**Федеральное государственное автономное образовательное  
учреждение высшего образования «Московский физико-технический институт**

**(национальный исследовательский университет)»**

**УТВЕРЖДЕНО**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(должность, ФИО лица, утвердившего программу)

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_г.

**Рабочая программа дисциплины (модуля)**

|  |  |
| --- | --- |
| **по дисциплине:** | **Основы программного моделирования вычислительных систем** |
| **по направлению:** | **010900 прикладная математика и физика** |
| **профиль подготовки:** | **Инфокоммуникационные и вычислительные системы и технологии**  Физтех-школа радиотехники и компьютерных технологий  кафедра Микропроцессорные технологии в интеллектуальных системах управления  (учебное подразделение – кафедра, департамент) |
| **курс:** | **4** |
| **квалификация:** | **бакалавр** |

Семестр, формы промежуточной аттестации: 8 (весенний) – дифференцированный зачет

Аудиторных часов: 34 всего, в том числе:

лекции: 34 час.

семинары: 0 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 6 час.

Подготовка к экзамену: 0 час.

Всего часов: 40 , всего зач. ед.: 1

Количество контрольных работ, заданий: 1

Программу составил: **Е.А. Юлюгин, кандидат технических наук**

Программа обсуждена на заседании кафедры (департамента) «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_г.

**Аннотация**

В курсе рассматриваются ключевые понятия и методы программного моделирования вычислительных систем. Прежде всего дается представление о проблемах, с которыми сталкиваются исследователи и инженеры при разработки новых вычислительных систем. Рассматриваются основные способы решения обозначенных проблем, основанные на моделировании. Даются знания о существующем спектре технологий моделирования, их возможностях и ограничениях.

Курс содержит в себе обзор и обсуждение существующих вычислительных систем, знание которых необходимо для глубокого понимания проблем, с которыми сталкиваются создатели новых устройств. Для успешного осовоения курса слушателю требуются базовые знания «Информатики» и «Архитектуры вычислительных систем».

* 1. **Цели и задачи**

Целью курса являетсяознакомление слушателей с современными технологиями создания программных моделей вычислительных систем, используемых для разработки программного обеспечения, тестирования, исследования производительности и свойств вычислительных систем на стадиях раннего проектирования, когда образцы соответствующей аппаратуры ещё недоступны.

Задачи курса:

* Получение представления о проблемах, встающих перед создателями новых вычислительных систем, и способах их решения, основанных на использовании моделирования.
* Приобретение знаний о существующем спектре технологий моделирования, их различий в скорости, точности работы, масштабируемости.
* Знакомство с основными алгоритмами, используемыми в существующих коммерческих и академических программных продуктах: симуляторах и виртуальных машинах.
  1. **Перечень формируемых компетенций**

|  |  |
| --- | --- |
| Код и наименование компетенции | Индикаторы достижения компетенции |
| ОК-1 | способность анализировать научные проблемы и физические процессы, использовать на практике фундаментальные знания, полученные в области естественных и гуманитарных наук |
| ОК-2 | способность осваивать новые проблематику, терминологию, методологию и овладевать научными знаниями, владеть навыками самостоятельного обучения |
| ОК-7 | способен понимать сущность и проблемы развития современного информационного общества |
| ПК-3 | способность понимать сущность задач, поставленных в ходе профессиональной деятельности, использовать соответствующий физико-математический аппарат для их описания и решения |
| ПК-5 | способность работать с современным программным обеспечением, приборами и установками в избранной области |

* 1. **Перечень планируемых результатов обучения по дисциплине (модулю)**

В результате освоения дисциплины обучающиеся должны

* знать определения ключевых понятий области программного моделирования и виртуализации, технические, теоретические и экономические причины использования данных технологий при разработке цифровой аппаратуры.
* уметь для каждой поставленной практической задачи определять первоочередные и второстепенные блоки, требующие моделирования при создании модели полной системы, степень точности, требуемой для оптимального решения поставленной задачи; понимать особенности, достоинства и ограничения существующих алгоритмов моделирования устройств.
* владеть навыками создания простых моделей центральных процессоров и периферийных устройств, а также программными средами для связывания их в модели полных систем.
  1. **Содержание дисциплины (модуля)**
  2. **Разделы дисциплины (модуля) и трудоемкость по видам учебных занятий**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Тема (раздел) дисциплины (модуля) | Трудоемкость по видам учебных занятий, включая самостоятельную работу, час. | | | |
| Лекции | Семинары | Лаборат. работы | Самост. работа |
|
| 1 | Назначение и возможности программных моделей | 6 |  |  |  |
| 2 | Функциональное моделирование центрального процессора | 6 |  |  | 2 |
| 3 | Аппаратная виртуализация | 6 |  |  | 2 |
| 4 | Потактовые модели | 4 |  |  | 1 |
| 5 | Моделирование полной платформы. | 6 |  |  |  |
| 6 | Параллельные модели | 4 |  |  |  |
| 7 | Сопряжение виртуальной среды и реальности | 2 |  |  | 1 |
| Итого часов | | 34 час. |  |  | 6 час. |
| Подготовка к экзамену | | 0 час. | | | |
| Общая трудоемкость | | 40 час., 1 зач. ед. | | | |

* 1. **Содержание дисциплины (модуля), структурированное по темам (разделам)**

1. **Назначение и возможности программных моделей**. Обоснование причин использования моделиования как основного средства изучения поведения цифровых систем. Назначение и роль моделирования в процессе разработки и выпуска программно-аппаратных систем. История вопроса. Виртуализация, симуляция, эмуляция: сходства и различия этих терминов. Функциональное и потактовое моделирование как основные сценарии использования. Моделирование полных систем и отдельных приложений. Точки сохранения (checkpointing). Исполнение в обратном направлении (reverse execution). Обзор существующих программных решений: Wind River Simics, Qemu, Oracle Virtualbox, Bochs, Parallels Desktop, Microsoft Hyper-V, VMware ESX и др.
2. **Функциональное моделирование центрального процессора**. Простейшая модель центрального процессора – интерпретатор. Основной цикл работы процессора: fetch-decode-dispatch-execute-write-back. Базовый принцип построения модели процессора. Оптимизации интерпретирующих моделей: сцепленная интерпретация, кэширование результатов декодирования. Двоичная трансляция. Моделирование как компиляция программы, написанной на машинном языке гостевой архитектуры, для хозяйской архитектуры. Ограничения возможностей двоичной трансляции по сравнению с компиляцией из языков высокого уровня. Трансляция трасс и трансляция базовых блоков. Проблема самомодифицирующегося кода. Прямое исполнение. Прямое исполнение. Особенности архитектуры Intel, мешающие эффективному использованию. Виртуализационные расширения архитектуры и их использование для симуляции. Использование круга защиты 1 для режимов круга защиты 0. Изменение способа моделирования на различных участках работы системы. Моделирование архитектурного состояния, вопросы его размещения (в памяти, на хозяйских регистрах), связь с производительностью системы. Оптимизация вычисления флагов.
3. **Аппаратная виртуализация**. Теоретическое обоснование достаточных условий построения монитора виртуальных машин. Обзор существующих технологий поддержки аппаратной виртуализации. Сопоставление теории и практических решений. Разбор существующих гипервизоров. Изучение случаев, в которых аппратная виртуализация оказывается медленне программного моделирования. Совмещение виртуализации и функционального моделирования процессора для достижения более высокой производительности. Описание проблемы вложенной виртуализации подходов к ее решению.
4. **Потактовые модели**. Модель портов. Сопряжение функциональной и потактовой модели. Трассы исполнения и их использование для потактовой симуляции. Концепция портов как абстракции для построения потактового симулятора с ведущей функциональной частью. Совмещение функциональных и программный моделй.
5. **Моделировние полной платформы**. Дискретные события. Исполняющие и неисполняющие виды моделей. Очереди событий. Применимость для исполняющихся устройств. Моделирование единого времени в системе со многими устройствами. Гипер-симуляция. Моделирование многопроцессорных систем. Классический подход к моделированию многопроцессорных систем. Квант времени. Детерминизм моделей, корректность. Достигаемая скорость работы. Языки описания моделей. Стандарты SystemC и TLM. DML как пример специализированного языка. Отличия от языков моделирования аппаратуры Verilog и VHDL. Отличия от языков высокого уровня (C, C++, Java).
6. **Параллельные модели**. Расширение идеи дискретного моделирования событий. Некорректность простейшего подхода. Консервативные и оптимистические модели. Барьерная синхронизация. Схемы с точками сохранения. Детерминизм параллельных моделей. Существующие реализации.
7. **Сопряжение виртуальной среды и реальности**. Периферийные устройства и сетевое взаимодействие Последовательный порт как самое простое устройство ввода-вывода. Уровни OSI ISO с точки зрения моделирования. Сопряжение реальной и симулируемой сети: NAT, проброс портов, TUN-TAP адаптеры, точки обслуживания (service points). Паравиртуализация. Отличие паравиртуализации от аппаратно поддерживаемой модели. Различия и ограничения реализаций проброса устройств: эксклюзивное использование гостем, общее устройство. USB и VGA устройства. Другие способы связи реального мира и симуляции.
   1. **Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)**

* аудитории, оснащенные компьютерной техникой с подключением к сети «Интернет»;
* необходимое оборудование для лекций : компьютер и проектор
* необходимое программное обеспечение: MS Office Power Point.
  1. **Перечень рекомендуемой литературы**

**Основная литература:**

1. *Г.С. Речистов, Е.А. Юлюгин, А.А. Иванов и др.* Основы программного моделирования ЭВМ – Учебное пособие. 2-е изд., исп. и доп. – Издательство МФТИ, 2013. ISBN 978-5-7417-0444-8.
2. *James E. Smith, Ravi Nair.* Virtual Machines — Versatile Platforms for Systems and Processes.
3. *Hennessy, John L., and David A.* Patterson. Computer Architecture: A Quantitative Approach.2019. **Дополнительная литература:**
4. *Aarno Daniel, Engblom Jakob.* Software and System Development using Virtual Platforms, 1st Edition. — Elsevier, 2014. — ISBN: 9780128008133.
5. *Bellard, Fabrice.* QEMU, a fast and portable dynamic translator // FREENIX Track: 2005 USENIX Annual Technical Conference. 2005.
6. *Richard L. Sites, Anton Chernoff, Matthew B. Kirk et al.* Binary translation // Communications of the ACM Vol. 36, № 2. Pp. 69–81. 1993.
7. *Popek, Gerald J, Robert P. Goldberg.* Formal requirements for virtualizable third generation architectures // Communications of the ACM Vol. 17. 1974.
8. *Edouard Bugnion, Scott Devine, Mendel Rosenblum et al.* Bringing Virtualization to the x86 Architecture with the Original VMware Workstation // ACM Trans. Comput. Syst. — 2012. — Vol. 30, no. 4. — P. 1–51.
   1. **Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)**
9. Intel Corporation. Intel® 64 and IA-32 Architectures Software Developer's Manual. <https://software.intel.com/content/www/us/en/develop/articles/intel-sdm.html>
10. Oracle VM VirtualBox User Manual. <https://www.virtualbox.org/manual/UserManual.html>
11. KVM. <http://www.linux-kvm.org/page/Main_Page>
12. Intel Software Development Emulator. <https://software.intel.com/en-us/articles/intel-software-development-emulator>
    1. **Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)**

На лекционных занятиях используются мультимедийные технологии, включая демонстрацию презентаций. Для демонстрации праткических аспектов работы исследуемых систем используется программное обеспечение Intel Simics, Oracle Virtualbox.

* 1. **Методические указания для обучающихся по освоению дисциплины (модуля)**

Успешное освоение курса требует самостоятельной работы студента с источниками. Самостоятельная работа включает в себя:

* чтение и конспектирование рекомендованной литературы,
* проработку учебного материала (по конспектам лекций),
* решение задач, предлагаемых студентам на лекциях,
* подготовку к контрольной работе.

Руководство и контроль за самостоятельной работой студента осуществляется в результате периодических опросов по материалу предыдущих лекций и анализа итогов контрольной работы.

* 1. **Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся по итогам обучения**

1. Функциональные модели: назначение, общие свойства, скорость работы, ограничения в точности
2. Симуляторы уровня приложения. Отличия от полноплатформенных моделей. Уровень моделирования системных вызовов операционной системы.
3. Сравнение моделей, основанных на интерпретации и на двоичной трансляции: сложность реализации, скорость работы, сценарии, при которых интерпретация выгоднее трансляции.
4. Соотношения скоростей работы реальных и виртуальных систем.
5. Квота исполнения в многопроцессорных моделях. Опасности при излишне большом значении квоты.
6. Причины создания гибридных симуляторов.
7. Источники недетерминизма при работе параллельных симуляторов.
8. Условия возможности использования прямого исполнения моделируемого кода.
9. Возможность обратного течения времени в моделях – необходимые условия.
10. Сходства и различия компиляции языков высокого уровня и двоичной трансляции.
11. Средства динамической двоичной инструментации программ.
    1. **Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности**

Во время проведения дифференциального зачета обучающиеся могут пользоваться программой дисциплины, а также справочной литературой, вычислительной техникой, конспектами лекций. Дифференциальный зачет проводится путем организации специального опроса, проводимого в устной форме.

Программу составил: Юлюгин Е.А. «\_\_»\_\_\_\_\_\_\_\_\_ 2020 г.