Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «**Информатика и системы управления**»

КАФЕДРА «**Программное обеспечение ЭВМ и информационные технологии**»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

НА ТЕМУ:

Разработка метода сжатия оперативной памяти в ядре Linux

Студент группы **ИУ7-83Б**		**А. В. Романов**
	(Подпись, дата)	(И.О. Фамилия)
Руководитель ВКР		**А. А. Оленев**
	(Подпись, дата)	(И.О. Фамилия)
Нормоконтролер		**Ю. В. Строганов**
	(Подпись, дата)	(И.О. Фамилия)

***2022** г.*

РЕФЕРАТ

Расчетно-пояснительная записка 20 с., 0 рис., 0 табл., X ист., X прил.

КЛЮЧЕВЫЕ СЛОВА

СОДЕРЖАНИЕ

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ВВЕДЕНИЕ

В данной научно-исследовательской работе должны быть проанализированы <Linux, zRam и оперативная память>.

1 Аналитическая часть

В данном разделе <...>.

1.1 Предпосылки

В последнее десятилетие центральное процессорное устройство – ЦПУ (англ. central processing unit – CPU [1]) достигли своей пиковой тактовой частоты – около 5 Ггц, и этот предел будет преодолён ещё не скоро [2]. Из-за того что ЦПУ стали настолько быстрыми, становится нередка ситуация, когда процессор не выполняет инструкции, а ждёт, пока данные переместятся с диска в оперативное запоминающее устройство – ОЗУ (англ. random access memory – RAM [3]) (или наоборот) [4]. Так, например, скорость работы системы с мощным ЦПУ, но с маленьким количеством ОЗУ может быть маленькой – несмотря на быстроту, ЦПУ будет часто будет ожидать подсистему ввода/вывода.

1.2 Способы увеличения количества оперативной памяти

Проблему объема оперативной памяти компьютера можно решить увеличив количество планок ОЗУ. Но, у данного подхода есть свои минусы:

- электроэнергия – каждая новая установленная планка увеличивает потребление энергии компьютера [5];
- физические ограничения – количество слотов для планок на материнской плате ограничено;
- стоимость – планка стоит денег.

Альтернативным решением является использование системы подкачки страниц (англ. paging [6]) – специальный механизм операционный системы, при котором отдельные фрагменты памяти (мало используемые или полностью не активные) перемещаются из оперативной памяти во вторичной хранилище, на-

пример, на жёсткий диск или другой внешний накопитель. Несмотря на то что количество ОЗУ в таком случае увеличивается, доступ к таким участкам памяти замедляется – системе приходится работать (читать и писать) с внешним запоминающим устройством, а такие устройства работают значительно медленнее ОЗУ [7].

Ещё одним решением является сжатие данных прямо в оперативной памяти. Данный подход выигрывает у первого рассмотренного решения – он реализуется программно и не требует закупки дополнительных планок ОЗУ. Так же, этот подход выигрывает и у второго рассмотренного решения – сжатие не требует работы с внешним накопителем, за счёт чего скорость обработки данных увеличивается. Но сжатие данных так же имеет свои минусы, главным из которых является потребность в вычислительных ресурсах – ЦПУ должен выполнять какую-то работу. Но, как было описано в разделе 1.1, обычно ЦПУ как раз простаивает и не выполняет никакой работы, ожидая ввод/вывод.

1.3 Сжатие данных

Сжатие основано на устранение избыточности, которая содержится в исходных данных. Часто повторяющиеся фрагменты данных заменяются другими более короткими фрагментами. Так, например, набор символов `abc def abc jhf abc` можно преобразовать в `x def x jhf x`, произведя замену символов `abc` на символ `x`.

Все методы сжатия данных делятся на два класса: без потерь и с потерями. Сжатие данных без потерь позволяет полностью восстановить сжатые данные, в отличии от сжатия с потерями. Первый вариант чаще всего используют для сжатия текстовых данных и компьютерных программ, а сжатие с потерями используют для сокращения аудио- или видеоданных – для таких данных не требуется полное соответствие исходным данным. Алгоритмы сжатия данных с потерями обладают большей эффективностью, чем алгоритмы сжатия без по-

терь [9].

<тут написать про степень сжатия>

1.3.1 Требования к алгоритмам сжатия

К алгоритмам сжатия могут применяться несколько требований:

- скорость сжатия;
- скорость декомпрессии;
- степень сжатия;
- эффективность программной реализации.

Для каждой конкретной задачи будут важны одни требования и менее важны другие. Так, например, для сжатия больших видеоданных с их последующим хранением, важнее будет степень сжатия данных, а не скорость работы алгоритма.

1.4 Сжатие памяти в ядре Linux

Ядро Linux [8] - ядро операционной системы с открытым исходным кодом, распространяющееся как свободное программное обеспечение. Именно из-за этого в данной **<научной-исследовательской надо писать или нет?>** работе будет рассмотрено сжатие оперативной памяти в ядре Linux.

Для сжатия данных в оперативной памяти, ядро берёт последовательность байтов в памяти, сжимает их, записывает сжатую версию обратно в ОЗУ и хранит их до тех пор, пока эти данные не потребуются системе. Пока данные находятся в сжатом состоянии, система не может прочитать или записать какие-либо отдельные байты из этой сжатой последовательности. Когда данные потребуется системе вновь, система распаковывает сжатую последовательность.

Современные алгоритмы могут сжимать любое количество последовательных байт. Несмотря на это, в ядре удобно использовать фиксированную

единицу сжатия памяти. Единицей хранения в ядре была выбрана страница памяти (англ. memory page [10]) – фиксированная константа `PAGE_SIZE`, которая на большинстве современных архитектур, поддерживаемых Linux, составляет 4 Кб [11].

Для достижения высокой степени сжатия требуется выполнение большого количества команд ЦПУ, тогда как менее эффективное сжатие может выполняться быстрее. В ядре необходимо добиться баланса между временем и степенью сжатия. Кроме того, важно чтобы выбор алгоритма оставался гибким. Например для выполнения одной задачи подойдёт один алгоритм сжатия, а для второй другой.

Из-за того что размер страницы достаточно большой (4 Кб), сжатие и распаковка данных – достаточно дорогостоящие операции, поэтому необходимо ограничить количество этих операций. Нужно тщательно выбирать, какие страницы стоит сжимать, а какие нет. Алгоритм, реализованный в ядре Linux, определяющий какие страницы нужно сжимать, выбирает те страницы, которые вероятно будут использоваться снова, но вряд ли будут использоваться в ближайшем будущем [4]. Такая реализация позволяет не тратить всё время ЦПУ на многократное сжатие и распаковку страниц. Кроме того, необходимо чтобы ядро могло идентифицировать сжатую страницу – иначе её невозможно найти и распаковать.

Размер страницы, к которой был применён алгоритм сжатия зависит от данных на исходной странице и его трудно предсказать. Степень сжатия страницы описывается следующей формулой (1):

$$compression\ ratio = \frac{PAGE\_SIZE}{zsize} \tag{1}$$

где `PAGE_SIZE` размер страницы памяти в системе, а `zsize` – размер сжатой страницы.

Обычно размер сжатой страницы меньше чем `PAGE_SIZE`, поэтому, обыч-

но, коэффициент сжатия меньше единицы. Но, в некоторых случаях, коэффициент сжатия может быть больше единицы, и для решения этой проблемы необходимо, чтобы алгоритм реализованный в ядре мог найти выход из такой ситуации.

1.4.1 Модуль ядра zram

2 Конструкторская часть

3 Технологическая часть

4 Исследовательская часть

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. What is CPU (Central Processing Unit)? - Computer Hope [Электронный ресурс]. – Режим доступа: https://www.computerhope.com/jargon/c/cpu.htm, свободный – (10.11.2021)

2. Why Aren’t CPUs Getting Faster? - Apple Gazette [Электронный ресурс]. – Режим доступа: https://applegazette.com/mac/why-arent-cpus-getting-faster/, свободный – (10.11.2021)

3. What Does Computer Memory (RAM) Do? [Электронный ресурс]. – Режим доступа: https://www.crucial.com/articles/about-memory/support-what-does-computer-memory-do, свободный – (10.11.2021)

4. In-kernel memory compression [Электронный ресурс]. – Режим доступа: https://lwn.net/Articles/545244/, свободный – (10.11.2021)

5. What are the negative effects on increasing RAM of a PC? [Электронный ресурс]. – Режим доступа: https://www.quora.com/What-are-the-negative-effects-on-increasing-RAM-of-a-PC, свободный – (10.11.2021)

6. Paging | Android Developers [Электронный ресурс]. – Режим доступа: https://developer.android.com/jetpack/androidx/releases/paging, свободный – (10.11.2021)

7. SSD vs. HDD | Speed, Capacity, Performance & Lifespan | AVG [Электронный ресурс]. – Режим доступа: https://www.avg.com/en/signal/ssd-hdd-which-is-best, свободный – (10.11.2021)

8. What is Linux? [Электронный ресурс]. – Режим доступа: https://www.linux.com/what-is-linux/, свободный – (10.11.2021)

9. Industrial Control Technology. Peng Zhang, 2008. с. 675 - 774.

10. Memory Management — The Linux Kernel documentation [Электронный ресурс]. – Режим доступа: https://www.kernel.org/doc/html/latest/admin-guide/mm/index.html, свободный – (10.11.2021)

11. Using 4KB Page Size for Virtual Memory is Obsolete [Электронный ресурс]. – Режим доступа: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5211562, свободный – (10.11.2021)

ПРИЛОЖЕНИЕ А