**Московский авиационный институт**

**(национальный исследовательский университет)**

**Факультет информационных технологий и прикладной математики**

**Кафедра вычислительной математики и программирования**

**Лабораторная работа №3 по курсу дискретного анализа**

Студентка: Ю. В. Алексюнина

Преподаватель: Н. А. Зацепин

Группа: М8О-407Б-17

Дата:

Оценка:

Подпись:

**Москва, 2020**

Условие

Провести исследование скорости выполнения и потребления оперативной памяти. В случае выявления ошибок или явных недочётов, требуется их исправить.

Результатом л.р. является отчёт, состоящий из:

* Дневника выполнения работы:

Что и когда делалось

Какие средства использовались

Какие результаты были достигнуты на каждом шаге выполнения лаборатор-ной работы

* Выводов о найденных недочётов.
* Сравнение работы исправленной программы с предыдущей версии.
* Общих выводов о выполнении лабораторной работы, полученном опыте.

Метод решения

Добавила ошибки и посмотрел, как можно проанализировать прогрмму.

Используемые утилиты

Дебаггер - valgrind.

* решила использовать его, так как он имеет консольный интерфейс, что означает, что можно запустить его по ssh, что, в свою очередь, открывает много возможностей.

Отладку программ следует проводить с ключами компиляции -g (добавляем инфор-мацию необходимую для дебаггера) и -O0 (выключает оптимизацию).

* + man старице к волгринду можно найти информацию о том, что можно создать файл .valgrindrc и туда добавлять ключи (вместо того, чтобы писать их каждый раз по новой)

Ключи к valgrind:

* --quit - вывод только информации об ошибках
* --leak-check=yes - проверка на утечки памяти (с помощью модуля memcheck)
* --vgdb=full,--vgdb-error=number - вызов gdb как только valgrind найдёт number

ошибок

gdb можно присоеденить командой target remote | /usr/bin/vgdb --pid=<pid> Типы ошибок при работе с memcheck:

1. Difinitely lost - не освободила память, при выходе указателя за область видимости.

1

1. Possibly lost - valgrind не уверена в том, что указатель на начало области памяти до сих пор существует.
2. Still reachable - нашла указатель на начало не освобождённого блока памяти (-- show-reachable=yes)

Профайлер - perf. Опять же использует командную строку. Перед использованием надо установить пакет linux-tools-<версия ядра> (можно найти с помощью команды uname -r)

Исследование программы

Ошибка освобождения памяти

Добавляем один элемент в дерево и выходим

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | $ | v a l g r i n d | −−l e a k −check=y e s . / a . out | 2> out . t x t |  |
| 2 | $ | c a t out . t x t | |  |  |
| 3 | ==166601== | | Memcheck , a memory e r r o r | d e t e c t o r |  |
| 4 | ==166601== Copyright (C) 2002 −2017 , | | | and GNU GPL’ d , | by J u l i a n Seward e t a l . |
| 5 | ==166601== | | Using Valgrind −3 . 1 6 . 1 and | LibVEX ; r e r u n | with −h f o r c o p y r i g h t i n f o |

* ==166601== Command : . / a . out

7 ==166601==

8 ==166601==

9 ==166601== HEAP SUMMARY:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | ==166601== | i n | us e a t | | e x i t : | | 122 ,9 | 5 2 b y t e s i n | 7 | b l o c k s |
| 11 | ==166601== | t o t a l | | heap | usage : | | 9 a l l o c s , 2 f r e e s , | | | 1 9 5 , 7 2 8 b y t e s a l l o c a t e d |
| 12 | ==166601== |  |  |  |  |  |  |  |  |  |
| 13 | ==166601== | 72 b y t e s | | i n 1 | b l o c k s | | a r e | d e f i n i t e l y | l o s t i n l o s s r e c o r d 1 o f 7 | |
| 14 | ==166601== | a t | 0x483ADEF : | | | o p e r a t o r | | new ( u n s i g n e d | | l o n g ) ( vg\_replace\_malloc . c : 3 4 2 ) |
| 15 | ==166601== | by | 0 x10A700 : | | | RBTree : : RBInsert ( s t d : : \_\_cxx11 : : b a s i c \_ s t r i n g <char , s t d : : c h a r \_ t r a | | | | |
| 16 | ==166601== | by | 0x10B5DA : | | | SRRBT : : Add ( ) ( i n /home/ k i p / f i l e s /tmp/ d a n i l a \_ i v a n o v /2/ a . out ) | | | | |
| 17 | ==166601== | by | 0x10A49C : | | | main | ( i n | /home/ k i p / f i l e s /tmp/ d a n i l a \_ i v a n o v /2/ a . out ) | | |

1. ==166601==
2. ==166601== LEAK SUMMARY:

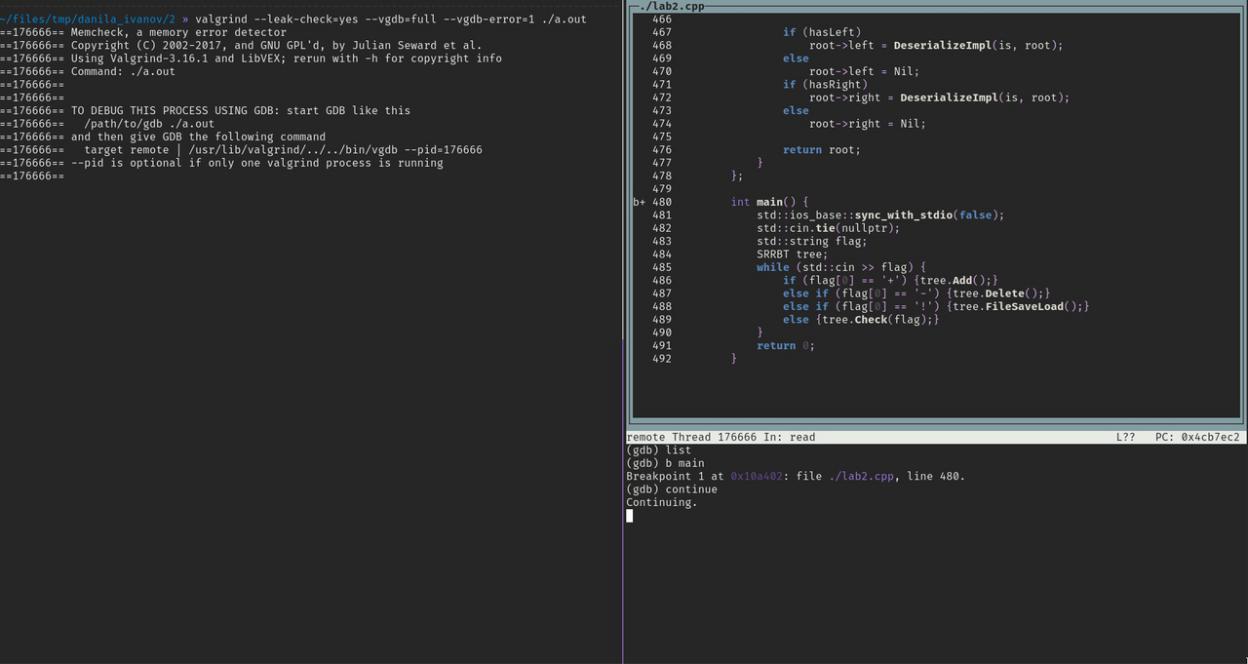
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 20 | ==166601== |  | d e f i n i t e l y | | l o s t : | 72 b y t e s | | i n | 1 | b l o c k s |  |  |
| 21 | ==166601== |  | i n d i r e c t l y | | l o s t : | 0 | b y t e s | i n | 0 | b l o c k s |  |  |
| 22 | ==166601== |  | p o s s i b l y | | l o s t : | 0 | b y t e s | i n | 0 | b l o c k s |  |  |
| 23 | ==166601== |  | s t i l l | r e a c h a b l e : | | 122 ,880 | | b y t e s | | i n 6 b l o c k s | |  |
| 24 | ==166601== |  | s u p p r e s s e d : | | | 0 | b y t e s | i n | 0 | b l o c k s |  |  |
| 25 | ==166601== | Reachable | | b l o c k s ( t h o s e t o which | | | | | | a p o i n t e r | was found ) | a r e not shown . |
| 26 | ==166601== | To | s e e them , | | r e r u n | with : −−l e a k −check= f u l l | | | | | −−show−l e a k −k i n d s=a l l | |
| 27 | ==166601== |  |  |  |  |  |  |  |  |  |  |  |
| 28 | ==166601== | For | l i s t s | o f | d e t e c t e d | | and s u p p r e s s e d e r r o r s , r e r u n with : −s | | | | | |
| 29 | ==166601== ERROR SUMMARY: 1 e r r o r s from | | | | | | | 1 | c o n t e x t s ( s u p p r e s s e d : | | | 0 from 0 ) |

Программа пишет что у нас 9 резервирований памяти и всего два освобождения, значит где-то мы не освобождаем память.

Далее все проблемы решаются аналогичным методом.

Раскладка окон при работе с gdb (размер окон можно варьировать по количеству информации)

2



Perf

Команды:

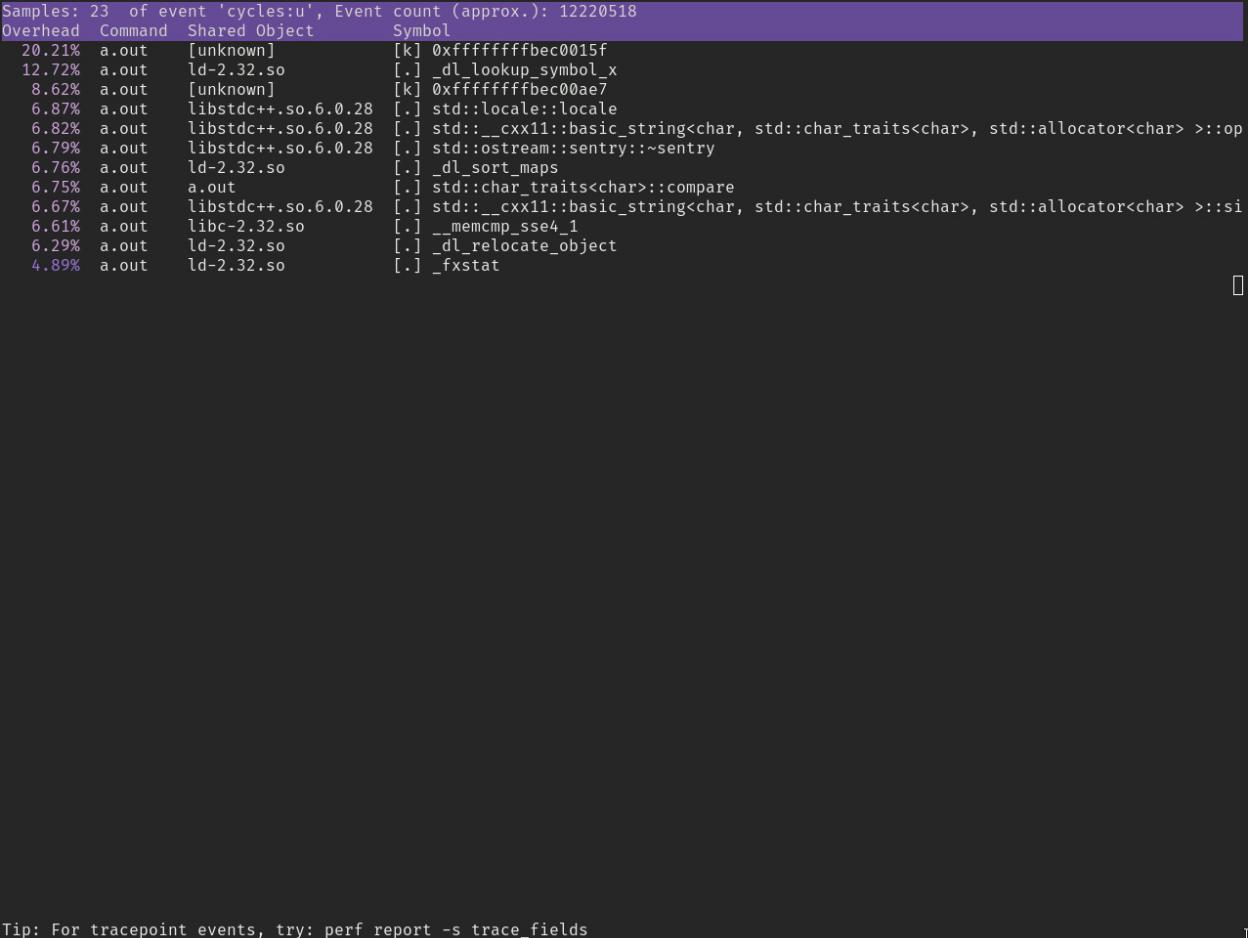
1. record <executable> <args\_to\_executable> - записываем результаты профили-ровщика в perf.data файл.
2. report - обработка perf.data файла и вывод информации в читаемом виде.

Ключи:

1. record -g - сохраняет call graph файл
2. report --hierarchy - вывод иерархически
3. record -e instructions:u - запись инструкций только на уровне пользователя Без добавления ключа -g получим такой вывод.

3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sampl es : | 23 of eve nt · cyc l es: u', Event | | cou nt (approx.) : 12220518 | |  |  |  |
| Overhead | Comma nd | Shared Ob j ect | Symbo1 | |  |  |  |
| 1 | a.OLJt | [L1пk1101m] | [k] | OxffffffffbecOO15f |  |  |  |
|  | a.oLJt | ld-2.32.50 | [.] \_d1\_1ook,1p\_sy,пbo1\_x | |  |  |  |
|  | d.OLJt | [LIПkllOl'ill] | [k] | OxffffffffbecOOae7 |  |  |  |
|  | a.0L1t | lilJ5tdc-+--+-,5o.6.0.28 | [.] | std::1oca1e::1oca1e |  |  |  |
|  | a.OLJt | lib5tdc-+--+-,5o.6.0.28 | [.] | std::cxxll: :basic\_strir1g<cl1ar, | std: :char\_traits<c(1ar>, | std: :allocator<cl1ar> | >: :ор |
|  | а. OLJt | lilJ5tdc++. 50. 6. О. 28 | [.] | 5td: :05tгеа111: :5епtгу: :~5е11tгу |  |  |  |
|  | а. OLJt | ld-2.32.50 | [.] \_dl\_5ort\_пiap5 | |  |  |  |
|  | а. OLJt | а. OLJt | [.] | 5td: :c(1aг\_trait5<cl1ar>: :co111pare |  |  |  |
|  | а. OLJt | libstdc-+--+-. 50. Ь. О. 28 | [.] | std: : \_\_ cxxll: :basic\_striпg<ct1ar, | 5td: :char\_traits<c(1ar>, | std: :allocator<ct1ar> | >: :si |
|  | д. OLJt | \il,c-2.32.so | [.] \_\_ 111еп1с111р\_55е4\_1 | |  |  |  |
|  | д, OLJt | \d-2.3).so | [.] \_d1\_re1ocate\_object | |  |  |  |
|  | а. OLJt | \d-2.3).so | [.] | fxstat |  |  |  |

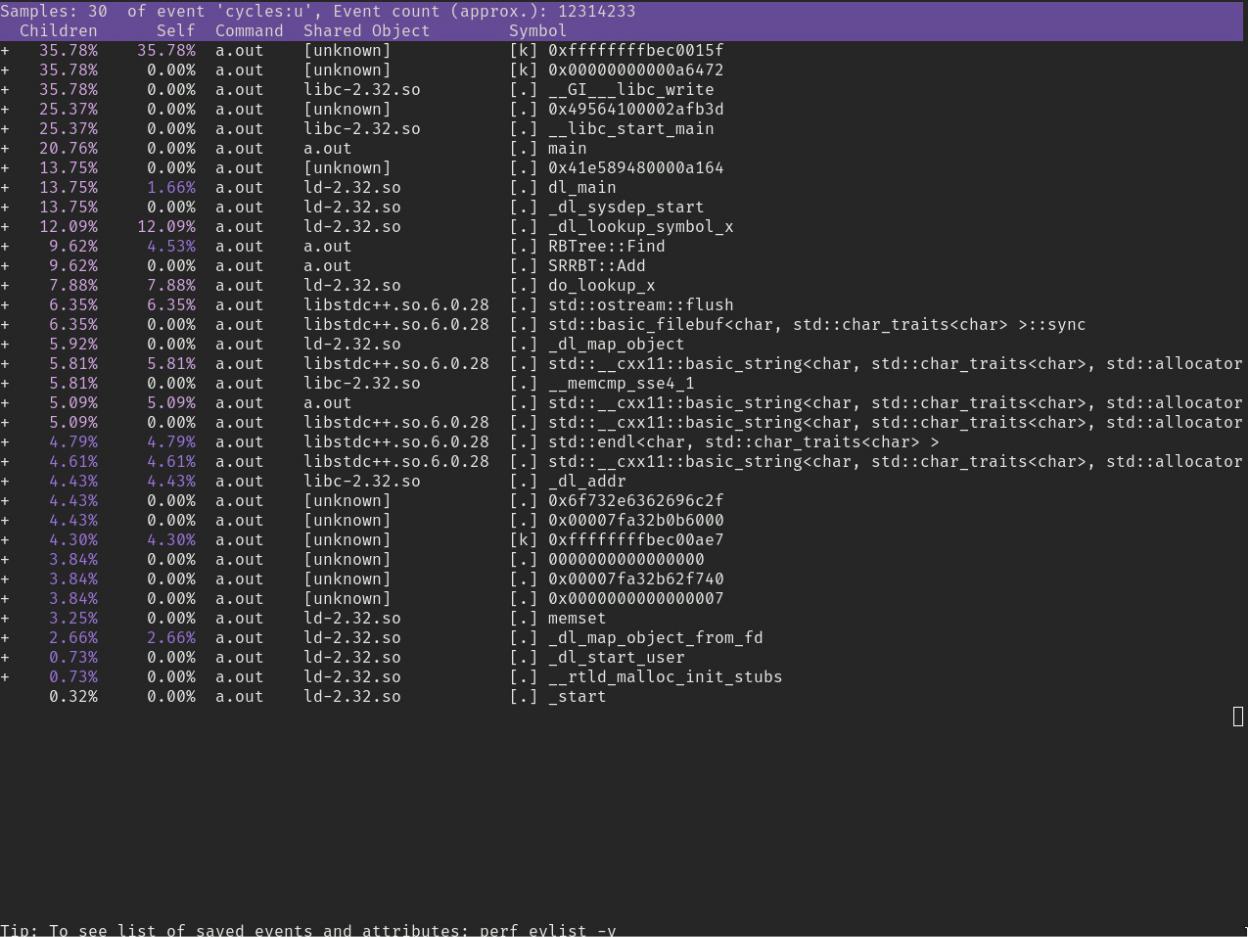


For tгасе oiпt eveпts tr erf ге огt -5 trace field5

Добавив ключ -g получим следующий вывод

4

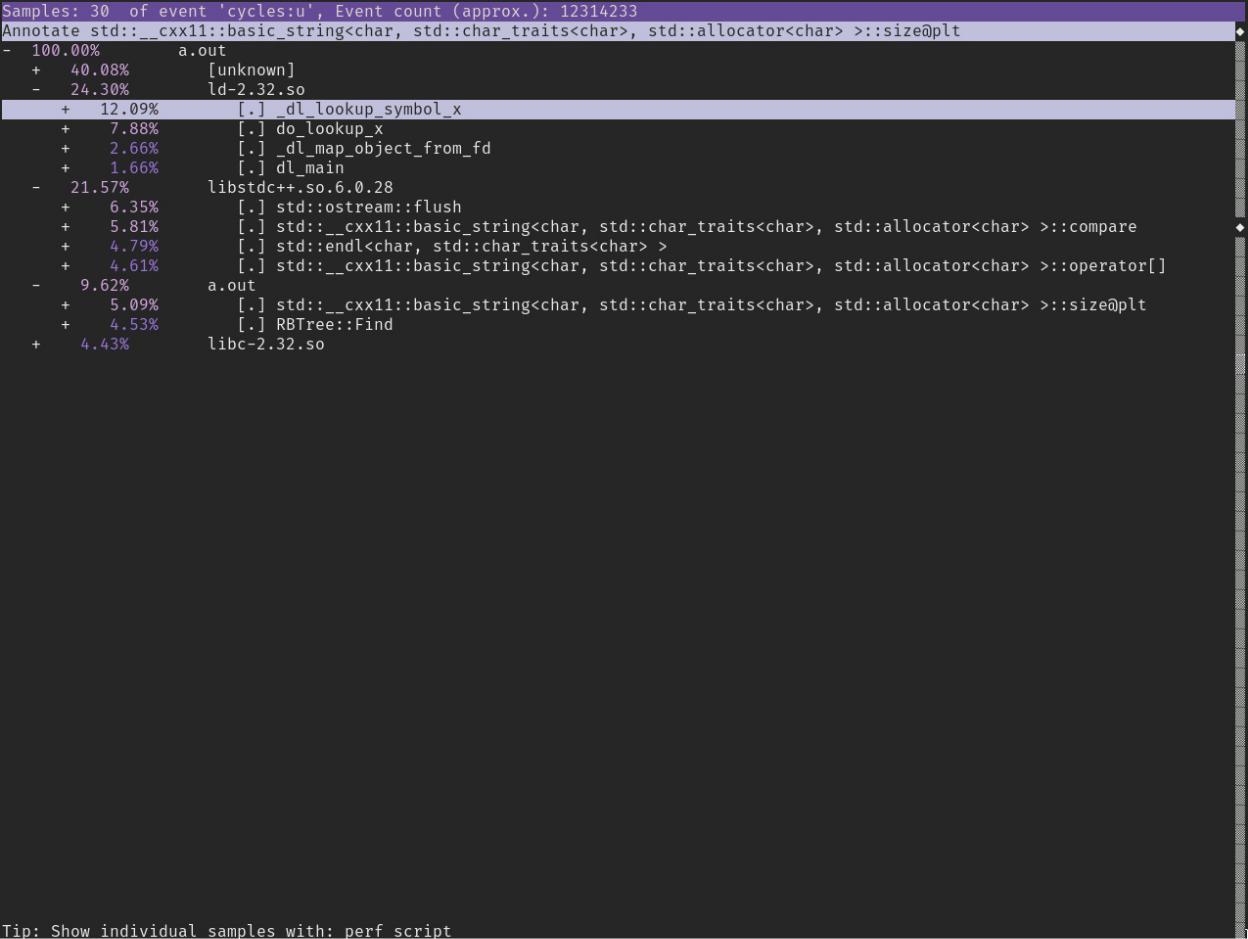
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sampl es: 30 | **of event** | **cycl es: u ',** Eveпt **coun t** (approx . ): | | | 12314233 | |  |  |  |
| Ch ildre n | Self | Comma nd | S hared Obj ect | Symbo l | |  |  |  |  |
|  |  | a .O ll t | [LJllkПOWll ] | [k] | Oxfff f ffffbec00 1 5f | |  |  |  |
|  | О. 00% | a.Oll t | [ LJП kпowп ] | [ k] | Ох0000000000Оа64 72 | |  |  |  |
|  | О. 00% | a.0 L1 t | 1 ibc - 2.32.so | [.] | GI | l ibc wr1te |  |  |  |
|  | О. 00% | а. Oll t | [ LJ пkпowп] | [.] | Ox49564100002afb3d | |  |  |  |
|  | О. 00% | а, Oll t | libc - 2.32.so | [.] | li bc | start п1а1~1 |  |  |  |
|  | О. 00% | а. Oll t | а. OLJ t | [ .] | 111аi п |  |  |  |  |
|  | О. 00% | а. O ll t | [ LJп kпowп ] | [.] | Ох4 1е589480000а164 | |  |  |  |
|  |  | а. Oll t | ld - 2.32.so | [.] | dl \_1пa111 |  |  |  |  |
|  | О. 00% | а. Otl t | l d - 2.32.so | [.] | \_ dl \_ sysdep\_ start | |  |  |  |
|  |  | d, Oll t | ld - 2.32.so | [.]d l\_look ,1p\_sy1пbol\_x | | |  |  |  |
|  |  | а. Oll t | а. OLJ t | [.] | RBTree::Fi пd | |  |  |  |
|  | О. 00% | d. Oll t | d. OLJ t | [.] SRRBT: : Add | | |  |  |  |
|  |  | а. Oll t | ld - 2.32.so | [.] | do \_l ook ll p\_ x | |  |  |  |
|  |  | а. Otl t | l ibstdc++.so . 6.0.28 | [ . ] | std: :ostream: : fl 11 sl1 | |  |  |  |
|  | О. 00% | d, Oll t | libstdc++.so.6.0.28 | [.] | std : : bas i c \_ fileb t1 f<c(1ar, | | std: :ct1ar\_ traits<c l1 ar> >: :sупс | |  |
|  | О. 00% | а. Oll t | l d - 2. 32. so | [ .] | dl \_ 111ap \_o l,ject | |  |  |  |
|  |  | d. Oll t | libstdc++.so.6.0.28 | *[.]* | std: : | cxxll::basic \_stri п g<cl1ar, | | std: с(1 а1 tra 1 ts<cl1ar>, | std::all ocato1· |
|  | О. 00% | а. Oll t | l i bc - 2.32.so | [ .) | \_\_ 1iн,111c111p \_sse4 1 | |  |  |  |
|  |  | а. Otl t | d . OlJ t | [ . ) | std:: | cxx l l: :basic \_stri п g<cllar, | | std: :c(l ar \_ traits<cll ar>, | std: :all ocator |
|  | О. 00% | d. OLl t | 1 ibstdc++. so. 6. О. 28 | [.] | std: : | cxxl l : : basic \_str iп g<cl1ar, | | std: : c(1ar \_ traits<c l1 ar>, | std: :all ocator |
|  |  | а. Oll t | l i bstdcн. so. 6. О. 28 | [.] | std : :e пdl <c(1ar, std::c llar | | tra i ts<c(1ar> > | |  |
|  |  | d. Oll t | libstdc++.so.6.0.28 | [.] | std: : | cxxll: :bas i c \_ str ir1 g<c l1 ar, | | std: :c(1a1· \_ tra i ts<c l1 ar>, | std: :all ocator |
|  |  | а. Oll t | li bc - 2.32.so | [ .] | d l addr | |  |  |  |
|  | О. 00% | d. Oll t | [ LJп kпowп ] | [ . ] Oxбf732e6362696c2f | | |  |  |  |
|  | О. 00% | d. OLl t | [ t111kпow11 ] | [.] | Ox00007fa32b0b6000 | |  |  |  |
|  |  | а. Oll t | [ LJ пkпowп] | [k] | Oxffffffffbec00ae7 | |  |  |  |
|  | О. 00% | а, Oll t | [ t1r1 k r1ow11 ] | [.] | 0000000000000000 | |  |  |  |
|  | О. 00% | а. Oll t | [LJllkПOl,'Jll] | [ .] | Ox00007fa32b62f740 | |  |  |  |
|  | О. 00% | а. O ll t | [ LJп kпowп ] | [ . ] Ох0000000000000007 | | |  |  |  |
|  | О. 00% | а. Oll t | ld - 2.32.so | [ .] | 11e111set |  |  |  |  |
|  |  | d, Oll t | ld - 2.32.so | [.] | \_ d l\_111ap \_object \_ fro11\_ fd | |  |  |  |
|  | О. 00% | а. Oll t | l d - 2.32.so | [ .] | \_ dl start LJser | |  |  |  |
|  | О. 00% | d. Oll t | ld - 2.32.so | *[.]* | rtld \_111al l oc \_ i r1 i t \_ stL1bs | |  |  |  |
| 0.32% | О. 00% | а. Oll t | l d - 2.32.so | [ .] | start |  |  |  |  |



Видим что в нём уже остлеживается какая функция загруженна своим контентом,

* какая по большей части загруженна внутренними функциями. Добавив к выводу ключ hierarchy получим:

5

Samples: 30 of event · cycles: u ·, Event count (approx.): 12314233

**nnotate std:: \_\_ cxxll::** bas1c\_stпn **<char, std:: char** \_ **traits<char>, std:: al locator<char>** >:: **sizeaJpl t**

д. Ollt

[ LJпkпowп]

ld-2. 32. so

* **12 .09%**[.] **\_dl\_lookup\_symbol \_x** [.] do\_ lookL1p\_x

[.] \_dl\_map\_object\_fro111\_fd

[.] dl\_111a i11

1ibstdc++.so.6.0.28

[.] std: :ostreaп1: :f1t1s(l

[.] std:: схх11: :basic\_strir1g<c(1a1·, std: :c(1ar\_t1·aits<cl1ar>, std: :allocator<ct1ar> >: :co111pare •

[.] std::eпdl<cl1ar, std::ct1дr\_tr<:1its<c(1aг> >

[.] std: · схх11: :basic\_striпg<c(1ar, std: :c(1ar\_traits<cl1ar>, std: :allocator<ct1ar> >: :opet·atoг[]

t1. OLJt

[.] std: · схх11: :basic\_striпg<c(1ar, std: :c(1ar\_traits<cl1ar>, std: :c1llou1tor<ct1ar> >: :sizeo)plt [.] RBTree::Fiпd

libc-2.32.so

Ti Sl10Y1 iпdividLJal sa111 les ~•1it(1: *erf* scri t

6

Выводы

Добавляя нужные ключи для valgrind’а в рабочую дерикторию можно значительно сократить количество печатаемых символов и повысить читаемость ошибок. Читая про valgrind, узнала много полезных ключей valgrind имеет много дополнительных модулей:

1. cachegrind - сбор статистики по попаданию в кэш первого и второго уровней процессора при выполнении операций чтения и записи данных и инструкций про-грамм, а также статистику по работе модуля предсказания ветвлений в програм-мах.
2. vallgrind - сбор информации о дереве вызова функций в программе.
3. massif - анализ выделения памяти.
4. helgrind - ошибки синхронизации.

Оказалось удобной команда (gdb) layout src. Также пригодились следующие "фишечки":

1. расставить breakpoint’ы в местах итерации критических местах
2. определить функцию в гдб, которая выполняет команду перемещения (next, continue, step)
3. один раз прописать вызов функции, потом жать enter до тех пор, пока не встре-тишь ошибку.

7