РОССИЙСКАЯ АКАДЕМИЯ НАУК

Федеральное государственное бюджетное учреждение науки
Институт системного программирования
Российской академии наук

«УТВЕРЖДАЮ»

Директор ИСП РАН
академик РАН,
д.ф.-м.н., профессор
_____ В.П.Иванников
«___»_____ 2012 г.

РАБОЧАЯ ПРОГРАММА

УЧЕБНОЙ ДИСЦИПЛИНЫ
«Динамическая компиляция»

для подготовки аспирантов по специальности
05.13.11 - Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

Москва 2012

1. АННОТАЦИЯ

Современные «облачные» технологии организации вычислений невозможны без виртуализации системного программного обеспечения, так как это основной способ быстрой настройки «облачного» сервера, выделенного для выполнения очередной программы, на контекст, который требуется для правильного выполнения этой программы. В результате виртуализации бинарный код программы пользователя фактически выполняется на интерпретаторе, что может существенно увеличить время выполнения. Динамическая компиляция бинарного кода такой программы позволяет исключить из интерпретации большую часть фактически выполняемого кода программы и тем самым существенно ускорить ее интерпретацию (примерно на два порядка). Кроме того, в процессе динамической компиляции интерпретируемая программа может быть оптимизирована, что еще больше ускорит ее интерпретацию.
Динамическая компиляция применяется не только к бинарным программам, но и к программам, написанным на таких языках высокого уровня, как Java, C# и др. В последнее время делаются успешные попытки применить динамическую компиляцию к скриптовым языкам (например, Java Script).

Методы динамической компиляции используются в ИСП РАН при выполнении ряда научно-исследовательских и коммерческих проектов. В частности во многих проектах активно используются такие свободно распространяемые среды как ***QEMU*** и ***Valgrind***.

К сожалению, изучение динамической компиляции и оптимизации не входит в учебные планы большей части университетов, готовящих специалистов по системному и прикладному программированию. Настоящий курс восполняет этот пробел. В нем изучаются современные проблемы динамической компиляции и методы их решения. Слушатели получат представление о таких средах, использующих динамическую компиляцию и оптимизацию как ***QEMU***, ***Valgrind***, ***Dynamo***.

2. ЦЕЛИ И ЗАДАЧИ КУРСА

Цель курса – изучение современных методов бинарной трансляции и динамической и адаптивной оптимизации аспирантами первого года обучения, специализирующимися по проблематике компиляторных технологий, методам обеспечения безопасности системного и прикладного ПО (как на уровне исходного кода, так и на уровне бинарного исполняемого кода).

Задачами данного курса являются:

- освоение студентами базовых знаний в области бинарной трансляции и динамической и адаптивной оптимизации программ;
- приобретение теоретических знаний в области теории графов, теории решеток, методов сбора статистики, используемых при разработке методов анализа и трансформации программ;
- оказание консультаций и помощи аспирантам в проведении собственных исследований и разработок в областях, использующих подходы, рассматриваемые в курсе.

3. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ПОСЛЕВУЗОВСКОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ (АСПИРАНТУРА))

Дисциплина «Динамическая компиляция» относится к дисциплинам по выбору учебного плана подготовки аспирантов по научной специальности 05.13.11 «Математическое обеспечение вычислительных машин, комплексов и компьютерных сетей».

Для успешного изучения курса аспиранту необходимо знать общесистемное программное обеспечение современных компьютеров и основы компиляторных технологий, а также уметь работать с персональной ЭВМ.

Основные положения дисциплины будут использованы при подготовке к кандидатскому экзамену по научной специальности 05.13.11 «Математическое обеспечение вычислительных машин, комплексов и компьютерных сетей», в научно-исследовательской работе и при выполнении диссертации на соискание ученой степени кандидата физико-математических наук.

4. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ СОДЕРЖАНИЯ ДИСЦИПЛИНЫ

В результате освоения дисциплины «Динамическая компиляция» обучающийся должен:

Знать:

- фундаментальные понятия современных компиляторных технологий;
- структуру и состав современных оптимизирующих компиляторных сред (примеры – GCC, LLVM и др.);
- цели, задачи и методы машинно-ориентированной динамической оптимизации программ в процессе их выполнения и компиляции;
- принципы применения бинарных трансляторов для ускорения эмуляции бинарных программ;
- принципы применения методов и средств динамической и адаптивной оптимизации программ на языках высокого уровня;

Уметь:

- разрабатывать, обосновывать и реализовывать новые методы и алгоритмы бинарной трансляции в виртуальных вычислительных средах;
- разрабатывать, обосновывать и реализовывать новые методы и алгоритмы динамической и адаптивной оптимизации программ;
- применять компиляторные методы и компиляторные среды для решения задач обратной инженерии, защиты программного кода, обнаружения дефектов в программах и др.;

Владеть:

- навыками самостоятельной работы в Интернете;
- культурой разработки и реализации системного программного обеспечения современных компьютеров;
- навыками грамотной разработки новых языков программирования и их программного обеспечения;

5. Содержание и структура курса

5.1 Содержание разделов курса

№	Наименование раздела	Содержание раздела	Форма текущего контроля
1	Введение	Виртуальные системы и их применение. Проблемы, рассматриваемые в данном курсе.	Т
2	Статическая и динамическая бинарная трансляция	Бинарная трансляция как средство оптимизации эмуляторов. Структура статического бинарного транслятора. Динамическая бинарная трансляция. Промежуточное представление программы, используемое в бинарных трансляторах. Генерация промежуточного представления. Оптимизация кода при его бинарной трансляции. Сбор профиля. Нумерация значений. Распределение регистров. Планирование кода. Использование динамической бинарной трансляции в мнгоплатформенном эмуляторе ***QEMU*** и среде ***Valgrind***.	Т
3	Динамическая компиляция интерпретируемых языков высокого уровня	Промежуточное представление программы для интерпретатора (байт-код, бит-код и т.п.) и генерация такого представления. Инструментирование программ и генерация профиля. Методы генерации профиля программы. Семплирование. Средства профилирования среды ***Java***. Профильно-ориентированная оптимизация программ. Выбор фрагментов для оптимизации. Связывание оптимизированных фрагментов. Кэширование оптимизированных фрагментов.	Т
4	Методы оптимизации кода, применяемые в динамических компиляторах	Методы динамической и адаптивной оптимизации. Перевод функции в квазиоптимальную SSA-форму. Выполнение профильно-ориентированных оптимизаций: открытая вставка функций, предварительная обработка виртуальных вызовов, оптимизация базовых блоков, распределение регистров, ускорение часто вызываемых функций, выявление и оптимизация наиболее часто выполняемых ветвей, отделение мертвого кода, оптимизация вызовов функций работы с кучей. Недостатки профильно-ориентированной оптимизации. Эволюционная компиляция и	Т

		многоэтапная оптимизация. Замещение на стеке. Выделение кода обработки исключительных ситуаций. Деоптимизация.	
5	Динамические компиляторы среды ***Java*** (***JIT***)	Динамические компиляторы (***JIT***) среды Java: ***HotSpot*** (SUN) - пример промышленного ***JIT***, ***Jikes*** (IBM) – пример экспериментального ***JIT***. Применение методов динамической оптимизации в экспериментальном ***JIT Jikes***. Взаимодействие ***Jikes*** и ***JavaVM*** при динамической компиляции и выполнении ***Java***-программы.	Т
6	Динамическая компиляция и оптимизация ***C/C++***-кода в среде ***LLVM***	Динамическая компиляция в среде ***LLVM***. Промежуточное представление ***LLVM*** (биткод). Построение профилей в среде ***LLVM***. Динамическая компиляция и оптимизация ***C/C++***-кода. Применение динамической компиляции ***C/C++***-кода для его аутсорсинга. Краткий обзор особенностей языка ***Java Script*** и его интерпретатора, влияющих на динамическую компиляцию. Особенности реализации динамической компиляции в ***JIT***-компиляторе языка ***Java Script***.	Т
7	Заключение	Динамическая компиляция, учитывающая требования пользователя.	Т

5.2 Структура курса

Общая трудоемкость курса составляет 2,5 зачетные единицы (90 часов).

Вид работы	Трудоемкость, часов
	2 курс
Общая трудоемкость	**90**
Аудиторная работа:	**32**
Лекции (Л)	32
Практические занятия (ПЗ)	-
Лабораторные работы (ЛР)	-
Самостоятельная работа:	**58**
Самостоятельное изучение разделов	-
Самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий, выполнение практических заданий)	58
Вид итогового контроля (зачет, экзамен)	**Кандидатский экзамен**

Трудоемкость отдельных разделов курса.

№	Наименование разделов	Количество часов				
		Всего	Аудиторная работа			Вне-ауд. работа СР
			Л	ПЗ	ЛР	
1	Введение	4	2	-	-	2
2	Статическая и динамическая бинарная трансляция	16	6	-	-	10
3	Динамическая компиляция интерпретируемых языков высокого уровня	14	4	-	-	10
4	Методы оптимизации кода, применяемые в динамических компиляторах	22	8	-	-	14
5	Динамические компиляторы среды *Java* (*JIT*)	20	8			12
6	Динамическая компиляция и оптимизация *C/C++*-кода в среде *LLVM*	8	2			6
7	Заключение	6	2			4
	Итого:	90	32	-	-	58

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы аспирантов

Форма контроля знаний:
- кандидатский экзамен по специальности.

Контрольно-измерительные материалы

На кандидатском экзамене аспирант должен продемонстрировать знания в объеме основной программы кандидатского экзамена по специальности 05.13.11 «Математическое обеспечение вычислительных машин, комплексов и компьютерных сетей», а также дополнительной программы, в которую, в зависимости от выбранной аспирантом специализации, могут входить вопросы, рассматриваемые в данном курсе.

Перечень контрольных вопросов для дополнительной программы:

1. Виртуальные системы программирования и их применение.

2. Статическая и динамическая бинарная трансляция. Промежуточные представления бинарного кода в современных бинарных трансляторах.

3. Современные методы профилирования бинарного кода. Способы ускорения методов профилирования.

4. Архитектура и возможности многоплатформенного эмулятора ***QEMU***. Методы профилирования, реализованные в ***QEMU***.

5. Эмуляция конкретных архитектур с помощью ***QEMU*** на примере ARM.

6. Расширение системы ***QEMU*** на примере добавления новых оптимизаций кода, получаемого в результате бинарной трансляции.

7. Динамическая компиляция интерпретируемых языков высокого уровня. Промежуточное представление программы для интерпретатора (байт-код, бит-код и т.п.) и его генерация.

8. Инструментирование программ и генерация профилей. Методы генерации профиля программы. Семплирование.

9. Средства профилирования среды ***Java***.

10. Профильно-ориентированная оптимизация программ. Выбор фрагментов для оптимизации. Связывание оптимизированных фрагментов. Кэширование оптимизированных фрагментов

11. Алгоритм построения квазиоптимальной SSA-формы функции в промежуточном представлении.

12. Профильно-ориентированные оптимизации: открытая вставка функций, предварительная обработка виртуальных вызовов.

13. Профильно-ориентированные оптимизации: оптимизация базовых блоков, перераспределение регистров, ускорение часто вызываемых функций.

14. Профильно-ориентированные оптимизации: выявление и оптимизация наиболее часто выполняемых ветвей, удаление мертвого кода.

15. Профильно-ориентированные оптимизации: оптимизация вызовов функций работы с кучей.

16. Эволюционная компиляция и многоэтапная оптимизация. Замещение на стеке.

17. Выделение кода обработки исключительных ситуаций. Деоптимизация.

18. Динамические компиляторы (***JIT***) среды Java. Применение методов динамической оптимизации в экспериментальном ***JIT Jikes***.

19. Взаимодействие ***Jikes*** и ***JavaVM*** при динамической компиляции и выполнении ***Java***-программы.

20. Динамическая компиляция в среде ***LLVM***. Промежуточное представление ***LLVM*** (биткод). Построение профилей в среде ***LLVM***.

21. Динамическая компиляция и оптимизация ***C/C++***-кода в среде ***LLVM***. Применение динамической компиляции ***C/C++***-кода для обеспечения его аутсорсинга.

22. Особенности реализации динамической компиляции в ***JIT***-компиляторе языка ***Java Script***.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ КУРСА

7.1 Рекомендуемая литература

7.1.1 Основная литература

1. Evelyn Duesterwald. Dynamic Compilation. (21 pages). Глава 10 книги The Compiler Design Handbook, Edited by Y.N. Srikant, and Priti Shankar. CRC Press Taylor & Francis Group, 2008, ISBN 978-1-4200-4382-2. (21 pages).
2. R. L. Sites, A. Chernoff, M. B. Kirk, M. P. Marks, and S. G. Robinson. Binary translation. // Communication of the ACM, 1993 http://130.203.133.150/viewdoc/summary?doi=10.1.1.225.7422
3. M. D. Smith. Overcoming the Challenges to Feedback-Directed Optimization. // Proceedings of the ACM SIGPLAN Workshop on Dynamic and Adaptive Compilation and Optimization (Dynamo'00), Boston, January 18, 2000. http://dl.acm.org/ft_gateway.cfm?id=351408.
4. Navneet Aron, Sorav Bansal. Dynamic Compilation, 2003. www.stanford.edu/class/ee392c/.../lec13.pdf
5. John Aycock. A Brief History of Just-In-Time. // ACM Computing Surveys, Vol. 35, No. 2, June 2003, pp. 97–113 http://web.csie.cgu.edu.tw/~jhchen/course/PL2/A%20brief%20history%20of%20just-in-time.pdf

7.1.2 Дополнительная литература

1. ***Valgrind*** User Manual. http://valgrind.org/docs/manual/manual.html
2. ***QEMU*** Manual http://wiki.qemu.org/Manual
3. M. Probst. Fast Machine-Adaptable Dynamic Binary Translation // UKUUG Linux Developers' Conference 4 - 7 July 2002 Bristol http://130.203.133.150/viewdoc/similar;jsessionid=C5ED62754161FDF03AEDAF294BBF7E7E?doi=10.1.1.100.160&type=ab
4. Evelyn Duesterwald et al. ***Dynamo***: A Transparent Dynamic Optimization System// Trans. Of ACM Conf. PLDI pp. 1-12, ACM, 2000 *http://personals.ac.upc.edu/vmoya/docs/bala.pdf*
5. Dries Buytaert, Andy Georges, Michael Hind, Matthew Arnold, Lieven Eeckhout, Koen De Bosschere. Using HPM-Sampling to Drive Dynamic Compilation. // Proc. OOPSLA'07, October 21–25, 2007, Montreal, Quebec, Canada

7.1.3. Пособия и методические указания.

1. Слайды лекций

8. Материально-техническое обеспечение курса

Для получения необходимой информации и самостоятельной работы аспирантов используются web-ресурсы Интернет и локальная библиотека электронных материалов.

В компьютерных классах ИСП РАН (ауд. 109) аспиранты могут самостоятельно ознакомиться с программным обеспечением, используемым для динамической компиляции программ.

Программу составил к.ф.–м.н., доцент Гайсарян С.С.

Программа принята на заседании Ученого Совета ИСП РАН

протокол № 2012-5 от 23 мая 2012 г.