

Национальный исследовательский университет

Высшая школа экономики

Концепция
Магистерской программы
«Системное программирование»

НАПРАВЛЕНИЕ 09.04.04

«Программная инженерия»

Степень (квалификация): магистр

Москва 2018

1. Общая характеристика

Магистерская программа «Системное программирование» предполагает ее освоение в течение двух учебных лет, трудоемкость составляет 120 зачетных единиц. Форма обучения очная. Прием ведется как на бюджетные места, так и на места с оплатой стоимости обучения.

Магистерская программа «Системное программирование» сочетает в себе академический подход к изучению предметной области с практической направленностью: ее основу составляет научное исследование принципов создания и функционирования программных средств, имеющих непосредственное практическое применение.

Преподавание ведется на русском языке с использованием зарубежных источников (преимущественно англоязычных, английский язык *de facto* является языком международного общения программистов).

По успешном окончании магистерской программы выдается диплом государственного образца РФ.

Академическим руководителем магистерской программы является профессор кафедр Системного программирования НИУ ВШЭ и ВМиК МГУ А.К.Петренко, д.ф.-м.н., профессор. (email: petrenko@ispras.ru).

В разработке магистерской программы «Системное программирование» принимали участие:

- В.П.Иванников (1940 - 2016), академик РАН, научный руководитель Института Системного Программирования РАН (ИСП РАН) (1994-2016);
- А.К.Петренко, профессор базовой кафедры Системного программирования НИУ ВШЭ и кафедры Системного программирования ВМиК МГУ, заведующий отделом технологий программирования ИСП РАН, д.ф.-м.н., профессор (email: petrenko@ispras.ru);
- С.М.Авдошин, руководитель Департамента Программной инженерии факультета Компьютерных наук НИУ ВШЭ, к.т.н., професор (email: savdishin@hse.ru);
- Е.М.Гринкруг, профессор Департамента Программной инженерии факультета Компьютерных наук НИУ ВШЭ, зам. заведующего базовой кафедрой Системного программирования, к.т.н.(email: egrinkrug@hse.ru).

2. Актуальность, цели и задачи магистерской программы

Системное программирование – важная часть компьютерных наук, направленная на исследование и разработку программных средств производства и поддержки функционирования программного обеспечения. Системное программирование является основой индустрии программного обеспечения, обеспечивающей производство средств производства программ. Все современное программное обеспечение создано и функционирует благодаря научным и практическим результатам, достигнутым в области системного программирования.

Современное производство программного обеспечения различного назначения дает в экономику ряда развитых и развивающихся стран основной вклад, превосходящий по объему вклад в экономику от других отраслей. При этом общие расходы на разработку и поддержку функционирования (сопровождение) программного обеспечения составляют значительную часть ВВП. Программное обеспечение становится основой современных достижений во всех областях науки, промышленности и обороны. Системное программирование при этом является основой создания любого программного обеспечения.

В настоящее время в нашей стране в развитии системного программирования наблюдается спад, вызванный доминированием зарубежных программных продуктов. Это обусловлено рядом причин, среди которых – острая нехватка квалифицированных специалистов в этой, наиболее наукоемкой, области программирования, требующей высокой квалификации и «высокого порога вхождения» в проблематику. В результате, появляется риск потери конкурентоспособности и самостоятельного развития в этой важнейшей области компьютерных наук. Вместе с тем, общеизвестно, что специалисты, получившие программистское образование в нашей стране, вносят заметный на мировом уровне вклад в развитие компьютерных наук и технологий.

Целью магистерской программы «Системное программирование» Департамента Программной инженерии Факультета компьютерных наук (ФКН) НИУ ВШЭ является создание предпосылок для исправления этого положения. Магистерская программа «Системное программирование» организуется базовой кафедрой Системного программирования, которая открыта при департаменте программной инженерии ФКН с сентября 2015 года Институтом системного программирования РАН (ИСП РАН) под руководством академика РАН В.П.Иванникова. В настоящее время В.П.Иванников является научным руководителем ИСП РАН и осуществляет руководство всей научной и научно-педагогической деятельностью ИСП РАН, в том числе – организацией данной магистерской программы. ИСП РАН – главная научно-исследовательская организация в области системного программирования в России, широко известная и уважаемая во всем мире, где не прерывалась и сохранилась научная школа в области Системного программирования. Одной из целей магистерской программы является участие студентов магистратуры НИУ ВШЭ в ее развитии и продолжении.

Магистерская программа «Системное программирование» предполагает обучение студентов-магистрантов (прежде всего тех, кто успешно овладел основными разделами Системного программирования в бакалавриате при базовой кафедре Системное программирование) с участием сотрудников базовой кафедры Системное программирование Департамента Программной инженерии ФКН НИУ ВШЭ и сотрудников ИСП РАН.

Целью магистерской программы является подготовка специалистов по системному программированию, способных проводить исследования и разработки в области системного программирования на современном мировом уровне. В задачи данной магистерской программы входит предоставление и использование образовательных и научно-исследовательских возможностей ИСП РАН для достижения этой цели.

3. Целевая аудитория магистерской программы

Основной целевой аудиторией магистерской программы Системное программирование являются выпускники бакалавриата базовой кафедры Системное программирование Департамента программной инженерии ФКН НИУ ВШЭ и выпускники кафедр Системного программирования других университетов.

Помимо выпускников кафедр Системного программирования магистерская программа направлена на выпускников смежных специальностей (прикладная математика и информатика, теоретическая информатика, технологии программирования), которые хотят приобрести знания и опыт в области системного программирования.

Величина предполагаемого потока студентов составляет ориентировочно 15 - 20 чел./год, в зависимости от результатов вступительного отбора. Целью программы является подготовка специалистов элитного, мирового уровня; при таких целях качество подготовки имеет безусловный приоритет над количеством обучающихся.

Целью магистерской программы является предоставление возможности углубленного изучения современного системного программирования тем, кто сознательно выбрал эту специализацию и намерен стать специалистом мирового уровня. Для этого в ИСП РАН предусмотрен и выверен путь становления высоко-квалифицированных научных работников – от бакалавриата до аспирантуры (включительно), частью которого является данная магистерская программа.

4. Международный и отечественный опыт

Программирование принято разделять на прикладное программирование и системное программирование. Прикладное программирование имеет целью разработку программных продуктов, непосредственно используемых пользователями для различных целей. Системное программирование является производством средств производства программных продуктов. Это – основа индустриализации программирования в целом, определяющая уровень развития компьютерной отрасли.

Обычно под проблематикой системного программирования понимают проблематику создания операционных систем и систем программирования, к которым относятся языки программирования, компиляторы и другие средства создания и сопровождения программного обеспечения. Иногда также к проблематике системного программирования относят вопросы создания больших программных комплексов, отличающихся повышенной сложностью и важностью, при разработке которых требуется отделять принципы организации вычислений от решения конкретных функциональных задач. При разработке сложных программных комплексов тоже происходит разграничение средств организации функционирования программ и собственно их выполнения (т.е. средств производства программного продукта от его конкретного функционального наполнения). Как и в любом производстве, разработка средств производства является определяющим фактором.

Лучшие университеты мира имеют факультеты компьютерных наук, где готовят специалистов в разных областях программирования, в том числе – и прежде всего - в области системного программирования. В их числе: Stanford University, MIT, University of California Berkeley, Princeton University, Carnegie Mellon University, Cornell University (USA), Toronto University (Canada), Oxford (England), Technion (Israel), ETH (Switzerland) и другие. Сотрудники и выпускники этих университетов известны своим вкладом в развитие компьютерных наук.

Во всем мире ощущается нехватка квалифицированных специалистов в области системного программирования – основополагающей области программирования с «высоким порогом вхождения». По данным из исследований рынка труда в области IT спрос на специалистов по Системному программированию удовлетворяется на 20 – 30% во всем мире. Особую актуальность эта проблема приобретает в связи с импортозамещением.

Институт Системного Программирования РАН – наиболее авторитетная научно-исследовательская организация в области системного программирования в России, имеет обширные международные связи и признана за рубежом и в нашей стране как лидер в научных исследованиях в области программирования. Научные издания ИСП РАН по признаны во всем мире как самые авторитетные издания в России по вопросам программирования (журнал Программирование, Труды ИСП РАН). С участием ИСП РАН в ведущих университетах – МГУ, МФТИ и НИУ ВШЭ – уже организованы базовые кафедры системного программирования; в ИСП РАН накоплен опыт выращивания специалистов в области системного программирования мирового уровня – от бакалавриата до аспирантуры включительно. Открытие магистерской программы Системное программирование при базовой кафедре ИСП РАН в НИУ ВШЭ является естественным шагом в этом направлении.

5. Характеристика сегмента рынка образовательных услуг

В нашей стране подготовкой квалифицированных специалистов в области Системного программирования занимаются несколько ведущих университетов и факультетов: ВМиК МГУ, МФТИ, СПбПУ, ИТМО, имеются смежные специализации в ряде других университетов.

В НИУ ВШЭ на ФКН имеется магистерская программа департамента Программной инженерии со специализациями «Управление разработкой программного обеспечения» и «Разработка мобильных приложений». Эти специализации имеют определенную связь с Системным программированием – именно в том смысле, что системное программирование, как указывалось, является основой всей программной отрасли вообще (как производство средств производства для нее). Однако, уровень, объем изучения и исследования вопросов системного программирования на указанных специализациях нельзя сопоставить с предлагаемой программой – они направлены на производство пользовательских приложений (их менеджеральных и технологических аспекты, соответственно).

Для формирования квалифицированного системного программиста как ученого-исследователя необходима научная школа и участие в современных разработках мирового

уровня. В ИСП РАН имеются для этого все условия, и специалисты по Системному программированию, выпускаемые ВМиК МГУ и МФТИ, проходят подготовку именно на соответствующих базовых кафедрах ИСП РАН. Их программы подготовки не являются конкурентами рассматриваемой магистерской программы (они скорее являются партнерами), так как преподавательские составы базовых кафедр пересекаются, а студенты занимаются исследованиями совместно (на базе ИСП РАН).

Преимуществом данной магистерской программы по сравнению со сходными программами ведущих университетов страны (исключая, возможно, партнеров по базовым кафедрам ИСП РАН) является наличие предварительной подготовки к магистерской программе при имеющейся в НИУ ВШЭ базовой кафедре ИСП РАН на двух старших курсах бакалавриата, а также естественная возможность продолжения исследований в аспирантуре ИСП РАН по окончании магистерской программы. Эти преимущества являются следствием апробированной образовательной политики ИСП РАН в области системного программирования и опыта ее реализации.

6. Востребованность и трудоустройство выпускников

Общеизвестно, что квалифицированные отечественные программисты легко находят себе трудоустройство в нашей стране и за рубежом, в том числе – в самых передовых компаниях и исследовательских центрах.

Квалифицированный системный программист относится к программистской элите, так как способен решать наиболее сложные и важные задачи программирования. Диплом магистратуры по Системному программированию является свидетельством освоения материала с «высоким порогом вхождения» и показателем такой квалификации. Как указывалось, спрос на таких специалистов удовлетворяется всего на 20 – 30% повсеместно, и крупные компании вынуждены сами заниматься переподготовкой специалистов смежных специальностей.

Успешный выпускник магистерской программы «Системное программирование» имеет реальные возможности трудоустройства в ведущих ИТ-компаниях и возможности продолжения сотрудничества с ИСП РАН для участия в исследованиях, публикациях результатов этих исследований в изданиях ИСП РАН и поступления в аспирантуру ИСП РАН.

7. Структура учебного плана

Учебный план рассчитан на два учебных года. На первом году обучения углубленно преподаются основополагающие теоретические дисциплины, служащие базисом для всей программы обучения, и основные разделы системного программирования. К ним относятся дисциплины:

- Прикладной системный анализ
- Формальные методы программной инженерии
- Конструирование ядра операционных систем

- Компьютерные сети и телекоммуникации
- Верификация программного обеспечения
- Параллельное программирование
- Перспективные системы управления базами данных
- Моделирование программных систем

На втором году обучения детально обсуждаются специальные разделы современного системного программирования:

- Конфигурирование и виртуализация облачных служб и их использование для обработки больших данных
- Формальные методы верификации и тестирования телекоммуникационных протоколов и сервисов
- Теоретическая криптография
- Научная визуализация

Учебный план предусматривает непрерывное участие студентов-магистрантов в научно-исследовательском семинаре по программе обучения. Студенты имеют персональных научных руководителей-кураторов и занимаются исследованиями и разработками в исследовательских группах, регулярно докладывая на семинарах результаты своей работы.

8. Концепция научно-исследовательского семинара

Научно-исследовательский семинар магистерской программы органично сопряжен с научной деятельностью ИСП РАН и является ее составной частью. Тематика научно-исследовательского семинара магистерской программы соответствует тематике работ основных научно-исследовательских отделов ИСП РАН, к которым прикрепляются студенты магистратуры, получая там научных руководителей, темы и задачи для исследований. Таким образом студенты непосредственно участвуют в передовых разработках по специальности, регулярно отчитываясь о полученных результатах на семинарах магистерской программы и регулярно проводимых семинарах ИСП РАН.

Одновременно с научно-исследовательской деятельностью студенты приобретают навыки участия в передовых разработках программных продуктов, проводимых в ИСП РАН с участием отечественных и зарубежных компаний. Именно такое участие является залогом подготовки специалистов мирового уровня с гарантией успешного трудоустройства при сохранении возможности продолжения научной работы в ИСП РАН.

9. Характеристика кадрового потенциала

Кадровый потенциал, задействованный в реализации магистерской программы, составляют сотрудники департамента Программной инженерии ФКН НИУ ВШЭ, сотрудники ИСП РАН и базовой кафедры ИСП РАН в НИУ ВШЭ.

Научный потенциал ИСП РАН в области системного программирования широко известен в нашей стране и за рубежом. Ниже приведены некоторые дополнительные сведения.

Список членов ученого совета ИСП РАН:

1. Аветисян Арутюн Ишханович, д.ф.-м.н., доцент, директор.
2. Белеванцев Андрей Андреевич, к.ф.-м.н., в.н.с.
3. Бурдонов Игорь Борисович, д.ф.-м.н., в.н.с.
4. Гайсарян Сергей Суменович, к.ф.-м.н., доцент, зав. отделом.
5. Карпов Леонид Евгеньевич, д.т.н., в.н.с.
6. Кузнецов Сергей Дмитриевич, д.т.н., профессор, гл.н.с.
7. Кузюрин Николай Николаевич, д.ф.-м.н., профессор, зав. отделом.
8. Кулямин Виктор Вячеславович, к.ф.-м.н., доцент, в.н.с.
9. Лаврищева Екатерина Михайловна, д.ф.-м.н., профессор, гл.н.с.
10. Падарян Вартаан Андроникович, к.ф.-м.н., в.н.с.
11. Пакулин Николай Витальевич, к.ф.-м.н., ученый секретарь института.
12. Петренко Александр Константинович, д.ф.-м.н., профессор, зав. отделом.
13. Семёнов Виталий Адольфович, д.ф.-м.н., профессор, в.н.с.
14. Турдаков Денис Юрьевич, к.ф.-м.н., зав. отделом.
15. Хорошилов Алексей Владимирович, к.ф.-м.н., в.н.с.
16. Томилин Александр Николаевич, д.ф.-м.н., профессор, гл.н.с.
17. Шнитман Виктор Зиновьевич, д.т.н., с.н.с., зав. отделом.
18. Шокуров Александр Владимирович, к.ф.-м.н., доцент, в.н.с.

Коллектив ИСП РАН насчитывает около 200 научных сотрудников, из них более 50 это кандидаты наук, и более 10 доктора наук.

ИСП РАН имеет долгие связи с крупнейшими исследовательскими центрами и промышленными компаниями как в России, так и за рубежом. Основными зарубежными партнерами являются:

- Университеты :
 - Пассау (Германия),
 - Оксфорд,
 - Кембридж (ВК),
 - MIT (США) и др.
- Компании:
 - Dell
 - HP
 - INRIA
 - Intel
 - Microsoft Research
 - NVIDIA
 - Rogue Wave Software
 - Samsung

Приложение

Содержательная часть программы

Первый курс

1. Прикладной системный анализ

Основной целью дисциплины "Прикладной системный анализ" является обсуждение и практическое применение некоторых систематических подходов, направленных на понимание и описание возникающих в инженерной и управленческой деятельности проблем (проблемных ситуаций) с их последующим анализом и определением приемлемых путей решения таких проблем. Неуклонно растущая сложность проблем, возникающих в различных областях деятельности человека (например, ИТ-индустрии, компьютерной и программной инженерии), неточность, неполнота или неопределенность информации о предмете рассмотрения (системе), а также конфликтующие точки зрения вовлеченных в проблемную ситуацию сторон (stakeholders), заставляют на практике искать возможные альтернативы решения проблем с использованием достаточно простых и понятных всем участникам проблемной ситуации моделей. В зависимости от характера рассматриваемых проблем, которые могут быть хорошо структурированными (well-structured), слабоструктурированными (ill-structured) или вовсе неструктурированными (non-structured), участники проблемной ситуации всегда стремятся смоделировать ситуацию, используя "жесткие" (hard) и/или "мягкие" (soft) методики системного исследования. Дисциплина направлена на формирование у студентов понимания роли системного подхода к решению проблем, связанных с их профессиональной деятельностью, получения навыков описания проблемных ситуаций, построения соответствующих формальных и «мягких» моделей реальных проблем, их анализа и формирования возможных вариантов решения.

Литература

1. Тарасенко Ф.П. Прикладной системный анализ: уч.пособ., М.:КНОРУС, 2010.
2. О'Коннор Дж., Макдермотт И. Искусство системного мышления. Необходимые знания о системах и творческом подходе к решению проблем, пер.с англ., М.: Альпина Бизнес Букс, 2008.
3. Волкова В.Н., Деннисов А.А., Теория систем и системный анализ.: учебник для вузов, М.: ИД Юрайт, 2010.
4. Von Bertalanffy L. General System Theory. Foundations, Development, Application, George Braziller, NY, 2009.
5. Медоуз Д., Азбука системного мышления, БИНОМ, Лаборатория знаний, 2010.
6. Flood R.L., Carson E.R., Dealing with Complexity. An Introduction to the Theory and

- Application of Systems Science, 2nd ed., Plenum Press, NY, 1993.
7. Boardman J., Sauser B., Systems Thinking: Coping with 21st Century Problems (Industrial Innovation), CRC Press, 2008.
 8. Edson R., Systems Thinking. Applied. A Primer, ASysT Institute, 2008.
 9. Ross T.J., Fuzzy Logic with Engineering Applications, 3rd ed., John Wiley & Sons, 2010.
 10. Демидова Л.А., Кираковский В.В., Пылькин А.Н., Принятие решений в условиях неопределенности, М.: Горячая Линия – Телеком, 2012.

2. Формальные методы программной инженерии

Курс посвящен формальной верификации программ и моделей компьютерных систем. Цели курса: познакомить студентов с базовыми принципами и методами формальной верификации; сформировать у студентов навыки необходимые для практического использования рассмотренных методов. Основу курса составляют: (1) методы формальной спецификации программ (пред- и постусловия, темпоральные утверждения); методы формализации поведения программ (формализация семантики языков программирования, использование формальных моделей); методы формальной верификации (дедуктивная верификация программ, проверка моделей).

Литература

1. Wolfgang Reisig, Understanding Petri Nets. Modeling Techniques, Analysis Methods, Case Studies. 2013. ISBN 978-3-642-33278-4.
2. Jensen K. and Kristensen L. M. Coloured Petri Nets Modelling and Validation of Concurrent Systems. Springer-Verlag, 2009.
3. Ben-Ari M. Principles of the Spin Model Checker. Springer-Verlag, 2008.
4. J. C. M. Baeten, T. Basten and M. A. Reniers: Process Algebra: Equational Theories of Communicating Processes. Cambridge Tracts in Theoretical Computer Science, Vol. 30. Cambridge University Press, 2010
5. Michael Fisher. An Introduction to Practical Formal Methods Using Temporal Logic. Wiley publisher, 2011. SBN-10: 0470027886 | ISBN-13: 978-0470027882
6. R.A. Milner. Calculus of communicating systems. Lecture Notes in Computer Science, v.92. Springer, 1980.
7. Карпов Ю.Г. MODEL CHECKING. Верификация параллельных и распределенных программ и систем. СПб.: БХВ-Петербург, 2010.
8. Rajeev Alur, Tom Henzinger. Invariant verification. Chapter II in manuscript "Computer-aided verification". <http://mtc.epfl.ch/courses/CAV2006/Notes/2.pdf> (электронная версия)
9. Ломазова И.А. Сети Петри и анализ поведенческих свойств распределенных систем. Ярославль: ЯрГУ, 2002.

10. Хопкрофт Дж., Мотвани Р., Ульман Дж. Введение в теорию автоматов, языков и вычислений: Пер. с англ. М.: Издательский дом "Вильямс", 2008.
11. Nielson H. R. and Nielson F. Semantics with Applications: An Appetizer. Springer-Verlag, 2007.
12. Schneider K . Verification of Reactive Systems. Springer-Verlag, 2004.

3. Конструирование ядра операционных систем

Курс направлен на изучение основных принципов внутреннего устройства ядра операционной системы, механизмов аппаратной поддержки работы ядра, а также получение навыков проектирования и программирования компонентов ядра операционной системы и отладки программ в привилегированном режиме работы процессора. Слушатели должны усвоить основные принципы внутреннего устройства ядра операционной системы, приобрести теоретические знания и практические навыки в проектировании и программировании компонентов ядра операционной системы и в отладке программ в привилегированном режиме работы процессора. В курсе рассматриваются следующие темы: принципы внутреннего устройства ядра операционной системы, механизмы аппаратной поддержки работы ядра, механизмы обеспечения защиты ядра операционной системы от приложений и приложений друг от друга, методы управления и распределения аппаратными ресурсам, методы и средства виртуализации аппаратных ресурсов.

Литература

1. А. Робачевский, С. Немнюгин, О. Стесик «Операционная система UNIX», БХВ Санкт-Петербург, 2010 г.
2. Э. Таненбаум “Архитектура компьютера”, Питер, 2012 г.
3. Э. Таненбаум “Современные операционные системы”, Питер, 2011 г.
4. Э. Таненбаум “Операционные системы. Разработка и реализация”, Питер, 2006 г.
5. IA-32 Intel Architecture Software Developer's Manual Volume 1: Basic Architecture. Basic 80x86 architecture and programming environment. <http://developer.intel.com>.
6. IA-32 Intel Architecture Software Developer's Manual Volume 3A: System Programming Guide. Operating system support, including segmentation, paging, tasks, interrupt and exception handling. <http://developer.intel.com>.
7. Tool Interface Standard Executable and Linking Format (ELF) Specification Version 1.2.
8. C. A. R. Hoare, Monitors: An Operating System Structuring Concept. Communications of the ACM, 17(10):549--557, 1974.
9. Thomas W. Doeppner “Operating Systems In Depth: Design and Programming” Wiley, 2010
10. Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman. “Linux Device Drivers”.

11. Ellen Siever, Stephen Figgins, Robert Love, Arnold Robbins. "Linux in a Nutshell".
O'Reilly Media, 2009

4. Компьютерные сети и телекоммуникации

Цель курса - познакомить студентов с основными средствами современных телекоммуникаций, используемых для построения распределённых систем. Курс начинается с базовых представлений о коммуникационных сетях, включая эталонную семиуровневую модель и протоколы стека TCP/IP. В настоящее время всё больше приложений используют веб-протоколы для обмена данными, поэтому далее в рамках курса будут рассмотрены общие механизмы Веба (протоколы HTTP/HTTPS и форматы обмена данными XML/JSON), и специализированные средства построения и вызова веб-сервисов. Один из краеугольных камней курса - обеспечение безопасности сетевых коммуникаций. В рамках этого модуля рассматриваются как общие вопросы безопасности (аутентификация, конфиденциальность, авторизация), так и конкретные решения для практических задач: протокол безопасности TLS/SSL, механизмы безопасности протокола HTTP и веб-сервисов, представление о разделении доступа в веб-приложениях.

Литература

1. Танненбаум Э. Компьютерные сети. СПб.: Питер, 2003.
2. Халсалл Ф. Передача данных, сети компьютеров и взаимосвязь открытых систем. М.: Радио и связь, 1995.
3. Семенов Ю.А. Протоколы и ресурсы Internet. М.: Радио и связь, 1996.
4. С. Бенет, С. Пэйн. Криптография. Официальное руководство RSA Security. М.: Бином-Пресс, 2002
5. Б. Шнайер. Прикладная криптография. М.: ТРИУМФ, 2003
6. Панасенко С.П. Алгоритмы шифрования. Специальный справочник. СПб.: БХВ-Петербург, 2009.
7. В.А. Галатенко. Основы информационной безопасности. М.: ИНТУИТ.РУ «Интернет-Университет», 2003.
8. В.А. Галатенко. Стандарты информационной безопасности. М.: ИНТУИТ.РУ «Интернет-Университет», 2004.
9. М.Р. Биктимиров, А.Ю. Щербаков. Избранные главы компьютерной безопасности. Казань: Издательство Казанского математического общества, 2004.
10. К.В. Ребриков, В.З. Шнитман. "Протоколы автоматического установления контекстов безопасности и управления ключами в Интернет", Препринт 19 ИСП РАН, М., 2007.

5. Верификация программного обеспечения

Курс представляет собой введение в методы верификации программного обеспечения. Цель курса -- познакомить с предметом верификации ПО, представить широкую палитру существующих методов и подходов, осветить преимущества и ограничения, присущие методам верификации. В рамках курса рассматриваются общее понятие качества ПО, подпроцессы обеспечения качества в рамках жизненного цикла ПО, методы статического анализа программ, методы проверки моделей (model checking), методы динамического анализа программ и различные варианты функционального тестирования. Задачами данного курса являются: формирование базовых знаний в области обеспечения качества программного обеспечения, как неотъемлемой части теории и практики разработки ПО, адресуемого к проблемам построения корректных и надежных программ, и имеющего важное методологическое значение как для подготовки специалистов в области современных информационных технологий, так и для поддержки разнообразных инновационных сфер деятельности; обучение студентов основам жизненного цикла программного обеспечения и задачам верификации, возникающим в ходе разработки, внедрения и эксплуатации ПО; обучение студентов методам функционального тестирования, применяемым в различных сценариях разработки ПО, включая модульное тестирование, случайное тестирование, тестирование с использованием моделей, а также методам оценки полноты тестирования; обучение студентов базовым методам анализа корректности программ; формирование теоретических подходов к верификации программного обеспечения для проведения исследований в рамках выпускных работ на степень магистра.

Литература

1. В.В. Липаев. Тестирование программ. М.: Радио и связь, 1986.
2. Г. Майерс. Искусство тестирования программ. М.: Финансы и статистика, 1982.
3. Б. Бейзер. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем. СПб.: Питер, 2004.
4. С. Сеницын, Н. Налютин. Верификация программного обеспечения. М.: Бином, 2008.
5. В.В. Кулямин. Методы верификации программного обеспечения. Всероссийский конкурс обзорно-аналитических статей по приоритетному направлению «Информационно-телекоммуникационные системы», 2008.
6. В.А. Непомнящий, О.М. Рязин. Прикладные методы верификации программ. М.: Радио и связь, 1988.
7. Ю.Г. Карпов. Model Checking. Верификация параллельных и распределенных программных систем. СПб.: БХВ-Петербург, 2010.
8. Э. Дейкстра. Дисциплина программирования. М.: Мир, 1978.
9. Д. Грис. Наука программирования. М.: Мир, 1984.
10. M. Broy, B. Jonsson, J.-P. Katoen, M. Leucker, A. Pretschner (eds.). Model Based Testing of Reactive Systems. LNCS 3472, Springer, 2005.

6. Параллельное программирование

Курс рассматривает принципы, архитектуры и методы параллельного программирования. Излагаются общие сведения об уровнях параллелизма в аппаратуре (команд, потоков, процессов), о подходах к параллельному программированию. Дается обзор моделей параллельного программирования, после чего основной акцент делается на методах параллельного программирования в потоках (модель OpenMP) и параллельного программирования в процессах (модель MPI). Рассматриваются способы организации параллельных вычислений и коммуникаций, способы доступа к общим ресурсам, способы синхронизации.

Литература

1. Randy Allen, Ken Kennedy. Optimizing Compilers for Modern Architectures: A Dependence-based Approach, 1st Edition. Morgan Kaufman, 2001. 790 pages. ISBN 978-1558602861.
2. Calvin Lin, Lawrence Snyder. Principles of Parallel Programming, 1st Edition. Pearson, 2008. 352 pages. ISBN 978-0321487902.
3. Brendan Gregg. Systems Performance: Enterprise and the Cloud, 1st Edition. Pearson, 2013. 792 pages. ISBN 978-0133390094
4. Barbara Chapman, Gabriele Jost, Ruud van der Pas. Using OpenMP: Portable Shared Memory Parallel Programming. The MIT Press, 2007. 384 pages. ISBN 978-0262533027
5. Michael Quinn. Parallel Programming in C with MPI and OpenMP. McGraw-Hill Education, 2008. 480 pages. ISBN 978-0071232654
6. David Paterson. Computer Architecture, Fifth Edition: A Quantitative Approach. Morgan Kaufman, 2011. 856 pages. ISBN 978-0123838728
7. David Kirk, Wen-mei Hwu. Programming Massively Parallel Processors, Second Edition: A Hands-on Approach. Morgan Kaufman, 2012. 514 pages. ISBN 978-0124159921
8. Paul McKenney. Is Parallel Programming Hard, And, If So, What Can You Do About It?
9. William Kahan. Why do we need a floating-point arithmetic standard?
10. Krste Asanović, Rastislav Bodik, James Demmel, Tony Keaveny, Kurt Keutzer, John Kubiatowicz, Nelson Morgan, David A. Patterson, Koushik Sen, John Wawrzynek, David Wessel, and Katherine Yelick. A View of the Parallel Computing Landscape. Communications of the ACM, October 2009.

7. Перспективные системы управления базами данных

Курс предназначен для студентов магистратуры имеющих базовую подготовку в области баз данных и систем управления базами данных: модели данных, реляционная модель данных и язык SQL, основы теории реляционных баз данных, методы проектирования реляционных баз данных, общая архитектура типовой SQL-ориентированной СУБД, методы хранения и индексации данных, методы управления транзакциями, методы обеспечения надежности хранения баз данных.

Данный курс обеспечивает студентам знания в областях методов и алгоритмов обработки запросов в SQL-ориентированных СУБД; оптимизации запросов; различных подходов к распараллеливанию запросов: архитектур параллельных SQL-ориентированных СУБД с общей основной памятью, с общей внешней памятью и без совместно используемых ресурсов; подходов к организации систем обработки данных на основе совместного использования технологий баз данных и распределенных систем; перспектив перехода к использованию новых видов аппаратных средств хранения данных (флеш-память, энергонезависимая основная память).

Курс начинается с обсуждения разных подходов к классификации СУБД: по используемой модели данных (реляционные, SQL-ориентированные, объектно-ориентированные и т.д.); транзакционные и аналитические; универсальные и специализированные; файл-серверные, клиент-серверные и встраиваемые; дисковые и сохраняющие данные в основной памяти (in-memory); однопроцессорные, параллельные с общей памятью (shared everything), параллельные с общими дисками (shared disks), параллельные без общих ресурсов (shared nothing); проприетарные и с открытыми кодами. Обосновывается особая актуальность SQL-ориентированных СУБД, описываются общая архитектура и основные функции таких СУБД.

Далее обсуждаются общие подходы к обработке запросов и операций обновления баз данных: методы выполнения «реляционных» операций, компиляция и построение планов выполнения операций SQL.

Следующая тема посвящена оптимизации запросов в SQL-ориентированных СУБД: методы перезаписи запросов, методы оценки стоимости планов выполнения запросов, методы оценки селективности предикатов, оценки распределений значений в столбцах таблиц, методы оптимизации запросов с многочисленными соединениями и т.д.

Затем обсуждаются аппаратно-программные архитектуры СУБД, обеспечивающие распараллеливание операций над базами данных.

Сначала рассматриваются параллельные СУБД категории shared everything, имеющиеся решения (например, в Oracle), особенности компиляции и оптимизации запросов, идеи, позволяющие добиваться горизонтальной масштабируемости при росте числа ядер.

Затем описываются подходы к построению параллельных СУБД категории shared disks, имеющиеся решения (Oracle RAC, Tibero), специальные требования к аппаратуре.

Обсуждаются параллельные СУБД категории shared nothing, в том числе, аналитические СУБД (Vertica, Greenplum, Asterdata), особенности компиляции и оптимизации запросов, методы выполнения распределенных запросов, методы хранения и разделения данных, методы параллельного серверного программирования, а также транзакционные СУБД, ориентация на высокий уровень доступности или на высокий темп обработки транзакций, методы уменьшения числа распределенных транзакций при заданной рабочей нагрузке.

После этого описываются подходы к организации систем обработки данных на основе совместного использования технологий баз данных и распределенных систем: аналитика и MapReduce, Hadoop и OpenStack, «транзакционные» СУБД категории NoSQL, «теорема» CAP и ее следствие – транзакции BASE (Basically Available, Soft-state, Eventual consistency – доступность в большинстве случаев, неустойчивое состояние, согласованность в конечном счете и другие ослабленные модели транзакций, NoSQL в мире SQL: применение теоремы CAP при реализации параллельных СУБД без использования общих ресурсов, поддерживающих высокий уровень доступности.

Наконец, в заключение курса рассматривается влияние на общую архитектуру СУБД, структуры данных и алгоритмы выполнения операций и оптимизацию запросов потенциального перехода к использованию новых видов аппаратных средств хранения данных – флеш-памяти (твердотельных дисков) и энергонезависимой основной памяти.

Литература

1. С.Д. Кузнецов. Базы данных. Академия, Серия: Университетский учебник, 2012 г.
2. Сергей Кузнецов. Базы данных. Вводный курс, 2008,
http://citforum.ru/database/advanced_intro/
3. C.J. Date. An Introduction to Database Systems (8th Edition). Addison-Wesley, 2003
Перевод: Крис Дж. Дейт. Введение в системы баз данных. Вильямс, 2006
4. Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom. Database Systems: The Complete Book (2nd ed.). Prentice Hall, 2008. Доступно по ссылке:
https://www.researchgate.net/publication/200034291_Database_Systems_The_Complete_Book.
Перевод первого издания: Гектор Гарсиа-Молина, Джеффри Д. Ульман, Дженнифер Уидом. Системы баз данных. Полный курс. Вильямс, 2004.
5. Jim Gray, Andreas Reuter. Transaction Processing: Concepts and Techniques. Morgan Kaufmann Publishers, 1992.
6. Gerhard Weikum, Gottfried Vossen, Transactional information systems: theory, algorithms, and the practice of concurrency control and recovery, Morgan Kaufmann, 2002.
7. Сергей Кузнецов, Петр Чардин. Семейство алгоритмов ARIES. 2004. Доступно по ссылке: <http://citforum.ru/database/articles/aries/>

8. System R. История проекта, публикации, участники проекта. Сайт Пола Макджонса. http://www.mcjones.org/System_R/
Переводы некоторых публикаций, в том числе, The 1995 SQL Reunion: People, Projects, and Politics доступны по ссылкам <http://citforum.ru/database/digest/>, <http://citforum.ru/database/classics/> и <http://citforum.ru/database/articles/index.shtml>
9. V. Markl, G. M. Lohman, V. Raman. LEO: An autonomic query optimizer for DB2. IBM SYSTEMS JOURNAL, VOL 42, NO 1, 2003
Перевод: Волкер Маркл, Гай Лохман, Виджайшанкар Раман. LEO: самонастраивающийся оптимизатор запросов для DB2. Открытые системы, #04/2003. Доступно по ссылке: <http://citforum.ru/database/articles/leo.shtml>
10. Сергей Кузнецов. Оптимизация запросов в коммерческих СУБД в начале 1990-х. Переводы статей из IEEE Computer Society Data Engineering Bulletin, Vol. 16 No. 4, 1993 об оптимизации запросов в IBM DB2, NonStop SQL и DEC Rdb. Доступно по ссылке: http://citforum.ru/database/classics/query_opt_1993/
11. Сергей Кузнецов. MapReduce: внутри, снаружи или сбоку от параллельных СУБД? Труды Института системного программирования, т. 19, М., ИСП РАН, 2010, стр. 35-40. Доступно по ссылке: http://citforum.ru/database/articles/dw_appliance_and_mr/
12. Сергей Кузнецов. Транзакционные параллельные СУБД: новая волна. Труды Института системного программирования, т. 20, М., ИСП РАН, 2011, стр. 189-251. Доступно по ссылке: http://citforum.ru/database/articles/kuz_oltp_2010/
13. С.Д. Кузнецов, А. Посконин. Распределенные горизонтально масштабируемые решения для управления данными. Труды Института системного программирования, т. 24, М., ИСП РАН, 2013, стр. 327-358. Доступно по ссылке: http://ispras.ru/ru/proceedings/docs/2013/24/isp_24_2013_327.pdf
14. С.Д. Кузнецов, А. Посконин. Системы управления данными категории NoSQL. Ж. “Программирование”, N6, 2014, стр. 34-47. Доступно по ссылке: http://www.ispras.ru/publications/nosql_data_management_systems.pdf

8. Моделирование программных систем

Курс посвящен методам и языкам моделирование программных, программно-аппаратных и гетерогенных (cyber-physical) систем. Охватывается широкий набор аспектов моделирования, включая моделирование статической структуры системы и динамических аспектов поведения. Помимо широко известного языка моделирования UML рассматриваются языки SysML и AADL. Наибольшее внимание уделяется вопросам моделирования поведения, спецификации характеристик, связанных с функциональностью, временем/скоростью выполнения, пропускной способностью линий связи, с объемом потребления других ресурсов системы. Рассматриваются задачи верификации модели, верификации целевой системы на основе модели, задачи анализа рисков, надежности, безопасности программно-аппаратных систем. Дается обзор распространенных инструментов для работы с моделями.

Литература

1. Вендров А.М., Малышко В.В. Объектно-ориентированный анализ и проектирование с использованием языка UML // М.: Издательский отдел факультета ВМиК МГУ, 2002. — 137 с.
2. Volter M. et al. Model-Driven Software Development: Technology, Engineering, Management. John Wiley & Sons, 2006 [PDF]
3. Warmer J. B., Kleppe A. G. The Object Constraint Language: Getting Your Models Ready for MDA. Addison-Wesley Professional, 2003
4. Gogolla M., Buttner F., Richters M. USE: A UML-Based Specification Environment for Validating UML and OCL
5. Voelter M. DSL Engineering. 2013 [HTML]
6. Brambilla M., Cabot J., Wimmer M. Model-Driven Software Engineering in Practice. Morgan & Claypool Publishers, 2012
7. Фаулер М. Предметно-ориентированные языки программирования - Вильямс, 2011.
8. Кознов Д. В. Основы визуального моделирования. – БИНОМ. Лаборатория знаний, ИНТУИТ. 2008
9. Kolovos D., Rose L., Garcia-Dominguez A., Paige R. The Epsilon Book, 2013.
10. Peter H. Feiler, David P. Gluch, John J. Hudak. The Architecture Analysis & Design Language (AADL): An Introduction // The Software Engineering Institute & Carnegie Mellon University, 2006, <http://www.sei.cmu.edu/reports/06tn011.pdf>.

Второй курс

9. Конфигурирование и виртуализация облачных служб и их использование для обработки больших данных

Данный курс будет посвящен проблемам обработки больших данных и эволюции подходов к их обработке. В курсе обсуждается ряд технологий распределенной и параллельной обработки данных, предназначенных, в частности, для использования в облаках и ориентированных на обработку данных большого объема. В ходе курса будут рассмотрены:

- технологии массивно-параллельных вычислений при помощи видеокарт общего назначения;
- технологии, предназначенные для распределения данных по вычислительным узлам вычислительных систем, начиная с распределенных файловых систем, заканчивая объектными хранилищами данных;
- технологии оркестрации и управления облачными ресурсами;
- технологии виртуализации и контейнеризации;
- архитектура и общее устройство существующих облачных сред;
- технологии обработки больших данных (в том числе и их применение в облачных средах) и их применение к вычислительным задачам.

Среди перечисленных семейств технологий будут представлены: Nvidia CUDA, OpenCL, iSCSI, Apache HDFS, GlusterFS, Ceph, Swift, Amazon S3, Chef, Puppet, Stacksalt, Ansible, KVM, Xen, LXC, Docker, Eucalyptus, Openstack, Amazon EC2/ECC, Apache Hadoop, Apache Spark, Apache Ignite.

Литература

1. Jason Sanders and Edward Kandrot. CUDA by Example: An Introduction to General-Purpose GPU Programming (1st ed.). Addison-Wesley Professional, 2010. ISBN: 978-0-131-38768-3
2. David B. Kirk and Wen-mei W. Hwu. Programming Massively Parallel Processors: A Hands-On Approach (2nd ed.). Morgan Kaufmann Publishers Inc., 2012. ISBN: 978-0-124-15992-1
3. Боресков А.В., Харланов А.А. Основы работы с технологией CUDA. ДМК Пресс, 2010. ISBN: 978-5-940-74578-5
4. Sébastien Goasguen. Docker Cookbook (1st ed.). O'Reilly Media, 2015. ISBN: 978-1-4919-1971-2
5. Scott Adkins, John Belamaric, Vincent Giersch, Denys Makogon, Jason E. Robinson. OpenStack Cloud Application Development. Wrox Press Ltd., 2015. ISBN: 978-1-119-19431-6
6. Madhuranjan Mohaan, Ramesh Raithatha. Learning Ansible. Packt Publishing, 2014. ISBN: 978-1-783-55063-0
7. Dan C. Marinescu. Cloud Computing: Theory and Practice (1st ed.). Morgan Kaufmann Publishers Inc., 2013. ISBN: 978-0-124-04627-6
8. Elizabeth K. Joseph, Matt Fischer. Common OpenStack Deployments: Real World Examples for Systems Administrators and Engineers. Prentice Hall, 2016. ISBN: 978-0-134-08623-1
9. Matthew Portnoy. Virtualization Essentials (1st ed.). SYBEX Inc., 2012. ISBN: 978-1-118-17671-9
10. Fabio Alessandro Locati. OpenStack Cloud Security. Packt Publishing, 2015. ISBN: 978-1-782-17098-3

10. Статический анализ и оптимизация программ - 1

Курс предназначен для обучения студентов основным алгоритмам анализа и оптимизации программ, использующихся в современных промышленных компиляторах. В курсе вводятся внутренние представления среднего и низкого уровней, на которых демонстрируется ряд алгоритмов машинно-независимой и машинно-зависимой оптимизации. Среди машинно-независимых оптимизаций излагаются классические итерационные методы оптимизаций, методы класса Mark and Sweep, нумерация значений на SSA-представлении, межпроцедурные оптимизации, в частности, межпроцедурное продвижение констант на основе аннотаций функций. В разделе машинно-зависимых оптимизаций излагаются вопросы кодогенерации, а также алгоритмы планирования команд и распределения регистров.

Литература

1. А.В. Ахо, М.С. Лам, Р. Сети, Дж.Д. Ульман. Компиляторы: принципы, технологии и инструментарий. Издание второе. / М.: ООО «И.Д. Вильямс», 2008, ISBN: 978-5-8459-1349-4
2. K.D. Cooper and L. Torczon. Engineering a Compiler. / Morgan Kaufman Publishers, 2004, ISBN: 1-55860-698-X
3. The **LLVM** Compiler Infrastructure. // <http://llvm.org/>
4. A GNU Manual. // <http://gcc.gnu.org/>
5. Massimiliano Poletto and Vivek Sarkar Linear Scan Register Allocation // ACM Transactions on Programming Languages and Systems, Vol. 21, No. 5, September 1999, Pages 895{913.
6. А.Ю.Дроздов, А.В.Кан Анализ “Межпроцедурная нумерация значений”// Компьютеры в учебном процессе №5, май 2005
7. Andrew W. Appel and Jens Palsberg Modern Compiler Implementation in Java, Second Edition Cambridge University Press, 2002 (501 pages) ISBN:052182060x
8. Meisam Booshehri, Abbas Malekpour, Peter Luksch An Improving Method for Loop Unrolling // International Journal of Computer Science and Information Security, Vol. 11, No. 5, May 2013
9. D. Novillo. Design and implementation of Tree SSA. In GCC Developers Summit, pages 119–130, 2004.
10. Ben Hardekopf, Calvin Lin Flow-Sensitive Pointer Analysis for Millions of Lines of Code // <https://www.cs.utexas.edu/users/lin/papers/cgo11.pdf>.

11. Безопасность программного обеспечения

Курс рассматривает проблемы информационной безопасности, обусловленные дефектами ПО, методы и подходы выявления таких дефектов, способы превентивного противодействия эксплуатации дефектов и смягчения последствий эксплуатации. Основной упор делается на технологии, применяемые на этапах разработки с целью улучшения безопасности ПО, а также на методы изучения программ в отсутствии исходных кодов. Рассматриваются динамический и статический подходы к анализу, возможности комбинирования этих подходов. Предметом изучения выступает как исходный код, так и исполняемый, включая ситуации, когда необходимо анализировать ПО всей вычислительной системы в целом. По окончании курса слушатели приобретают знания о моделях безопасного жизненного цикла ПО, причинах нарушения информационной безопасности в разработанном ПО, методах улучшения безопасности и оценки безопасности ПО в отсутствии исходных кодов, а также о методах расследования инцидентов.

Литература

1. Крис Касперски, Ева Рокко. Искусство дизассемблирования. — СПб.: БХВ-Петербург, 2008. — 896 с.

2. Brian Chess, Jacob West. Secure Programming with Static Analysis. — Addison-Wesley Professional, 2007. — 624 с.
3. Flemming Nielson, Hanne Riis Nielson, Chris Hankin. Principles of Program Analysis. — Springer, 2010. — 450 с.
4. Somesh Jha, Anupam Datta, Ninghui Li, David Melski, Thomas Reps. Analysis Techniques for Information Security. — Morgan and Claypool Publishers, 2010. — 164 с.
5. J. Seward, N. Nethercote, J. Weidendorfer. Valgrind 3.3 — Advanced Debugging and Profiling for GNU/Linux Applications. — Network Theory Ltd., 2008. — 164 с.
6. Chris Eagle. The IDA Pro Book, 2nd Edition. — No Starch Press, 2011. — 676 с.
7. Амини П., Саттон М., Грин А. Fuzzing: исследование уязвимостей методом грубой силы. — Символ-Плюс, 2009.
8. Frank Tip. A survey of program slicing techniques. — Journal of Programming Languages, 3(3):121-189, September 1995.
9. G. Balakrishnan, T. Reps. Analyzing Memory Accesses in X86 Executables. — Proceedings of 13th International Conference on Compiler Construction, 2004.
10. Edward J. Schwartz, Thanassis Avgerinos, David Brumley. All You Ever Wanted to Know about Dynamic Taint Analysis and Forward Symbolic Execution (but might have been afraid to ask). — Proceedings of the 2010 IEEE Symposium on Security and Privacy (Oakland, California, USA, May 16-19, 2010). IEEE, New York, NY, 317-331.

12. Научная визуализация

Цель курса — освоение студентами фундаментальных знаний в области визуализации и связанных с ней разделах компьютерной графики и вычислительной геометрии. Важное внимание в курсе уделяется базовым принципам визуализации, особенностям постановок задач, возникающих в разных прикладных областях, в том числе в математическом моделировании, программной инженерии, управлении проектами. В рамках курса студенты осваивают широкую палитру методов и приемов визуализации, а также знакомятся с современными средствами и технологиями их реализации.

Литература

1. Д.Роджерс. Алгоритмические основы машинной графики. М.:Мир, 1989.
2. Е.В. Шикин, А.В. Боресков. Компьютерная графика. М.: “Диалог МИФИ”, 1997.
3. Ю. Тихомиров, Программирование трехмерной графики, Спб.: BHV, 1998.
4. М. Ласло, Вычислительная геометрия и компьютерная графика на C++. Москва. “БИНОМ”, 1997.
5. Ф.Препарата, М.Шеймос. Вычислительная геометрия. Введение. М.:Мир, 1989.
6. Hanan Samet, Foundations of Multidimensional and Metric Data Structures. Morgan Kaufmann publishers, 2011.
7. Семенов В.А. Открытая система для математического моделирования и научной визуализации. Учебно-методическое пособие. М.: МФТИ, 2005.
8. Е. Ю. Ечкина, С. Б. Базаров, И. Н. Ионовенков «Визуализация в научных исследованиях. Учебное пособие». М.: МАКС ПРЕСС, 2006.

9. H. Senay, E. Ignatus, Rules and Principles of Scientific Data Visualization, Tech. Report, George Washington University, Department of Electrical Engineering and Computer Science, February 1999.
10. В.Н. Касьянов, В.А. Евстигнеев. Графы в программировании: обработка, визуализация и применение. СПб.: БХВ-Петербург, 2003.

Статический анализ и оптимизация программ - 2

Курс предназначен для обучения студентов методам анализа исходного кода программ, применяемых в ходе промышленных циклов разработки безопасного программного обеспечения (как правило, при ежедневном ночном тестировании программной системы). Излагаются требования к статическим анализаторам, непосредственно следующие из условий применения (время работы, количество истинных срабатываний), дается понятие об устройстве анализаторов (разбор кода, анализ кода, выдача предупреждений, ведение истории анализа). После этого студенты изучают используемые методы внутрипроцедурного анализа программ (анализ указателей, анализ интервалов значений переменных, нумерация значений), устройство системы межпроцедурного анализа, устройство системы чувствительного к путям выполнения анализа, получают понятие о применении SAT/SMT-решателей для проведения анализа. Изложение ведется с использованием примеров реальных программ на языке Си. Отдельно обращается внимание на особенности анализа программ на языке Си++ (необходимость анализа классов, конструкторов, деструкторов и т.п.), на языке Java (преобразование байткода в промежуточное представление для анализа, алгоритмы девиртуализации, алгоритмы обработки потока управления, создаваемого при наличии блоков try/catch).

Литература

1. А.В. Ахо, М.С. Лам, Р. Сети, Дж.Д. Ульман. Компиляторы: принципы, технологии и инструментарий. Издание второе. / М.: ООО «И.Д. Вильямс», 2008, ISBN: 978-5-8459-1349-4.
2. K.D. Cooper and L. Torczon. Engineering a Compiler. / Morgan Kaufman Publishers, 2004, ISBN: 1-55860-698-X
3. The LLVM Compiler Infrastructure. // <http://llvm.org/>
4. A GNU Manual. // <http://gcc.gnu.org/>
5. Massimiliano Poletto and Vivek Sarkar Linear Scan Register Allocation // ACM Transactions on Programming Languages and Systems, Vol. 21, No. 5, September 1999, Pages 895{913.
6. А.Ю.Дроздов, А.В.Кан Анализ “Межпроцедурная нумерация значений”// Компьютеры в учебном процессе №5, май 2005.
7. Andrew W. Appel and Jens Palsberg Modern Compiler Implementation in Java, Second Edition Cambridge University Press, 2002 (501 pages) ISBN:052182060x.
8. Meisam Booshehri, Abbas Malekpour, Peter Luksch An Improving Method for Loop Unrolling // International Journal of Computer Science and Information Security, Vol. 11, No. 5, May 2013.
9. D. Novillo. Design and implementation of Tree SSA. In GCC Developers Summit, pages 119–130, 2004.
10. Ben Hardekopf, Calvin Lin Flow-Sensitive Pointer Analysis for Millions of Lines of Code // <https://www.cs.utexas.edu/users/lin/papers/cgo11.pdf>.

Концепция научно-исследовательского семинара “Системное программирование”

В рамках научно-исследовательского семинара будут проводиться доклады, мастер-классы, обсуждения исследовательских работ, связанных с актуальными вопросами современного системного программирования.

Научно-исследовательский семинар сопряжен с семинарами ИСП РАН, которые базируются на современных разработках и исследованиях, проводимых как непосредственно в ИСП РАН, так и совместно с передовыми исследовательскими центрами и ведущими мировыми компаниями в области Системного программирования.