

1 Вопросы по темам

1.1 Операционные системы.

Классификация ядер.

1. (2.1) Классификация операционных систем. Особенности ОС определенных типов. Виртуальная машина и иерархическая машина – декомпозиция системы на уровни иерархии, иерархическая структура Unix BSD, архитектуры ядер ОС – определение, примеры.

2. (10.1) Классификация ядер операционных систем. Особенности ОС с микроядром. Три состояния блокировки процесса при передаче сообщений. Достоинства и недостатки микро-ядерной архитектуры, операционная система Mach: основные абстракции.

1.2 Режимы работы. Защищенный режим. Системные таблицы.

3. (1.2) Три режима работы компьютера на базе процессоров Intel(x86). Адресация аппаратных прерываний в защищенном режиме: таблица дескрипторов прерываний (IDT) – формат дескриптора прерывания, типы шлюзов. Пример заполнения IDT из лабораторной работы.

4. (2.2 + 2009) Три режима работы вычислительной системы с архитектурой x86: особенности. Реальный режим: линия A20 – адресное заворачивание. Перевод компьютера в защищенный режим. Линия A20 в защищенном режиме: включение и выключение линии A20 (код из лабораторной работы). XMS.

5. (3.2) Защищенный режим: назначение системных таблиц – глобальной таблицы дескрипторов (GDT), таблицы дескрипторов прерываний (IDT), теневых регистров (структуры, описывающие дескрипторы GDT и IDT и заполнение дескрипторов в лабораторной работе по защищенному режиму).

6. (18.2) Защищенный режим: перевод компьютера в защищенный режим – последовательность действий; реализация – пример кода из лабораторной работы.

1.3 Прерывания

7. (3.1) Прерывания: классификация. Последовательность действий при выполнении запроса ввода-вывода. Обработчики аппаратных прерываний: виды и особенности. Функции обработчика прерывания от системного таймера (в ОС семейства Windows и семейства Linux).

8. (8.2) Аппаратные прерывания: задачи обработчика прерывания от системного таймера в защищенном режиме.

9. (10.2) Обработчик прерывания ДОС int 8h: функции; контроллер прерываний – схема, маскируемые и немаскируемые прерывания; запрет и разрешение маскируемых прерываний в обработчике int 8h, префиксная команда lock. пример кода обработчика прерывания от системного таймера из лабораторной работы по защищенному режиму.

10. (14.2) Аппаратные прерывания: типы аппаратных прерываний; особенности. Прерываний от устройств ввода-вывода: назначение и аппаратная реализация. Прерывание от системного таймера в защищенном режиме. Пример кода обработчика прерывания от системного таймера из лабораторной работы по защищенному режиму.

11. (17.2 + 2020) Прерывание от системного таймера в защищенном режиме: функции (по материалам лабораторной работы). Адресация прерываний от системного таймера в защищенном режиме (схема).

12. (20.2) Пересчет динамических приоритетов в ОС UNIX и Windows (лабораторная работа).

1.4 Процессы

1.4.1 Общее

13. (1.1) Определение ОС. Ресурсы вычислительной системы. Режимы ядра и задачи: переключение в режим ядра – классификация событий. Процесс, как единица декомпозиции системы, диаграмма состояний процесса с демонстрацией действий, выполняемых в режиме ядра. Переключение контекста. Потоки: типы потоков, особенности каждого типа потоков.

14. (6.1) Понятие процесса. Процесс как единица декомпозиции системы. Диаграмма состояний процесса с демонстрацией действий, выполняемых

в режиме ядра. Планирование и диспетчеризация. Классификация алгоритмов Планирования. Примеры алгоритмов планирования, соотнесенные с типами ОС. Процессы и потоки. Типы потоков.

15. (12.1) ОС с монолитным ядром. Переключение в режим ядра. Диаграмма состояний процесса и переход из одного состояния в другое – причины каждого перехода. Диаграмма состояний процесса в UNIX. Переключение контекста. Система прерываний.

1.4.2 UNIX. Системные вызовы

16. (9.2) UNIX: концепция процессов: иерархия процессов, процессы – "сироты процессы" "зомби демоны; примеры из лабораторной работы (5 программ).

17. (16.1) Процессы в UNIX: системные вызовы `fork()`, `exec()`, `wait()`, `signal()` – примеры из лабораторных работ.

18. (27.2) Лабораторная работа по UNIX: системные вызовы `fork()`, `wait()`, `exec()`, `pipe()`, `signal()`. Примеры программ из лабораторной работы.

19. (19.1) Процессы Unix: создание процесса в ОС Unix и запуск новой программы. Примеры программ из лабораторных работ, демонстрирующие эти действия. Системные вызовы `wait()` и `pipe()`: назначение, примеры из лабораторных работ. Процессы "сироты" "зомби" и "демоны".

20. (21.2) Процессы Unix: создание процесса в ОС Unix и запуск новой программы. Примеры из лабораторной работы (коды).

1.4.3 Взаимодействие параллельных процессов

21. (11.1) Параллельные процессы: взаимодействие, обоснование необходимости монопольного доступа к разделяемым переменным, способы взаимного исключения. Мониторы: определение; примеры – простой монитор и монитор кольцевой буфер.

22. (20.1) Процессы: взаимодействие параллельных процессов – монопольный доступ и взаимное исключение; программная реализация взаимного исключения – примеры, семафоры – определение, виды семафоров, примеры использования множественных семафоров из лабораторных работ "производство-потребление" и "читатели-писатели"

23. (22.1 + 2020) Процессы: взаимодействие параллельных процессов – монополюный доступ и взаимоисключение; аппаратная реализация взаимного исключения, спин-блокировка – реализация. Семафоры Дейкстры: определение, взаимное исключение с помощью семафоров, алгоритм "Производство-потребление" – решение Дейкстры.

1.4.4 Параллельные процессы. Средства UNIX

24. (11.2) Средства межпроцессного взаимодействия (IPC) операционной системы UNIX System V: очереди сообщений и программные каналы – сравнение, примеры (для программных каналов пример из лабораторной работы с сигналами)

25. (15.1) Межпроцессное взаимодействие в Unix System V (IPC): сигналы, программные каналы, семафоры и разделяемая память; примеры использования из лабораторных работ.

1.4.5 Производство-потребление. Читатели-писатели.

26. (5.2) Задача "Производство-потребление": алгоритм Эд. Дейкстры, реализация на семафорах UNIX (код из лабораторной работы).

27. (8.1) Взаимное исключение и синхронизация процессов и потоков. Семафоры: определение, виды. Семафор, как средство синхронизации и передачи сообщений. Семафоры UNIX: примеры решения задач с помощью семафоров: "Производство-потребление" и "Читатели-писатели" в UNIX (пример реализации в лабораторной работе).

28. (6.2) Обеспечение монополюного доступа к разделяемым данным в задаче "писатели-читатели": реализация на базе Win32 API (пример кодов лабораторной работы "читатели-писатели" для ОС Windows). (Сравнение мьютексов и семафоров).

29. (12.2) Задача: читатели-писатели – монитор Хоара, решение с использованием семафоров Unix и разделяемой памяти, пример реализации из лабораторной работы.

30. (19.2) Взаимодействие параллельных процессов: мониторы – определение; монитор Хоара "читатели-писатели" реализация в ОС Windows – пример из лабораторной работы.

31. (21.1 + 2020) Взаимодействие параллельных процессов: монопольное использование – реализация; типы реализации взаимодействия. Мониторы – определение, примеры: простой монитор, монитор "кольцевой буфер" и монитор "читатели-писатели". Пример реализации монитора "читатели-писатели" для ОС Windows. Алгоритм Э. Дейкстры "Алгоритм банкира" и алгоритм Хабермана с примеров определения состояния системы.

1.4.6 Проблемы распараллеливания. Тупики. Философы. Булочная. Лампорт.

32. (4.1) Тупики: Обнаружение тупиков для повторно используемых ресурсов методом редукации графа, способы представления графа, алгоритмы обнаружения тупиков. Пример анализа состояния системы методом редукации графа. Методы восстановления работоспособности системы.

33. (15.2 2020) Тупики: классификация ресурсов и их особенности. Четыре условия возникновения тупика. Методы исключения тупиков.

34. (4.2) Задача "Обедающие философы" – модели распределения ресурсов вычислительной системы. Множественные семафоры UNIX: системные вызовы, поддержка в системе, пример использования из лабораторной работы "производство-потребление".

35. (18.1) Процессы: взаимодействие параллельных процессов – монопольный доступ и взаимное исключение; программная реализация взаимного исключения – флаги, алгоритм Деккера, алгоритм Лампорта "Булочная".

36. (24.1) Процессы: взаимодействие параллельных процессов – монопольный доступ и взаимное исключение; алгоритм Лампорта "Булочная" и "Логические часы" Лампорта.

37. (22.2) Процессы: бесконечное откладывание, зависание, тупиковая ситуация – анализ на примере задачи об обедающих философах и примеры аналогичных ситуаций в ОС. Множественные семафоры в Linux: системные вызовы и поддержка в ОС Linux; примеры из лабораторных работ.

38. (24.2) Процессы: бесконечное откладывание, зависание, тупиковая ситуация – анализ на примере задачи об обедающих философах и примеры аналогичных ситуаций в ОС. Множественные семафоры в Linux: системные вызовы и поддержка в системе; пример из лабораторной работы "производство-потребление".

1.4.7 Параллельные процессы в распределенных системах

39. (13.2) Синхронизация и взаимное исключение параллельных процессов в распределенных системах: централизованный и распределенный алгоритмы, алгоритм Token-ring; сравнение алгоритмов. Транзакции: определение, особенности, двухфазный протокол фиксации.

40. (14.1) Процессы: взаимодействие процессов в распределенных системах; централизованный и распределенный алгоритмы, синхронизация логических часов (алгоритм Лампорта); RPC – механизм.

41. (15.2) Синхронизация и взаимное исключение параллельных процессов в распределенных системах: централизованный и распределенные алгоритмы: сравнение.

42. (16.1) Взаимодействие параллельных процессов: проблемы; монопольный доступ и взаимное исключение (определения); взаимодействие параллельных процессов в распределенных системах – особенности; централизованный алгоритм, распределенный алгоритм; синхронизация логических часов (алгоритм Лампорта).

1.4.8 Виртуальная память

43. (5.1) Виртуальная память: распределение памяти страницами по запросам, схема с гиперстраницами, обоснование использования данной схемы. Управление памятью страницами по запросам в архитектуре x86 – расширенное преобразование (PAE) – схема преобразования. Анализ страничного поведения процессов: свойство локальности, рабочее множество.

44. (9.1 + 2009) Виртуальная память: управление памятью страницами по запросу – три схемы преобразования; реализация страничного преобразования в компьютерах на базе процессоров Intel (x86): стандартное преобразование и PAE в защищенном режиме – схемы, размеры таблиц и их количество на каждом этапе преобразования. Сегментно-страничное распределение памяти по запросам (сегментами, разделенными на страницы по запросу???)

45. (13.1) Виртуальная память: управление памятью страницами по запросам – три схемы. Алгоритмы вытеснения страниц: демонстрация особенностей на модели траектории страниц. Рабочее множество – определение, гло-

бальное и локальное размещение. Флаги в дескрипторах страниц, предназначенные для реализации замещения страниц.

46. (17.1) Виртуальная память: распределение памяти страницами по запросам, свойство локальности, рабочее множество, анализ страничного поведения процессов. Схема страничного преобразования в процессорах Intel (x86) PAE – размеры таблиц и дескрипторов. Обоснование использования многоуровневого преобразования, кэш TLB – структура (процессор 486).

47. (27.1) Виртуальная память: управление памятью страницами по запросам – три схемы преобразования виртуального адреса к физическому. Алгоритмы вытеснения страниц: демонстрация особенностей на модели траектории страниц. Рабочее множество – определение, глобальное и локальное замещение. Флаги в дескрипторах страниц, предназначенные для реализации замещения страниц.

48. (7.1) Управление виртуальной памятью. Распределение памяти сегментами по запросам: схема преобразования виртуального адреса, способы организации таблиц сегментов, стратегии выбора разделов памяти для загрузки сегментов, алгоритмы и особенности замещения сегментов.

49. (7.2) Управление памятью сегментами по запросам в архитектуре x86. Организация таблиц сегментов в защищенном режиме. Формат дескриптора сегмента в таблицах дескрипторов сегментов (GDT и LDT) (заполнение полей дескрипторов GDT из лабораторной работы по защищенному режиму).