

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Саратовский государственный университет имени Н.Г. Чернышевского

Факультет компьютерных наук и информационных технологий

УТВЕРЖДАЮ

" " _____ 20__ г.

Рабочая программа дисциплины

ОПЕРАЦИОННЫЕ СИСТЕМЫ

Направление подготовки

230100 Информатика и вычислительная техника

Профиль подготовки

Вычислительные машины, комплексы, системы и сети

Квалификация (степень) выпускника

Бакалавр

Форма обучения

очная

Саратов,
2011 год

1. Цели освоения дисциплины

Целями освоения дисциплины «Операционные системы» являются обучение студентов систематизированному представлению о принципах построения, проектирования и функционирования современных операционных систем (ОС), сред и систем программирования, и практическим навыкам создания и использования эффективного программного обеспечения для управления вычислительными ресурсами в многопользовательских ОС.

2. Место дисциплины в структуре ООП бакалавриата

Данная учебная дисциплина входит в раздел «Профессиональный цикл. Базовая часть» ФГОС-3.

Для изучения дисциплины необходимы компетенции, сформированные у обучающихся в результате изучения дисциплин «Математическая логика и теория алгоритмов», «Технологии программирования».

Сформированные в процессе изучения дисциплины «Операционные системы» компетенции необходимы студенту при изучении дисциплин «Системное программное обеспечение», «Методы и средства защиты компьютерной безопасности», «Параллельное программирование», «Системы реального времени».

3. Компетенции обучающегося, формируемые в результате освоения дисциплины «Операционные системы»

Данная дисциплина способствует формированию следующих компетенций:

- стремится к саморазвитию, повышению своей квалификации и мастерства (ОК-6);
- разрабатывать интерфейсы «человек - электронно-вычислительная машина» (ПК-3);
- готовить конспекты и проводить занятия по обучению сотрудников и применению программно-методических комплексов, используемых на предприятии (ПК-8);
- участвовать в настройке и наладке программно-аппаратных комплексов (ПК-9);
- устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем (ПК-11);
- понимание концепций, базовых алгоритмов, принципов функционирования современных операционных систем и систем реального времени (ПК-16);
- владение методами и навыками использования и конфигурирования операционных систем и платформенных окружений (ПК-17).

В результате освоения дисциплины обучающийся должен:

Знать:

- принципы управления вычислительными ресурсами в микропроцессорных системах;
- принципы организации и функционирования аппаратно-программных комплексов под управлением операционными системами;
- принципы проектирования и реализации драйверов;
- особенности управления вычислительными ресурсами в режиме разделения времени;
- понятия процессов, потоков и контекстов исполнения, модели их организации и алгоритмы планирования;
- организацию и основные модели управления оперативной памятью, а также принципы организации виртуальной памяти;
- принципы работы механизмов синхронизации межпроцессного взаимодействия;
- принципы контроля вычислительных процессов и обеспечения безопасности в многопользовательских операционных системах;
- основные программные приёмы по наладке программно-аппаратных комплексов под управлением операционными системами.

Уметь:

- использовать основы системного подхода и критерии эффективной организации вычислительного процесса для постановки и решения задач организации оптимального функционирования вычислительных систем;
- обоснованно выбирать оптимальные алгоритмы управления вычислительными ресурсами;
- применять системные средства операционных систем при разработке приложений;
- пользоваться сервисными функциями ОС и специализированными утилитами для при оценке качества функционирования алгоритмов управления вычислительными ресурсами.

Владеть:

- методами и средствами реализации приложений на уровне взаимодействия с системными вызовами операционной системы;
- методами и средствами отладки приложений средствами операционной системы и специализированных утилит;
- методами проектирования и реализации драйверов устройств;
- методами и средствами защиты информации, предоставляемыми ОС
- методами и средствами программной реализации параллельных взаимодействующих процессов.

4. Структура и содержание дисциплины «Операционные системы»

Общая трудоемкость дисциплины составляет 5 зачетных единиц, 180 часов.

№ п/п	Раздел дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)			Формы текущего контроля успеваемости (по неделям семестра) Формы промежуточной аттестации (по семестрам)
				Лекции	Лабораторная работа	Самостоятельная работа	
1	2	3	4	5	6	7	8
1	Основные принципы построения ОС	4	1	1		1	
2	Эволюция ОС	4	1	1	2	2	
3	Структурные компоненты ОС	4	2	2	4	4	
4	Модели процесса, потока и контекста исполнения	4	3	2	6	4	
5	Управление памятью и модель виртуальной памяти	4	4	2	4	4	
6	Архитектуры ОС	4	5	1	2	2	
7	Обработка прерываний в операционных системах	4	5	1	2	4	
8	Проблемы синхронизации и механизмы межпроцессного взаимодействия	4	6	2	6	6	
9	Управление вводом-выводом	4	7 - 8	4	4	6	Контрольная работа № 1 на 8 неделе
10	Файловые системы	4	9	2	2	4	
11	Управление и планирование процессами, потоками и контекстами исполнения	4	10	2	2	4	
12	Принципы создания параллельных взаимодействующих процессов	4	11	2	4	4	
13	Особенности реализаций	4	12 - 13	4	4	4	

	современных ОС						
14	Форматы исполняемых файлов и загрузка программ	4	14	2	2	2	
15	Модель сетевых интерфейсов и поддержка сетевых служб в ОС	4	15	2	2	2	
16	Защитные механизмы ОС	4	16	2	2	2	Контрольная работа № 2 на 16 неделе
Промежуточная аттестация							Экзамен
ИТОГО:			180	32	48	55	45

РАЗДЕЛ 1. Основные принципы построения ОС

Тема 1.1. Определение операционной системы

Роль операционной системы (ОС) в общей структуре аппаратно-программного обеспечения вычислительных систем. ОС как виртуальная машина, ОС как система управления ресурсами.

Тема 1.2. Основные принципы построения ОС

Принцип модульности. Принцип функциональной избыточности. Принцип генерируемости ОС. Принцип функциональной избирательности. Принцип виртуализации. Принцип независимости программ от внешних устройств. Принцип совместимости. Принцип открытой и наращиваемой ОС. Принцип переносимости. Принцип обеспечения безопасности вычислений. Принцип разделения политик и механизмов.

Тема 1.3. Требования, предъявляемые к многопользовательским ОС

Мультипрограммность и многозадачность. Отзывчивость

РАЗДЕЛ 2. Эволюция ОС

Тема 2.1. Первые поколения вычислительных машин

Лампы и коммутационные панели. Транзисторы и системы пакетной обработки. Задания, управляемые оператором, и управляющие программы.

Тема 2.2. Появление первых многозадачных ОС

Интегральные схемы, увеличение объёма доступной оперативной памяти и появление первых мультипрограммных ОС для мэйнфреймов.

Тема 2.3. Операционные системы с разделением времени

Режим разделения времени. ОС CTSS (Compatible Time Sharing System — совместимая система разделения времени) и MULTICS (MULTIplexed Information and Computing Service — мультиплексная информационная и вычислительная служба), как прототипы UNIX. Первые две основных версии UNIX — System V

(корпорация AT&T) и BSD (Berkley Software Distribution). Разработка первого стандарта POSIX.

Тема 2.4. Эпоха персональных компьютеров и встроенных устройств

Микропроцессорные системы и виды управляющих ими ОС. Серверные ОС. Многопроцессорные операционные системы. Системы с разделением времени. Операционные системы для персональных компьютеров. Операционные системы реального времени. Встроенные операционные системы. Особенности современного этапа развития ОС.

РАЗДЕЛ 3. Структурные компоненты ОС

Тема 3.1. Операционная система, как операционная среда и оболочка

Понятие операционной среды. Операционная система, как оболочка для запуска приложений.

Тема 3.2. Основные понятия ОС

Понятие вычислительного процесса и ресурса. Реализация понятия последовательного процесса в ОС. Функциональные компоненты ОС. Управление прерываниям. Управление процессами. Управление памятью. Управление файлами и внешними устройствами.

Тема 3.3. Взаимодействие с ОС и понятие системного вызова

Взаимодействие процессов с операционной системой, системные вызовы. Системные вызовы для управления процессами. Системные вызовы для управления файлами. Системные вызовы для управления каталогами. Стандарты на интерфейсы к ядру ОС.

РАЗДЕЛ 4. Модели процесса, потока и контекста исполнения

Тема 4.1. Модели контекстов исполнения.

Понятие процесса и потока. Модель процесса. Модель потока. Модель контекстов исполнения, как параллельно исполняемых процессов с общим набором ресурсов. Модель потока, как контекста исполнения с заданным набором общих ресурсов.

Тема 4.2. Ресурсы разделяемые контекстами исполнения

Виды общих ресурсов: оперативная память, файловые дескрипторы, переменные окружения, сигналы. Ресурсы потока и контекста исполнения: идентификатор, набор регистров, стек, код завершения. Идентификация процессов и потоков, как контекстов исполнения.

РАЗДЕЛ 5. Управление памятью и модель виртуальной памяти

Тема 5.1. Функции ОС по управлению памятью.

Проблемы управления памятью. Фрагментация. Функции ОС по управлению памятью. Алгоритмы распределения памяти. Простое непрерывное распределение и распределение с перекрытием (оверлейные структуры). Распределение статическими и динамическими разделами. Разделы с фиксированными и подвижными границами.

Тема 5.2. Виртуальная память

Сегментный, страничный, сегментно-страничный виды организации управления памятью. Виртуальная память, как модель страничной организации управления памятью, виртуальное адресное пространство. Алгоритмы замещения страниц. Механизмы и алгоритмы подкачки страниц. Алгоритмы обработки запросов на выделение памяти. Защита памяти.

Тема 5.3. Особенности управления памятью в пространстве ядра

Разделение физической памяти на зоны. Страничная организация памяти в пространстве ядра. Механизмы кеширования памяти.

Тема 5.4. Особенности управления памятью в пространстве пользователя

Адресное пространство процесса. Области памяти в адресном пространстве процесса и их типы.

РАЗДЕЛ 6. Архитектуры ОС

Тема 6.1. Понятие ядра ОС

Ядро и вспомогательные модули ОС. Ядро в привилегированном режиме. Ограничивающие ресурсы. Совместимость и множественные прикладные среды. Способы реализации прикладных программных сред.

Тема 6.2. Основные архитектуры ОС

Монолитные и микроядерные архитектуры ОС. Концепция микроядра, преимущества и недостатки. Распределение и использование ресурсов в ОС.

Тема 6.3. Другие виды архитектур

Многоуровневые ОС. Виртуальные машины. Экзоядро. Распределённые, клиент-серверные архитектуры.

РАЗДЕЛ 7. Обработка прерываний в ОС

Тема 7.1. Аппаратные средства управления прерываниями

Контроллеры прерываний. Вектора обработчиков прерываний. Совместное

использование прерываний несколькими устройствами. Прерывания в мультипроцессорных системах.

Тема 7.2. Механизмы обработки прерываний

Отложенные прерывания и нижние половины. Контекст прерываний и особенности синхронизации нижних половин. Спин-блокировки и их разновидности.

РАЗДЕЛ 8. Проблемы синхронизации и механизмы межпроцессного взаимодействия

Тема 8.1. Основная проблема синхронизации

Атомарные и не атомарные операции. Необходимость синхронизации, состояние состязания или «гонок». Понятие критической секции. Понятия повторно используемых, системных и потребляемых ресурсов.

Тема 8.2. Основная проблема синхронизации

Отложенные прерывания и нижние половины. Контекст прерываний и особенности синхронизации нижних половин

Тема 8.3. Механизмы синхронизации и примитивы межпроцессного взаимодействия

Запрет прерываний и взаимное исключение с активным ожиданием. Семафоры Дейкстры. Мьютексы. Мониторы Хоара. Условные переменные. Передача сообщений. Барьеры. Аппаратные средства поддержки реализации примитивов синхронизации.

Тема 8.4. Проблемы использования механизмов синхронизации

Классические проблемы межпроцессного взаимодействия. Взаимоблокировки или тупиковые ситуации. Обнаружение, избежание и предотвращение тупиков.

РАЗДЕЛ 9. Управление вводом-выводом

Тема 9.1. Принципы программного обеспечения ввода-вывода

Управляемый прерываниями ввод-вывод. Ввод-вывод с использованием механизмов прямого доступа к памяти. Программные уровни ввода-вывода.

Тема 9.2. Организация связи ядра ОС с драйверами

Символьные и блочные устройства ввода-вывода. Очереди запросов блочного ввода-вывода. Подсистема буферизации и планировщики ввода-вывода. Алгоритмы планирования ввода-вывода.

Тема 9.3. Управление устройствами ввода-вывода

Символьные устройства и терминалы. Устройства графического ввода-вывода и графические интерфейсы пользователя. Сетевые интерфейсы и устройства ввода-вывода.

Тема 9.4. Управление специальными устройствами

Программные и аппаратные таймеры. Программные интерфейсы управления таймерами. Управление энергосбережением.

РАЗДЕЛ 10. Файловые системы

Тема 10.1. Файлы

Именованые файлов. Типы файлов. Доступ к файлам и списки контроля доступа. Атрибуты файлов.

Тема 10.2. Каталоги

Дерево файлов и каталогов ОС. Операции с файлами и каталогами. Имена путей к файлам.

Тема 10.3. Реализация файловой системы

Виртуальная файловая система и дерево объектов ОС. Реализация файловой системы, как структуры данных. Виды файловых систем.

РАЗДЕЛ 11. Управление и планирование процессами, потоками и контекстами исполнения

Тема 11.1. Управление и реализация процессов и потоков

Создание и завершение процессов. Состояния процессов. Иерархия процессов. Создание и завершение потоков. Создание потоков, как контекстов исполнения. Планирование и диспетчеризация. Состояния потока, процесса и контекста исполнения. Реализация процессов и потоков, как контекстов исполнения с общим набором ресурсов.

Тема 11.2. Планирование и реализация многозадачности

Вытесняющая и кооперативная многозадачность. Вытесняющие и невытесняющие алгоритмы планирования. Задачи ограниченные скоростью ввода-вывода и скоростью процессора. Стратегии и алгоритмы планирования. Алгоритмы планирования, основанные на квантовании. Алгоритмы планирования, основанные на приоритетах. Дисциплины диспетчеризации. Качество диспетчеризации и гарантии обслуживания. Диспетчеризация задач с использованием динамических приоритетов. Балансировка нагрузки в многопроцессорных системах. Моменты перепланировки.

РАЗДЕЛ 12. Принципы создания параллельных взаимодействующих процессов

Тема 12.1. Задачи параллельных взаимодействующих процессов

Независимые и взаимодействующие вычислительные процессы и потоки. Цели и средства синхронизации.

Тема 12.2. Реализация параллельных взаимодействующих процессов

Критические секции в приложениях. Блокирующие переменные. Средства синхронизации и связи при проектировании взаимодействующих вычислительных процессов. Использование блокировки памяти при синхронизации. Использование семафоров и мьютексов при проектировании взаимодействующих вычислительных процессов и потоков. Ожидание и передача событий при использовании условных переменных.

РАЗДЕЛ 13. Особенности реализаций современных ОС

Тема 13.1. Программные интерфейсы современных операционных систем.

Программные интерфейсы и системные вызовы. Стандарты на программные интерфейсы ОС. Состав базовых системных вызовов стандарта POSIX. Распределение оперативной памяти в реализациях современных ОС.

Тема 13.2. ОС, как среда взаимодействующих процессов

Классический подход к построению операционной системы как операционной среды взаимодействующих процессов. Понятие операционной оболочки. Языки пакетной обработки, как оболочки системы. Аргументы команд, стандартные потоки ввода-вывода, перенаправление ввода-вывода.

Тема 13.3. Рассмотрение конкретных реализаций ОС

Семейство операционных систем UNIX. Особенности архитектуры и межпроцессного взаимодействия в ОС семейства UNIX. Операционные системы GNU/Linux. Операционная система OpenSolaris. Семейство операционных систем Microsoft Windows NT. Операционная система Minix 3 на основе микроядра.

РАЗДЕЛ 14. Форматы исполняемых файлов и загрузка программ

Тема 14.1. Форматы исполняемых файлов

Исполняемые файлы. Стандарты на форматы исполняемых файлов. Форматы файлов ELF, COFF и PE, LE.

Тема 14.2. Загрузка исполняемых файлов

Связь формата исполняемых файлов и логикой загрузки и запуска приложений. Динамические загружаемые библиотеки, как разделяемые объекты

кода в памяти.

РАЗДЕЛ 15. Модель сетевых интерфейсов и поддержка сетевых служб в ОС

Тема 15.1. Концепции распределённой обработки

Принципы идентификации узлов и приложений. Механизмы передачи сообщений. Буферизация и синхронизация примитивов передачи сообщений.

Тема 15.2. Реализация сетевого взаимодействия

Механизм сокетов Беркли. Модели сетевых служб и распределённых приложений. Сетевые файловые системы.

РАЗДЕЛ 16. Защитные механизмы ОС

Тема 16.1. Локальные механизмы контроля доступа

Понятия аутентификации и авторизации. Механизмы контроля доступа к ресурсам на основе идентификаторов владельцев, запущенных процессов. Списки контроля доступа.

Тема 16.2. Модели управления доступом

Избирательное управление доступом. Управление доступом на основе ролей. Мандатное управление доступом.

Тема 16.3. Сетевые механизмы контроля доступа

Протоколы сетевой аутентификации. Использование несимметричных алгоритмов шифрования для организации безопасной передачи сообщений.

На лабораторных занятиях студенты получают индивидуальные и групповые задания, связанные с тематикой соответствующей занятию недели, пример которых приведен в разделе 6 настоящей программы. Задания выполняются в компьютерном классе с использованием программного обеспечения, указанного в разделе 7. Результатом выполнения индивидуальных и групповых заданий являются документация с описанием результатов работы, образы виртуальных машин, в которых сохранены текущие установки и состояние исследуемых операционных систем, а также программный код, представленный в системе контроля версий.

5. Образовательные технологии

В учебном процессе, при реализации компетентного подхода, используются такие активные и интерактивные формы проведения занятий как модельный метод обучения, метод развивающей кооперации, разбор конкретных

ситуаций, командное выполнение заданий с распределением ролей, тестирование. Широко используются мультимедийные презентации при представлении лекционного материала, а также технологии для совместного взаимодействия через интернет.

6. Учебно-методическое обеспечение самостоятельной работы студентов.

Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины

Самостоятельная работа студентов заключается в углубленном изучении материала курса по соответствующей тематике недели с использованием научной и учебно-методической литературы. В рамках самостоятельной работы студент также готовит отчет о проделанных лабораторных работах.

Примеры тем для курсовых работ для дисциплины «Операционные системы»

- 1 Реализация быстрых механизмов синхронизации в пространстве пользователя на примере системного вызова `futex()` в ядре Linux. *Работа направлена на исследование механизмов синхронизации между контекстами исполнения с общей памятью, например потоками, не требующих, для проверки занятости ресурса в критической секции, переключения контекста, в связи обращением к ядру через механизмы системных вызовов;*
- 2 Исследование механизмов аутентификации и авторизации в GNU/Linux через сетевую службу Active Directory под управлением Windows Server. *Работа направлена на исследование механизмов локальной проверки пароля при входе в систему и получению авторизационной информации о пользователях и группах через сетевую службу Active Directory;*
- 3 Влияние алгоритмов планирования ввода-вывода на производительность вычислительной системы. *Целью работы является исследование влияния различных планировщиков ввода-вывода в Linux на производительность вычислительной системы, поиск компромиссных вариантов настройки операционной системы в зависимости, решаемых задач;*
- 4 Реализация работы с образом виртуального жёсткого диска VirtualBox в виде файловой системы в пространстве пользователя. *Исследование направлено на изучение формата образа файловой системы и механизмов реализации файловой системы в виде службы. Использование такого механизма предусмотрено модулем FUSE под Linux, и применяется, в частности, для создания зашифрованных файловых систем.*
- 5 Особенности выбор компилятора для сборки ядра операционной системы. *Работа направлена на выявление текущих тенденций в области использования компиляторов, пригодных или не пригодных по тем или иным причинам для сборки ядер операционных систем. Предполагается исследовать особенности тех версий семейства компиляторов GCC,*

которые используются для сборки ядер ОС в текущих выпусках дистрибутивов GNU/Linux, исследовательской ОС Minix 3 и других Unix-подобных систем и оценить причины и критерии, на основании которых выбраны именно эти версии компилятора.

- 6 *Использование системного кеша ключей для доступа к сетевым службам. В работе предполагается исследование возможностей пакета `keyutils` и кеша удостоверений в ядре Linux для доступа различных приложений к различным сетевым службам, используя одни и те же ключи доступа.*
- 7 *Реализация загрузки исполняемого модуля из памяти. Работа направлена на изучение механизмов загрузки динамически загружаемых библиотек. В работе необходимо исследовать особенности процесса загрузки модулей исполняемого кода из динамических библиотек, описать и реализовать упрощённый вариант этого механизма для модуля размещённого в данных или полученного по сети. В работе также можно рассмотреть альтернативные варианты загрузки исполняемого кода в память. Работа может выполняться как под Linux, так и под Windows.*
- 8 *Реализация загрузки динамической библиотеки в формате PE под Linux. Цель работы состоит в изучении формата запускаемых файлов Windows и механизмов размещения и запуска загруженного кода библиотеки в GNU/Linux. Примером применения аналогичной задачи является загрузка под Linux видео кодеков для Windows.*
- 9 *Сборка пакета со специализированным ядром в операционной системе GNU/Linux. Целью работы является исследование процесса конфигурирования, сборки и ядра операционной системы под задачу (уменьшение объёма занимаемого пространства в памяти, поддержка расширенного режима адресации памяти 32-битном режиме работы,), а также упаковки результат в виде тиражируемого решения.*
- 10 *Реализация альтернативного механизма входа в систему Windows. Работа направлена на исследование механизмов локальной аутентификации и авторизации в семействе ОС Windows NT, основанной на использовании расширяемого набора модулей, используемых через интерфейс библиотеки установки контекста для запускаемых процессов GINA.*
- 11 *Исследование современных реализаций системы запуска ОС GNU/Linux. В работе предполагается разобрать традиционный подход к загрузке ОС SystemV, провести его сравнение с современными альтернативами (`upstart`, `initng`, `systemd`) и дать оценку рассмотренным альтернативам в соответствии с теми задачами, для которых применение этих альтернатив оправдано.*
- 12 *Исследование специализированных возможностей ядра Linux. Работа направлена на исследование специализированных возможностей предоставляемых ядром Linux, но редко используемых традиционных в приложениях с силу кроссплатформенности и совместимости, которой придерживаются*

разработчики приложений и GNU/Linux дистрибутивов. Предполагается рассмотреть системный вызов создания контекста исполнения `clone()`, средства управления привилегиями (`capabilities`), средство группового управления ресурсам процессов (`cgroups`), виртуализатор уровня ядра, использующий механизмы аппаратной поддержки виртуализации (`KVM`), модуль поддержки запуска сторонних форматов исполняемых файлов (`binfmt_misc`), расширяемый механизм уведомлений о событиях (`epoll`), быстрый механизм синхронизации процессов (`futex` — *fast mutex*), модуль создания файловой системы в пространстве пользователя в виде службы (`FUSE`), расширенный механизм контроля доступа (`SELinux`), Механизм быстрой перезагрузки ядра (`kexec`) и т.д.

- 13 Исследование специализированных возможностей ОС OpenSolaris. Работа направлена на исследование специализированных возможностей предоставляемых ОС OpenSolaris. Предполагается рассмотреть файловую систему ZFS, предоставляющую широкие возможности для резервного копирования, распределения дисковых ресурсов и управления их состоянием, встроенный механизм динамической трассировки приложений уровня ядра ОС (`DTrace`), технология виртуализации на уровне операционной системы (`Solaris Containers`), механизм создания множества виртуальных сетевых интерфейсов (`Crossbow`) и т.д.
- 14 Исследование специализированных возможностей ОС FreeBSD. Работа направлена на исследование специализированных возможностей предоставляемых ОС FreeBSD. Модули совместимости с приложениями, предназначенным для выполнения в других ОС (включая Linux, SCO UNIX, NetBSD и BSD/OS), поддержка файловой системы ZFS, предоставляющей широкие возможности для резервного копирования, распределения дисковых ресурсов и управления их состоянием, поддержка встроенного механизма динамической трассировки приложений уровня ядра ОС (`DTrace`), механизм виртуализации уровня операционной системы (`Jail`) и т.д.
- 15 Исследование расширяемого микропрограммного интерфейса EFI. Работа направлена на исследование интерфейса между операционной системой и микропрограммами (`EFI` — *Extensible Firmware Interface*), управляющими низкоуровневыми функциями оборудования, предназначенного корректно инициализировать оборудование при включении системы и передать управление загрузчику операционной системы. EFI предназначен на замену BIOS — традиционного интерфейса используемого всеми IBM PC-совместимыми персональными компьютерами.
- 16 Современный формат размещения таблиц разделов GPT. Работа направлена на исследование нового формата таблицы разделов (`GUID Partition Table`), являющегося частью расширяемого микропрограммного интерфейса (`EFI`). Новый формат таблицы разделов предназначен в качестве замены главной загрузочной записи (`MBR`).

- 17 Программные интерфейсы и режимы работы современных графических адаптеров. Работа направлена на исследование программных интерфейсов современных графических адаптеров, реализуемых в драйверах ОС, и режимов работы, определяющих отображение памяти графических адаптеров в адресное пространство управляющих модулей ядра и приложений.
- 18 Двоичный интерфейс приложений x32-psABI для архитектуры x86_64 в ОС GNU/Linux. Работа направлена на исследование реализации режима поддержки двоичного интерфейса x32-psABI в ядре Linux и базовых библиотеках ОС, в котором используются практически все преимущества x86_64, но предлагает 32-битный размер указателей, что позволяет сократить объём используемой оперативной памяти в два раза.
- 19 Реализация Wine-сервера в ядре Linux. Исследование решения по переносу реализации службы поддержки среды для запуска Windows-приложений в Unix-подобных операционных системах в ядро Linux. Реализация службы wine-сервер позволяет уменьшить накладные расходы на трансляцию системных вызовов, за счёт уменьшения числа переключений контекста процессов, и ускорить работу запускаемых приложений в несколько раз. В результате, некоторые приложения смогут работать быстрее, чем в родной операционной системе.
- 20 Архитектура улучшенного программируемого контроллера прерываний APIC. Работа направлена на исследование возможностей предоставляемых улучшенным программируемым контроллером прерываний (Advanced Programmable Interrupt Controller) для многопроцессорных систем, а также изучение работы контроллера на примере модулей использующих его ОС.
- 21 Реализация архитектурно-зависимого приложения в ОС Android. Работа направлена на исследование интерфейса для реализация «родных» приложений (NDK — native development kit) в ОС Android, с использованием. Основной целью использования этого интерфейса, обычно, является реализация высокоэффективного кода игровых приложений, способного запускаться на современных мобильных телефонах, планшетных компьютерах, нетбуках и смартбуках.
- 22 Объектная модель ядра Linux и её представление в файловой системе /sys. Работа направлена на исследование объектной модели используемой для организации внутренней структуры ядра и её динамического отображения в пространство пользователя через файловую систему /sys. Результатом работы может быть набор модулей ядра, дающих представление об использовании структур ядра в драйверах устройств.
- 23 Установка GNU/Linux с корневой файловой системой JFFS2. Работа направлена на изучение файловой системы для флеш-накопителей JFFS2, и возможностям её использования в качестве корневой файловой системы для ОС GNU/Linux.

- 24 Алгоритм вычисления списка пакетов по замыканию дерева сборочных зависимостей для заданного множества пакетов. *Работа направлена на исследование системы установки ПО в ОС GNU/Linux, применяющих пакетные менеджеры с зависимостями. Целью работы является составление эффективного алгоритма по вычислению минимального списка пакетов необходимых для пересборки заданного множества.*
- 25 Конфликты компоновки динамически загружаемых библиотек на уровне двоичных интерфейсов. *В работе рассматривается проблема загрузки в память одной и той же библиотеки разных версий с разными бинарными интерфейсами. Проблема, в основном, встречается при не явной загрузке модулей расширения скомпонованных с разными версиями библиотек. Целью работы является описание причин, определение потенциальных последствий, проблемы и варианты её избежания.*

Примеры тем для лабораторных работ по курсу «Операционные системы»

- 1 Создание процессов и запуск приложений с помощью системных вызовов
- 2 Изучение механизмом межпроцессного взаимодействия (не именованные каналы, именованные каналы, локальные сокеты, общая память, очереди сообщений)
- 3 Создание скриптов для командной оболочки, изучение передачи аргументов командам и приложениям
- 4 Реализация событийно-ориентированных приложений с асинхронным вводом-выводом на основе цикла обработки сообщений
- 5 Изучение программных интерфейсов работы с потоками соответствующих примитивов синхронизации
- 6 Реализация модулей ядра (системных модулей, загружаемых в пространство ядра, и драйверов)

Примеры заданий для лабораторных работ по курсу «Операционные системы»

- 1 Исследовать и разобрать особенности работы учебного образа жёсткого диска операционной системы Linux в виртуальной машине VirtualBox. В ходе работы необходимо выполнить следующие задания:
 - освоить базовые команды и утилиты работы в командной оболочке, познакомиться с документацией на соответствующие команды и утилиты;
 - задать настройки для работы над заданиями с помощью системы контроля версий Git;
 - выполнить сборку и запуск примера приложения, на основе шаблона git-репозитория;
 - исследовать особенности вариантов настройки сети в виртуальной машине;
 - скопировать ранее созданные или сгенерировать новые ssh-ключи и

проверить доступ для отправки результатов работы на сервер `git.toiit.sgu.ru`.

2 Исследовать работу системных вызовов `fork()`, `exec()` и `pipe()` на примере кода приложения `fork` (<http://git.toiit.sgu.ru/projects/os/fork.git>), реализованного под Linux, в рамках стандарта POSIX. В ходе работы необходимо выполнить следующие задания:

- проследить с помощью системы контроля версий **Git** историю изменения исходного кода примера приложения;
- разобрать исходный код, скомпилировать и запустить пример приложения;
- исследовать работу приложения с помощью утилиты **strace**;
- найти и исправить ошибку, допущенную в приложении;
- добавить, в отдельной ветке **exec** системы контроля версия **Git**, запуск нового приложения в дочернем процессе с помощью одного из вызовов семейства `exec()`;
- реализовать, в отдельной ветке **pipe**, взаимодействие дочернего процесса с родительским, с помощью системного вызова `pipe()`.

3 Написать приложение, использующее механизм ожидания событий на открытых файловых дескрипторах (системные вызовы **select()**, **poll()** или **epoll()**) для организации цикла обработки сообщений. Реализация может включать в себя различные варианты использования этого механизма. В простом случае, это может быть асинхронная обработка сообщений через уже рассмотренные средства межпроцессного взаимодействия, а также для реализации простых сетевых сервисов, с использованием сетевых сокетов.

4 Написать приложение с использованием POSIX Threads, реализующее обработку запросов из задания №3 в многопоточном режиме. В простом случае, в качестве стратегии обработки запросов, достаточно реализовать обработку каждого запроса в новом потоке. Кроме того, разрешается выбрать иную стратегию обработки запросов. Например, пул-потоков.

Примеры контрольных вопросов по курсу «Операционные системы»

1. Истина или Ложь?
 - (а) И / Л : Таблицы трансляции физической памяти в виртуальную могут перемещаться в файл подкачки
 - (б) И / Л : Спин-блокировки требуются только мультипроцессорных системах, а в одно процессорных компилируются в пустой код
 - (в) И / Л : В обработчиках нижних половин можно использовать блокируемые семафоры и мьютексы
 - (г) И / Л : Планирование задач происходит только по прерыванию таймера, после окончания назначенному задаче кванта времени
2. Задайте путь к файлу.
 - А. Укажите пример относительного пути к файлу `/usr/lib/game/bin/doom` из

каталога `/usr/bin`: _____

Б. Дан относительный путь `../lib/libutil.so.1` в каталоге `/usr/lib/`, укажите полный путь к файлу `libutil.so.1`: _____

3. Определите вывод команды.

Файл `script` содержит следующие строки:

```
echo first
echo second
```

Вы вводите следующую команду

```
$ sh < script > out
```

Укажите вывод этой команды.

4. Представьте реализацию функции переименования файла `rename()` с помощью системных вызовов `link()` и `unlink()` и кратко объясните свою реализацию.
5. Объясните, что может произойти, если во время выполнения вашей функции `rename("oldfile", "newfile")`, выключить компьютер. Будем считать, что файлы находятся в одном и том же каталоге, но в разных блоках файла каталога во время вызова `rename()`.
6. Представьте условия возникновения тупика на примере использования двух мьютексов для синхронизации двух потоков.
7. Укажите компромиссы между которыми приходится выбирать при реализации планировщиков ввода-вывода. Поясните это на примере планировщика с равноправными очередями (CFQ) и планировщика с лимитом времени.
8. Поясните, на примере системного вызова `fork()`, механизм «копирования при записи» (copy on write).
9. Объясните механизм работы мониторов на примере использования условной переменной и мьютекса.
10. Поясните проблемы синхронизации в обработчиках прерываний. Какие виды объектов синхронизации существуют для обработчиков прерываний и «нижних половин».
11. Поясните разницу между обработкой прерываний и обработкой «нижних половин». Приведите примеры видов «нижних половин».
12. Представьте виды областей памяти в пространстве пользователя на примере вывода следующей команды:

```
$ cat /proc/$$/maps
08045000-080bd000 r-xp 00000000 08:02 117758 /bin/bash
080bd000-080be000 r--p 00077000 08:02 117758 /bin/bash
080be000-080c3000 rw-p 00078000 08:02 117758 /bin/bash
080c3000-0825f000 rw-p 00000000 00:00 0 [heap]
482b4000-482ce000 r-xp 00000000 08:02 378155 /lib/ld-2.10.1.so
482ce000-482cf000 r--p 0001a000 08:02 378155 /lib/ld-2.10.1.so
482cf000-482d0000 rw-p 0001b000 08:02 378155 /lib/ld-2.10.1.so
```

```
482d2000-4841a000 r-xp 00000000 08:02 378156 /lib/libc-2.10.1.so
4841a000-4841b000 ---p 00148000 08:02 378156 /lib/libc-2.10.1.so
4841b000-4841d000 r--p 00148000 08:02 378156 /lib/libc-2.10.1.so
4841d000-4841e000 rw-p 0014a000 08:02 378156 /lib/libc-2.10.1.so
.....
```

Для получения допуска к экзамену по данной дисциплине обучающемуся необходимо выполнить четыре лабораторных и две контрольных работы

Лабораторные работы оформляются в виде компилируемых в рабочее приложение или библиотеку исходных кодов в распределённой системе контроля версий Git, а также образов виртуальных машин с заданным состоянием. Репозитории с исходными кодами публикуются на кафедральном сервере (<http://git.toiit.sgu.ru>), в интернете или предоставляются локально на внешнем накопителе. Уведомление о публикации результатов проводится через систему дистанционного образования на основе Moodle (<http://course.sgu.ru>), через почтовую рассылку курса в интернет, либо во время проведения занятий по выполнению лабораторных работ.

Контрольные работы оформляются в виде теста с использованием контрольных вопросов.

Экзаменационные вопросы по курсу «Операционные системы»

- 1 Понятие операционной системы и её место в программном обеспечении. Основные требования, предъявляемые к ОС. ОС, как расширенная или виртуальная машина. ОС, как менеджер ресурсов.
- 2 Определение процесса. Состояние процесса. Контекст процесса. Операции над процессами. Переключение контекста.
- 3 Синхронизация процессов. Понятие критической секции. Программные решения проблемы критической секции.
- 4 Определение потока исполнения кода. Особенности синхронизации потоков. Контекст исполнения.
- 5 Семафоры. Типы семафоров (бинарный и общий или считающий). Использование семафоров для синхронизации процессов. Реализация семафоров.
- 6 Виды семафоров. Семафор, как основной инструмент синхронизации. Классические задачи синхронизации: поставщик-потребитель, читатель-писатель.
- 7 Условные переменные и мониторы. Реализация мониторов с помощью семафоров. Языковые средства, предоставляющие механизм мониторов.
- 8 Определение тупиков. Условия появления тупиков. Основные подходы к проблеме тупиков.
- 9 Планирование процессов. Основные алгоритмы планирования: «каруселька», приоритетное планирование, «кратчайшая задача первая».

- 10 Планирование процессов с приоритетами. Алгоритмы планирования с динамическими приоритетами.
- 11 Управление прерываниями. Контроллеры прерываний. Отложенные прерывания и нижние половины. Контекст прерываний и особенности синхронизации нижних половин. Спин-блокировки и их разновидности.
- 12 Управление памятью. Типы памяти (по скорости доступа). Стратегии распределения памяти: разделы фиксированного размера, разделы переменного размера, динамическое распределение памяти. Проблема уплотнения памяти.
- 13 Виртуальная память. Страничная организация виртуальной памяти. Трансляция виртуальных адресов в реальные. Сегментная организация памяти. Аппаратные механизмы поддержки виртуальной памяти.
- 14 Стратегии замещения страниц виртуальной памяти. Механизмы виртуализации памяти, как ресурса. Организация подкачки страниц.
- 15 Особенности управления памятью в пространстве ядра. Разделение физической памяти на зоны. Страничная организация памяти в пространстве ядра. Механизмы кеширования памяти.
- 16 Особенности управления памятью в пространстве пользователя. Адресное пространство процесса. Области памяти в адресном пространстве процесса и их типы.
- 17 Управление блочными устройствами ввода-вывода. Компромисс между пропускной способностью и временем доступа в операциях ввода-вывода. Планировщик и алгоритмы планирования ввода-вывода.
- 18 Файловые системы. Понятие файла. Основные операции над файлами. Имена файлов. Организация файлов и доступ к ним.
- 19 Модель виртуальной файловой системы. Дерево объектов, как универсальный механизм организации доступа к глобальным объектам. Специальные виды файлов.
- 20 Форматы исполняемых файлов и загрузка их в адресное пространство процесса. Загрузка динамически загружаемых библиотек
- 21 Сетевые интерфейсы и организация поддержки сети. Идентификация узлов и приложений в сети.
- 22 Защитные механизмы операционных систем. Идентификация и аутентификация. Авторизация. Механизмы ограничения доступа к глобальным объектам. Списки контроля доступа.

7. Учебно-методическое и информационное обеспечение дисциплины «Операционные системы»

а) основная литература

1. *Таненбаум, Эндрю С.* Современные операционные системы. 2-е изд. — СПб. Питер, 2007.
2. *Олифер В.Г., Олифер Н. А.* Сетевые операционные системы — СПб.: Питер, 2008
3. *Синельников Е. А.* Курс. Операционные системы. — 2010.
<http://course.sgu.ru/course/view.php?id=206>

б) дополнительная литература

1. *Терехов А. Н.* Технологии программирования: учебное пособие — М.: Интернет-Ун-т Информ. Технологий, 2007 ; М.: БИНОМ. Лаб. знаний, 2007.
2. *Липаев В. В.* Программная инженерия. Методологические основы. — М.: ТЕИС, 2006.г
3. *Богачёв К. Ю.* Основы параллельного программирования — М.: БИНОМ. Лаб. знаний, 2010.
4. *Таненбаум, Эндрю С.* Архитектура компьютера. 5-е изд. — СПб. Питер, 2010
5. *Таненбаум, Эндрю С.* Операционные системы. Разработка и реализация — СПб. Питер, 2006
6. *Соловьёв В. М.* Практикум по операционным системам. Подготовка системных администраторов информационной компьютерной сети Саратовского государственного университета. Linux-системы — Саратов: Изд-во Саратов. ун-та, 2009.
7. *Гласс Г., Эйблс К.* UNIX для программистов и пользователей. — СПб.: БХВ-Петербург, 2004.
8. *Курячий Г. В., Маслинский К. А.* Операционная система Linux. — 2005.
<http://www.intuit.ru/department/os/linux/>

в) программное обеспечение и Интернет-ресурсы

- 1 ОС GNU/Linux или ОС Windows XP/7;
- 2 консольный файловый менеджер (Far под Windows или mc под Linux);
- 3 утилита удалённого доступа к терминалам Putty (только под Windows);
- 4 утилита удалённого доступа к файлам WinSCP (только под Windows);
- 5 виртуализатор VirtualBox (версии 4.0.4 или старше);
- 6 система контроля версий Git (средства интеграции опционально).

8. Материально-техническое обеспечение дисциплины

Лекционная аудитория, оснащенная мультимедийным оборудованием для организации презентаций (компьютер с проектором и акустической системой).
Лабораторная аудитория, оснащенная персональными компьютерами с необходимым программным обеспечением, подключенными к локальной сети и имеющими доступ в глобальную сеть Интернет.

Программа составлена в соответствии с требованиями ФГОС ВПО с учетом рекомендаций и Примерной ООП ВПО по направлению и профилю подготовки «Вычислительные машины, комплексы, системы и сети».

Автор

Директор саратовского подразделения
ООО «Этерсофт»

_____ Е. А. Синельников

Программа одобрена на заседании кафедры дискретной математики и информационных технологий от «___» _____ 2011 года, протокол № ____.

Заведующий кафедрой
дискретной математики и
информационных технологий,
доцент

_____ Л. Б. Тяпаев

Декан факультета КНиИТ,
доцент

_____ А. Г. Федорова