**Сравнительный анализ эффективности библиотек динамического выделения памяти**

И. Киреев

Научный руководитель – М. Г. Курносов, д.т.н., профессор кафедры ВС, СибГУТИ; старший научный сотрудник федерального государственного бюджетного учреждения науки Института физики полупроводников им. А.В. Ржанова Сибирского отделения Российской академии наук

Проведено экспериментальное исследование библиотек динамического выделения памяти на вычислительном устройстве с процессором Ryzen 5. Определена оптимальная библиотека. Эксперименты проведены на вычислительном устройстве с процессором Ryzen 5.

*Ключевые слова*: динамическая память, вычислительные системы.

1. Введение

Библиотеки динамического выделения памяти — инструменты для управления динамическим выделением памяти в соответствии с задачами приложения, в котором используются. Библиотеки динамического выделения памяти предоставляют эффективные алгоритмы контроля памяти, а также функции управления блоками памяти, её фрагментации для оптимизации производительности. Одними из наиболее распространённых библиотек динамического выделения памяти являются: *ptmalloc2*, *jemalloc*, *tcmalloc*.

Цель исследования — выполнить сравнительный анализ эффективности библиотек динамического выделения памяти: *ptmalloc2*, *jemalloc*, *tcmalloc*.

2. Библиотеки динамического выделения памяти

Основными пользовательскими функциями в библиотеках динамического выделения памяти являются: функция выделения (резервирования) памяти, функция освобождения памяти. В общем случае функция malloc выделения памяти резервирует блок памяти размера *n* и возвращает указатель на *void,* который указывает на начало блока зарезервированной памяти.

void \*malloc(size\_t n)

Основной функцией для очистки памяти является функция free*,* которая принимает указатель *ptr* на начало выделенной памяти и освобождает занимаемый блок

void free(void \*ptr)

Рассматриваемые библиотеки динамического выделения памяти имеют собственные механизмы контроля памяти. Каждая из исследуемых библиотек использует механизм выделения блоков памяти для каждого потока, что уменьшает вероятность возникновения конкуренции за общие ресурсы. Из рассматриваемых библиотек динамического выделения памяти, стандартной для большинства систем на базе Linux является ptmalloc2.

Jemalloc — аллокатор памяти, разработанный для использования в многопоточных приложениях. Jemalloc использует стратегию выделения памяти, основанную на «arena-based memory allocation», это означает то, что он выделяет память блоками, которые называются аренами. Каждый поток использует свою собственную арену для выделения памяти. Это позволяет избежать конкуренции за память между потоками и улучшить производительность.

Ptmalloc2 — аллокатор памяти, который используется в стандартной библиотеке GNU C для управления памятью в многопоточных приложениях. Он использует стратегию распределения памяти, основанную на «bins», где каждый обьект *bin* содержит блоки памяти фиксированного размера. Это уменьшает вероятность фрагментации памяти и улучшает производительность.

Tcmalloc — аллокатор памяти, разработанный Google для использования в многопоточных приложениях с высокой производительностью. Он также использует стратегию распределения памяти на основе арен, как и Jemalloc. Однако tcmalloc имеет несколько оптимизаций, которые делают его более эффективным, в том числе использование потоковых кэшей, предварительное выделение памяти и т. д.

3. Результаты экспериментов

Эксперименты выполнены на вычислительной машине, которая содержит процессор Ryzen 5 (6 ядер). Операционная система GNU/Linux. Размер страницы памяти – 4 Кбайт. Сравниваемые библиотеки: Jemalloc 5.3.0, Ptmalloc2, Tcmalloc 2.10, версия glibc – 2.35, версия ядра Linux – 5.19.0. Обьём свободной памяти определяется путём обращения к файлу, который имеет абсолютный путь /proc/vmstat, первая строка файла содержит название обьекта «nr\_free\_pages» и количество свободных страниц памяти. В качестве теста производительности использованы функции выделения и освобождения памяти для каждой библиотеки, а так же функция

void \*memset(void \*ptr, int value, size\_t num)

, которая принимает указатель *ptr,* который указывает на начало памяти, количество *num* элементов которой необходимо установить в значение *value*.

На рис. 1 показано время выполнения операции выделения 512 Мбайт памяти в зависимости от количества потоков, а также количество тактов процессора в зависимости от количества потоков. Библиотека tcmalloc показывает худшую производительность, так как время выполнения операции динамического выделения памяти из библиотеки tcmalloc линейно зависит от числа *p* потоков. Операция выделения памяти из библиотеки jemalloc показывает наименьшее время выполнения для выделения памяти.

Рекомендуется использовать библиотеку jemalloc в многопоточном режиме, так как она показывает лучшее время, по сравнению с tcmalloc и ptmalloc2.

|  |  |
| --- | --- |
| а) | б) |

Рис. 1. Время выполнения и количество тактов процессора операций выделения памяти: а) время; б) количество тактов.

На рис. 2 показано количество свободной памяти в Мбайтах при выделении памяти размером 512 Мбайт блоками по 1 Кбайт, а также количество свободной памяти в Мбайтах при выделении памяти размером 512 Мбайт блоками по 4 Кбайта. На эксперименте выделения памяти блоками размера 4 Кбайт tcmalloc показывает более стабильные результаты по сравнению с jemalloc и ptmalloc2. На эксперименте выделения памяти блоками размера 1 Кбайт различие в библиотеках незначительное.

Для размеров блоков 4 Кбайт рекомендуется использовать библиотеку tcmalloc, так как она показывает лучшую стабильность в выделении памяти блоками в сравнении с jemalloc и ptmalloc2.

|  |  |
| --- | --- |
| а) | б) |

Рис. 1. Количество свободной памяти при блочном выделении 512 Мбайт памяти с размерами блоков: а) 4 Кбайт; б) 1 Кбайт.

**4. Заключение**

Проведено экспериментальное исследование библиотек динамического выделения памяти Jemalloc 5.3.0, Ptmalloc 2, Tcmalloc 2.10 на вычислительном устройстве с процессором Ryzen 5. На тестах времени выполнения операции выделения памяти показано, что рекомендуется использовать библиотеку jemalloc, так как она показывает лучшие результаты в сравнении с ptmalloc2 и tcmalloc. Также, на тестах потребления памяти при её блочном выделении, где размер блока равен 4 Кбайт, рекомендуется использовать библиотеку Tcmalloc.

**Литература**

1. libc Standard // URL: https://www.gnu.org/software/libc/manual/html\_node/ The-GNU-Allocator.html (дата обращения: 20.03.2022).
2. Jemalloc // URL: https://jemalloc.net (дата обращения: 20.03.2022).
3. Jemalloc experiments // URL: https://github.com/jemalloc/jemalloc-experiments (дата обращения 20.03.2022)
4. Tcmalloc // URL: https://github.com/google/tcmalloc/blob/master/README.md (дата обращения 20.03.2022)
5. Testing memory allocators: ptmalloc2 vs tcmalloc vs hoard vs jemalloc while trying to simulate real-world loads // URL: http://ithare.com/testing-memory-allocators-ptmalloc2-tcmalloc-hoard-jemalloc-while-trying-to-simulate-real-world-loads (дата обращения 20.03.2022)

**Киреев Илья Анатольевич**

студент группы ИВ-121 СибГУТИ (630102, Новосибирск, ул. Кирова, 86), тел. (383) 269-83-82, e-mail: [ilya\_kireev\_2004@bk.ru](mailto:ilya_kireev_2004@bk.ru).

**Курносов Михаил Георгиевич**

д.т.н., профессор, профессор кафедры вычислительных систем Сибирского государственного университета телекоммуникаций и информатики (630102, г. Новосибирск, ул. Кирова, 86), тел. (383) 269-83-82, e-mail: [mkurnosov@sibguti.ru](mailto:mkurnosov@sibguti.ru); старший научный сотрудник федерального государственного бюджетного учреждения науки Института физики полупроводников им. А.В. Ржанова Сибирского отделения Российской академии наук (630090, г. Новосибиpск, пp. Ак. Лавpентьева, 13), тел. (383) 330-56-26, e-mail: mkurnosov@isp.nsc.ru.

**Comparative analysis of the efficiency of dynamic memory allocation libraries**

**I. Kireyev, M. Kurnosov**

An experimental study was conducted on a computing device with a Ryzen 5 processor to evaluate dynamic memory allocation libraries. The optimal library was identified. The experiments carried out on a computing device with a Ryzen 5 processor.

*Keywords*: dynamic memory, computing systems.