = 0; i < count; ++i) {
57              cout << "- " << test_data[i].first << "\n";
58          }
59          std::shuffle(test_data.begin(), test_data.end(), rng);
60          for (size_t i = 0; i < count; ++i) {
61              cout << "- " << test_data[i].first << "\n";
62          }
63          std::cout << "! Load tree_file\n";
64          for (size_t i = 0; i < count; ++i) {
65              cout << test_data[i].first << "\n";
66          }
67      }
68      return 0;
69  }
```

С помощью файлов, генерируемых этой программой, можно проверить вставку и удаление в дерево(даже в тех случаях, когда ключ для вставки уже есть в дереве или когда ключа для удаления в дереве нет), поиск, сохранение и загрузку дерева из бинарного файла

Протестируем программу с помощью valgrind.

```
julia@julia21novo:~/CLionProjects/da_02/cmake-build-debug$ ./text_test 100 >test_file
Generate: 2 ms
julia@julia21novo:~/CLionProjects/da_02/cmake-build-debug$ head -10 test_file
+ ldyycjvrrvspqfywgxkplcqegvfegkusdshsmrvgytluxcswvzjooximbgfmwioigolxkcfmxjlwkcpcvgyp
4864557196841833101
+ ymvvvkuumkvisgbiotigxraqnwqjmt 17289865313646074648
+ zwieqryproiocurcxbxqazbgdwbyrezdcyohqpfulljodfrzsxvyziqfwumxzqlyhcykkyijizuylpplxczb
```

```
14343040435473772830
+ vbadtxyzmzhrbrjoljojyxhngznvygaosatmuqoparsmipwpyhkiqhkrlyhofwtlfkhfrsfnveigfkenxznz
5435407115923534065
+ xoprdvyngakjicuadiplbigfvgfauseismhpwyfmqtn 9018525952244026488
+ ouuoohyuvyzyhsgpzovkidqysypsxpvammaxuxkswqgtdipb 9614688308841786507
+ ozupwkcduuahmkxaafhaicirjvosrdbyfaieukfjbcefyfkkkiyoegcthdqslcilubiakzpfhbdzq
15870777454782644898
+ uqlinghmvehhhliscbrxlpkadgavkcftgllkghdvtndghxqqsbwmsszksiozhqoaqpllojghhabiijedippl
97140334622290269
+ lugowkoqahliltkhbudiwkiawnrlkzkxppoxououpabvy 11688839788771377031
+ grrjveowcafcfnsyxhdlqzifolucbnttrxkygcbowyofrcpuelxopyjkfzmjkzwulmdlwteyytphchcyucwo
2826873418547064160
julia@julia21ovo:~/CLionProjects/da_02/cmake-build-debug$ valgrind ./da_02
<test_file >/dev/null
==6744== Memcheck,a memory error detector
==6744== Copyright (C) 2002-2017,and GNU GPL'd,by Julian Seward et al.
==6744== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==6744== Command: ./da_02
==6744==
==6744==
==6744== HEAP SUMMARY:
==6744==     in use at exit: 122,880 bytes in 6 blocks
==6744==   total heap usage: 4,402 allocs,4,396 frees,433,318 bytes allocated
==6744==
==6744== LEAK SUMMARY:
==6744==    definitely lost: 0 bytes in 0 blocks
==6744==    indirectly lost: 0 bytes in 0 blocks
==6744==      possibly lost: 0 bytes in 0 blocks
==6744==    still reachable: 122,880 bytes in 6 blocks
==6744==         suppressed: 0 bytes in 0 blocks
==6744== Rerun with --leak-check=full to see details of leaked memory
==6744==
==6744== For lists of detected and suppressed errors,rerun with: -s
==6744== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Как видно, явных утечек нет, но достижима утечка 122,880 байт. Попробуем получить больше информации с помощью ключа –leak-check-full.

```
julia@julia21novo:~/CLionProjects/da_02/cmake-build-debug$ valgrind --leak-check=full
--show-leak-kinds=all ./da_02 <test_file >/dev/null
```

```
==7112== Memcheck,a memory error detector
==7112== Copyright (C) 2002-2017,and GNU GPL'd,by Julian Seward et al.
==7112== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==7112== Command: ./da_02
==7112==
==7112==
==7112== HEAP SUMMARY:
==7112==     in use at exit: 122,880 bytes in 6 blocks
==7112==   total heap usage: 4,400 allocs,4,394 frees,425,126 bytes allocated
==7112==
==7112== 8,192 bytes in 1 blocks are still reachable in loss record 1 of 6
==7112==    at 0x4C30B5B: operator new[](unsigned long) (vg_replace_malloc.c:433)
==7112==    by 0x4F2D097: std::basic_filebuf<char,std::char_traits<char>>::_M_allocate
(in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.25)
==7112==    by 0x4F2AE72: ??? (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.25)
==7112==    by 0x4EE0B80: std::ios_base::sync_with_stdio(bool) (in /usr/lib/x86_64-lin
==7112==    by 0x10985C: main (main.cpp:16)
==7112==
==7112== 8,192 bytes in 1 blocks are still reachable in loss record 2 of 6
==7112==    at 0x4C30B5B: operator new[](unsigned long) (vg_replace_malloc.c:433)
==7112==    by 0x4F2D097: std::basic_filebuf<char,std::char_traits<char>>::_M_allocate
(in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.25)
==7112==    by 0x4F2AE72: ??? (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.25)
==7112==    by 0x4EE0BA1: std::ios_base::sync_with_stdio(bool) (in /usr/lib/x86_64-lin
==7112==    by 0x10985C: main (main.cpp:16)
==7112==
==7112== 8,192 bytes in 1 blocks are still reachable in loss record 3 of 6
==7112==    at 0x4C30B5B: operator new[](unsigned long) (vg_replace_malloc.c:433)
==7112==    by 0x4F2D097: std::basic_filebuf<char,std::char_traits<char>>::_M_allocate
(in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.25)
==7112==    by 0x4F2AE72: ??? (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.25)
==7112==    by 0x4EE0BC2: std::ios_base::sync_with_stdio(bool) (in /usr/lib/x86_64-lin
==7112==    by 0x10985C: main (main.cpp:16)
==7112==
==7112== 32,768 bytes in 1 blocks are still reachable in loss record 4 of 6
==7112==    at 0x4C30B5B: operator new[](unsigned long) (vg_replace_malloc.c:433)
==7112==    by 0x4F2EE7A: std::basic_filebuf<wchar_t,std::char_traits<wchar_t>>::_M_al
(in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.25)
==7112==    by 0x4F2B052: ??? (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.25)
==7112==    by 0x4EE0C37: std::ios_base::sync_with_stdio(bool) (in /usr/lib/x86_64-lin
==7112==    by 0x10985C: main (main.cpp:16)
```

```
==7112==
==7112== 32,768 bytes in 1 blocks are still reachable in loss record 5 of 6
==7112==    at 0x4C30B5B: operator new[](unsigned long) (vg_replace_malloc.c:433)
==7112==    by 0x4F2EE7A: std::basic_filebuf<wchar_t,std::char_traits<wchar_t>>::_M_al
(in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.25)
==7112==    by 0x4F2B052: ??? (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.25)
==7112==    by 0x4EE0C51: std::ios_base::sync_with_stdio(bool) (in /usr/lib/x86_64-lin
==7112==    by 0x10985C: main (main.cpp:16)
==7112==
==7112== 32,768 bytes in 1 blocks are still reachable in loss record 6 of 6
==7112==    at 0x4C30B5B: operator new[](unsigned long) (vg_replace_malloc.c:433)
==7112==    by 0x4F2EE7A: std::basic_filebuf<wchar_t,std::char_traits<wchar_t>>::_M_al
(in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.25)
==7112==    by 0x4F2B052: ??? (in /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.25)
==7112==    by 0x4EE0C6B: std::ios_base::sync_with_stdio(bool) (in /usr/lib/x86_64-lin
==7112==    by 0x10985C: main (main.cpp:16)
==7112==
==7112== LEAK SUMMARY:
==7112==    definitely lost: 0 bytes in 0 blocks
==7112==    indirectly lost: 0 bytes in 0 blocks
==7112==      possibly lost: 0 bytes in 0 blocks
==7112==    still reachable: 122,880 bytes in 6 blocks
==7112==         suppressed: 0 bytes in 0 blocks
==7112==
==7112== For lists of detected and suppressed errors,rerun with: -s
==7112== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Исходя из полученной информации, можно установить, что причиной возможных утечек являются методы $sync_with_stdio$, попробуем убрать их из программы и запустить valgrind снова.

```
julia@julia21novo:~/CLionProjects/da_02/cmake-build-debug$ valgrind --leak-check=full
--show-leak-kinds=all ./da_02 <test_file >/dev/null
==7232== Memcheck,a memory error detector
==7232== Copyright (C) 2002-2017,and GNU GPL'd,by Julian Seward et al.
==7232== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==7232== Command: ./da_02
==7232==
==7232==
==7232== HEAP SUMMARY:
```

```
==7232==     in use at exit: 0 bytes in 0 blocks
==7232==   total heap usage: 4,396 allocs,4,396 frees,310,438 bytes allocated
==7232==
==7232== All heap blocks were freed --no leaks are possible
==7232==
==7232== For lists of detected and suppressed errors,rerun with: -s
==7232== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

2 gprof

Проверим время выполнения нашей программы с помощью Gprof. Для этого надо скомпилировать программу с флагом -pg(добавим этот флаг в CMakeLists.txt).

```
julia@julia21novo:~/CLionProjects/da_02/cmake-build-debug$ gprof -p ./da_02
<test_file  gmon.out
Flat profile:

Each sample counts as 0.01 seconds.
%   cumulative   self              self     total
time   seconds   seconds    calls  ms/call  ms/call  name
22.23      0.02     0.02  9081875     0.00     0.00  TVector<unsigned char>::operator
long) const
22.23      0.04     0.02  1354061     0.00     0.00  getNthBit(TVector<unsigned
char>const&,unsigned long)
11.11      0.05     0.01  6440425     0.00     0.00  TVector<unsigned char>::PushBack
char const&)
11.11      0.06     0.01  5806083     0.00     0.00  TVector<unsigned char>::Size()
const
11.11      0.07     0.01  1288056     0.00     0.00  TVector<unsigned char>::operator
long)
11.11      0.08     0.01    54510     0.00     0.00  TPatricia<unsigned long
long>::SearchKey(TVector<unsigned char>const&,TPatricia<unsigned long long>::Node**)
const
11.11      0.09     0.01    50000     0.00     0.00  strToVec(char const*)
0.00      0.09     0.00  1209461     0.00     0.00  unsigned char* std::__niter_base<u
char*>(unsigned char*)
0.00      0.09     0.00   799660     0.00     0.00  unsigned char* std::__miter_base<u
char*>(unsigned char*)
0.00      0.09     0.00   409801     0.00     0.00  TVector<unsigned char>::begin()
0.00      0.09     0.00   409801     0.00     0.00  unsigned char* std::__copy_move<fa
char>(unsigned char const*,unsigned char const*,unsigned char*)
0.00      0.09     0.00   399830     0.00     0.00  TVector<unsigned char>::end()
0.00      0.09     0.00   399830     0.00     0.00  unsigned char* std::__copy_move_a<
char*,unsigned char*>(unsigned char*,unsigned char*,unsigned char*)
0.00      0.09     0.00   399830     0.00     0.00  unsigned char* std::__copy_move_a2
char*,unsigned char*>(unsigned char*,unsigned char*,unsigned char*)
0.00      0.09     0.00   399830     0.00     0.00  unsigned char* std::copy<unsigned
char*,unsigned char*>(unsigned char*,unsigned char*,unsigned char*)
0.00      0.09     0.00   233834     0.00     0.00  unsigned long const& std::max<unsi
long>(unsigned long const&,unsigned long const&)
```

```
0.00      0.09     0.00   114394     0.00     0.00  TVector<unsigned char>::~TVector()
0.00      0.09     0.00    69795     0.00     0.00  TVector<TPatricia<unsigned
long long>::Node*>::operator[](unsigned long)
0.00      0.09     0.00    59826     0.00     0.00  std::remove_reference<unsigned
long&>::type&& std::move<unsigned long&>(unsigned long&)
0.00      0.09     0.00    43501     0.00     0.00  std::remove_reference<TVector<unsi
char>&>::type&& std::move<TVector<unsigned char>&>(TVector<unsigned char>&)
0.00      0.09     0.00    39999     0.00     0.00  bool operator==<unsigned
char>(TVector<unsigned char>const&,TVector<unsigned char>const&)
0.00      0.09     0.00    39887     0.00     0.00  TVector<TPatricia<unsigned
long long>::Node*>::Size() const
0.00      0.09     0.00    29913     0.00     0.00  std::remove_reference<unsigned
char*&>::type&& std::move<unsigned char*&>(unsigned char*&)
0.00      0.09     0.00    24510     0.00     0.00  TVector<unsigned char>::TVector(TV
char>&&)
0.00      0.09     0.00    20000     0.00     0.00  TPatricia<unsigned long
long>::Erase(TVector<unsigned char>const&)
0.00      0.09     0.00    20000     0.00     0.00  TPatricia<unsigned long
long>::Insert(TVector<unsigned char>,unsigned long long)
0.00      0.09     0.00    19942     0.00     0.00  TVector<unsigned char>::TVector()
0.00      0.09     0.00    19942     0.00     0.00  TPatricia<unsigned long
long>::Node::Node()
0.00      0.09     0.00    19942     0.00     0.00  TPatricia<unsigned long
long>::Node::~Node()
0.00      0.09     0.00    19942     0.00     0.00  unsigned char const* std::__miter_
char const*>(unsigned char const*)
0.00      0.09     0.00    19942     0.00     0.00  unsigned char const* std::__niter_
char const*>(unsigned char const*)
0.00      0.09     0.00    19942     0.00     0.00  std::enable_if<std::__and_<std::__
long>>,std::is_move_constructible<unsigned long>,std::is_move_assignable<unsigned
long>>::value,void>::type std::swap<unsigned long>(unsigned long&,unsigned
long&)
0.00      0.09     0.00    18991     0.00     0.00  TVector<unsigned char>::operator=
char>&&)
0.00      0.09     0.00    14481     0.00     0.00  TPatricia<unsigned long
long>::SearchParentNode(TPatricia<unsigned long long>::Node*) const
0.00      0.09     0.00    13530     0.00     0.00  std::remove_reference<unsigned
long long&>::type&& std::move<unsigned long long&>(unsigned long long&)
0.00      0.09     0.00    10000     0.00     0.00  TOptional<unsigned long
long>::TOptional(unsigned long long const&)
0.00      0.09     0.00    10000     0.00     0.00  TOptional<unsigned long
```

```
long>::operator*()
0.00      0.09     0.00    10000     0.00     0.00  TOptional<unsigned long
long>::operator bool() const
0.00      0.09     0.00    10000     0.00     0.00  TPatricia<unsigned long
long>::operator[](TVector<unsigned char>const&) const
0.00      0.09     0.00    10000     0.00     0.00  bool operator!=<unsigned
char>(TVector<unsigned char>const&,TVector<unsigned char>const&)
0.00      0.09     0.00     9971     0.00     0.00  TVector<TPatricia<unsigned
long long>::Node*>::PushBack(TPatricia<unsigned long long>::Node* const&)
0.00      0.09     0.00     9971     0.00     0.00  TVector<unsigned char>::TVector(T
char>const&)
0.00      0.09     0.00     9971     0.00     0.00  TVector<unsigned char>::TVector(u
long)
0.00      0.09     0.00     9971     0.00     0.00  TVector<unsigned char>::operator=
char>const&)
0.00      0.09     0.00     9971     0.00     0.00  TVector<unsigned char>::end()
const
0.00      0.09     0.00     9971     0.00     0.00  TVector<unsigned char>::begin()
const
0.00      0.09     0.00     9971     0.00     0.00  unsigned char* std::__copy_move_a<
char const*,unsigned char*>(unsigned char const*,unsigned char const*,unsigned
char*)
0.00      0.09     0.00     9971     0.00     0.00  unsigned char* std::__copy_move_a2
char const*,unsigned char*>(unsigned char const*,unsigned char const*,unsigned
char*)
0.00      0.09     0.00     9971     0.00     0.00  unsigned char* std::copy<unsigned
char const*,unsigned char*>(unsigned char const*,unsigned char const*,unsigned
char*)
0.00      0.09     0.00     9971     0.00     0.00  std::enable_if<std::__and_<std::__
char*>>,std::is_move_constructible<unsigned char*>,std::is_move_assignable<unsigned
char*>>::value,void>::type std::swap<unsigned char*>(unsigned char*&,unsigned
char*&)
0.00      0.09     0.00     4510     0.00     0.00  std::enable_if<std::__and_<std::__
char>>>,std::is_move_constructible<TVector<unsigned char>>,std::is_move_assignable<TVe
char>>>::value,void>::type std::swap<TVector<unsigned char>>(TVector<unsigned
char>&,TVector<unsigned char>&)
0.00      0.09     0.00     4510     0.00     0.00  std::enable_if<std::__and_<std::__
long long>>,std::is_move_constructible<unsigned long long>,std::is_move_assignable<uns
long long>>::value,void>::type std::swap<unsigned long long>(unsigned long
long&,unsigned long long&)
0.00      0.09     0.00       45     0.00     0.00  TPatricia<unsigned long
```

```
long>::Node** std::__niter_base<TPatricia<unsigned long long>::Node**>(TPatricia<unsig
long long>::Node**)
0.00      0.09     0.00       30     0.00     0.00  TPatricia<unsigned long
long>::Node** std::__miter_base<TPatricia<unsigned long long>::Node**>(TPatricia<unsig
long long>::Node**)
0.00      0.09     0.00       15     0.00     0.00  TVector<TPatricia<unsigned
long long>::Node*>::end()
0.00      0.09     0.00       15     0.00     0.00  TVector<TPatricia<unsigned
long long>::Node*>::begin()
0.00      0.09     0.00       15     0.00     0.00  TPatricia<unsigned long
long>::Node** std::__copy_move<false,true,std::random_access_iterator_tag>::__copy_m<T
long long>::Node*>(TPatricia<unsigned long long>::Node* const*,TPatricia<unsigned
long long>::Node* const*,TPatricia<unsigned long long>::Node**)
0.00      0.09     0.00       15     0.00     0.00  TPatricia<unsigned long
long>::Node** std::__copy_move_a<false,TPatricia<unsigned long long>::Node**,TPatricia
long long>::Node**>(TPatricia<unsigned long long>::Node**,TPatricia<unsigned
long long>::Node**,TPatricia<unsigned long long>::Node**)
0.00      0.09     0.00       15     0.00     0.00  TPatricia<unsigned long
long>::Node** std::__copy_move_a2<false,TPatricia<unsigned long long>::Node**,TPatrici
long long>::Node**>(TPatricia<unsigned long long>::Node**,TPatricia<unsigned
long long>::Node**,TPatricia<unsigned long long>::Node**)
0.00      0.09     0.00       15     0.00     0.00  TPatricia<unsigned long
long>::Node** std::copy<TPatricia<unsigned long long>::Node**,TPatricia<unsigned
long long>::Node**>(TPatricia<unsigned long long>::Node**,TPatricia<unsigned
long long>::Node**,TPatricia<unsigned long long>::Node**)
0.00      0.09     0.00        3     0.00     0.00  std::remove_reference<TPatricia<un
long long>::Node*&>::type&& std::move<TPatricia<unsigned long long>::Node*&>(TPatricia
long long>::Node*&)
0.00      0.09     0.00        2     0.00     0.00  TVector<TPatricia<unsigned
long long>::Node*>::~TVector()
0.00      0.09     0.00        2     0.00     0.00  TPatricia<unsigned long
long>::DeleteTree(TPatricia<unsigned long long>::Node*)
0.00      0.09     0.00        2     0.00     0.00  TPatricia<unsigned long
long>::~TPatricia()
0.00      0.09     0.00        2     0.00     0.00  std::operator|(std::_Ios_Openmode
0.00      0.09     0.00        1     0.00     0.00  _GLOBAL__sub_I__Z8strToVecPKc
0.00      0.09     0.00        1     0.00     0.00  _GLOBAL__sub_I__Z9printCharh
0.00      0.09     0.00        1     0.00     0.00  __static_initialization_and_destru
0.00      0.09     0.00        1     0.00     0.00  __static_initialization_and_destru
0.00      0.09     0.00        1     0.00     0.00  TVector<TPatricia<unsigned
long long>::Node*>::TVector(unsigned long)
```

```
0.00      0.09     0.00        1     0.00    10.00  TPatricia<unsigned long
long>::ScanFromFile(char const*)
0.00      0.09     0.00        1     0.00     0.02  TPatricia<unsigned long
long>::PrintToFile(char const*) const
0.00      0.09     0.00        1     0.00     0.00  TPatricia<unsigned long
long>::CountIds(TPatricia<unsigned long long>::Node*,int&,TVector<TPatricia<unsigned
long long>::Node*>&) const
0.00      0.09     0.00        1     0.00     0.00  std::enable_if<std::__and_<std::_
long long>::Node*>>,std::is_move_constructible<TPatricia<unsigned long long>::Node*>,s
long long>::Node*>>::value,void>::type std::swap<TPatricia<unsigned long long>::Node*>
long long>::Node*&,TPatricia<unsigned long long>::Node*&)

%         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) 2012-2018 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.
```

В дерево было вставлено 10000 элементов, которые позже были сохранены в файл, удалены, загружены из файла, так же для каждого элемента была проведена попытка его поиска в дереве. Утилита показывает время работы каждой функции. Благодаря gprof можно легко искать медленно работающие участки программы и немедленно приступать к их ускорению.

3 gcov

Для использования утилиты gcov, необходимо сперва скомпилировать программу с ключом –coverage. После запуска полученной программы и завершения ее работы, будет сгенерирован файл с расширением gcda, содержащий информацию о покрытии кода.

```
julia@julia21novo:~/CLionProjects/da_02$ g++ --coverage main.cpp TPatricia.cpp
-o da_02_gcov
julia@julia21novo:~/CLionProjects/da_02$ ls
cmake-build-debug  main.cpp   test_generator.cpp       TPatricia.gcno
CMakeLists.txt     main.gcno  text_test_generator.cpp  TPatricia.h
copy_test.cpp      profile.h  TOptional.h              TVector.h
da_02_gcov         report     TPatricia.cpp
julia@julia21novo:~/CLionProjects/da_02$ ./da_02_gcov <cmake-build-debug/test_file
>/dev/null
julia@julia21novo:~/CLionProjects/da_02$ ls
cmake-build-debug  main.cpp   report                   TPatricia.cpp   tree_file
CMakeLists.txt     main.gcda  test_generator.cpp       TPatricia.gcda  TVector.h
copy_test.cpp      main.gcno  text_test_generator.cpp  TPatricia.gcno
da_02_gcov         profile.h  TOptional.h              TPatricia.h
julia@julia21novo:~/CLionProjects/da_02$ gcov main.gcda
File 'main.cpp'
Lines executed:92.68% of 41
Creating 'main.cpp.gcov'

File '/usr/include/c++/7/iostream'
Lines executed:100.00% of 1
```

```
Creating 'iostream.gcov'

File '/usr/include/c++/7/bits/stl_algobase.h'
Lines executed:100.00% of 20
Creating 'stl_algobase.h.gcov'

File '/usr/include/c++/7/bits/cpp_type_traits.h'
Lines executed:100.00% of 2
Creating 'cpp_type_traits.h.gcov'

File 'TVector.h'
Lines executed:93.44% of 61
Creating 'TVector.h.gcov'

File 'TPatricia.h'
Lines executed:86.92% of 237
Creating 'TPatricia.h.gcov'

File '/usr/include/c++/7/bits/move.h'
Lines executed:100.00% of 7
Creating 'move.h.gcov'

File 'TOptional.h'
Lines executed:75.00% of 8
Creating 'TOptional.h.gcov'

File '/usr/include/c++/7/bits/ios_base.h'
Lines executed:100.00% of 2
Creating 'ios_base.h.gcov'
```

Как видно, код программы при выполнении покрывается достаточно полно, но его можно улучшить, удалив некоторые участки кода. Обычно целью тестирования покрытия кода является выявление невыполнившихся областей кода. Полученные данные можно использовать для регрессионного тестирования(направленного на обнаружение ошибок в уже протестированных участках кода), тщательно проверяющего исходный код. Покрытие кода особенно важно для программ с высокими требованиями к безопасности. Удалим из TPatricia.h несколько строк кода, написанного при раннем тестировании и не необходимых более и запустим программу еще раз.

```
File 'main.cpp'
```

```
Lines executed:92.68% of 41
Creating 'main.cpp.gcov'

File '/usr/include/c++/7/iostream'
Lines executed:100.00% of 1
Creating 'iostream.gcov'

File '/usr/include/c++/7/bits/stl_algobase.h'
Lines executed:100.00% of 20
Creating 'stl_algobase.h.gcov'

File '/usr/include/c++/7/bits/cpp_type_traits.h'
Lines executed:100.00% of 2
Creating 'cpp_type_traits.h.gcov'

File 'TVector.h'
Lines executed:93.44% of 61
Creating 'TVector.h.gcov'

File 'TPatricia.h'
Lines executed:88.36% of 232
Creating 'TPatricia.h.gcov'

File '/usr/include/c++/7/bits/move.h'
Lines executed:100.00% of 7
Creating 'move.h.gcov'

File 'TOptional.h'
Lines executed:75.00% of 8
Creating 'TOptional.h.gcov'

File '/usr/include/c++/7/bits/ios_base.h'
Lines executed:100.00% of 2
Creating 'ios_base.h.gcov'
```

Покрытие увеличилось на несколько процентов. Можно считать, что программа стала чуть лучше.

3 Выводы

Выполнив третью лабораторную работу, я познакомилась со множеством утилит, крайне полезных для всестороннего тестированияи отладки своей программы. С помощью gprof я познакомилась с профилированием,которое позволяет изучить, где программа расходует свое время и какие функции вызывали другие функции, пока программа исполнялась. Эта информация может указать на ту часть программы, которая исполняется медленнее, чем ожидалось, и которая может быть кандидатом на переписывание, чтобы ускорить выполнение программы. Эта информация также подсказывает, какие функции вызывались чаще или реже, чем ожидалось. Это может помочь отметить ошибки, которые иначе остались бы незамеченными. Полученные знания наверняка помогут мне в будущем быстрее и эффективнее разрабатывать различные программы.