
Операционные системы

Отчетность по дисциплине

Ревизия: 1.4

История изменений

- 17.02.2010 – Версия 0.1. Первичный документ. Влад Ковтун.
- 20.02.2010 – Версия 0.2. Добавлены вопросы к первому модулю.
- 12.04.2010 – Версия 0.3. Добавлены вопросы ко второму модулю. Добавлены темы для курсовой работы. Влад Ковтун.
- 17.05.2010 – Версия 0.4. Добавлены вопросы к третьему модулю. Влад Ковтун.
- 28.05.2010 – Версия 0.5. Добавлены рекомендации по оформлению курсовой работы, описан пакет материалов для сдачи в рамках курсовой работы. Влад Ковтун.
- 07.11.2010 – Версия 0.6. Добавлены вопросы к пятому модулю. Влад Ковтун.
- 20.11.2010 – Версия 0.7. Добавлены вопросы к шестому модулю. Влад Ковтун.
- 24.11.2010 – Версия 0.8. Добавлены задания и варианты расчетно-графической работы. Влад Ковтун.
- 01.12.2010 – Версия 0.9. Добавлены билеты к шестому модулю. Влад Ковтун.
- 07.02.2011 – Версия 1.0. Обновлен список тем курсовых работ. Влад Ковтун.
- 24.03.2011 – Версия 1.1. Обновлены варианты вопросов к модулю 1. Влад Ковтун.
- 11.03.2011 – Версия 1.2. Обновлены варианты вопросов к модулю 2. Влад Ковтун.
- 22.05.2011 – Версия 1.3. Обновлены варианты вопросов к модулю 3. Влад Ковтун.
- 27.01.2012 – Версия 1.4. Обновлены темы курсовых работ и задания для расчетно-графической работы. Влад Ковтун.

Содержание

История изменений	2
Содержание	3
Список вопросов	6
Модуль 1. Общая архитектура операционных систем и исполнение потока команд	6
Лекция 1. История. Назначение, функции и классификация ОС	6
Лекция 2. Основные понятия, системные вызовы, структура ОС	6
Лекция 3.1. Процессы и потоки. Межпроцессное и межпоточное взаимодействие. Часть 1. Процессы и потоки в современных ОС	7
Лекция 3.2. Процессы и потоки. Межпроцессное и межпоточное взаимодействие. Часть 2. Механизмы синхронизации процессов и потоков	7
Лекция 3.3. Процессы и потоки. Межпроцессное и межпоточное взаимодействие. Часть 3. Механизмы синхронизации в распределенных системах	8
Модуль 2. Управление ресурсами, памятью и ввод-вывод	9
Лекция 4. Распределение ресурсов и проблемы взаимоблокировок	9
Лекция 5.1. Управление оперативной памятью. Виртуальная память. Часть 1	10
Лекция 5.2. Управление оперативной памятью. Виртуальная память. Часть 2	10
Лекция 5.3. Управление оперативной памятью. Виртуальная память. Часть 3	11
Лекция 6.1. Ввод-вывод. Часть 1	12
Лекция 6.2. Ввод-вывод. Часть 2	13
Модуль 3. Файлы, параллельные, распределенные вычисления и безопасность	14
Лекция 7.1. Файловая система. Файлы и каталоги. Часть 1	14
Лекция 8.1. Операционные системы для многопроцессорных и многомашинных вычислительных систем. Часть 1	16
Лекция 8.2. Операционные системы для многопроцессорных и многомашинных вычислительных систем. Кластеры. MPI. Часть 2	17
Лекция 9. Организация сетевого взаимодействия	17
Расчетно-графическая работа	17
Модуль 4. Операционные системы семейства Microsoft Windows	17
Лекция 10. История. Архитектура. Структура. Программный интерфейс приложений	17
Лекция 11. Управление потоками и процессами. Управление памятью	18
Лекция 12. Управление вводом выводом. Файловая система. Безопасность	20
Модуль 5. Операционные системы семейства UNIX	22
Лекция 13. История. Архитектура. Структура. Особенности программирования системных вызовов. Управление потоками и процессами	22
Лекция 14. Управление памятью. Управление вводом выводом. Файловая система. Безопасность	23
Модуль 6. Операционные системы семейства QNX	25
Лекция 15. История. Архитектура. Структура. Особенности программирования системных вызовов. Управление потоками и процессами	25
Лекция 16. Управление памятью. Управление вводом выводом. Файловая система. Безопасность	25
Модуль 7. Курсовая работа	25
Лекция 17 - Курсовая работа	25
Темы курсовых работ	26
Задание	26

Перечень тем курсовых работ	26
Требования к содержанию пояснительной записки	31
Перечень материалов к сдаче	32
Темы заданий для расчетно-графической работы	33
Задание	33
Пояснения к проведению эксперимента	33
Обработка результатов эксперимента	35
Перечень материалов к сдаче	36
Билеты к модулям	37
Модуль 1	37
Вариант 1	37
Вариант 2	37
Вариант 3	37
Вариант 4	38
Вариант 5	38
Вариант 6	38
Модуль 2	39
Вариант 1	39
Вариант 2	39
Вариант 3	40
Вариант 4	40
Вариант 5	40
Вариант 6	41
Модуль 3	41
Вариант 1	41
Вариант 2	41
Вариант 3	42
Вариант 4	42
Вариант 5	42
Вариант 6	43
Модуль 4	43
Вариант 1	43
Вариант 2	43
Вариант 3	43
Вариант 4	43
Вариант 5	44
Вариант 6	44
Модуль 5	44
Вариант 1	44
Вариант 2	44
Вариант 3	45
Вариант 4	45
Вариант 5	45
Вариант 6	46

Модуль 6	46
Вариант 1	46
Вариант 2	46
Вариант 3	46
Вариант 4	47
Вариант 5	47
Вариант 6	47
Модуль 7	48
Вариант 1	48
Вариант 2	48
Вариант 3	48
Вариант 4	48
Вариант 5	48
Вариант 6	48

Список вопросов

Модуль 1. Общая архитектура операционных систем и исполнение потока команд

Лекция 1. История. Назначение, функции и классификация ОС

Основные понятия

Понятие ОС. Место ОС в структуре вычислительной системы. Назначение ОС. Особенности функционирования ОС. Функции ОС: расширение возможностей компьютера, управление ресурсами. Распределение (мультиплексирование) ресурсов во времени и пространстве.

История

Первое поколение компьютеров.

Второе поколение компьютеров. Пакетная система обработки.

Третье поколение компьютеров. Многозадачность. Использование дисковых массивов. Режим разделения времени. Причины успеха системы CTSS. Причины краха системы MULTICS. Тенденции в развитии вычислительных систем публичного пользования. Появление UNIX и его ветви: System V, BSD. Стандарт POSIX.

Четвертое поколение компьютеров. Персональные компьютеры. Выход Intel 8080. Причины успеха и краха системы CP/M. Причины успеха DOS. Появление графического интерфейса. Развитие ОС Windows. Сетевые и распределенные ОС. Кластеры.

Классификация ОС

Классификация ОС, согласно решаемым задач.

Лекция 2. Основные понятия, системные вызовы, структура ОС

Системные вызовы

Системные вызовы. Стандартизация системных вызовов: неопределенные обобщения, специализация. Разница между системным вызовом и обычной подпрограммой. Пояснить механизм выполнения СВ (11 шагов).

СВ управления процессами

СВ управления процессами в ОС Windows и в ОС FreeBSD/Linux/QNX.

Базовые понятия ОС

Процесс. Ресурсы, ассоциированные с процессом. Таблица процессов. Состояния процесса.

Взаимоблокировки. Причины. Подходы к предупреждению.

Память. Управление и защита. Размещение программ в оперативной памяти.

Ввод-вывод данных.

Файловая система: файлы и каталоги. Иерархия каталогов. Управление иерархией. Рабочий каталог процесса. Дескриптор файла. Монтирование файловой системы: описание, особенности. Специальные файлы: символьные и блочные. Каналы.

Безопасность. Разграничение доступа в файловой системе.

Оболочка (режим командной строки).

Структура ОС

Монолитные ОС. Идея. Особенности. Достоинства и недостатки. Модель.

Многоуровневые ОС. Разделение на уровни. Достоинства и недостатки.

Виртуальные машины. Предпосылки. Достоинства и недостатки TSS/360. Особенности CP/CMS. Монитор виртуальной машины. Современная интерпретация виртуальной машины: в процессорах семейства Intel, ОС Windows. Подходы к реализации виртуальной машины: особенности, достоинства и недостатки. Java и Java Virtual Machine.

Экзоядро. Особенности. Достоинства и недостатки.

Модель клиент-сервер. Тенденции. Серверные и клиентские процессы. Достоинства и недостатки. Способы обмена сообщениями в распределенных системах (особенности выполнения низкоуровневых команд).

Лекция 3.1. Процессы и потоки. Межпроцессное и межпоточное взаимодействие. Часть 1. Процессы и потоки в современных ОС

Процессы

Предпосылки многозадачности. Псевдопараллелизм.

Модель процесса. Модель последовательных процессов. Процесс. Создание процесса. Завершение процесса и причины. Иерархия процессов. Состояние процессов и переходы между ними. Планировщик. Преимущества и недостатки.

Реализация процессов. Таблица процессов. Элемент таблицы. Иллюзия многозадачности. Схема обработки прерывания.

Потоки

Понятие потока. Модель потока. Отличие потока и процесса. Модель потоков и модель процессов. Многопоточность. Примеры. Понятие безопасности и защиты в модели потоков. Стек данных в потоке. Состояния. «Вежливость» потоков. Преимущества и недостатки в использовании потоков. Пример использования потоков в текстовом редакторе.

Реализация потоков: режим ядра, режим пользователя их смесь.

Реализация потоков в режиме пользователя. Преимущества и недостатки реализации потоков. Блокировка потоков и подходы к решению проблемы блокировки. Отсутствие необходимой страницы в памяти. Переключение потоков. Особенности построения эффективных многопоточных приложений.

Реализация потоков в режиме ядра. Преимущества и недостатки реализации потоков. Реактивация потоков.

Смешанная реализация. Особенность использования потоками уровня пользователя потоков уровня ядра. Достоинства и недостатки.

Активация планировщика – один из подходов смешанной реализации. Преимущества и недостатки. Основа механизма.

Всплывающие потоки. Особенности применения. Достоинства и недостатки.

Подходы к распараллеливанию. Подходы к доступу переменных глобальных для потока. Подходы к решению проблемы реентерабельности. Сигналы. Управление стеками.

Лекция 3.2. Процессы и потоки. Межпроцессное и межпоточное взаимодействие. Часть 2. Механизмы синхронизации процессов и потоков

Межпроцессное и межпоточное взаимодействие

Проблемы межпроцессного взаимодействия. Проблемы межпоточного взаимодействия.

Условия реализации совместной работы параллельных процессов и эффективного использования данных.

Состояние состязания. Пример спулера печати.

Критические области. Основная задача. Метод решения основной задачи. Пример исходного кода. Функции ОС Windows реализующие механизмы критических секций.

Взаимное исключение с активным ожиданием

Запрещение прерываний. Достоинства и недостатки.

Переменные блокировки. Достоинства и недостатки. Подходы используемые в ОС Windows. Пример исходного кода.

Строгое чередование. Достоинства и недостатки. Пример исходного кода. Использование в ОС Windows.

Алгоритм Петерсона. Пример исходного кода.

Команда TSL. Особенности использования. Команды процессора 0x86, реализующая данную команду.

Примитивы межпроцессного взаимодействия.

Основной недостаток активного ожидания. Проблема инверсии приоритета. Простейшие примитивы. Пример использования.

Проблема производителя и потребителя. Пример исходного кода. Подход к решению проблемы в ОС Windows.

Семафоры. Идея. Решение проблемы производителя и потребителя с помощью семафоров. Пример исходного кода. Функции ОС Windows реализующие функциональность семафоров.

Мьютексы. Идея. Пример исходного кода. Преимущества мьютексов над критическими секциями. Управление мьютексами в ОС Windows.

Различие в подходах к синхронизации потоков и процессов. Решение проблемы совместного доступа к общему адресному пространству, для проверки семафора (мьютекса).

Мониторы. Предпосылки. Основная идея. Подходы к переключению процессов. Пример исходного кода. Преимущества и недостатки.

Лекция 3.3. Процессы и потоки. Межпроцессное и межпоточное взаимодействие. Часть 3. Механизмы синхронизации в распределенных системах

Передача сообщений

Передача сообщений. Механизм передачи сообщений. Достоинства и недостатки. Решение проблемы производителя и потребителя. Пример исходного кода.

Барьеры. Условия применения. Идея.

Классические проблемы межпроцессного взаимодействия

Проблема обедающих философов. Описание. Решение задачи. Пример исходного кода.

Проблема читателей и писателей. Описание. Подходы к решению задачи. Пример исходного кода.

Проблема спящего бравобрея. Описание. Решение задачи. Пример исходного кода.

Планирование

Планировщик и алгоритм планирования. Суть проблемы планирования заданий.

Ограничения возможностей. Актуальность планирования. Узкие места повышения производительности за счет совершения планирования.

Ситуации планирования. Алгоритмы планирования с переключениями и без них.

Категории алгоритмов планирования. Задачи алгоритмов планирования. Показатели эффективности функционирования планировщика для систем: пакетной обработки, интерактивных систем, системы реального времени.

Планирование в системах пакетной обработки

Первым пришел – первым ушел. Кратчайшая задача – первая. Наименьшее оставшееся время выполнения. Трехуровневое планирование. Критерии планировщика при трехуровневом планировании.

Планирование в интерактивных системах

Циклическое планирование. Приоритетное планирование. Несколько очередей. Самый короткий процесс – следующий. Гарантированное планирование. Лотерейное планирование. Справедливое планирование.

Планирование в системах реального времени

Гибкие и жесткие системы реального времени. Периодические и непредсказуемые события. Условие планируемости. Политика и механизм планирования. Планирование потоков.

Модуль 2. Управление ресурсами, памятью и ввод-вывод

Лекция 4. Распределение ресурсов и проблемы взаимоблокировок

Взаимоблокировки

Ресурсы. Выгружаемые и невыгружаемые ресурсы. Тупиковая ситуация (взаимоблокировка). Условия возникновения взаимоблокировки.

Моделирование взаимоблокировок посредством направленных графов. Стратегии борьбы с взаимоблокировками.

Страусовый алгоритм.

Обнаружение и устранение взаимоблокировок

Обнаружение взаимоблокировки при наличии одного ресурса каждого типа. Пример ситуации с 7 процессами и 6 ресурсами: граф ресурсов. Алгоритм обнаружения взаимоблокировки.

Обнаружение блокировок при наличии нескольких ресурсов каждого типа. Алгоритм обнаружения взаимоблокировки.

Восстановление при помощи принудительной выгрузки ресурса: Метод восстановления через откат. Метод восстановления путем уничтожения процессов.

Избежание взаимоблокировок

Суть подхода аккуратного предоставления ресурсов. Концепция безопасных состояний. Пример траектории ресурсов, для 2-х ресурсов. Безопасные и небезопасные состояния. Пример безопасного и небезопасного состояния. В чем разница между безопасным и небезопасным состоянием.

Алгоритм банкира для одного вида ресурсов. Алгоритм банкира для нескольких видов ресурсов.

Предотвращение взаимоблокировок

Атака условия взаимного исключения.

Атака условия удержания и ожидания. Проблемы применения такой атаки. Подходы к решению проблем.

Атака условия отсутствия принудительной выгрузки ресурса.

Атака условия циклического ожидания. Способы устранения.

Сводная таблица условий взаимоблокировок и методы их устранения.

Лекция 5.1. Управление оперативной памятью. Виртуальная память. Часть 1

Введение

Иерархия памяти в компьютерах. Менеджер памяти. Классы систем управления памятью.

Однозадачная система без подкачки с диска: модели организации памяти.

Многозадачность с фиксированными разделами: модель с отдельными входными очередями к разделам, с единой очередью к разделам. Недостатки и преимущества. Подходы к решению возникших проблем.

Моделирование многозадачности

Степень загрузки CPU. Степень многозадачности. Графическая интерпретация. Рост производительности компьютера в зависимости от увеличения ОЗУ.

Настройка адресов и защита

Проблема перемещения программ в памяти (настройки адресов). Метод модификации команд: описание, достоинства и недостатки. Метод настройки адресов и защиты OS/360. Альтернативный подход двух проблем: описание, достоинства и недостатки.

Подкачка

Предпосылки в возникновении подкачки. Подходы к управлению памятью.

Swapping. Разделение памяти на фиксированные разделы и разделы переменного размера: отличие, достоинства и недостатки. Фрагментация памяти. Проблема роста области данных процессов.

Управление памятью с помощью битовых массивов

Способы учета используемой памяти. Единичные блоки. Размер единичного блока. Способ битовых массивов. Достоинства и недостатки.

Управление памятью с помощью связанных списков

Способ учета используемой памяти. Методы сортировки списков. Достоинства и недостатки.

Алгоритм «первого подходящего участка». Достоинства и недостатки.

Алгоритм «следующего подходящего участка». Достоинства и недостатки.

Алгоритм «самый подходящий участок». Достоинства и недостатки.

Алгоритм «самый неподходящий участок». Достоинства и недостатки.

Подход к ускорению алгоритмов выделения памяти – отдельных списков. Как изменится картина скорости алгоритмов.

Алгоритм «быстрый подходящий». Достоинства и недостатки.

Виртуальная память

Метод оверлейного управления памятью. Виртуальная память: основная идея.

Страничная организация памяти. Диспетчер памяти и его место в компьютере. Виртуальный адрес и физический адрес. Страница и страничный блок. Связь между виртуальными и физическими адресами посредством таблицы страниц. Страничное прерывание.

Лекция 5.2. Управление оперативной памятью. Виртуальная память. Часть 2

Виртуальная память

Таблица страниц: описание, требования, достоинства и недостатки. Многоуровневые таблицы страниц: описание, достоинства и недостатки. Структура элемента таблицы.

Буфер быстрого преобразования адреса: предпосылки, описание, структура записи, принцип действия.

Инвертированные таблицы страниц

Предпосылки. Описание. Преимущества и недостатки. Способы усовершенствования: буфер быстрого преобразования, хеш-таблицы.

Алгоритмы замещения страниц

Проблема страничного обмена: предпосылки, описание.

Оптимальный алгоритм: описание, достоинства и недостатки, подходы к реализации.

Алгоритм NRU: описание, достоинства и недостатки, подходы к реализации.

Алгоритм FIFO: описание, достоинства и недостатки, подходы к реализации.

Алгоритм «вторая попытка»: описание, достоинства и недостатки, подходы к реализации.

Алгоритм «clock»: описание, достоинства и недостатки, подходы к реализации.

Алгоритм LRU: описание, достоинства и недостатки, подходы к реализации.

Алгоритм «рабочий набор»: описание, достоинства и недостатки, подходы к реализации. Пробуксовка. Рабочий набор. Модели рабочего набора. Опережающая подкачка. Методы формирования рабочего набора.

Алгоритм WSClock: описание, достоинства и недостатки, подходы к реализации.

Сводная таблица алгоритмов размещения страниц и комментариев по их применениям.

Лекция 5.3. Управление оперативной памятью. Виртуальная память. Часть 3

Моделирование алгоритмов замещения страниц

Аномалия Билэди.

Магазинные алгоритмы. Последовательность обращений. Ограничения (условия). Характеристики систем со страничной организацией памяти. Свойства магазинной модели.

Строка расстояний: описание, от чего зависит, влияние на производительность алгоритма замещения страниц.

Прогнозирование частоты страничных прерываний: описание, зависимости.

Реализация систем управления страничной организацией памяти

Политика распределения памяти. Предпосылки локальной и глобальной политики. Локальная политика: описание, графическая интерпретация, особенности применения. Глобальная политика: описание, графическая интерпретация, особенности применения. Использование алгоритмов замещения страниц в рамках различных политик: подходы, достоинства и недостатки.

Регулирование нагрузки. Симптомы пробуксовки. Подходы к решению проблемы пробуксовки: выгрузка процессов на диск. Степень многозадачности.

Размер страницы. Большой и небольшой размер страницы: достоинства и недостатки. Внутренняя фрагментация. Математическое обоснование оптимального размера страницы.

Разделение памяти для данных и для команд: предпосылки, описание, достоинства и недостатки.

Совместно используемые страницы: предпосылки, описание механизма, достоинства и недостатки, проблемы реализации.

Политика очистки страниц: предпосылки, особый сервис в ОС, гарантии. Алгоритмы замещения страниц, применяемые для очистки страниц.

Интерфейс виртуальной памяти. Функции ОС Windows по управлению виртуальной памятью.

Участие ОС в подкачке страниц. Основные ситуации участия ОС: создание нового процесса, исполнение процесса, страничное прерывание, завершение процесса.

Алгоритм страничного прерывания. Перезапуск прерванной команды CPU: суть проблемы, подходы к ее решению.

Блокирование страниц в памяти: предпосылки, подходы к решению проблемы.

Хранение страничной памяти на диске: дисковая подкачка, область подкачки, инициализация области подкачки для процесса, стратегии резервирования областей в файле подкачки для процессов, недостатки и преимущества. Методы размещения страниц в файле подкачки.

Разделение политики от механизма

Пример разделения политики и механизма. Графическая интерпретация. Механизм управления памятью. Описание процесса выделения памяти в рамках разделения на 3 подсистемы: драйвер, обработчик прерываний и внешний обработчик страниц. Достоинства и недостатки.

Сегментация

Предпосылки разделения единого адресного пространства на несколько различных сегментов. Достоинства и недостатки сегментации. Сравнение сегментной виртуальной и единой памяти. Внешняя фрагментация.

Сегментация в CPU i486

Таблицы дескрипторов LDT, GDT: назначение. Механизм доступа к виртуальному и физическому адресу посредством пары (селектор, смещение). В каких случаях линейный адрес интерпретируется как физический, а когда нет. Отображение линейного адреса в физический, при страничной организации памяти. Использование TLB.

Уровни защиты CPU. Механизм защиты памяти. Шлюз вызова.

Лекция 6.1. Ввод-вывод. Часть 1

Базовые понятия

Классификация устройств: блочные и символьные, их отличия и особенности. Контроллер устройства и его назначение. Интерфейс взаимодействия устройства и контроллера. Исправление ошибок.

Ввод-вывод, отображаемый в память

Взаимодействие контроллера устройства и CPU. Механизм использования портов ввода-вывода и областей памяти. Достоинства и недостатки.

Прямой доступ к памяти: описание. Режимы передачи информации при прямом доступе к памяти. Достоинства и недостатки. Пример на основе дискового ввода-вывода.

Прерывания

Использование прерываний для организации ввода-вывода. Вектор прерываний. Механизм обработки прерываний. Достоинства и недостатки. Особенности реализации механизма прерываний на современных компьютерах с конвейерной архитектурой. Точные (жесткие) и неточные (мягкие) прерывания. Свойства точных прерываний. Недостатки неточных прерываний для CPU конвейерной архитектурой.

Принципы ПО ввода-вывода

Ключевая концепция ПО ввода-вывода. Независимость от устройств. Обработка и восстановление после ошибок. Способ переноса данных: синхронный и асинхронный. Буферизация. Выделенные устройства и устройства коллективного использования.

Реализация ввода-вывода

Способы осуществления операций ввода-вывода.

Программный ввод-вывод. Описание механизма. Пример. Достоинства и недостатки.

Ввод-вывод посредством прерываний. Описание механизма. Пример. Достоинства и недостатки.

Ввод-вывод посредством DMA. Описание механизма. Пример. Достоинства и недостатки.

Программные уровни ввода-вывода

Уровни организации ввода-вывода. Структура системы ввода-вывода: уровни и функции.

Обработчики прерываний: описание механизма.

Драйвер устройства: описание механизма, место в ОС, классификация, динамическая подгрузка драйвера, функции драйвера, общая структура драйвера (последовательность действий по управлению устройством). Реентерабельность драйвера. Особенности функционирования драйвера для PnP устройств.

Независимое от устройств ПО ввода-вывода: предпосылки, место в ОС, функции.

Интерфейс драйверов устройств и ОС: аспекты универсального интерфейса.

Буферизация: предпосылки, двойная буферизация, достоинства и недостатки, пример частого копирования данных – снижения производительности.

Сообщения об ошибках: классы ошибок.

Захват и освобождение выделенных устройств: подходы к эксклюзивному предоставлению устройств.

Соккрытие конструктивных особенностей устройств.

ПО ввода-вывода уровня пользователя: библиотечные процедуры, спулинг.

Лекция 6.2. Ввод-вывод. Часть 2

Контроллер магнитных дисков

Функции. Популярные интерфейсы. Перекрывающийся поиск. Соккрытие геометрии: физическая и виртуальная геометрия, предпосылка, особенности. Представление информации на диске: цилиндр, голоска, сектор. Режим HBA, LBA.

RAID

Идея. Уровни. RAID 0: описание, достоинства и недостатки. RAID 1: описание, достоинства и недостатки. RAID 2: описание, достоинства и недостатки. RAID 3: описание, достоинства и недостатки. RAID 4: описание, достоинства и недостатки. RAID 5: описание, достоинства и недостатки.

Форматирование дисков

Предпосылка. Описание полей сектора. Форматирование диска. Перекос цилиндров: описание предпосылка, величина перекося, графическая иллюстрация. Перекос головок. Путаница в емкости HDD: форматированная и неформатированная емкость. Нумерация секторов: предпосылки, без чередования, однократное чередование, двукратное чередование. Таблица разделов: назначение, содержимое. Разбивка на разделы. MBR. Активный раздел. Высокоуровневое форматирование: основные операции. Процесс загрузки компьютера.

Алгоритмы планирования головок

Факторы, влияющие на скорость записи и чтения данных с HDD. Алгоритмы обработки запросов к драйверу. Использование связанных списков для формирования последовательности отработки запросов. Алгоритм планирования «ближайший цилиндр первым»: описание, пример, достоинства и недостатки. Алгоритм планирования «ближайший запрос первым»: описание, пример, достоинства и недостатки. Элеваторный алгоритм планирования: описание, пример, достоинства и недостатки. Кеширование данных при считывании и записи.

Обработка ошибок

Производственные дефекты. Запасные и дефектные блоки. Способы организации замены дефектных блоков на запасные: пометка и сдвиг, иллюстрация. Обработка ошибок контроллером и ОС. Создание резервных копий: по-файлово и по-секторно, достоинства и недостатки.

Ошибки поиска цилиндра: предпосылки, подходы к исправлению.

Стабильное запоминающее устройство

Стабильное запоминающее устройство: предпосылки, описание. Модель стабильного запоминающего устройства. Операции реализующие стабильное запоминающее устройство, их описание. Влияние сбоя CPU стабильного запоминающего устройства – пять вариантов. Подходы по оптимизации и усовершенствованию.

Оптические диски

Процесс создания CD. Представление информации на CD. Достижение совместимости различных стандартов CD.

CD-ROM. Исправление ошибок. Логическое расположение данных на CD-ROM. Файловая система CD-ROM. Процесс создания CD-R. Условие успешной записи диска. Подходы к защите авторских прав.

Процесс создания CD-RW.

Таймеры

Аппаратная часть. Типы таймеров. Режимы функционирования одновибратора. Генератор прямоугольных импульсов.

Основные функции драйвера часов. Подходы к реализации функции времени суток. Ограничение времени работы CPU. Учет использования CPU. Моделирование нескольких виртуальных таймеров. сторожевые таймеры. Сбор статистики.

«Мягкие таймеры». Подходы к организации ввода-вывода посредством «мягких» таймеров: предпосылки, идея, достоинства и недостатки.

Модуль 3. Файлы, параллельные, распределенные вычисления и безопасность

Лекция 7.1. Файловая система. Файлы и каталоги. Часть 1

Файловые системы

Предпосылки появления долгосрочных хранилищ данных – 3 проблемы. Требования к долговременным хранилищам. Файлы. Устойчивость (персистентность). Файловая система.

Файлы

Именованье файлов. Особенности именования в MS-DOS, UNIX, Windows 2000. Расширение.

Структура файла – 3 типа: Файлы – последовательность байт. Файлы – образы перфокарт и строк. Файлы – деревья.

Типы файлов. Регулярные файлы и каталоги. Символьные и двоичные файлы. Структура исполняемого файла UNIX.

Последовательный и произвольный доступ к файлам. Категории функций по работе с файлами в ОС Windows. Дополнительная информация о файлах в ОС Windows.

Атрибуты файла. Атрибуты файлов и их установка в ОС Windows.

Операции с файлами в UNIX. Создание. Удаление. Открытие. Закрытие. Чтение. Запись. Добавление. Поиск. Получение и установка атрибутов.

Операции с файлами в Windows.

Файлы, отображаемые в память.

Файлы, отображаемые в адресное пространство. Общая идея. Механизм функционирования файлов, отображаемых в память. Трудности применения.

Функции ОС Windows по работе с файлами, отображаемыми в адресное пространство.

Каталоги

Однокаталоговые файловые системы: описание, достоинства и недостатки. Двухуровневые файловые системы: описание, достоинства и недостатки. Иерархия каталогов: описание, достоинства и недостатки.

Имя пути. Абсолютное и относительное.

Функции ОС UNIX по работе с каталогами. Создание. Удаление. Открытие. Закрытие. Чтение следующего элемента открытого каталога. Переименовать. Связать. Удалить ссылку.

Операции с каталогами в Windows.

Реализация файловых систем

Структура файловой системы. Главная загрузочная запись. Таблица разделов. Структура раздела.

Функции ОС Windows по работе с дисками.

Функции ОС Windows по работе с разделами.

Размещение файлов на диске. Непрерывные файлы: описание, преимущества и недостатки, особенности применения. Связанные списки: описание, преимущества и недостатки, особенности применения. Связанные списки с таблицей в памяти (FAT): описание, преимущества и недостатки, особенности применения. I-узлы: описание, преимущества и недостатки, особенности применения.

Реализация каталогов: структура каталоговой записи, размещение атрибутов файла. Именование файлов: подходы к реализации: фиксированный размер записи, альтернативный подход, записи переменной длины. Поиск файла посредством хеш-таблиц: описание, достоинства и недостатки. Кеширование результатов поиска.

Организация дискового пространства

Стратегии выделения дискового пространства под файлы. Размер блока хранения файла: описание проблемы, подходы к решению. Зависимость скорости чтения/записи даны с диска и эффективность использования дискового пространства от размера блока.

Учет свободных блоков. Метод связанных списков блоков диска: описание, достоинства и недостатки. Метод битовых массивов: описание, достоинства и недостатки, подходы к усовершенствованию.

Дисковые квоты: описание, механизм реализации. Таблица квот пользователей. Гибкие и жесткие лимиты.

Надежность файловой системы

Резервное копирование. Причины и способы их предотвращения: аварии и ошибки пользователей, удаление. Трудности при создании резервных копий: резервирование всей файловой системы или ее части, резервирование изменений с момента последнего резервирования, архивирование (сжатие резервной копии), создание резерва при функционирующей ОС, хранение резервных копий.

Стратегии резервного копирования: физическая и логическая архивация. Физическая архивация: описание, достоинства и недостатки. Логическая архивация: описание, достоинства и недостатки, особенности применения. Алгоритм логической архивации – четыре этапа Специфика применения логической архивации.

Функции ОС Windows по работе с архивами.

Производительность файловой системы

Кеширование. Алгоритмы страничной организации памяти, как прототипы управления блочным дисковым кешем. Нежелательность алгоритма LRU. Модифицированный алгоритм LRU. Подходы к кешированию в UNIX, MS-DOS.

Опережающее чтение блока. Подходы к поиску компромисса между последовательным доступом и произвольным.

Снижение времени перемещения блока головок. Подход к реализации. Кластеризация блоков файлов. Подходы к размещению i -узлов и блоков данных: размещение i -узлов в середину диска, разбиение диска на группы цилиндров.

Файловые системы с журнальной структурой

Предпосылки создания ФС с журнальной системой. Проблема операции записи мелких блоков данных. Подходы к решению проблемы операции записи мелких блоков в LFS. Проблема обнаружения i -узла в LFS и подходы к ее решению. Проблема повторного использования блоков в LFS и подходы к ее решению.

Лекция 8.1. Операционные системы для многопроцессорных и многомашинных вычислительных систем. Часть 1

Стандартные подходы к построению высокопроизводительных систем. Многопроцессорные системы с общей памятью. Многопроцессорные системы с передачей сообщений. Многомашинные системы. Распределенные системы (слабосвязанные и сильносвязанные).

Мультипроцессор с общей памятью: описание, особенность, основа межпроцессорного обмена. Особенности мультипроцессорных ОС.

UMA-мультипроцессоры с общей шиной. Основная идея. UMA-мультипроцессоры: без кеша, с кешем, с кешем и локальной памятью. Ограничение использования такой архитектуры.

UMA-мультипроцессоры с координатными коммутаторами. Основная идея. Координатный коммутатор: описание, достоинства и недостатки. Неблокирующая сеть.

UMA-мультипроцессоры с многоступенчатыми коммутаторными сетями. Основная идея. Сложные коммутаторы (несколько входов и выходов). Блокирующая сеть. Структура сообщения межпроцессорного обмена. Пример сети омега для 8 CPU, 8 RAM модулей и 12 коммутаторов. Параметры сети омега в общем случае для n CPU, n RAM модулей. Идеальное тасование. Алгоритм маршрутизации сообщений в сети. Чередующаяся система памяти.

NUMA-мультипроцессоры. Основная идея. Ключевые характеристики. Основное преимущество перед UMA-системами. Системы с когерентным (CC-NUMA) и некогерентным (NU-NUMA) кешем. Каталогный мультипроцессор в CC-NUMA: описание, достоинства и недостатки.

Типы мультипроцессорных ОС

Персональная копия ОС на каждом CPU. Описание. Графическая интерпретация. Основные замечания по данной схеме. Достоинства и недостатки.

Персональная копия ОС лишь CPU-хозяину, остальные – CPU-подчиненные. Описание. Графическая интерпретация. Основные замечания по данной схеме. Достоинства и недостатки.

ОС для симметричных мультипроцессоров. Описание. Графическая интерпретация. Достоинства и недостатки.

Синхронизация в мультипроцессорах. Мьютекс-протокол. Взаимное исключение. Кеширование и проблема конкуренции за шину. Решение проблемы кеширования: итерация пробного чтения, двоичный экспоненциальный откат, опрос собственной переменной блокировки.

Проблема ожидания в цикле либо переключение: описание, поиск компромисса: простое ожидание, спин-блокировка, учет предыдущих интервалов ожидания.

Планирование мультипроцессора

Вопросы планирования: какой процесс запустить следующим, зависимость процессов между собой.

Разделение времени. Единая структура данных для планирования: описание, достоинства и недостатки. Умное планирование. Приоритетное планирование. Родственное планирование. Двухуровневое планирование.

Совместное использование пространства. Простейший алгоритм. Кратчайшее задание – первое. Разбиение процессов на группы. Учет работающих и желающих работать процессов.

Бригадное планирование – планирование во времени и пространстве: описание, достоинства и недостатки.

Лекция 8.2. Операционные системы для многопроцессорных и многомашинных вычислительных систем. Кластеры. MPI. Часть 2

Лекция 9. Организация сетевого взаимодействия

Расчетно-графическая работа

Задание для расчетно-графической работы приведены в разделе «[Темы расчетно-графических работ](#)».

Модуль 4. Операционные системы семейства Microsoft Windows

Лекция 10. История. Архитектура. Структура. Программный интерфейс приложений

Основные понятия

Процессы, потоки и задания. Понятие процесса. Что включает? Графическая интерпретация процесса и его ресурсов. Набор приложений для мониторинга и управления ресурсами. Понятие потока. Что включает? Понятие контекста потока. Маркер доступа процесса. Дескриптор виртуальных адресов. Понятие задания.

Виртуальная память. Организация памяти. Механизм трансляции виртуальных адресов в физические. Подкачка страниц. Теоретический максимум общего виртуального пространства для x86 и x64 платформ. Какие возможности преодоления 4 Гб барьера. Структура адресного пространства для x86 систем. Структура адресного пространства для x64 систем.

Понятие пользовательского и пользовательского режима. Механизм Driver Verifier. Переключение между пользовательским режимом и режимом ядра.

Понятие службы терминалов. Нулевой сеанс. Удаленные соединения. Какие сборки Windows и в каком количестве позволяют удаленные подключения.

Понятие объекта. Понятие описателя. Атрибуты объекта. Отличие объекта от структуры. Какие важные функции реализуются объектами?

Базовые возможности Windows согласно Common Criteria for Information Technology Security Evaluation. Виды контроля доступа к объектам. Управление избирательным доступом. Управление привилегированным доступом.

Понятие реестра. Управление ОС через реестр.

Понятие Unicode и поддержка разными версиями ОС. Особенности.

Архитектура Windows

Модель ОС. Поддержка режимов функционирования современных процессоров, платформ. Модель ОС. Общая архитектура ОС и ее графическая интерпретация. Типы пользовательских процессов. Компоненты режима ядра.

Понятие переносимости. Какие платформы поддерживаются. За счет чего обеспечивается переносимость.

Реализация многозадачности. Симметричная многопроцессорная обработка. Ассиметричная многопроцессорная обработка. Какие многопроцессорные системы поддерживаются Windows. Особенности поддержки логических процессоров. Особенности поддержки NUMA-систем.

Понятие масштабируемости. Особенности многопроцессорной Windows.

Различия между клиентскими и серверными версиями.

Отладочная версия ОС Windows. Назначение. Что позволяет?

Ключевые подсистемы ОС Windows. Архитектура ОС Windows. Подсистема окружения. Подсистема Windows. Подсистема POSIX. Ntdll.dll. Исполнительная система. Основные функции исполнительной системы. Основные компоненты исполнительной системы. Основные группы функций поддержки.

Общее понятие ядра. Объекты ядра. Управляющие объекты и объекты диспетчера. Поддержка оборудования – абстрагирование. Примеры абстрагирования.

Понятие уровня абстрагирования от аппаратуры. Чего позволяет достигнуть. Как реализуется.

Понятие драйвера устройства. Контексты исполнения драйверов. Типы драйверов устройств и их описание. Усовершенствования в модели драйвера Windows.

Понятие системного процесса. Основные системные процессы.

Лекция 11. Управление потоками и процессами. Управление памятью

Процессы

Понятие процесса в Windows. Понятие блока процесса и потока. Графическое представление структур данных процесса, потока и переменных окружения. Структура EPROCESS. Структура KPROCESS (PCB).

Основные переменные уровня ядра, связанные с процессами.

Основные счетчики производительности связанные с процессами.

Основные функции Win32 API по работе с процессами.

Варианты Win32 API функции CreateProcess. Основные этапы создания процесса в Windows. Графическая интерпретация этапов создания процесса в Windows. Этап 1: открытие образа, подлежащего выполнению. Этап 2: создание объекта процесс. Этап 3: создание первичного потока, стека и контекста. Этап 4: уведомление подсистемы Windows о новом процессе. Этап 5: запуск первичного потока. Этап 6: инициализация в контексте нового процесса.

Потоки

Понятие потока в Windows. Структура данных описывающая поток ETHREAD. Ключевые поля структуры ETHREAD. Структура W32THREAD. Структура KTHREAD. Ключевые поля структуры KTHREAD.

Основные переменные уровня ядра, связанные с потоками.

Основные счетчики производительности связанные с потоками.

Основные функции Win32 API по работе с потоками.

Этапы создания потока в Windows.

Планирование потоков. Вытесняющее планирование потоков на основе приоритета. Привязка к CPU. Диспетчер ядра. События, вызывающие диспетчеризацию потоков. Когда изменяется поток приоритета. Изменение привязки к CPU. Переключение контекста. Пример переключения контекста.

Уровни приоритетов в Windows. Взгляд на уровни приоритетов со стороны уровня пользователя и ядра. Приоритеты потоков в рамках процесса. Текущий и базовые приоритеты потоков. Изменение приоритета потока.

Функции Win32 API связанные с планированием потоков. Приоритеты реального времени. Состояния потоков в клиентской версии Windows. Состояния потоков в серверной версии Windows.

База данных диспетчера ядра потоков. Отличие БД диспетчера ядра потоков в клиентских и серверных версиях Windows. Очередь готовых потоков. Сводка готовности.

Квант. Отличие кванта в клиентских и серверных версиях Windows. Учет квантов. Управление величиной кванта. Динамическое увеличение кванта. Настройка длины кванта через реестр.

Сценарии планирования потоков. Самостоятельное переключение. Вытеснение. Завершение кванта. Завершение кванта. Переключение контекста.

Поток простоя. Его действия. Динамическое повышение приоритета потока.

Особенности планирования потоков для многопроцессорных систем. БД диспетчера ядра в многопроцессорных системах. Маска активных CPU. Сводка простоя. Синхронизация процессоров при доступе к общим структурам данных уровня ядра. Состояние потока – готов, но отложен. Поддержка Hyperthreading.

Системы NUMA. Привязка потоков к процессорам. Идеальный и последовательный процессор. Особенности поддержки Hyperthreading. Особенности поддержки NUMA процессоров.

Алгоритмы планирования в многопроцессорных системах. Выбор CPU для потока при наличии простаивающего CPU. Выбор CPU для потока в отсутствие простаивающего CPU. Выбор потока для выполнения на конкретном CP: клиентская и серверная версия Windows.

Задания

Понятие задания. Функции Win32 API для заданий. Ограничения на задания.

Управление памятью

Размер памяти, поддерживаемый ОС Windows для разных платформ. Задачи диспетчера памяти. Трансляция (проецирование). Рабочий набор. Подкачка. Базовый набор сервисов. Компоненты диспетчера памяти. Системные потоки режима ядра диспетчера памяти.

Синхронизация доступа к общесистемным ресурсам (например номерам фреймов страниц).

Настройка диспетчер памяти.

Основные сервисы диспетчера памяти, которые предоставляются через функции Win32 API. Функции Win32 API для работы с основными сервисами диспетчера памяти. Разбиение виртуального адресного пространства на страницы. Преимущества и недостатки больших и малых страниц.

Резервирование и передача страниц: описание, преимущества и недостатки. Возврат и освобождение страниц: описание, преимущества и недостатки.

Блокировка памяти на уровне пользователя и ядра. Гранулярность выделения памяти в режиме пользователя и ядра.

Разделяемая память и проецируемые файлы: описание, преимущества и недостатки. Закрытые области памяти. Объект – проекция файла. Разделы с поддержкой страничных файлов. Функции Win32 API для работы с файлами, проецируемыми в память.

Защита памяти в Windows. Основные способы защиты памяти. Атрибуты защиты страниц памяти. Атрибут – запрет на выполнение: описание, применение и его поддержка различными аппаратными платформами. Атрибут – копирование при записи: описание, применение, алгоритм отложенной оценки. Диспетчер куч: описание, назначение, функции Win32 API.

Механизм Address Windowing Extension: описание механизма, назначение, ограничения, преимущества и недостатки, какие версии ОС Windows поддерживают.

Системные пулы памяти: подкачиваемый и неподкачиваемые пулы. Реализация системных пулов в однопроцессорных системах. Реализация системных пулов в многопроцессорных системах. Настройка размеров пулов. Максимальные размеры пулов для 32-х и 64-х разрядных архитектур. Мониторинг размеров системных пулов.

Ассоциативные списки: описание, назначение.

Структуры виртуального адресного пространства.

Лекция 12. Управление вводом выводом. Файловая система. Безопасность

Безопасность в ОС Windows

Классы безопасности. Trusted Computer System Evaluation Criteria (TCSEC). Рейтинг безопасности TCSEC. Требования, которым удовлетворяет уровень безопасности C2. Common Criteria (CC). Преимущества CC перед TCSEC. Концепции CC. Требования Controlled Access Protection Profile к Windows.

Компоненты и БД защиты в ОС Windows.

Защита объектов и протоколирование обращений к ним. Олицетворение: описание, назначение. Проверка прав доступа. Описание механизма проверки при открытии объекта по имени: стандартная и нестандартная защита (пример доступа к файлу). Описание механизма проверки при ссылке на объект по существующему описателю.

Идентификаторы защиты – SID: структура, варианты, объекты для которых генерируется.

Маркер и альтернативный маркер. Структура маркера. Список прав доступа. Список привилегий. Список управления избирательным доступом. Основной маркер. Олицетворяющий маркер. Локально уникальный идентификатор. Идентификатор аутентификации.

Механизм олицетворения. Поддержка механизма олицетворения в ОС Windows. Механизм делегирования.

Ограниченный маркер. Дескриптор защиты. Структура дескриптора защиты.

Алгоритмы определения прав доступа к объекту.

Права и привилегии учетных записей. Суперпривилегии.

Вход в систему. Пакет аутентификации. Winlogon. GINA. Схема взаимодействия между компонентами в процессе регистрации пользователя. Унифицированная регистрация.

Инициализация Winlogon.

Политики ограниченного использования программ. Места применения такой политики.

Управление вводом-выводом

Компоненты подсистемы ввода-вывода. Графическая интерпретация подсистемы ввода-вывода. Функциональность подсистемы ввода-вывода.

Диспетчер ввода-вывода. Пакеты ввода-вывода. Механизм организации ввода-вывода диспетчером.

Типичная обработка ввода-вывода. Базовая схема обработки запроса ввода-вывода.

Драйверы устройств

Драйверы устройств. Типы драйверов устройств: пользовательский режим и режим ядра. WDM-драйверы. Многоуровневые драйверы. Пример взаимодействия драйвера файловой системы и драйвера диска. Промежуточные драйверы, например драйвер тома диска.

Структура драйвера. Основные процедуры драйвера устройства. Объект драйвер. Объект устройство. Открытие устройств, на примере объекта файла.

Обработка ввода-вывода

Типы ввода-вывода. Синхронный и асинхронный ввод-вывод. Быстрый ввод-вывод. Ввод-вывод в проецируемый файл. Ввод-вывод по механизму scatter/gather. Пакеты запроса ввода-вывода.

Запрос ввода-вывода к одноуровневому драйверу. Обслуживание прерывания. Завершение обработки запроса ввода-вывода. Синхронизация.

Запрос ввода-вывода к многоуровневому драйверу. Обслуживание прерывания. Завершение обработки запроса ввода-вывода. Синхронизация.

Порты завершения ввода-вывода: актуальность, описание. Порт завершения. Пакет завершения. Использование портов завершения. Описание механизма портов завершения ввода-вывода.

Диспетчер Plug & Play

Диспетчер PnP: описание, назначение, возможности. Уровень поддержки PnP. Поддержка PnP со стороны драйвера. Переходы PnP-состояний устройств.

Загрузка, инициализация и установка драйвера. Параметр Start. Перечисление устройств (узел устройств, дерево устройств). Внутренняя структура узла устройств. Загрузка драйвера для узла устройств. Установка драйвера. Компоненты участвующие в установке драйвера. Идентификатор оборудования и идентификатор совместимых устройств. INF-файл.

Диспетчер электропитания

Стандарт ACPI. Состояния с различным энергопотреблением. Переход системы между различными состояниями энергопотребления. Состояния энергопотребления устройств.

Работа диспетчера электропитания. Первичная информация необходимая для перехода в другое состояние энергопотребления. Сопоставление между режимами энергопотребления системы и состояния устройства.

Участие драйверов в управлении электропитанием. Управление электропитанием устройства драйвером.

Управление внешней памятью

Диск. Сектор. Раздел. Простой том. Составной том.

Драйверы дисков. Стек драйверов устройств внешней памяти в Windows. Ntldr. Порт и минипорт драйверы. Драйверы iSCSI. Архитектура сети устройств хранения данных. MPIO драйверы. Стек устройств внешней памяти Windows MPIO.

Объекты устройств для дисков.

Диспетчер разделов.

Управление томами.

Базовые диски. Разбиение на разделы по схеме MBR. Разбиение на разделы по схеме GPT. Диспетчер томов на базовых дисках.

Динамические диски. База данных LDM: описание, назначение, структура. Организация динамических дисков. Диспетчер томов на динамических дисках.

Управление составными томами. Типы составных томов. Перекрытые тома: описание, назначение, структура. Зеркальные тома: описание, назначение, структура. Чередующиеся тома: описание, назначение, структура. RAID-5: описание, назначение, структура.

Пространство имен томов. Диспетчер монтирования. Точки монтирования. Монтирование томов. Поток ввода-вывода на смонтированном томе.

Операции ввода-вывода на томах. Служба виртуального диска: описание, назначение, архитектура. Служба теневого копирования тома: описание, назначение, архитектура.

Файловые системы Windows

Базовые понятия. Сектор. Формат файловой системы. Кластер. Основные файловые системы ОС Windows: CDFS, FAT12, FAT16, FAT32, UDF, NTFS.

Архитектура драйвера файловой системы (ФС). Особенности драйверов ФС. Драйверы локальных ФС: описание, назначение, архитектура. Драйверы сетевых ФС: описание, назначение, архитектура.

Схема взаимодействия компонентов ФС при вводе-выводе. Явный файловый ввод-вывод. Подсистема записи модифицированных и спроецированных страниц. Подсистема отложенной записи. Поток опережающего чтения. Обработчик ошибок страниц.

Драйверы фильтров файловой системы.

Цели разработки и особенности NTFS. Требования к надежной файловой системе. Восстанавливаемость. Защита. Избыточность данных и отказоустойчивость.

Дополнительные возможности NTFS: множественные потоки данных, имена на основе Unicode, универсальный механизм индексации, динамическое переназначение плохих кластеров, жесткие связи и точки соединения, сжатие и разреженные файлы, протоколирование изменений, квоты томов, отслеживание ссылок, шифрование, поддержка POSIX, дефрагментация, поддержка доступа только для чтения.

Драйвер ФС NTFS. NTFS и связанные с ней компоненты.

Структуры данных NTFS. Структура NTFS на диске. Тома. Кластеры. Главная таблица файлов (MFT). Структура записи MFT. Структура файловых ссылок. Записи о файлах. Имена файлов. Резидентные и нерезидентные атрибуты: описание, назначение, преимущества и недостатки, особенности представления файлов и каталогов различного размера.

Сжатие данных и разреженные файлы. Общие положения. Сжатие разреженных файлов. Сжатие неразреженных файлов.

Файл журнала изменений. Индексация имен файлов. Взаимосвязь между идентификатором файла и номером файловой ссылки. Отслеживание квот. Консолидированная защита. Точки повторного разбора.

Поддержка восстановления в NTFS. Файловые системы с точной записью. Файловые системы с отложенной записью. Восстанавливаемые файловые системы.

Протоколирование. Сервис файла журнала. Типы записей журнала.

Восстановление. Проход анализа. Проход повтора. Проход отмены.

Восстановление плохих кластеров в NTFS. Запись MFT для пользовательского файла с плохим кластером. Переназначение плохих кластеров.

Шифрованная файловая система (EFS): описание. Архитектура шифрованной файловой системы. Первое шифрование файла. Создание связки ключей.

Шифрование файлов данных. Схема работы EFS. Сводная схема процесса шифрования.

Расшифровывание данных. Кеширование расшифрованного ключа шифрования файла (FEK). Расшифровка файловых данных. Резервное копирование шифрованных файлов.

Функции Win32 API по работе с EFS.

Модуль 5. Операционные системы семейства UNIX

Лекция 13. История. Архитектура. Структура. Особенности программирования системных вызовов. Управление потоками и процессами

Архитектура UNIX

Общая архитектура ОС UNIX. Интерфейсы в ОС UNIX. Оболочка UNIX. Категории утилит ОС UNIX и их стандартизация. Структура ядра ОС UNIX.

Процессы в UNIX

Понятие процесса в ОС UNIX. Демоны в UNIX. Дочерние и родительские процессы. Идентификаторы процессов. Межпроцессное взаимодействие: Сигналы. Каналы. Системные вызовы управления процессами. Состояние зомби. Функционирование процесса в режиме ядра. Ядро: Структура пользователя. Машинные регистры. Состояние системного вызова. Таблица дескрипторов файлов. Учетная информация. Стек ядра. Таблица процессов. Параметры планирования процесса. Образ памяти процесса. Сигналы. Механизм создания дочернего процесса. Механизм создания нового процесса. Механизм копирования при записи. Пример выполнения команды Is, как дочернего процесса. Пример выполнения команды Is, как нового процесса.

Потоки

Проблемы реализации потоков в UNIX. Копирование всех потоков при создании нового процесса. Некопирование всех потоков при создании нового процесса. Обработка сигналов. Файловый ввод-вывод.

Особенности реализации потоков в Linux. Системный вызов clone. Особенности в реализации таблицы процессов и структуры пользователя.

Преимущества и недостатки в реализации потоков в пользовательском режиме. Преимущества и недостатки в реализации потоков в режиме ядра. Системные вызовы управления потоками. Механизм синхронизации посредством мьютексов. Системные вызовы управления мьютексами. Механизм синхронизации посредством условных переменных. Системные вызовы управления условными переменными.

Планирование в UNIX

Механизм планирования процессов. Низкоуровневый механизм планирования. Понятие многоуровневой очереди.

Планирование в Linux

Особенность механизм планирования потоков. Классы потоков. Механизм планирования потоков реального времени FIFO. Механизм планирования потоков реального времени циклически. Механизм планирования потоков с разделением времени. Приоритет планирования. Квант времени. Джиффи. Добродетель. Алгоритм планирования. Условия отнятия CPU у потока. Свойства алгоритма планирования.

Лекция 14. Управление памятью. Управление вводом выводом. Файловая система. Безопасность

Управление памятью в UNIX

Адресное пространство процесса. Текстовый (программный) сегмент. Пример отображения виртуального адресного пространства процесса на физическую память системы. Сегмент данных. Инициализированные и неинициализированные данные. Динамическое изменение размера сегмента данных. Сегмент стека. Пример управлением память при запуске одного и того же приложения разными пользователями.

Отображение файлов в адресное пространство памяти. Пример совместного использования файла отображаемого в адресное пространство разных процессов.

Системные вызовы управления памятью.

Реализация управления памятью. Свопинг (подкачка). Свопер. События запроса на свопинг. Механизм функционирования свопера. Учет свободного места в памяти и файле подкачки.

Постраничная подкачка. Механизм постраничной подкачки. Реализация постраничной подкачки: ядро и страничный демон Организация памяти при страничной организации памяти. Карта памяти. Механизм обработки страничного прерывания.

Алгоритм замещения страниц. Условия, при которых страничный демон производит замену страниц. Описание, преимущества и недостатки алгоритма замещения страниц: основной алгоритм часов. Описание, преимущества и недостатки алгоритма замещения страниц: алгоритм часов с двумя стрелками.

Алгоритм свопинга при страничной организации виртуальной памяти. Отличие алгоритма страничной подкачки System V от 4BSD.

Управление памятью в Linux

Организация виртуального адресного пространства. Описание области памяти через структуру `vm_area_struct`. Управление памятью дочернего процесса. Трехуровневая система подкачки страниц. Динамически загружаемые модули. «Дружественный» алгоритм выделения памяти. Алгоритм нарезания мелких фрагментов. Алгоритм выделения непрерывных областей в виртуальной памяти.

Механизм предоставления страниц по требованию. Организация пула страниц, для предоставления страниц по требованию. Процедуры которые вызываются в цикле страничного демона.

Ввод-вывод в UNIX

Концепция организации ввода-вывода. Специальные файлы. Блочные и символьные специальные файлы. Номер младшего и старшего устройства.

Работа с сетью. Сокеты. Типы сокетов. Механизм установления сетевого соединения через сокеты.

Системные вызовы ввода-вывода.

Реализация системы ввода-вывода в BSD. Архитектура драйвера устройства. Система ввода-вывода с блочными специальными файлами. Буферный кэш для чтения и записи. Концепция подгружаемых модулей в Linux для драйверов. Процесс загрузки подгружаемого модуля. Система ввода с символьными специальными файлами. С-списки. Дисциплина линии связи. Необработанный и обработанный символьный поток. Концепция ввода-вывода посредством потоков данных в System V.

Файловая система UNIX

Файл. Каталог. Основные каталоги. Концепция монтируемых файловых систем дисков в единую файловую систему. Блокировка, как механизм совместного использования файлов. Блокировка с монополизацией и без монополизации. Множественная блокировка.

Системные вызовы для управления файлами. Системные вызовы для управления каталогами и файловой системой.

Реализация традиционной файловой системы ОС UNIX. Расположение на диске. Суперблок. Индекс-узлы. Пример открытия файла в каталоге по относительному пути. Пример открытия файла в каталоге по абсолютному пути. Структура индекс-узла в System V. Массив дескрипторов файлов. Механизм поиска индекс-узла файла по дескриптору файла – таблица открытых файлов. Одинарный косвенный блок. Двойной косвенный блок. Тройной косвенный блок.

Реализация файловой системы Berkley Fast. Основные усовершенствования в Berkley Fast. Длинные имена. Структура каталоговой записи. Кэширование имен. Разбиение диска на группы цилиндров. Использование блоков различного размера.

Файловая система ОС Linux. История развития файловых систем для Linux. Реализация файловой системы Ext2. Разбиение диска на группы блоков. Структура записи группы блоков. Отличие файловой системы Ext2 от Berkley Fast.

Файловая система proc в ОС Linux.

Файловая система NFS. Архитектура файловой системы NFS. Протоколы файловой системы NFS. Протокол. Протокол монтирования каталогов: статическое монтирование и автосмонтирование, преимущества и недостатки. Протокол доступа к файлам и каталогам. Системные вызовы чтения файлов в NFS. Механизм блокировки при совместном доступе к файлам.

Реализация файловой системы NFS. Трехуровневая реализация файловой системы NFS. Структура уровней файловой системы. NFS: уровень системных вызовов, уровень VFS, уровень клиента. Локальные и удаленные индексные узлы. Опережающее чтение на клиенте. Кэширование данных на сервере. Кэширование данных на стороне клиента. Решение проблем обновления кэша.

Безопасность в UNIX

UID. GID. Основной механизм безопасности. Суперпользователь. Проблема регулируемого доступа к устройствам ввода-вывода. Решение проблемы регулируемого доступа к устройствам ввода-вывода.

Системные вызовы безопасности ОС UNIX.

Реализация механизмов безопасности в ОС UNIX. Вход в систему (регистрация) пользователя.

Модуль 6. Операционные системы семейства QNX

Лекция 15. История. Архитектура. Структура. Особенности программирования системных вызовов. Управление потоками и процессами

Лекция 16. Управление памятью. Управление вводом выводом. Файловая система. Безопасность

Модуль 7. Курсовая работа

Лекция 17 - Курсовая работа

Вопросы будут задаваться в индивидуальном порядке, согласно [теме курсовой работы](#).

Темы курсовых работ

Задание

Курсовая работа посвящена изучению особенностей ОС, а также разработке системного ПО. Ниже приведен перечень групп, на которые разбиты темы:

1. Разработка приложений с использованием библиотек и технологий параллельного программирования:

- [OpenMP](#). Данная библиотека поддерживается многими [компиляторами](#).
- [Intel TBB](#). Данная библиотека является независимой от компилятора.
- Встроенных в языки.

2. Разработка приложений с использованием библиотек и технологий распределенного программирования:

- RPC. Технология распределенного вычисления, например в реализации [Microsoft](#).
- [OpenMPI](#). Данная библиотека поддерживается многими ОС.
- Sockets. Технология межсетевого обмена.
- Web-сервисы. Например, можно воспользоваться библиотекой [gSOAP](#).
- Microsoft COM+/ Enterprise Services.

3. Разработка приложений, которые напрямую взаимодействуют с устройствами и подсистемами ОС:

- Накопителями на жестких магнитных дисках.
- Съёмными накопителями.
- Устройствами, подключаемыми через интерфейс USB.
- Процессорами.
- Оперативной памятью.

4. Разработка приложений – вспомогательных приложений для ОС (системных утилит):

- Синхронизация каталогов файловой системы.
- Гарантированное удаление файлов/каталогов файловой системы.
- Восстановление удаленных файлов/каталогов файловой системы.
- Работа с файлами-образами CD/DVD/Blue-Ray.
- Дефрагментация раздела/логического диска.

5. Эмуляция работы подсистем ОС:

- Эмуляция работы с файловой системой через командную строку.
- Эмуляция файловой системы.
- Эмуляция работы менеджера памяти.
- Эму

Перечень тем курсовых работ

Перечислим темы для курсовых работ:

1. Windows-приложение для реализации простейшей виртуальной файловой системы. Необходимо разработать консольное Windows-приложение для реализации простейшей обычной файловой системы, например FAT, в виде виртуальной файловой системы расположенной в файле.

2. Сервер Windows-приложения Интернет-пейджера. Необходимо разработать сервер Windows-приложения Интернет-пейджера по типу ICQ. Сервер обеспечивает подключение до 100 клиентов. Поддерживается передача, как текстовых сообщений, так и файлов размером до 10 Мб.

3. Клиент Windows-приложения Интернет-пейджера. Необходимо разработать клиент Windows-приложения Интернет-пейджера по типу ICQ. Клиент обеспечивает подключение к заданному серверу (указывается IP-адрес и порт). Поддерживает получение списка подключенных клиентов и позволяет передавать, как текстовые сообщения, так и файлов размером до 10 Мб.

4. Windows-приложение для мониторинга состояния SMART регистров жесткого диска. Разработать консольное Windows-приложение для мониторинга состояния SMART регистров жесткого диска. Приложение выводит список всех дисковых устройств в системе и выдает полную информацию об указанном жестком диске как устройстве, его идентификационные данные, модель, состояние SMART регистров и т.д.

5. Linux-приложение для мониторинга состояния SMART регистров жесткого диска. Разработать консольное Linux-приложение для мониторинга состояния SMART регистров жесткого диска. Приложение выводит список всех дисковых устройств в системе и выдает полную информацию об указанном жестком диске как устройстве, его идентификационные данные, модель, состояние SMART регистров и т.д.

6. Windows-сервер и Windows-клиент для хранения резервных копий файлов в сети. Необходимо разработать протокол для приема файлов по протоколу TCP от удаленных клиентов, на основе протокола следует разработать консольный Windows-сервер и консольный Windows-клиент, который будет архивировать принятые файл с помощью библиотеки zlib и раскладывать по каталогам клиентов для хранения. В каталоге клиента следует создавать подкаталоги с названиями дней, когда файлы были приняты. При передаче, следует оставлять исходное имя и расширение принятого файла, однако через точку указывать время приема, далее через точку следует указывать расширение архива (zip, gzip и т.д.).

7. Многопоточное Windows-приложение для анализа логов web-сервера с использованием технологии Intel TBB. Необходимо разработать консольное Windows-приложение в рамках лабораторной работы 2, с использованием технологии [Intel TBB](#).

8. Linux-сервер и Linux-клиент для хранения резервных копий файлов в сети. Необходимо разработать протокол для приема файлов по протоколу TCP от удаленных клиентов, на основе протокола следует разработать консольный Linux -сервер и консольный Linux-клиент, который будет архивировать принятые файл с помощью библиотеки zlib и раскладывать по каталогам клиентов для хранения. В каталоге клиента следует создавать подкаталоги с названиями дней, когда файлы были приняты. При передаче, следует оставлять исходное имя и расширение принятого файла, однако через точку указывать время приема, далее через точку следует указывать расширение архива. (zip, gzip и т.д.).

9. Распределенное Windows-приложение для анализа логов web-сервера с использованием технологии web-сервисов Microsoft Web-сервисов. Необходимо разработать консольное распределенное Windows-приложение в рамках лабораторной работы 2 с использованием Microsoft Web-сервисов. Предполагается, что приложение обрабатывает файлы логов, которые находятся на общедоступном сетевом ресурсе посредством различных компьютеров сети. Результаты работы приложения доступны через клиентское приложение. **(Зарезервировано за преподавателем)**.

9.Оконное Windows-приложение, которое позволяет построить графическую карту раздела диска, посредством которой можно определить какой файл, где находится (какие кластеры занимает). Необходимо разработать оконное Windows-приложение аналогичное Sysinternals DiskView.

10. Распределенное Linux-приложение для анализа логов web-сервера с использованием технологии web-сервисов gSOAP. Необходимо разработать консольное распределенное Linux-приложение в рамках лабораторной работы 2 с использованием web-сервисов gSOAP. Предполагается, что приложение обрабатывает файлы логов, которые находятся на общедоступном сетевом ресурсе посредством различных компьютеров сети. Результаты работы приложения доступны через клиентское приложение.

11. Windows-приложение для просмотра активных процессов в памяти. Необходимо разработать консольное Windows-приложение для просмотра активных процессов в памяти. Для каждого процесса следует показывать объем оперативной памяти, число открытых ресурсов, число чтений, число записей, число прочитанных байт, число записанных байт, возможность уничтожить выбранный процесс и все его дочерние процессы.

12. Linux-приложение для просмотра активных процессов в памяти. Необходимо разработать консольное Linux-приложение для просмотра активных процессов в памяти. Для каждого процесса следует показывать объем оперативной памяти, число открытых ресурсов, число чтений, число записей, число прочитанных байт, число записанных байт, возможность уничтожить выбранный процесс и все его дочерние процессы.

13. Windows-приложение для получения полной информации о процессоре, модулях памяти и материнской плате. Необходимо разработать оконное Windows-приложение для получения полной информации о процессоре, модулях памяти и материнской плате. За основу следует взять информацию, которую можно получить посредством SiSoftware Sandra.

14. Linux-приложение для получения полной информации о процессоре, модулях памяти и материнской плате. Необходимо разработать оконное Linux-приложение для получения полной информации о процессоре, модулях памяти и материнской плате. За основу следует взять информацию, которую можно получить посредством lshw, dmidecode.
15. Windows-приложение для создания виртуального CD/DVD диска. Необходимо разработать Windows-приложение для создания виртуального CD/DVD диска, с возможностью монтирования файлов ISO в качестве CD/DVD.
16. Windows-приложение для записи CD/DVD диска посредством Image Mastering API. Необходимо разработать Windows-приложение для записи CD/DVD диска посредством Image Mastering API.
17. Windows-приложение для мониторинга обращений к заданной директории/файлу файловой системы. Необходимо разработать Windows-приложение для мониторинга обращений к заданной директории/файлу. Все действия ОС по доступу к тому или ному файлу/директории необходимо заносить в журнал аудита расположенный в памяти, по окончании мониторинга результаты следует заносить в текстовый файл.
18. Linux-приложение для мониторинга обращений к заданной директории/файлу файловой системы. Необходимо разработать Linux-приложение для мониторинга обращений к заданной директории/файлу. Все действия операционной системы по доступу к тому или ному файлу/директории необходимо заносить в журнал аудита расположенный в памяти, по окончании мониторинга результаты следует заносить в текстовый файл.
19. Windows-приложение для мониторинга обращений к заданной ветке/значению реестра. Необходимо разработать консольное Windows-приложение для мониторинга обращений к заданной ветке/значению реестра. Все действия ОС по доступу к тому или ному значению/ветке реестра необходимо заносить в журнал аудита (текстовый файл).
20. Windows-приложение для гарантированного уничтожения заданного файла/директории. Необходимо разработать консольное Windows-приложение для гарантированного уничтожения заданного файла/директории. Удаленную информацию невозможно восстановить посредством специализированного ПО.
21. Linux-приложение для гарантированного уничтожения заданного файла/директории. Необходимо разработать консольное Linux-приложение для гарантированного уничтожения заданного файла/директории. Удаленную информацию невозможно восстановить посредством специализированного ПО.
22. Windows-приложение для криптографической защиты файлов и каталогов с использованием алгоритма AES. Необходимо разработать консольное Windows-приложение, которое позволяет генерировать ключи, зашифровывать и расшифровывать с помощью алгоритма AES файлы и каталоги файловой системы. Доступ к секретному ключу (в файле контейнере), следует организовать посредством пароля. Криптографические преобразования выполнять посредством криптопровайдера Microsoft Windows CSP.
23. Windows-приложение для криптографической защиты файлов и каталогов с использованием алгоритма AES. Необходимо разработать консольное Windows-приложение, которое позволяет генерировать ключи, зашифровывать и расшифровывать с помощью алгоритма AES файлы и каталоги файловой системы. Доступ к секретному ключу (в файле контейнере), следует организовать посредством пароля. Криптографические преобразования использовать из Intel Performance Primitives Cryptography Package.
24. Windows-приложение для криптографической защиты файлов и каталогов с использованием алгоритма AES. Необходимо разработать консольное Windows-приложение, которое позволяет генерировать ключи, зашифровывать и расшифровывать с помощью алгоритма AES файлы и каталоги файловой системы. Доступ к секретному ключу (в файле контейнере), следует организовать посредством пароля. Криптографические преобразования использовать из библиотеки Crypto++.
25. Windows-приложение для защищенного обмена файлами в сети Ethernet. Необходимо разработать консольное Windows-приложение для защищенного обмена файлами в сети Ethernet между двумя компьютерами. Один иницирует соединение, другой ожидает это соединение. Предполагается, что на стороне клиента и сервера уже имеются секретные ключи для организации защищенного файлового обмена. За основу

следует взять socket соединения. Криптографические преобразования следует выполнять посредством криптопровайдера Microsoft Windows CSP.

26. Windows-приложение для просмотра содержимого файлов образов *.iso для CD/DVD. Необходимо разработать консольное или оконное Windows-приложение для просмотра содержимого файлов образов дисков *.iso для CD/DVD.

27. Распределенное Windows-приложение для анализа логов web-сервера с использованием библиотеки [Open MPI](#). Необходимо разработать консольное распределенное Windows-приложение в рамках лабораторной работы 2 с использованием библиотеки [Open MPI](#). Предполагается, что приложение обрабатывает файлы логов, которые находятся на общедоступном сетевом ресурсе посредством различных компьютеров сети. Результаты работы приложения доступны через основное приложение.

28. Распределенное Linux-приложение для анализа логов web-сервера с использованием библиотеки [Open MPI](#). Необходимо разработать консольное распределенное Linux-приложение в рамках лабораторной работы 2 с использованием библиотеки [Open MPI](#). Предполагается, что приложение обрабатывает файлы логов, которые находятся на общедоступном сетевом ресурсе посредством различных компьютеров сети. Результаты работы приложения доступны через основное приложение.

29. Windows-приложение, для восстановления файлов после удаления в файловых системах NTFS и FAT. Необходимо разработать оконное Windows-приложение, для восстановления файлов после удаления. Поддерживаются следующие файловые системы: NTFS, FAT.

30. Linux-приложение для защищенного обмена файлами в сети Ethernet. Необходимо разработать консольное Linux-приложение для защищенного обмена файлами в сети Ethernet между двумя компьютерами. Один иницирует соединение, другой ожидает это соединение. Предполагается, что на стороне клиента и сервера уже имеются секретные ключи для организации защищенного файлового обмена. За основу следует взять socket соединения. Криптографические преобразования следует выполнять посредством библиотеки OpenSSL.

31. Linux-приложение, для восстановления файлов после удаления в файловых системах Ext2 и Ext4. Необходимо разработать оконное Linux-приложение, для восстановления файлов после удаления. Поддерживаются следующие файловые системы: Ext2, Ext4.

32. Windows-приложение, для моделирования менеджера управления памятью со страничной организацией на основе алгоритма WSClock. Необходимо разработать Windows-приложение, для моделирования менеджера управления памятью со страничной организацией на основе алгоритма WSClock. Исследовать заданный алгоритм для различного размера страницы, общего размера памяти вычислительной системы, и т.д.

33. Windows-приложение, для моделирования менеджера управления памятью со страничной организацией на основе алгоритма «Рабочий набор». Необходимо разработать Windows-приложение, для моделирования менеджера управления памятью со страничной организацией на основе алгоритма «Рабочий набор». Исследовать заданный алгоритм для различного размера страницы, общего размера памяти вычислительной системы, и т.д.

34. Windows-приложение для побайтного сравнения содержимого двух каталогов, с поддержкой операций файлового ввода-вывода с нестабильным устройством хранения данных. Необходимо разработать Windows-приложение, с поддержкой операций файлового ввода-вывода с нестабильным устройством хранения данных. Предполагается, что могут происходить сбои во время открытия/закрытия, чтения/записи данных в файл.

35. Windows-драйвер для подсчета нажатий клавиш и сохранения их скан-кодов в отдельный файл. Необходимо разработать WDF-драйвер, который фиксирует скан-коды всех нажатых клавиш в текстовом файле.

36. Windows-приложение для чтения физических адресов оперативной памяти компьютера. Необходимо разработать оконное Windows-приложение для чтения физических адресов оперативной памяти компьютера. Приложение позволяет осуществлять: переход по заданному адресу, поиск заданной последовательности. Информация представляется в 16-ричной системе исчисления и в виде ASCII символов.

37. Windows-приложение для чтения физических адресов памяти заданного накопителя компьютера. Необходимо разработать оконное Windows-приложение для чтения физических адресов памяти заданного накопителя компьютера. Приложение позволяет осуществлять: переход по заданному адресу, поиск заданной последовательности. Информация представляется в 16-ричной системе исчисления и в виде ASCII символов.
38. Windows-приложение, которое использует старшие адреса памяти (выше 2-го Гбайта) с использованием Address Windowing Extensions API для 32-х разрядных ОС. Необходимо разработать консольное Windows-приложение, которое использует старшие адреса памяти (выше 2-го Гбайта) с использованием Address Windowing Extensions API (AWE API) для адресации виртуального 32-битного адресного пространства при наличии 3-х и более Гбайт физической оперативной памяти. Произвести замеры времени копирования различных блоков памяти в старшие адреса и из старших адресов.
39. Linux-приложение для чтения физических адресов оперативной памяти компьютера. Необходимо разработать оконное Linux-приложение для чтения физических адресов оперативной памяти компьютера. Приложение позволяет осуществлять: переход по заданному адресу, поиск заданной последовательности. Информация представляется в 16-ричной системе исчисления и в виде ASCII символов.
40. Linux-приложение для чтения физических адресов памяти заданного накопителя компьютера. Необходимо разработать оконное Linux-приложение для чтения физических адресов памяти заданного накопителя компьютера. Приложение позволяет осуществлять: переход по заданному адресу, поиск заданной последовательности. Информация представляется в 16-ричной системе исчисления и в виде ASCII символов.
41. Windows-приложение для планирования выполнения задач, используя Windows Task Scheduler. Необходимо разработать оконное Windows-приложение для управления (планирования) задач используя Windows Task Scheduler. Приложение позволяет: создавать, редактировать и удалять заданиями для выполнения их ОС.
42. Windows-приложение для анализа сетевого трафика используя Network Monitor API. Необходимо разработать оконное Windows-приложение для анализа сетевого трафика используя Network Monitor API (входит в Platform SDK).
43. Windows-приложение для синхронизации заданных каталогов на 2-х компьютерах между собой. Необходимо разработать консольное Windows-приложение для синхронизации заданных папок на 2-х компьютерах между собой. Другими словами, на этих компьютерах запущено приложение, которое по команде проверяет содержимое заданной папки и производит синхронизацию с содержимым папки (папки не являются публичными) на другом компьютере (пример такого приложения GoodSync).
44. Windows-приложение для защищенного обмена файлами используя Named Pipe. Необходимо разработать консольное Windows-приложение для защищенного обмена файлами используя Named Pipe между двумя компьютерами. Один иницирует соединение, другой ожидает это соединение. Предполагается, что на стороне клиента и сервера уже имеются секретные ключи для организации защищенного файлового обмена. За основу следует взять socket соединения. Криптографические преобразования следует выполнять посредством криптопровайдера Microsoft Windows CSP.
45. Консольное Linux-приложение для просмотра содержимого файлов образов дисков *.iso для CD/DVD. Разработать консольное (оконное) Linux-приложение для просмотра содержимого файлов образов *.iso для CD/DVD.
46. Консольное Windows-приложение для установки хуков для Диспетчера задач Windows. Windows-приложение которое устанавливает hook, который позволяет изменить заголовок окна Диспетчера задач Windows, например на «Курсовая работа студента 431 группы Николаева Петра Васильевича».
47. Консольное Linux-приложение для управления (планирования) задач. Необходимо разработать консольное Linux-приложение, которое позволяет: создавать, редактировать и удалять заданиями для выполнения их ОС, например cron-скрипты.
48. Консольное Linux-приложение для синхронизации заданных папок на 2-х компьютерах между собой по сети. Необходимо разработать консольное Linux-приложение для синхронизации заданных каталогов на 2-х компьютерах между собой по сети. Другими словами, на этих компьютерах запущено приложение, которое по команде проверяет содержимое заданной папки и производит синхронизацию с

содержимым папки (папки не являются публичными) на другом компьютере (пример такого приложения GoodSync).

49. Консольное Linux-приложение для побайтного сравнения содержимого двух каталогов, с поддержкой операций файлового ввода-вывода с нестабильным устройством хранения данных. Необходимо разработать Linux-приложение, с поддержкой операций файлового ввода-вывода с нестабильным устройством хранения данных. Предполагается, что могут происходить сбои во время открытия/закрытия, чтения/записи данных в файл.

50. Консольное Linux-приложение для защищенного обмена файлами в сети. Необходимо разработать консольное Linux-приложение для защищенного обмена файлами в сети. За основу следует взять socket соединения. Криптографические преобразования следует выполнять посредством библиотеки OpenSSL. Клиент и сервер обязаны обладать соответствующими сертификатами, с длиной ключа RSA не менее 1024 бит.

51. Консольное Linux-приложение для получения информации о каталоге. Необходимо разработать Linux-приложение, которое может получить размер всех файлов, число файлов, число каталогов, которые находятся в заданном каталоге. В случае необходимости, может быть произведено рекурсивное сканирование всех каталогов, которые находятся в заданном каталоге. За пример следует взять приложение Sysinternals Disk Usage.

52. Консольное Linux-приложение для записи CD/DVD диска. Необходимо разработать консольное Linux-приложение для записи CD/DVD дисков используя библиотеки libburn/brasero/xfburn/cdw.

53. Консольное Windows-приложение для получения списка загруженных динамических DLL в оперативную память. Необходимо разработать консольное Windows-приложение, для отображения списка всех DLL загруженных в память, с указанием имени процесса, который ее загрузил, а также версию загруженной библиотеки. За пример следует взять приложение Sysinternals ListDLL.

54. Консольное Windows-приложение для получения информации о каталоге. Необходимо разработать Windows-приложение, которое может получить размер всех файлов, число файлов, число каталогов, которые находятся в заданном каталоге. В случае необходимости, может быть произведено рекурсивное сканирование всех каталогов, которые находятся в заданном каталоге. За пример следует взять приложение Sysinternals Disk Usage.

55. Консольное Windows-приложение для дефрагментации наиболее часто используемых файлов. Необходимо разработать консольное windows-приложение, которое может дефрагментировать (последовательно разместить все кластеры) файл. За пример следует взять приложение Sysinternals Contig.

56. Разработать консольное Linux-приложение для сравнения аллокаторов памяти glibc, ptmalloc, hoard, concur, tcmalloc, jemalloc, по типу: <http://www.facebook.com/notes/facebook-engineering/scalable-memory-allocation-using-jemalloc/480222803919>

Требования к содержанию пояснительной записки

При оформлении курсовой работы следует пользоваться [методическими рекомендациями разработанными кафедрой и университетом](#).

Титульный лист – содержит название ВУЗа, института, кафедры, которая читает данную дисциплину. Указывается группа и ФИО студента. Указывается кафедра, ФИО, ученое звание и степень преподавателя.

Задание на выполнение курсовой работы – указывается задание на выполнение курсовой работы, место для подписи студента, получившего задание и преподавателя, выдавшего задание.

Реферат – дается краткая характеристика всей работы. Указывается задача, которая решалась, каким образом она решалась, и какие результаты были получены.

Содержание – автоматически собирается на основе стилей заголовков 1, 2 и 3-го уровней.

Список условных обозначений, символов, единиц, сокращений и терминов – описываются основные сокращения, которые используются в работе, малоупотребимые

(обладающие различной трактовкой) символы, даются расшифровки терминов используемых в работе.

Введение – описывается необходимость (актуальность) проводимых исследований (разработки). Формулируется: цель, объект, предмет исследований (проекта) и задачи. Другими словами: зачем необходимо все это делать.

Раздел 1 – описывается общая теоретическая база, необходимая для реализации данной работы (проекта). Описывается общая архитектура решения, которое используется при разработке. Например, если используется архитектура «клиент-сервер», то необходимо описать, что это за архитектура и описать, какие технологии применяются для ее реализации. Например: COM+, RPC, web-сервис и т.д.

Раздел 2 – описывается общий алгоритм приложения, с соответствующими блок-схемами и пояснениями к ним.

Раздел 3 – описываются особенности реализации алгоритма в рамках заданной платформы. Какие функции, технологии и библиотеки применялись для реализации того или иного алгоритма.

Раздел 4 – описываются эксперименты проведенные в рамках курсовой работы. Указываются условия эксперимента, ход выполнения эксперимента, приводятся полученные результаты эксперимента, проводится анализ результатов эксперимента, делаются выводы.

Выводы – описываются выводы в рамках поставленных целей и задач.

Приложение А – методика тестирования приложения. Описываются: какие тесты проводились, какие входные данные, и какие результаты были получены.

Приложение Б – руководство пользователя по работе с приложением.

Перечень материалов к сдаче

Для сдачи курсовой работы студенту необходимо подготовить:

1. Пояснительную записку к курсовой работе.
2. Исходный код приложения. Обязательным является: модульность, объектно-ориентированный подход, детальное комментирование исходного кода, наличие комментариев для распространенных систем автоматической генерации документации к исходному коду. Например, для C++, считается приемлемым Doxygen. Для C# используется встроенный в интегрированную систему разработки – генератор.
3. Документация к исходному коду, сгенерированная с помощью систем автоматической генерации документации.
4. Функциональная спецификация к приложению (для получения дополнительного балла).

Темы заданий для расчетно-графической работы

Основу задания для расчетно-графической работы составляет исследование временных характеристик многопоточного приложения в ОСРВ QNX во время обработки информации от периферийных устройств.

Задание

Целью задания является:

- Исследование времени реакции сервера на запросы клиентов, формируемыми в реальном масштабе времени.

Объект исследования – процесс обработки пакетов сервером, полученных от удаленных клиентов.

Для достижения цели необходимо решить следующие задачи:

- Разработать серверное приложение, которое получает запросы по протоколу TCP|UDP от некоторого количества клиентов и выполняет обработку этих запросов.
- Разработать консольное клиентское приложение, которое производит подключение к заданному серверу и через некоторые промежутки времени формирует запрос к нему по протоколу TCP|UDP.
- Исследовать время передачи и обработки пакетов серверным приложением.

Пояснения к проведению эксперимента

Перед проведением эксперимента, следует подготовить указанное число компьютеров (в максимальном подключении), а также компьютер сервер и запустить на каждом из них соответствующее приложение под управлением ОС QNX (рекомендуется использование виртуальных машин).

Каждое клиентское приложение после запуска начинает формировать пакеты для передачи на сервер по заданному закону, кроме того размер пакетов также изменяется по заданному закону.

Содержимое пакетов формируется посредством генератора псевдослучайных чисел, встроенным в язык программирования.

Информация о принятых пакетах на стороне сервера фиксируется в лог файл, формат которого приведен в таблице 4.

Таблица 1. Варианты заданий

Вариант	Протокол	Максимальное количество одновременных подключений	Промежутки времени формирования запросов, вариант	Объем информации в запросе, кбайт
1	TCP	5	1	1
2	UDP	4	2	2
3	TCP	2	3	3
4	UDP	4	4	4
5	TCP	5	5	5
6	UDP	4	6	6
7	TCP	3	5	5
2	UDP	2	4	4
9	TCP	3	3	3
10	UDP	4	2	2

11	TCP	5	1	1
12	UDP	4	2	2
13	TCP	3	3	3
14	UDP	2	4	4
15	TCP	3	5	5
16	UDP	4	6	6
17	TCP	5	5	5
19	UDP	4	4	4
20	TCP	3	3	3
21	UDP	2	2	2
22	TCP	3	1	1
23	UDP	4	2	2
24	TCP	5	3	3
25	UDP	4	4	4
26	TCP	3	5	5
27	UDP	2	6	6
22	TCP	3	5	5
29	UDP	4	4	4
30	TCP	5	3	3

Таблица 2. Варианты закона изменения между концом и началом передачами пакетов от клиентов к серверу

Вариант	Закон изменения	Параметры
1	$n = (k \cdot t + b) \bmod \maxcon$, $\maxcon = 30$ -максимальный промежуток времени, мс	$k = v$, где v -номер по журналу; $b = 10$; величина t увеличивается на 1 каждые 50 мс; период моделирования 20 мин
2	$n = (k \cdot t + b)^2 \bmod \maxcon$, $\maxcon = 35$ -максимальный промежуток времени, мс	$k = v$, где v -номер по журналу; $b = 30$; величина t увеличивается на 1 каждые 30 мс; период моделирования 15 мин
3	$n = (k \cdot t + b)^3 \bmod \maxcon$, $\maxcon = 25$ -максимальный промежуток времени, мс	$k = v$, где v -номер по журналу; $b = 50$; величина t увеличивается на 1 каждые 65 мс; период моделирования 20 мин
4	$n = \exp(k \cdot t + b) \bmod \maxcon$, $\maxcon = 15$ -максимальный промежуток времени, мс	$k = v$, где v -номер по журналу; $b = 70$; величина t увеличивается на 1 каждые 45 мс; период моделирования 15 мин
5	$n = \exp(k \cdot t + b) \bmod \maxcon$,	$k = v$, где v -номер по журналу; $b = 40$;

	$maxcon = 20$ -максимальный промежуток времени, мс	величина t увеличивается на 1 каждые 35 мс; период моделирования 20 мин
6	$n = \exp(k \cdot t + b) \bmod maxcon$, $maxcon = 30$ -максимальный промежуток времени, мс	$k = v$, где v -номер по журналу; $b = 20$; величина t увеличивается на 1 каждые 40 мс; период моделирования 15 мин

Таблица 3. Варианты закона изменения количества передаваемой информации от клиента к серверу

Вариант	Закон изменения, кбайт	Параметры, кбайт
1	$n = (k \cdot t + b) \bmod maxsize$, $maxsize = 3072$ -максимальный объем	$k = v \cdot 256$, где v -номер по журналу; $b = 64$; величина t увеличивается на 1 после каждой передачи
2	$n = (k \cdot t + b)^2 \bmod maxsize$, $maxsize = 4096$ -максимальный объем	$k = v \cdot 512$, где v -номер по журналу; $b = 192$; величина t увеличивается на 1 после каждой передачи
3	$n = (k \cdot t + b)^3 \bmod maxsize$, $maxsize = 5120$ -максимальный объем	$k = v \cdot 128$, где v -номер по журналу; $b = 256$; величина t увеличивается на 1 после каждой передачи
4	$n = (k \cdot t + b) \bmod maxsize$, $maxsize = 3072$ -максимальный объем	$k = v \cdot 64$, где v -номер по журналу; $b = 128$; величина t увеличивается на 1 после каждой передачи
5	$n = (k \cdot t + b) \bmod maxsize$, $maxsize = 4096$ -максимальный объем	$k = v \cdot 256$, где v -номер по журналу; $b = 128$; величина t увеличивается на 1 после каждой передачи
6	$n = (k \cdot t + b) \bmod maxsize$, $maxsize = 7168$ -максимальный объем	$k = v \cdot 32$, где v -номер по журналу; $b = 128$; величина t увеличивается на 1 после каждой передачи

Таблица 4. Формат лог файла

Поле	Описание
Дата и время начала приема пакета	
Дата и время окончания приема пакета	
IP адрес	
Период времени, за который принимался пакет	
Размер пакета	

Обработка результатов эксперимента

После окончания проведения эксперимента следует проанализировать записи лог файла:

- Построить график зависимости интервала времени приема и обработки пакета, от размера пакета.
- Построить график загрузки процессора за весь период моделирования.

- Оценить пропускную способность сервера при заданных условиях моделирования.

Сделать выводы о возможностях оптимизации серверного приложения, для увеличения его пропускной способности.

Перечень материалов к сдаче

Для сдачи курсовой работы студенту необходимо подготовить:

1. Пояснительную записку к расчетно-графической работе.
2. Исходный код приложения. Обязательным является: модульность, объектно-ориентированный подход, детальное комментирование исходного кода, наличие комментариев для распространенных систем автоматической генерации документации к исходному коду. Например, для С++, считается приемлемым Doxygen. Для С# используется встроенный в интегрированную систему разработки – генератор.
3. Документация к исходному коду, сгенерированная с помощью систем автоматической генерации документации.
4. Функциональная спецификация к приложению (для получения дополнительного балла).

Билеты к модулям

Модуль 1

Вариант 1

1. Раскройте суть понятия ОС.
2. Дайте характеристику операционным системам второго поколения компьютеров. Опишите пакетная система обработки заданий.
3. Опишите многоуровневые ОС. Как происходит разделение на уровни. Достоинства и недостатки.
4. В чем состоит суть модели последовательных процессов. Что такое процесс. Когда происходит создание процесса. Когда происходит завершение процесса и причины. Иерархия процессов. Состояние процессов и переходы между ними. Планировщик процессов. Преимущества и недостатки такой модели.
5. Опишите всплывающие потоки. Особенности их применения. Достоинства и недостатки.
6. Перечислите проблемы межпроцессного взаимодействия. Перечислите проблемы межпоточного взаимодействия. Условия реализации совместной работы параллельных процессов и эффективного использования данных.
7. Перечислите основные примитивы межпроцессного взаимодействия. Опишите основной недостаток активного ожидания. Укажите проблему инверсии приоритетов.
8. Перечислите механизмы планирования в системах пакетной обработки. Опишите механизм: первым пришел – первым ушел. Опишите механизм: кратчайшая задача – первая. Опишите механизм: наименьшее оставшееся время выполнения. Подход: трехуровневое планирование.

Вариант 2

1. Укажите место ОС в структуре вычислительной системы.
2. Дайте характеристику операционным системам третьего поколения компьютеров. Раскройте понятие многозадачность.
3. Архитектура ОС с экзоядром. Особенности. Достоинства и недостатки.
4. Опишите монолитные ОС. Идея. Особенности. Достоинства и недостатки. Разделение на уровни.
5. Как происходит реализация модели процессов. Таблица процессов. Элемент таблицы. Иллюзия многозадачности. Схема обработки прерывания.
6. Понятие потока. Модель потока. Отличие потока и процесса. Модель потоков в модели процессов. В чем суть многопоточности. В чем суть безопасности и защиты в модели потоков.
7. Опишите проблему обедающих философов. Подходы к решению задачи.
8. Перечислите механизмы планирования в интерактивных системах. Опишите механизм: справедливое планирование.

Вариант 3

1. Укажите назначение ОС.
2. Охарактеризуйте операционные системы четвертого поколения компьютеров.
3. Приведите классификацию ОС, согласно решаемых задач.
4. Понятие процесса. Предпосылки многозадачности. Псевдопараллелизм.
5. Раскройте суть понятия поток. Стек данных в потоке. Состояния потока. «Вежливость» потоков. Преимущества и недостатки в использовании потоков.
6. Перечислите основные механизмы взаимного исключения с активным ожиданием. Опишите механизм с запрещением прерываний. Достоинства и недостатки. Опишите механизм с применением переменных блокировки. Достоинства и недостатки. Подходы используемые в ОС Windows.

7. Семафоры как механизм синхронизации. Идея. Решение проблемы производителя и потребителя с помощью семафоров.

8. Перечислите механизм планирования в интерактивных системах. Опишите механизм: несколько очередей. Опишите механизм: самый короткий процесс – следующий. Опишите механизм: гарантированное планирование. Опишите механизм: лотерейное планирование.

Вариант 4

1. Раскройте функцию ОС: расширение возможностей компьютера.

2. Охарактеризуйте операционные системы третьего поколения компьютеров. Какие тенденции в развитии вычислительных систем публичного пользования.

3. Технология виртуальной машины. Поддержка технологии виртуальных машин в процессорах семейства Intel и ОС Windows. Особенности, достоинства и недостатки.

4. Реализация потоков в режиме пользователя. Преимущества и недостатки реализации потоков. Проблема блокировка потоков и подходы к ее решению. Исключение - отсутствие необходимой страницы в памяти. Механизм переключение потоков. Особенности построения эффективных многопоточных приложений.

5. Перечислите основные механизмы взаимного исключения с активным ожиданием. Механизм Петерсона.

6. Мьютексы как механизм синхронизации. Идея. Преимущества мьютексов над критическими секциями.

7. Передача сообщений как механизм синхронизации. Механизм передачи сообщений. Достоинства и недостатки. Решение проблемы производителя и потребителя.

8. Перечислите механизм планирования в интерактивных системах. Опишите механизм: циклическое планирование. Опишите механизм: приоритетное планирование.

Вариант 5

1. Раскройте функцию ОС: управление ресурсами.

2. Раскройте причины появления UNIX и его ветвей в третьем поколении компьютеров. Стандарт POSIX.

3. Охарактеризуйте сетевые и распределенные ОС, кластеры. Отличие кластера от сетевых и распределенных ОС.

4. Раскройте суть реализации потоков в режиме ядра. Преимущества и недостатки реализации потоков. Механизм реактивация потоков.

5. Перечислите основные механизмы взаимного исключения с активным ожиданием. Механизм строгого чередование. Достоинства и недостатки.

6. Различие в подходах к синхронизации потоков и процессов. Решение проблемы совместного доступа к общему адресному пространству, для проверки семафора (мьютекса).

7. Опишите проблема производителя и потребителя. Какие существуют подходы к решению проблемы.

8. Перечислите механизм планирования в системах реального времени. Опишите условие планируемости. Политика и механизм планирования. Планирование потоков.

Вариант 6

1. Распределение (мультиплексирование) ресурсов во времени и пространстве.

2. Охарактеризуйте операционные системы третье поколение компьютеров. Охарактеризуйте режим разделения времени.

3. Операционные системы с архитектурой клиент-сервер. Тенденции. Серверные и клиентские процессы. Достоинства и недостатки. Способы обмена сообщениями в распределенных системах (особенности выполнения низкоуровневых команд).

4. Технология виртуальной машины. Предпосылки. Достоинства и недостатки. Монитор виртуальной машины.

5. Механизм реализации потоков, посредством активации планировщика – один из подходов смешанной реализации. Преимущества и недостатки. Основа механизма.
6. Механизм реализации взаимного исключения с активным ожиданием через команду TSL. Особенности использования.
7. Механизм синхронизации - мониторы. Предпосылки. Основная идея. Подходы к переключению процессов. Преимущества и недостатки.
8. Опишите проблему читателей и писателей. Подходы к решению задачи.

Модуль 2

Вариант 1

1. Дайте определения следующим понятиям: Ресурсы, выгружаемые и невыгружаемые ресурсы, взаимоблокировка. Приведите условия возникновения взаимоблокировки.
2. Опишите алгоритм банкира для одного видов ресурсов.
3. Опишите иерархию типов памяти в компьютерах. Дайте определение менеджеру памяти. Какие классы систем управления памятью вы знаете? Опишите однозадачную систему без подкачки с диска и присущие для нее модели организации памяти.
4. Управление памятью с помощью связанных списков: Способ учета используемой памяти. Методы сортировки списков, достоинства и недостатки. Опишите следующие алгоритмы и укажите их достоинства и недостатки: «первого подходящего участка», «следующего подходящего участка».
5. Место буфера быстрого преобразования адреса при виртуальной организации памяти: предпосылки, описание, структура записи, принцип действия.
6. Опишите суть аномалии Билэди. Моделирование алгоритмов замещения страниц посредством магазинных алгоритмов. Что такое последовательность обращений. Укажите характеристики систем со страничной организацией памяти. Приведите свойства магазинной модели.
7. Приведите классификацию устройств ввода-вывода на блочные и символьные, их отличия и особенности. Что такое контроллер устройства и его назначение. Интерфейс взаимодействия устройства и контроллера. Что такое исправление ошибок и зачем оно необходимо в устройствах ввода-вывода.

Вариант 2

1. Опишите процесс моделирования взаимоблокировок с помощью направленных графов.
2. Дайте определение понятиям безопасного и небезопасного состояния, приведите пример.
3. Опишите систему с поддержкой многозадачности с фиксированными разделами и присущие для нее модели организации памяти: модель с отдельными входными очередями к разделам, с единой очередью к разделам. Недостатки и преимущества. Подходы к решению возникших проблем.
4. Опишите следующие алгоритмы управления памятью с помощью связанных списков, укажите их достоинства и недостатки: «самый подходящий участок», «самый неподходящий участок».
5. Подход инвертирования таблицы страниц при организации виртуальной памяти: Предпосылки. Описание. Преимущества и недостатки. Опишите один из способов усовершенствования виртуальной организации виртуальной памяти посредством: буфера быстрого преобразования, хеш-таблицы.
6. Моделирование алгоритмов замещения страниц: Строка расстояний: описание, от чего зависит, влияние на производительность алгоритма замещения страниц. Прогнозирование частоты страничных прерываний: описание, зависимости.
7. Ввод-вывод, отображаемый в память: Взаимодействие контроллера устройства и CPU. Механизм использования портов ввода-вывода и областей памяти. Достоинства и недостатки.

Вариант 3

1. Опишите «страусовый алгоритм», как один из подходов к решению проблемы взаимоблокировок.
2. Суть подхода аккуратного предоставления ресурсов. Концепция безопасных состояний. Пример траектории ресурсов, для 2-х ресурсов.
3. Опишите процесс моделирования многозадачности, а именно: оценка степени загрузки CPU; степень многозадачности. Дайте графическую интерпретацию: рост производительности компьютера в зависимости от увеличения ОЗУ.
4. Опишите подходы к ускорению алгоритмов управления памятью с помощью связанных списков. Как изменится картина скоростных характеристик алгоритмов?
5. Опишите проблему страничного обмена: предпосылки, описание. Опишите алгоритмы замещения страниц, их достоинства и недостатки: Оптимальный алгоритм. Алгоритм NRU. Алгоритм FIFO.
6. Реализация систем управления страничной организацией памяти: Политика распределения памяти. Предпосылки локальной и глобальной политики. Дайте описание, графическую интерпретацию и особенности применения локальной и глобальной политики.
7. Использование прерываний для организации ввода-вывода. Что такое вектор прерываний. Опишите механизм обработки прерываний. Достоинства и недостатки. Особенности реализации механизма прерываний на современных компьютерах с конвейерной архитектурой, в виде точных (жесткие) и неточных (мягкие) прерываний. Свойства точных прерываний. Недостатки неточных прерываний для CPU конвейерной архитектурой.

Вариант 4

1. Опишите алгоритм обнаружения взаимоблокировок при наличии одного ресурса каждого типа. Приведите пример решения задачи для 7 процессов и 6 ресурсов.
2. Опишите алгоритм банкира для нескольких видов ресурсов.
3. Опишите проблемы, возникающие при реализации многозадачности. Какие подходы к решению этой проблемы существуют.
4. Один из первых подходов к организации виртуальной памяти: оверлейное управление памятью. Опишите основную идею виртуальной памяти и страничной организации памяти. Диспетчер памяти и его место в вычислительной системе.
5. Опишите алгоритмы замещения страниц, их достоинства и недостатки: «вторая попытка», «clock», LRU.
6. Реализация систем управления страничной организацией памяти: Регулирование нагрузки. Симптомы пробуксовки. Подходы к решению проблемы пробуксовки: выгрузка процессов на диск.
7. Принципы разработки ПО ввода-вывода: Ключевая концепция ПО ввода-вывода. Обработка и восстановление после ошибок. Способ переноса данных: синхронный и асинхронный. Буферизация. Выделенные устройства и устройства коллективного использования.

Вариант 5

1. Опишите алгоритм обнаружения блокировок при наличии нескольких ресурсов каждого типа.
2. Опишите подход к решению проблемы взаимоблокировок посредством: атаки условия взаимного исключения; атака условия удержания и ожидания. Укажите проблемы применения такой атаки, подходы к ее решению.
3. Подход к управлению памятью - подкачка. Опишите подходы к разделению памяти на фиксированные разделы и разделы переменного размера, в чем отличие, их достоинства и недостатки.
4. Основные понятия: виртуальный адрес и физический адрес; страница и страничный блок. Укажите связь между виртуальными и физическими адресами посредством таблицы страниц. Что такое страничное прерывание, когда оно возникает.

5. Опишите алгоритм замещения страниц «рабочий набор», его достоинства и недостатки. Что такое пробуксовка? Понятие рабочий набор. Модели рабочего набора. Опережающая подкачка. Методы формирования рабочего набора.
6. Реализация систем управления страничной организацией памяти: Размер страницы. Большой и небольшой размер страницы: достоинства и недостатки. Внутренняя фрагментация. Математическое обоснование оптимального размера страницы.
7. Способы осуществления операций ввода-вывода, их описание, достоинства и недостатки: Программный ввод-вывод. Ввод-вывод посредством прерываний. Ввод-вывод посредством DMA.

Вариант 6

1. Опишите следующие подходы к восстановлению, при помощи принудительной выгрузки ресурса: метод восстановления через откат, метод восстановления путем уничтожения процессов.
2. Опишите подход к решению проблемы взаимоблокировок посредством: атаки условия отсутствия принудительной выгрузки ресурса; атака условия циклического ожидания.
3. Управление памятью с помощью битовых массивов: Способы учета используемой памяти. Единичные блоки. Размер единичного блока. Достоинства и недостатки.
4. Дайте описание, требования, достоинства и недостатки для таблицы страниц, при организации виртуальной памяти. Многоуровневые таблицы страниц: описание, достоинства и недостатки. Опишите структуру элемента таблицы.
5. Опишите алгоритм WSClock замещения страниц, его достоинства и недостатки. Сводная таблица алгоритмов размещения страниц и комментариев по их применения.
6. Разделение памяти для данных и для команд: предпосылки, описание, достоинства и недостатки. Совместно используемые страницы: предпосылки, описание механизма, достоинства и недостатки, проблемы реализации.
7. Программируемые уровни ввода-вывода: Структура системы ввода-вывода: уровни и функции. Обработчики прерываний. Драйвер устройства. Независимое от устройств ПО ввода-вывода. Интерфейс драйверов устройств и ОС. Буферизация. Сообщения об ошибках. Захват и освобождение выделенных устройств. Скрытие конструктивных особенностей устройств.

Модуль 3

Вариант 1

1. Опишите три основные проблемы послужившие появлению долгосрочных хранилищ данных.
2. Опишите общую идею и механизм функционирования файлов, отображаемых в память. Какие трудности применения вы знаете.
3. Опишите механизм поиска файла посредством хеш-таблиц, его достоинства и недостатки, который используется при реализации каталогов. Механизм кеширования результатов поиска.
4. Какие существуют стратегии выделения дискового пространства под файлы. Укажите какие проблемы связаны с определением размера блока хранения файла, какие подходы к решению. Как зависит скорость чтения/записи данных с диска и эффективность использования дискового пространства от размера блока.
5. Что такое NUMA-мультипроцессоры? Укажите основную идею и ключевые характеристики. Какие основные преимущества перед UMA-системами. Системы с когерентным (CC-NUMA) и некогерентным (NU-NUMA) кешем. Опишите каталоговый мультипроцессор в CC-NUMA, каковы его достоинства и недостатки.
- 6.

Вариант 2

1. Сформулируйте требования к долговременным хранилищам данных.
2. Опишите структуру классической файловой системы.

3. Опишите функции ОС Windows по работе с файлами, отображаемыми в адресное пространство.

4. Опишите структуру каталоговой записи, размещения атрибутов файла при реализации каталогов. Опишите подходы к именованию файлов: подходы к реализации: фиксированный размер записи, альтернативный подход, записи переменной длины.

5. Опишите механизм учета свободных блоков методом связанных списков блоков диска, его достоинства и недостатки. Опишите механизм учета свободных блоков методом битовых массивов, его достоинства и недостатки. Какие известны подходы к усовершенствованию.

6. Опишите UMA-мультипроцессоры с многоступенчатыми коммутаторными сетями. В чем их основная идея. Сложные коммутаторы (несколько входов и выходов). Суть блокирующей сети. Как происходит межпроцессорный обмен посредством сообщений, структура сообщений.

Вариант 3

1. Дайте определение терминам: файл и устойчивость.

2. Опишите пять основных требований к надежному резервированию.

3. Опишите однокаталоговые файловые системы, их достоинства и недостатки.

4. Размещение файлов на диске посредством связанных списков с таблицей в памяти (FAT): описание, преимущества и недостатки, особенности применения. I-узлы: описание, преимущества и недостатки, особенности применения.

5. Опишите механизм организации дисковых квот. Что такое таблица квот пользователей. Какие бывают лимиты?

6. Опишите UMA-мультипроцессоры с координатными коммутаторами, в чем состоит основная идея. Опишите координатный коммутатор, его достоинства и недостатки. Опишите неблокирующую сеть.

Вариант 4

1. Опишите файловую систему с журнальной структурой. Как происходит запись файла.

2. Опишите стратегии к построению резервных копий.

3. Дайте описание двухуровневых файловых систем, достоинства и недостатки. Иерархия каталогов: описание, достоинства и недостатки.

4. Опишите механизм размещения файлов на диске посредством связанных списков, его преимущества и недостатки, особенности применения.

5. Опишите механизм кеширования для повышения производительности файловой системы.

6. Опишите UMA-мультипроцессоры с общей шиной, в чем состоит основная идея. Опишите UMA-мультипроцессоры: без кеша, с кешем, с кешем и локальной памятью. Какие существуют ограничения использования такой архитектуры.

Вариант 5

1. Структура исполняемого файла в ОС.

2. Перечислите и опишите основные функции по работе с файлами.

3. Подходы к размещению файлов на диске. Опишите непрерывные файлы, преимущества и недостатки, особенности применения.

4. Опишите стратегии резервного копирования: физическая и логическая архивация. Физическая архивация: описание, достоинства и недостатки. Логическая архивация: описание, достоинства и недостатки, особенности применения.

5. Опишите механизм опережающего чтения для повышения производительности файловой системы.

6. Опишите мультипроцессор с общей памятью, его особенность, основа межпроцессорного обмена. Особенности мультипроцессорных ОС.

Вариант 6

1. Опишите механизмы организации структуры файлов: файлы – последовательность байт, файлы – образы перфокарт и строк, файлы – деревья.
2. Перечислите и опишите основные функции по работе с каталогами.
3. Опишите особенности реализации разреженных файлов в современных файловых системах.
4. Опишите структуру раздела классической файловой системы.
5. Кеширование для повышения производительности файловой системы. Алгоритмы страничной организации памяти, как прототипы управления блочным дисковым кешем. Нежелательность алгоритма LRU. Модифицированный алгоритм LRU. Подходы к кешированию в UNIX, MS-DOS.
6. Стандартные подходы к построению высокопроизводительных систем. Многопроцессорные системы с общей памятью. Многопроцессорные системы с передачей сообщений. Многомашинные системы. Распределенные системы (слабосвязанные и сильносвязанные).

Модуль 4

Вариант 1

1. Опишите подход к планированию в мультипроцессорах посредством разделения времени. В чем суть единой структуры данных для планирования, каковы достоинства и недостатки.
2. В чем суть механизма умного планирования, приоритетного планирования, родственного планирования, двухуровневого планирования.
3. Опишите подход функционирования ОС на каждом CPU - метод персональной копии. Дайте графическую интерпретацию и основные замечания. В чем состоят достоинства и недостатки.
4. Организация коллективных операций в MPI.

Вариант 2

1. Опишите подход к функционированию ОС на мультипроцессорах - персональная копия ОС лишь CPU-хозяину, остальные – CPU-подчиненные.
2. Дайте графическую интерпретацию и основные замечания по схеме персональная копия ОС лишь CPU-хозяину, остальные – CPU-подчиненные. В чем состоят достоинства и недостатки.
3. Опишите подход к планированию в мультипроцессорах посредством совместного использования адресного пространства.
4. Опишите простейший алгоритм планирования, кратчайшее задание – первое, разбиение процессов на группы. Как осуществляется учет работающих и желающих работать процессов.

Вариант 3

1. Опишите ОС для симметричных мультипроцессоров. Дайте графическую интерпретацию, в чем состоят достоинства и недостатки.
2. Репликация страниц. Преимущества и недостатки. Последовательная непротиворечивость.
3. Балансировка нагрузки посредством детерминистического графового алгоритма.
4. Функции MPI по обмену информацией между узлами: «точка»-«точка», радиовещательные сообщения.

Вариант 4

1. Механизм синхронизации в мультипроцессорах посредством мьютекс-протокола. Взаимное исключение.

2. Кеширование и проблема конкуренции за шину. Решение проблемы кеширования: итерация пробного чтения, двоичный экспоненциальный откат, опрос собственной переменной блокировки.

3. Опишите механизм бригадного планирования – планирование во времени и пространстве, в чем его достоинства и недостатки.

4. Балансировка нагрузки посредством распределенного эвристического алгоритма (с инициативой отправителя, так и получателя).

Вариант 5

1. В чем состоит суть проблема ожидания в цикле либо переключение, поиск компромисса: простое ожидание, спин-блокировка, учет предыдущих интервалов ожидания.

2. Опишите программный интерфейс OpenMP и принципы распараллеливания в приложениях с его помощью.

3. Механизм передачи сообщений - вызов удаленных процедур. Маршalling данных.

4. Возможные проблемы в процессе реализации вызова удаленных процедур.

Вариант 6

1. Вопросы планирования: какой процесс запустить следующим, зависимость процессов между собой.

2. Блокирующие и неблокирующие вызовы. Решение проблемы неблокирующих вызовов.

3. Опишите механизм MPI и принципы построения распределенных приложений с его помощью.

4. Балансировка нагрузки посредством алгоритма торгов.

Модуль 5

Вариант 1

1. Модель ОС. Поддержка режимов функционирования современных CPU.

2. Понятие процесса в Windows. Понятие блока процесса и потока. Графическое представление структур данных процесса, потока и переменных окружения.

3. Понятие потока в Windows. Структура данных описывающая поток ETHTHREAD. Структура KTHREAD.

4. Размер памяти, поддерживаемый ОС Windows для разных платформ. Задачи диспетчера памяти. Трансляция (проецирование). Рабочий набор. Подкачка.

5. Классы безопасности. Trusted Computer System Evaluation Criteria (TCSEC). Рейтинг безопасности TCSEC. Требования, которым удовлетворяет уровень безопасности C2. Common Criteria (CC).

6. Компоненты подсистемы ввода-вывода. Графическая интерпретация подсистемы ввода-вывода.

7. Драйверы дисков. Стек драйверов устройств внешней памяти в Windows.

8. Архитектура драйвера файловой системы (ФС). Особенности драйверов ФС.

Вариант 2

1. Общая архитектура ОС Windows.

2. Основные этапы создания процесса в Windows.

3. Особенности планирования потоков для многопроцессорных систем. БД диспетчера ядра в многопроцессорных системах. Маска активных CPU.

4. Базовый набор сервисов. Компоненты диспетчера памяти.

5. Компоненты и БД защиты в ОС Windows.

6. Функциональность подсистемы ввода-вывода.

7. Базовые диски. Разбиение на разделы по схеме MBR. Разбиение на разделы по схеме GPT. Диспетчер томов на базовых дисках.

8. Драйверы локальных ФС: описание, назначение, архитектура.

Вариант 3

1. Понятие масштабируемости. Особенности многопроцессорной Windows.

2. Графическая интерпретация этапов создания процесса в Windows.

3. Основные функции Win32 API по работе с потоками.

4. Основные сервисы диспетчера памяти, которые предоставляются через функции Win32 API. Функции Win32 API для работы с основными сервисами диспетчера памяти.

5. Защита объектов и протоколирование обращений к ним. Олицетворение: описание, назначение.

6. Драйверы устройств. Типы драйверов устройств: пользовательский режим и режим ядра. WDM-драйверы. Многоуровневые драйверы.

7. Пространство имен томов. Диспетчер монтирования. Точки монтирования. Монтирование томов. Поток ввода-вывода на смонтированном томе.

8. Схема взаимодействия компонентов ФС при вводе-выводе. Явный файловый ввод-вывод. Подсистема записи модифицированных и спроецированных страниц.

Вариант 4

1. Ключевые подсистемы ОС Windows и их описание.

2. Этапы создания процесса. Этап 1: открытие образа, подлежащего выполнению. Этап 2: создание объекта процесс.

3. Этапы создания потока в Windows.

4. Разбиение виртуального адресного пространства на страницы. Преимущества и недостатки больших и малых страниц.

5. Идентификаторы защиты – SID: структура, варианты, объекты для которых генерируется.

6. Структура драйвера. Основные процедуры драйвера устройства. Объект драйвер. Объект устройство.

7. Динамические диски. База данных LDM: описание, назначение, структура. Организация динамических дисков. Диспетчер томов на динамических дисках.

8. Драйверы фильтров файловой системы.

Вариант 5

1. Общее понятие ядра. Объекты ядра. Управляющие объекты и объекты диспетчера.

2. Этапы создания процесса. Этап 3: создание первичного потока, стека и контекста. Этап 4: уведомление подсистемы Windows о новом процессе.

3. Планирование потоков. Вытесняющее планирование потоков на основе приоритета.

4. Резервирование и передача страниц: описание, преимущества и недостатки. Возврат и освобождение страниц: описание, преимущества и недостатки.

5. Структура маркера. Список прав доступа. Список привилегий. Список управления избирательным доступом. Основной маркер. Олицетворяющий маркер.

6. Типы ввода-вывода. Синхронный и асинхронный ввод-вывод. Быстрый ввод-вывод. Ввод-вывод в проецируемый файл. Ввод-вывод по механизму scatter/gather. Пакеты запроса ввода-вывода.

7. Управление составными томами. Типы составных томов. Перекрытые тома: описание, назначение, структура.

8. Цели разработки и особенности NTFS. Требования к надежной файловой системе. Восстанавливаемость. Защита. Избыточность данных и отказоустойчивость.

Вариант 6

1. Понятие уровня абстрагирования от аппаратуры. Чего позволяет достигнуть. Как реализуется.
2. Этапы создания процесса. Этап 5: запуск первичного потока. Этап 6: инициализация в контексте нового процесса.
3. Привязка потока к CPU. Диспетчер ядра. События, вызывающие диспетчеризацию потоков. Когда изменяется приоритет поток.
4. Защита памяти в Windows. Основные способы защиты памяти. Атрибуты защиты страниц памяти: описание, применение и его поддержка различными аппаратными платформами.
5. Алгоритмы определения прав доступа к объекту.
6. Запрос ввода-вывода к одноуровневому драйверу. Обслуживание прерывания. Завершение обработки запроса ввода-вывода. Синхронизация.
7. Операции ввода-вывода на томах. Служба виртуального диска: описание, назначение, архитектура. Служба теневого копирования тома: описание, назначение, архитектура.
8. Драйвер ФС NTFS. NTFS и связанные с ней компоненты.

Модуль 6

Вариант 1

1. Общая архитектура ОС UNIX.
2. Системные вызовы управления процессами.
3. Особенности реализации потоков в Linux: реализация таблицы процессов и структуры пользователя.
4. Механизм планирования процессов в ОС UNIX.
5. Адресное пространство процесса в ОС UNIX.
6. Работа с сетью в ОС UNIX. Сокеты. Типы сокетов. Механизм установления сетевого соединения через сокеты.
7. Реализация традиционной файловой системы ОС UNIX. Расположение на диске.
8. Основной механизм безопасности в ОС UNIX.

Вариант 2

1. Интерфейсы в ОС UNIX. Оболочка ОС UNIX.
2. Функционирование процесса в режиме ядра.
3. Особенности реализации потоков в Linux. Системный вызов clone.
4. Механизм планирования потоков в ОС Linux.
5. Отображение файлов в адресное пространство памяти в ОС UNIX. Пример совместного использования файла отображаемого в адресное пространство разных процессов.
6. Реализация системы ввода-вывода в BSD. Архитектура драйвера устройства.
7. Реализация файловой системы Berkley Fast. Основные усовершенствования в Berkley Fast.
8. Проблема регулируемого доступа к устройствам ввода-вывода в ОС UNIX.

Вариант 3

1. Структура ядра ОС UNIX.
2. Ядро: Параметры планирования процесса. Образ памяти процесса. Сигналы.
3. Проблемы реализации потоков в ОС UNIX.
4. Планирование процессов в ОС UNIX: Низкоуровневый механизм планирования. Понятие многоуровневой очереди.

5. Пример управлением памятью в ОС UNIX при запуске одного и того же приложения разными пользователями.
6. Система ввода-вывода с блочными специальными файлами. Буферный кэш для чтения и записи.
7. Файловая система ОС Linux. История развития файловых систем для Linux. Реализация файловой системы Ext2.
8. Решение проблемы регулируемого доступа к устройствам ввода-вывода в ОС UNIX.

Вариант 4

1. Понятие процесса в ОС UNIX. Идентификаторы процессов.
2. Ядро: Структура пользователя. Машинные регистры. Состояние системного вызова. Таблица дескрипторов файлов. Учетная информация. Стек ядра. Таблица процессов.
3. Особенности реализации потоков в ОС UNIX: Копирование всех потоков при создании нового процесса.
4. Механизм планирования процессов в ОС UNIX: Механизм вычисления приоритета процесса. Высокоуровневый механизм планирования.
5. Реализация управления памятью в ОС UNIX. Свопинг (подкачка). Свопер. События запроса на свопинг. Механизм функционирования свопера. Учет свободного места в памяти и файле подкачки.
6. Концепция подгружаемых модулей в ОС Linux для драйверов. Процесс загрузки подгружаемого модуля.
7. Файловая система NFS. Архитектура файловой системы NFS.
8. Системные вызовы безопасности ОС UNIX.

Вариант 5

1. Демоны в ОС UNIX. Дочерние и родительские процессы.
2. Механизм создания дочернего процесса.
3. Особенности реализации потоков в ОС UNIX: Некопирование всех потоков при создании нового процесса.
4. Механизм планирования в ОС Linux потоков реального времени FIFO.
5. Постраничная подкачка. Механизм постраничной подкачки. Реализация постраничной подкачки: ядро и страничный демон Организация памяти при страничной организации памяти. Карта памяти.
6. Системные вызовы ввода-вывода в ОС UNIX.
7. Трехуровневая реализация файловой системы NFS. Структура уровней файловой системы.
8. Реализация механизмов безопасности в ОС UNIX.

Вариант 6

1. Межпроцессное взаимодействие. Системные вызовы управления процессами.
2. Механизм создания нового процесса.
3. Особенности реализации потоков в ОС UNIX: Обработка сигналов. Файловый ввод-вывод.
4. Механизм планирования в ОС Linux потоков реального времени циклически.
5. Описание, преимущества и недостатки алгоритма замещения страниц: основной алгоритм часов. Описание, преимущества и недостатки алгоритма замещения страниц: алгоритм часов с двумя стрелками.
6. Система ввода с символьными специальными файлами. С-списки. Дисциплина линии связи.
7. NFS: Опережающее чтение на клиенте. Кэширование данных на сервере. Кэширование данных на стороне клиента. Решение проблем обновления кэша.
8. Вход в систему (регистрация) пользователя.

Модуль 7

Вариант 1

Вариант 2

Вариант 3

Вариант 4

Вариант 5

Вариант 6