

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего
профессионального образования «Московский физико-технический институт
(государственный университет)»
Факультет радиотехники и кибернетики
Кафедра «Микропроцессорные технологии»

На правах рукописи
УДК **TODO 645.687.5**

Фамилия Имя Отчество

Название работы

Выпускная работа **TODO бакалавра/магистра**
Направление подготовки: **TODO 010900 Прикладные математика и физика**

Заведующий кафедрой _____ / _____/

Научный руководитель _____ / _____/

Студент _____ / _____/

г. Москва
2015

Содержание

1. Введение	3
1.1. Симуляторы вычислительных машин	3
1.2. Актуальность тематики	3
1.3. Структура работы	3
2. Обзор литературы	5
3. Реализация	6
4. Представление графической информации	7
5. Постановка эксперимента	8
6. Результаты	9
6.1. Обсуждение результатов	9
7. Печать	10
Список литературы	11
Приложения	12
А. Блок-схема алгоритма compile_capsules	13

Список иллюстраций

1. Суть работы	6
--------------------------	---

Список таблиц

1. Скорость симуляции для различных фаз экспериментов с Amber	9
---	---

1. Введение

Разумный человек
приспосабливается к миру;
неразумный — упорно пытается
приспособить мир к себе.
Поэтому прогресс зависит от
неразумных людей.

(Джордж Бернард Шоу)

Используйте данный файл как основу для написания своего диплома. Различные советы по структуре, оформлению и представлению даны далее.

1.1. Симуляторы вычислительных машин

В последнее время значительный интерес компаний¹, занимающихся разработкой аппаратного обеспечения для вычислительных комплексов, вызывает задача разработки и верификации систем на программных симуляторах, ещё до выхода серийных и даже экспериментальных образцов аппаратуры. Подобный подход позволяет уменьшить время, затрачиваемое на разработку, стоимость процесса, и сократить промежуток между моментами выхода нового процессора и готовности программного обеспечения для него. Однако для эффективности данного подхода скорость симуляции должна быть достаточно высока, а сама симуляция — точна в рамках решаемой задачи. Для достижения этих целей используются различные подходы построения симуляторов.

1.2. Актуальность тематики

Проблема моделирования ЭВМ на ЭВМ не нова — вопросы о возможности получения «машины в машине» возникали с начала эры вычислительной техники [2].

1.3. Структура работы

В данной работе приводятся и решаются следующие задачи:

- Анализ существующего кодогенератора, выявление недостатков в предыдущей его реализации.
- Разработка спецификации механизма двоичной трансляции нового поколения.
- **TODO Напиши меня**
- PROFIT!

Секция 1 обосновывает важность симуляционных решений в задачах вычислительной техники.

¹Таких как Intel, IBM, Samsung и др.

В секции 2 описываются подходы и существующие решения.

В секции 3 приводится информация о деталях реализации.

Секция 5 определяет методику проведения экспериментов, осуществлённых на разработанной системе.

Секция 6 суммирует результаты всей работы.

2. Обзор литературы

Для обозначения размеров задачи и интереса исследовательского сообщества к вопросам как эмуляции, так и бинарной трансляции, а также трудностей, возникающих перед разработчиками полноплатформенного симулятора, приведём здесь обзор подходов, применяемых на практике.

Симулятор SoftSDV [1].

3. Реализация

Сущность работы проиллюстрирована на рис. 1.

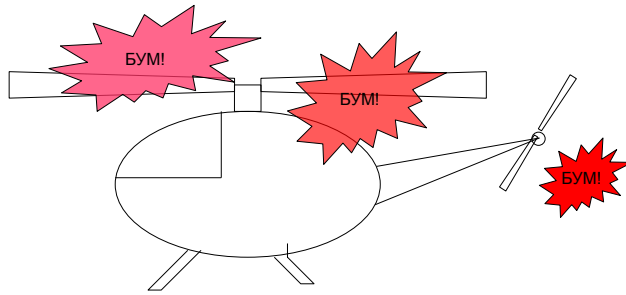


Рис. 1: Суть работы

Исходные данные взяты на Github <https://github.com/torvalds/linux>.

4. Представление графической информации

Иллюстрации и графики могут быть выполнены как средствами самого ЛАТ_EX(с подключением пакетов TIKZ и PGFPLOTS), так и внешними программами.

Для создания иллюстраций (картинок) внешними программами рекомендуются следующие пакеты.

1. Microsoft Visio <http://office.microsoft.com/en-us/visio/>.
2. LibreOffice Draw <http://libreoffice.org/> или Apache OpenOffice <http://www.openoffice.org/>.
3. Dia <http://dia-installer.de/>

Для создания графиков внешними программами рекомендуются следующие пакеты.

1. GNUPLOT www.gnuplot.info/.
2. Matplotlib <http://matplotlib.org/>.
3. QTILOT soft.proindependent.com/qtiplot.html .
4. Origin <http://www.originlab.com/>.
5. Matlab, Mathcad, Maple и другие математические пакеты.
6. Microsoft Excel (не рекомендуется). Пожалейте своё время, этот инструмент — для экономистов.

Настоятельно не рекомендуется использовать Microsoft PowerPoint и Microsoft Paint.

Поддерживаемые ЛАТ_EXом форматы изображений: PDF (рекомендуется), PNG, JPEG (не рекомендуется).

5. Постановка эксперимента

Для каждого теста выполнялись следующие действия:

1. В гостевой ОС запускался скрипт. Основной задачей его являлось исполнение команды `runspec` со следующими опциями: один прогон теста, нерепортируемый запуск¹, размер данных теста – тренировочный, конфигурация и имя теста – соответствующие запуску. Обрамляющие её команды `ssc_mark` – точки прерывания и возвращения управления в эмулятор.
2. После первого перехода в отладчик в среде отладчика запускался Perl-скрипт. Его задачи: включить кэш, затем каждые 100 миллионов инструкций проверять, не закончил ли `runspec` работу и не были ли выключены кэши; если это так, то вывести финальную статистику и вернуть систему в изначальное состояние.
3. Во время выполнения гостя методы класса `Hash` изменяли значения переменных-счётчиков, а также проверяли условия локального максимума значения α_{load} и выводили статистику.
4. Сообщения о значениях параметров выводились в отдельный файл журнала, а также на консоль. После завершения всех тестов файлы журналов отдельных запусков были пропущены через дополнительные скрипты для агрегации необходимых данных (они приведены в приложении А).

Основная формула для анализа данных:

$$FLOPS_{async}(N) = N \min(FLOPS_1, FLOPS_2, FLOPS_3)$$

¹То есть указание того, что его результаты не будут использованы для «официального» сравнения производительности каких-либо компиляторов или аппаратуры. Использование этого режима позволяет смягчить требования пакета и запускать тесты по отдельности, а не только весь пакет целиком.

6. Результаты

Результаты эксперимента Ω даны в таблице 1.

Таблица 1: Скорость симуляции для различных фаз экспериментов с Amber

Фаза	Длит., сим. с	Скорость, MIPS	Комментарий
Загрузка ОС	200	5500	Прямое исполнение
Кэши	200+200	60	Подключены иерархии
Сбор трасс MPI	200	140	Подключен <code>mpi-tracer</code>

При решении задачи создания новой системы капсул для бинарного транслятора были достигнуты все цели, поставленные при формулировании работы.

1. Анализ старой версии системы капсул показал её недостатки.
2. Выявлены причины найденных проблем: стремление к быстрой разработке языка капсул привело к крайне негибкому и неудобному в поддержке решению.
3. Предложено решение, свободное от вышеупомянутых проблем.
4. Это решение реализовано и протестировано.
5. Показано, что точность симуляции при переходе на новые капсулы не уменьшилась.

6.1. Обсуждение результатов

Воплощение описанных выше принципов позволило добиться гибкости записи капсул и избавиться от недостатков, присущих предыдущей их версии. Кроме того, был упразднён сложный многоступенчатый процесс сборки капсул благодаря объединению процесса в один скрипт.

7. Печать

Диплом следует печатать на бумаге А4, односторонняя печать. Минимум одна копия диплома должна быть подшита (или переплетена), подписана на титульном листе всеми обозначенными на нём лицами и представлена комиссии в день защиты.

Список литературы

- [1] Uhlig R., Fishtein R., Gershon O., Hirsh I., Wang H. SoftSDV: A Presilicon Software Development Environment for the IA-64 Architecture // Intel Technology Journal. — 1999. — P. 112 — 126.
- [2] Gerald J. Popek, Robert P. Goldberg. Formal requirements for virtualizable third generation architectures. Communications of the ACM Volume 17 , Issue 7 (July 1974)

Приложения

A. Блок-схема алгоритма `compile_capsules`

Приложения к диплому опциональны и могут содержать дополнительную справочную информацию, не обязательную для понимания основного содержания: текст программы, спецификация языка, особенности кодировки и и т.п.

