

Учебные материалы для 2 курса

[Главная страница](#)

[Занятия по языку Си](#)

[Занятия по](#)

[программированию в](#)

[Unix](#)

[Материалы для](#)

[дополнительного чтения](#)

[Конспекты занятий по](#)

[языку Си++](#)

[Опции компилятора на сервере \(C\)](#)

[Опции компилятора на сервере \(C++\)](#)

[Стиль форматирования программ](#)

[О вещественных числах](#)

[О рекурсивном спуске](#)

[О написании Makefile](#)

[Задание на](#)

[моделирование кеша](#)

[Документация по STL](#)

[Ссылки](#)

[Памятка по работе в Unix](#)

[Лекции по ОС для КазФ МГУ](#)

Задание: реализация модели системы кеширования ОЗУ

В данном задании необходимо реализовать программу, моделирующую работу кеш-памяти процессора (далее симулятор кеша). Модель кеша должна поддерживать следующие стратегии кеширования:

- полноассоциативный кеш и кеш прямого отображения;
- стратегии замещения: случайная ;
- стратегии записи: сквозная, отложенная.

Параметры моделируемого кеша задаются в конфигурационном файле. В аргументах командной строки задаются опции для симулятора кеша и имя конфигурационного файла. Трасса обращений к памяти поступает на стандартном потоке ввода.

Симулятор кеша должен вывести на стандартный поток вывода результаты моделирования в соответствии с опциями, переданными ему в командной строке.

Параметры командной строки

Скомпилированная программа симулятора кеша должна называться `cachesim`. Должны поддерживаться опции: `--dump-memory`, `--statistics`, `--disable-cache`, `--print-config`.

Если указана опция `--print-config`, симулятор должен вывести на стандартный поток вывода параметры, заданные в конфигурационном файле, и завершить работу с кодом завершения 0 (см. задачу `c01`).

Если указана опция `--dump-memory`, симулятор должен распечатать содержимое памяти после выполнения трассы.

Если указана опция `--statistics`, должна выводиться статистика работы кеш-памяти.

Если указана опция `--disable-cache`, кеширование отключается, и модель кеширования не создается.

Если указано несколько параметров командной строки, результаты работы выводятся в порядке `--dump-memory`, `--statistics`. Обязательным параметром командной строки является путь к конфигурационному файлу. Примеры командной строки:

```
cachesim --dump-memory trace.cfg
cachesim --dump-memory --statistics trace.cfg
cachesim trace.cfg
```

Конфигурационный файл

Конфигурационный файл — это текстовый файл. В текстовом файле из байтов с кодами, меньшими ' ' допускаются только байты \r, \n, \t. Длина строки конфигурационного файла не превышает 1000 байт (с учетом символов-завершителей строки \r, \n). Комментарии в файле начинаются с символа '#' и идут до конца строки. После отбрасывания комментариев пробельные символы в конце строки игнорируются. Пустые строки в конфигурационном файле игнорируются.

Непустые строки в конфигурационном файле имеют следующий вид:

NAME=VALUE

Перед NAME, между NAME и =, = и VALUE может находиться произвольное количество пробельных символов. NAME — это непустая последовательность заглавных и строчных латинских букв, цифр и знаков '_' или '-', которая начинается с буквы или '_'. VALUE — это произвольная (возможно пустая) последовательность символов (включая пробел) до конца строки.

Файл трассы

Файл трассы — текстовый файл. В текстовом файле из байтов с кодами, меньшими ' ' допускаются только байты \r, \n, \t. Длина строки файла трассы не превышает 1000 байт (с учетом символов-завершителей строки \r, \n). Комментарии в файле начинаются с символа '#' и идут до конца строки. После отбрасывания комментариев пробельные символы в конце строки игнорируются. Пустые строки в файле игнорируются.

Непустые строки в файле определяют трассу либо в кратком, либо в полном формате.

Строка трассы в кратком формате имеет вид:

OM ADDR

Где O — это операция — символ 'R' или 'W', M — тип памяти — символ 'D' или 'T'. ADDR — это адрес в памяти, задаваемый в шестнадцатеричном виде и представимый как неотрицательное число в 32-битном целом типе.

Строка трассы в полном формате имеет вид:

OM ADDR SIZE VALUE

Где дополнительно SIZE — это размер операции с памятью (1, 2,

4 или 8), VALUE – знаковое целое число (в десятичном виде) соответствующего размера.

Если для выполнения моделирования требуется трасса в полном формате, но на вход подается трасса в кратком формате, размер данных предполагается равным 1 байту, а значение — 0.

Оперативная память

Размер оперативной памяти задается параметром `memory_size`. Размер оперативной памяти не больше гигабайта и выражается целым количеством килобайт. Например,

```
memory_size = 65536
```

задает размер памяти 64 килобайта.

Параметр `memory_read_time` определяет количество тактов, за которое выполняется чтение из ОЗУ. Параметр `memory_write_time` определяет количество тактов, за которое выполняется запись в ОЗУ. Параметр `memory_width` определяет максимальное количество байт, пересылаемых в/из ОЗУ за одну операцию и за указанное время. Если из памяти считывается/в память записывается больше байт, чем значение параметра `memory_width`, то чтение/запись будет требовать пропорционально больше времени.

Начальное значение памяти неизвестно. Целые числа хранятся в памяти в порядке `big-endian`.

Кеш

Параметр `cache_size` задает общий размер кеш-памяти в байтах. Параметр `block_size` задает размер одного блока в байтах и может принимать значения 16, 32, 64. Общий размер кеша кратен размеру одного блока. Параметр `associativity` задает тип ассоциативности кеша и может принимать значения `full` для полноассоциативного кеша и `direct` для кеша прямого отображения.

Для полноассоциативного кеша параметр `replacement_strategy` определяет стратегию замещения и может принимать значения `random` для случайного замещения, `lfu` для стратегии `least frequently used`, `lru` для стратегии `least recently used`. Стратегия `lru` заключается в замещении блока кеша, находящегося на последнем месте в списке занятых блоков.

При использовании стратегии `lfu` должны обрабатываться следующие конфигурационные параметры: `lfu_count_size` – размер счетчика использований (в битах), счетчик работает с насыщением; `lfu_init_value` – начальное значение счетчика использований при загрузке блока из ОЗУ; `lfu_aging_interval` – число операций чтения, после которого выполняется операция

«старения»; `lfu_aging_shift` – количество бит, на которые сдвигается вправо значение счетчика при выполнении операции «старения». При замещении наименее редко используемого блока кеша из блоков с минимальным значением счетчика использований выбирается случайное.

Параметр `write_strategy` задает стратегию записи. Параметр может принимать два значения: `write-through` для сквозной записи и `write-back` для отложенной записи. В случае отложенной записи в ОЗУ записывается блок целиком.

Параметр `cache_read_time` задает время выполнения чтения из кеша. Это время добавляется каждый раз при выполнении операции чтения из кеша и в случае попадания, и в случае промаха. В случае промаха, однако, добавляется еще и время чтения из нижележащей памяти.

Параметр `cache_write_time` задает время выполнения записи в кеш. Это время добавляется каждый раз при выполнении операции записи. В случае сквозной записи отдельно добавляется время записи в память. В случае отложенной записи время записи в память добавляется при выталкивании грязного блока из кеша.

Для кеша со сквозной записью если при записи ячейка не находится в кеше, то блок не загружается из памяти в кеш. Для кеша с отложенной записью если при записи ячейка не находится в кеше, то блок загружается из памяти в кеш (с учетом времени загрузки из памяти).

Случайные числа

Для получения псевдослучайных чисел должны использоваться функции `srand` и `rand` стандартной библиотеки. Параметр `seed` конфигурационного файла задает начальное значение для генератора псевдослучайных чисел. Для получения целого псевдослучайного числа, равномерно распределенного на множестве $\{0, 1, \dots, N - 1\}$ должна использоваться формула

$$r = (\text{int}) (\text{rand}() / (\text{RAND_MAX} + 1.0) * N);$$

Псевдослучайные числа, равномерно распределенные на других множествах получаются отображением из псевдослучайных чисел на указанном выше множестве.

Вывод дампа памяти

Если в командной строке указана опция `--dump-memory`, на стандартный поток вывода должен быть напечатан дамп памяти в формате, показанном ниже.

Первый столбец (8 шестнадцатеричных цифр - это адрес). Далее идут 16 столбцов - байты в шестнадцатеричном виде. Если значение байта не известно, выводится `??`. В качестве

шестнадцатеричных цифр используйте заглавные буквы. Столбцы разделяются одним символом пробела.

```
00000000 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
00000010 00 00 00 C8 FF F0 ?? ?? ?? ?? ?? ?? ?? ?? ??
00000020 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
00000030 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
```

Вывод статистики выполнения трассы

Если в командной строке указана опция `--statistics`, на стандартный поток вывода должна быть напечатана статистика выполнения трассы как показано ниже:

```
clock count: CLOCK_COUNT
reads: READS
writes: WRITES
read hits: READ_HITS
cache block writes: CBW
```

Где `CLOCK_COUNT` – общее количество тактов, которые потребовались для выполнения операций с памятью в данной трассе. `READS` – общее количество операций чтения в данной трассе. `WRITES` – общее количество операций записи в данной трассе. `READ_HITS` – количество попаданий в кеш при выполнении операций чтения (только если кеш включен). `CBW` – количество операций записи блоков кеша в ОЗУ (только если кеш включен и для write-back кеша).

Организация исходных файлов

Исходные файлы проекта должны находиться в каталоге `cachesim`. Архив этого каталога в формате `.tar.gz` сдается на проверку.

Следование [стилю оформления программ](#) обязательно.

В каталоге должен находиться Makefile, оформленный в соответствии с требованиями, изложенными в [методическом пособии](#).

Общая структура проекта и заготовки некоторых файлов доступны в виде [документации](#) (сгенерировано Doxygen) и [исходных файлов](#).

Выбор между всеми возможными стратегиями реализации кеша (ассоциативность, стратегия замещения, стратегия записи) должен происходить на этапе создания объекта кеша в функциях `cache_create_*` с помощью подстановок соответствующих указателей на функции в структуру операций. В реализациях операций не должны присутствовать условные операторы и выражения, проверяющие стратегии реализации кеша. Для этого может потребоваться определить дополнительную структуру `CacheOps` с указателями на функции, которые настраиваются в зависимости от стратегии реализации кеша.

Главный файл должен называться cachesim.c.

Сообщения об ошибках

Все сообщения об ошибках выводятся на стандартный поток ошибок. Программа прекращает выполнение после обнаружения первой ошибки.

В случае, если аргументы командной строки не удовлетворяют спецификации должно выводиться сообщение

`Invalid arguments`

Ошибки разбора конфигурационного файла

В случае, если конфигурационный файл не может быть открыт на чтение должна выводиться следующая ошибка. FILE заменяется на имя файла, указанное в командной строке.

`Failed to open FILE for reading`

В случае, если в строке LINE конфигурационного файла FILE была обнаружена ошибка разбора файла должно выводиться следующее сообщение. Строки в файле нумеруются с 1.

`Syntax error in line LINE of FILE`

В случае, если конфигурационный параметр NAME уже определен должно выводиться следующее сообщение

`Duplicate parameter NAME in FILE`

Last modified: Friday, 21-Jun-2013 16:47:51 MSK

[Alexander Chernov](#)