

**Федеральное государственное автономное образовательное учреждение
высшего образования
"Национальный исследовательский университет
"Высшая школа экономики"**

Факультет компьютерных наук
Департамент программной инженерии

ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Конструирование ядра операционных систем

для магистерской образовательной программы «Системное программирование»
направления подготовки 09.04.04 «Программная инженерия» уровень - магистр

Разработчик программы
Хорошилов А.В, к.ф.-м.н., khoroshilov@ispras.ru

Одобрена на заседании департамента программной инженерии «__»_____ 2017 г.
Руководитель департамента Авдошин С.М. _____

Утверждена Академическим советом образовательной программы «__»_____ 2017 г.,
№ протокола _____

Академический руководитель образовательной программы Петренко А.К. _____

Москва, 2017

Настоящая программа не может быть использована другими подразделениями университета и другими вузами без разрешения подразделения-разработчика программы.

1. Область применения и нормативные ссылки

Настоящая программа учебной дисциплины устанавливает минимальные требования к знаниям и умениям студента и определяет содержание и виды учебных занятий и отчетности. Программа предназначена для преподавателей, ведущих данную дисциплину, учебных ассистентов и студентов образовательной программы «Системное программирование» направления подготовки 09.04.04 «Программная инженерия», изучающих дисциплину "Конструирование ядра операционных систем". Программа разработана в соответствии с образовательным стандартом Национального исследовательского университета «Высшая школа экономики» по направлению 09.04.04 «Программная инженерия».

2. Цели освоения дисциплины

Цель курса – Целью освоения дисциплины является 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.

Задачами данного курса являются:

- освоение студентами базовых современных достижений в области проектирования операционных систем;
- формирование практических навыков проектирования и программирования компонентов ядра операционной системы и отладки программ в привилегированном режиме работы процессора.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины студент должен:

1. Знать:

- фундаментальные понятия, теории современного системного программирования;
- 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.

2. Уметь:

- проектировать компоненты ядра операционной системы;
- программировать на языке Си и на языке ассемблера с использованием привилегированных инструкций процессора;
- отлаживать программы, работающие в привилегированном режиме работы процессора.

3. Иметь навыки (приобрести опыт):

- освоения большого объема информации;
- самостоятельной работы с документацией по архитектуре и машинным инструкциям процессора;
- культурой разработки и реализации системного программного обеспечения современных компьютеров;
- разработки и отладки программ, работающих в привилегированном режиме работы процессора.

В результате освоения дисциплины студент должен обладать следующими компетенциями:

Системные компетенции:

- Способен рефлексировать (оценивать и перерабатывать, анализировать и синтезировать) освоенные научные методы и способы деятельности для применения на практике (СК-М1).
- Способен предлагать концепции, модели, создавать и апробировать новые способы и инструменты профессиональной деятельности для применения на практике (СК-М2).
- Способен к самостоятельному освоению новых методов исследований, изменению научного и производственного профиля своей деятельности (СК-М3).
- Способен анализировать, верифицировать, оценивать полноту информации, найденной и полученной из различных источников в ходе профессиональной деятельности, при необходимости восполнять и синтезировать недостающую информацию (СК-М6).

Инструментальные компетенции:

- Способен проводить анализ, синтез, оптимизацию решений с целью обеспечения качества объектов профессиональной деятельности (ИК-М1.2.НИД (ПИ)).
- Способен планировать, управлять и контролировать выполнение требований (ИК-М2.1.АД (ПИ)).
- Способен выполнять проектную деятельность в области программной инженерии на основе системного подхода, уметь строить и использовать модели для описания и прогнозирования различных явлений, осуществлять их качественный и количественный анализ (ИК-М3.1.ПД (ПИ)).
- Способен оценить и выбрать методологию проектирования объектов профессиональной деятельности (ИК-М3.3.ПД (ПИ)).
- Способен применять современные технологии разработки программных комплексов с использованием автоматизированных систем планирования и управления, осуществлять контроль качества разрабатываемых программных продуктов (ИК-М4.1.ПТД_ПИ2 (ПИ)).

4. Место дисциплины в структуре образовательной программы

Изучение данной дисциплины базируется на знаниях, полученных студентами при освоении учебных дисциплин:

- «Дискретная математика»,
- «Программирование»,
- «Построение и анализ алгоритмов»,
- «Архитектура вычислительных систем»,
- «Операционные системы».

5. Тематический план учебной дисциплины

№	Название раздела	Всего часов	Аудиторные часы			Самостоятельная работа
			Лекции	Семинары	Практические занятия	
1.	Введение. Карта физической памяти x86. Процесс загрузки и инициализации PC. BIOS, инициализация основных устройств. Загрузчик JOS. Загрузка ядра.	12	2	0	2	8
2.	Устройство ядра JOS. Отладка кода ядра JOS. Компиляция первой собственной функции, вывод строк на консоль.	12	2	0	2	8
3.	Описатели процессов в JOS. Создание процессов в JOS, загрузка приложений в память из бинарных секций образа ядра. Переключение контекстов. Кооперативное разделение времени. Примитивный планировщик FIFO без приоритетов.	20	2	0	4	14
4.	Прерывания в x86. Инициализация IDT. Обработка прерываний таймера. Вытесняющее разделение времени. Примитивный планировщик Round Robin без приоритетов.	22	2	0	4	16
5.	Обработка вложенных прерываний в x86. Средства синхронизации, состояние гонок, дедлоки. Запрет прерываний, семафоры.	22	2	0	4	16
6.	Управление распределением физических страниц. Виртуальная память. Сегментная и страничная трансляция. Таблицы трансляции.	24	2	0	6	16
7.	Переключение между режимами работы процессора. Прерывания и системные вызовы. Вложенные прерывания. Изменения в создании процессов, переключении между контекстами. Передача данных между программой и ядром.	26	4	0	6	16
8.	Управление процессами. Системный вызов fork(). Механизмы межпроцессного взаимодействия.	26	4	0	6	16
9.	Примитивная файловая система. Реализация системных вызовов open(), close(), read(), write(), exec().	26	4	0	6	16
Итого:		190	24	0	40	126

6. Формы контроля знаний студентов

Тип контроля	Форма контроля	4 год			
		1 модуль	2 модуль	3 модуль	4 модуль
Текущий (месяц)		1-6 неделя	1-6 неделя	1-6 неделя	1-6 неделя
	Лабораторные работы			*	*
	Домашняя работа				*
Итоговый	Экзамен				*

Критерии оценки знаний, навыков

В рамках курса слушателям предлагается выполнить 12 лабораторных работ. Каждая лабораторная работа сдаётся преподавателю или ассистенту посредством демонстрации результатов и устного ответа на вопросы. За сдачу лабораторных работ в срок начисляются баллы (максимум 33 балла). Домашняя работа заключается в выполнении индивидуального задания, которое можно получить после сдачи 9-ой лабораторной работы (максимум 42 балла). Оценка за экзамен выставляется по 25-ти балльной шкале.

Порядок формирования оценок по дисциплине

Оценка по курсу состоит из оценки за выполнение лабораторных работ $O_{\text{лаб}}$ (33 балла), домашней работы (42 балла) и оценки за итоговый устный экзамен (25 баллов). В диплом выставляется результирующая оценка по учебной дисциплине, которая формируется по следующей формуле:

$$O_{\text{результ}} = 0,1 * O_{\text{лаб}} + 0,1 * O_{\text{дом}} + 0,1 * O_{\text{экс}}$$

7. Содержание дисциплины

Тема 1. Введение. Карта физической памяти x86. Процесс загрузки и инициализации персональных ЭВМ архитектуры x86. Задачи, решаемые BIOS, инициализация основных устройств. Функции загрузчика. Загрузка ядра. Эмулятор ЭВМ Qemu.

Тема 2. Виды архитектур ядра ОС. Монолитные и микроядерные архитектуры. Устройство ядра JOS.

Тема 3. Описатели процессов в JOS. Создание процессов в JOS, загрузка приложений в память из бинарных секций образа ядра. Переключение контекстов. Кооперативное разделение времени. Примитивный планировщик FIFO без приоритетов.

Тема 4. Прерывания в x86. Инициализация IDT. Обработка прерываний таймера. Вытесняющее разделение времени. Примитивный планировщик Round Robin без приоритетов. Обзор алгоритмов планирования процессорного времени.

Тема 5. Обработка вложенных прерываний в x86. Средства синхронизации, состояние гонок, дедлоки. Запрет прерываний, спинлоки, мьютексы, семафоры. Read-Copy-Update.

Тема 6. Управление распределением физических страниц. Виртуальная память. Сегментная и страничная трансляция x86. Таблицы трансляции.

Тема 8. Переключение между режимами работы процессора. Прерывания и системные вызовы. AMD syscall/sysreturn и Intel sysenter/sysexit. Выполнение системных вызовов без переключения в привелигированный режим (VDSO - Virtual Dynamically linked Shared Objects).

Тема 9. Управление процессами. Системные вызовы fork() и exec(). Механизмы межпроцессного взаимодействия. Обзор механизмов межпроцессного взаимодействия в ОС Linux (сигналы, разделяемая память, семафоры, очереди сообщений, программные каналы, сетевые интерфейсы, взаимодействие на основе файловых систем, KDBUS).

Тема 10. Файловые системы. Основные задачи файловых систем. Организации работы файловых систем в ОС Linux: VFS, драйвера файловых систем, bio, кэширование. Обзор современных файловых систем ОС Linux (ext4, btrfs, XFS, jffs2, f2fs). Прimitивная файловая система JOS. Реализация системных вызовов open(), close(), read(), write(), exec().

Тема 11. Механизмы и виды виртуализации. Аппаратная поддержка виртуализации (Intel VT-x и AMD AMD-V). Вложенные таблицы трансляции и идентификаторы виртуальных процессоров.

8. Оценочные средства для текущего контроля и аттестации студента

1.1 Тематика заданий текущего контроля

1. Процесс загрузки и инициализации персональных ЭВМ архитектуры x86. Функции загрузчика. Загрузка ядра JOS.
2. Переключение контекстов. Кооперативное разделение времени.
3. Прерывания в x86. Инициализация таблицы дескрипторов прерываний IDT.
4. Обработка прерываний таймера. Вытесняющее разделение времени. Алгоритмы планирования процессорного времени.
5. Обработка вложенных прерываний в x86. Средства синхронизации, состояние гонок, взаимные блокировки. Запрет прерываний, спинлоки, мьютексы, семафоры. Read-Copy-Update.
6. Виртуальная память. Модели организации оперативной памяти. Сегментная и страничная трансляция x86. Таблицы трансляции.
7. Переключение между режимами работы процессора. Прерывания и системные вызовы. Выполнение системных вызовов без переключения в привилегированный режим.
8. Аппаратно-программные средства поддержки мультипрограммного режима – система прерываний, защита памяти, привилегированный режим.
9. Файловые системы. Основные задачи файловых систем.

1.2 Примеры контрольных вопросов для экзамена

1. Как обеспечивается защита памяти приложений друг от друга? других ресурсов? Как обеспечивается взаимодействие приложений друг с другом?
2. Как обеспечивается защита памяти ОС от приложений? других ресурсов? Как обеспечивается взаимодействие приложений с ОС?
3. Что такое системный вызов? Какими способами можно реализовать этот механизм?
4. Как возможно выполнить системный вызов без переключения в привилегированный режим?
5. Какие средства синхронизации применяются в операционных системах?
6. Как происходит переключение процессов? Как передается управление от процесса к ядру, от ядра к процессу?
7. Как процессор транслирует виртуальные адреса в физические? Каким образом ядро контролирует этот процесс?

9. Учебно-методическое и информационное обеспечение дисциплины

1.3 Базовый учебник

1. Д.В. Ефремов, Н.Ю. Комаров, А.В. Хорошилов. "Конструирование ядра операционной системы", Издательский отдел факультета ВМК МГУ имени М.В. Ломоносова, Москва, 2015. ISBN: 978-5-89407-549-5

1.4 Основная литература

1. Э. Таненбаум "Архитектура компьютера", Питер, 2012 г.
2. Э. Таненбаум "Современные операционные системы", Питер, 2011 г.
3. Э. Таненбаум "Операционные системы. Разработка и реализация", Питер, 2006 г.
4. А. Робачевский, С. Немнюгин, О. Стесик «Операционная система UNIX», БХВ Санкт-Петербург, 2010 г.

1.5 Дополнительная литература

1. C. A. R. Hoare, Monitors: An Operating System Structuring Concept. Communications of the ACM, 17(10):549--557, 1974.
2. Thomas W. Doeppner "Operating Systems In Depth: Design and Programming" Wiley, 2010
3. Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman. "Linux Device Drivers".
4. Ellen Siever, Stephen Figgins, Robert Love, Arnold Robbins. "Linux in a Nutshell". O'Reilly Media, 2009.
5. Скотт Шакон. «Про Git». Apress. Перевод на русский. (<http://git-scm.com/book/ru>).
6. Веб-страница дисциплины: <http://forge.ispras.ru/projects/oscourse-hse-YEAR/wiki>
7. IA-32 Intel Architecture Software Developer's Manual Volume 1: Basic Architecture. Basic 80x86 architecture and programming environment. <http://developer.intel.com>.
8. 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>.
9. Tool Interface Standard Executable and Linking Format (ELF) Specification Version 1.2.