

Программа экзамена

Архитектура компьютера и операционные системы

Мехмат, фундаментальная информатика и информационные технологии, вечернее отделение

Архитектура компьютера

1. Физический уровень. Транзисторы. Комбинационная логика, создание электронных схем реализующих булевы функции. Критический путь, время прохождения сигнала по схеме и частота работы микропроцессора. Ячейки памяти и секвенциальная логика. Способы организации памяти.
2. Типичная архитектура микропроцессора. Арифметико-логическое устройство, регистры, управляющее устройство, шина. Основные типы архитектур процессора: RISC, CISC, VLIW. Параллельные архитектуры, SIMD/MIMD, многоядерные процессоры.
3. Микропрограммный уровень, организация работы модулей микропроцессора с помощью микропрограммы.
4. Представление чисел. Знаковые и беззнаковые целые числа. Числа с плавающей точкой. Числа с фиксированной точкой.
5. Уровень набора инструкций. Ассемблер и его преимущества. Основные особенности ассемблера: виды операций, способы указания операндов и обращения к памяти. Ассемблер X86. Краткая история развития архитектуры X86, ее основные особенности.
6. Современные архитектуры. Суперскалярные процессоры, конвейерные вычисления, векторные расширения. Предсказание ветвлений. Уровни кэшей, особенности доступа к оперативной памяти.

Операционные системы

1. Определение и основные особенности операционной системы. Основные компоненты операционных систем. Ядро, его функции. Процесс, адресное пространство, многозадачность. Виды многозадачности. Представление процесса внутри операционной системы. Планирование исполнения задач, поток, его преимущества. Прерывание. Виды прерываний. Режимы исполнения процессов. Пример: уровни привилегий в архитектуре X86. Виды управления памятью. Виртуальная память, ее преимущества. Отображения файла в

- память. Драйвер устройства. Концепции организации ядра, их особенности, примеры реализаций. Избирательное управление доступом. Источники определения прав доступа. Краткая характеристика и история развития операционных систем на примерах OS/360, CP/M, MS-DOS, UNIX, BSD, Linux, OS X, Windows. Проект GNU, стандарт POSIX. Операционные системы, совместимые с POSIX.
2. Интерфейс прикладных программ и процессы. Код возврата процесса, стандартные потоки ввода/вывода. Виды библиотек и способы подключения разделяемых библиотек. Проблемы использования разделяемых библиотек. Сценарии завершения процесса. Состояния процесса. Прикладные программные интерфейсы. Создание и завершения процесса в интерфейсах Windows API и POSIX. Сигнал POSIX. Дерево процессов.
 3. Виртуальная файловая система POSIX. Виртуальная файловая система. Канал. Идентификатор пользователя/группы POSIX. Основные особенности структуры POSIX-совместимой виртуальной файловой системы. Абсолютные и относительный путь. Краткая характеристика физических файловых систем. Специальная файловая система, примеры (pipefs, sockfs, shm, devfs, proc, sysfs). Жесткая и символьная ссылка, ограничения жестких ссылок. Типы файлов виртуальной файловой системы. Виды файлов устройств. Классы пользователей, права классов пользователей. Открытие файла, получение информации об открытом файле, чтение/запись информации из/в открытый файл, перемещение указателя внутри открытого файла, закрытие открытого файла в интерфейсе POSIX. Отображение файла в память в интерфейсе POSIX.
 4. Командная оболочка bash и утилиты командной строки POSIX. Командная оболочка. Режимы исполнения команд командной оболочкой. Оболочка bash. Синтаксис запуска команды в bash. Операции связывания нескольких команд. Перенаправления команд. Переменные. Назначение команд wc, ps, kill, man, echo, pwd, cd, ls, chmod, rm, rmdir, cp, mv, touch, cat, head, tail, find, grep. Формат регулярного выражения команды grep. Специальные символы, специальные конструкции и специальные переменные оболочки. Расширения оболочки: фигурных скобок, тильды, переменной или параметра, команды, арифметического выражения, процесса, имен файлов. Команды if, [, [[, case, for, while, until, break, continue, exit.
 5. Потоки. Основные области применения потоков. Основной цикла обработки сообщений в программе с графическим интерфейсом пользователя. Политика планирования, виды планирования. Критерии планирования. Голодная смерть. Простейшие алгоритмы планирования и их с точки зрения критериев планирования. Классы приоритетов и уровни приоритетов в различных ОС. Диспетчер ядра. Приоритеты

- потоков в Windows. Алгоритм «полностью честного планировщика». Создание и ожидание завершения потоков в интерфейсах Windows API и POSIX.
6. Синхронизация потоков. Функции задержки потоков в интерфейсах Windows API и POSIX. События в Windows API, области их применения и проблемы. Условие гонки, атомарность, критическая секция, взаимная блокировка, Взаимные блокировки и мьютексы в интерфейсах Windows API и POSIX. Блокировка с двойной проверкой. Семафоры в интерфейсах Windows API и POSIX. Барьер в интерфейсе POSIX. Задача о читателях и писателях. Блокировки чтения/записи в интерфейсах Windows API и POSIX. Задача производителей и потребителей средства ее решения.
 7. Алгоритмы блокировок. Сеть Петри. Функции инцидентности, порожденные сетью Петри. Правила срабатывания переходов в сети Петри. Функционирование сети Петри. Примеры сети Петри. Последовательность срабатываний. Живучесть переходов и сети. Мертвая и живая блокировка. Задача об обедающих философах. Требования к реализации взаимных исключений. Программные алгоритмы критической секции со строгим чередованием, попаданием в раздел двух процессов одновременно, мертвой и живой блокировкой. Алгоритмы Деккера и Петерсона. Алгоритмы с аппаратной реализацией атомарных операций «проверка и установка» (вариант с проверкой и «проверкой и установкой») и «обмен». Преимущества и недостатки алгоритмов с аппаратными атомарными операциями.
 8. Межпроцессное взаимодействие. Межпроцессный обмен дескрипторами в Windows. Совместное использование объектов синхронизации несколькими процессами в интерфейсах Windows API и POSIX. Использование разделяемой памяти в интерфейсах Windows API и POSIX. Использование неименованных каналов в интерфейсах Windows API и POSIX. Использование именованных каналов в POSIX.

Список вопросов.

1. Основные этапы развития вычислительной техники. Сверхбольшие интегральные схемы, наиболее известные исторические компьютерные архитектуры и поколения микропроцессоров.
2. Создание транзисторных электронных схем, реализующих булевы функции по их таблицам истинности.

3. Типичные вентили, из которых строятся логические схемы. NAND, NOR, XOR.
4. Время прохождения сигнала по эл. схеме, влияние на максимальную частоту работы микропроцессора.
5. Основные виды ячеек памяти и их особенности.
6. Способы организации памяти из отдельных ячеек.
7. Способы представления целых чисел в компьютере: беззнаковые, знаковые числа, представление в виде дополнения до степени двойки.
8. Простейшее устройство микропроцессора. АЛУ, регистры, управляющее устройство.
9. Типичные архитектуры микропроцессоров и их отличия. RISC, CISC, VLIW. Примеры.
10. Параллельные архитектуры. SIMD/MIMD, примеры.
11. Уровень микропрограммы в архитектуре вычислительной системы. Организация алгоритма работы микропроцессора.
12. Место набора инструкций в иерархическом устройстве вычислительной системы. Назначение ассемблера.
13. Краткая история развития архитектуры X86 на примере 8,16,32 битных ассемблеров.
14. Ассемблер X86. Основные инструкции, способы адресации, способы организации циклов, переходов, таблиц. самые распространенные программы-ассемблеры.
15. Суперскалярные и конвейерные вычисления в современных микропроцессорах, примеры доступных функций.
16. Организация работы с памятью в современном микропроцессоре. Уровни кешей. NUMA архитектуры.
17. История развития операционных систем. Самые распространенные ОС и этапы их развития.
18. Основные компоненты ОС. Основные функции ОС. Ядро, драйверы, прикладной уровень, прикладные программные интерфейсы.
19. Многозадачность в ОС. Виды многозадачности. Процессы и потоки, их различия и преимущества.
20. Планирование исполнения задач в ОС. Алгоритмы планирования, примеры.
21. Прерывания и их функции. Виды прерываний. Режимы исполнения процессов, уровни привилегий в архитектуре X86.
22. Виды управления памятью ОС. Виртуальная память, ее преимущества. Сегментная и страничная организация памяти.
23. Стандарт POSIX и его предназначение. POSIX-совместимые ОС. Основные составляющие стандарта и реализация на примере ОС Linux.
24. Права доступа в POSIX. Примеры оперирования правами доступа в ОС Linux.

25. Организация работы процесса. Потoki ввода/вывода. Состояния процесса, завершение работы процесса. Дерево процессов. Сигналы POSIX.
26. Виды библиотек и способы подключения разделяемых библиотек.
27. Виртуальная файловая система. Основные особенности POSIX-совместимой виртуальной системы. Типы файлов, виды файлов и устройств. Ссылки, специальные файлы.
28. Основные функции работы с файлами в POSIX. Открытие, закрытие, чтение и запись, позиционирование, отображение файлов в память.
29. Самые часто используемые команды оболочки bash. Примеры использования.
30. Расширения оболочки bash: {}, \$, `, арифметические выражения, переменные. Перенаправления команд.
31. Организация потока управления в оболочке bash. Команды if, [, [[, case, for, while, until, break, continue, exit.
32. Потoki в ОС. Основные области применения и особенности. Потoki ОС и пользовательские. Функции организации потоков в POSIX, примеры.
33. Планирование потоков в ОС. Политика и критерии планирования. Голодная смерть. Простейшие алгоритмы планирования. Приоритеты в Windows и Linux.
34. Особенности алгоритмов планирования в Windows и Linux. Очереди приоритетов, «полностью честный планировщик».
35. Особенности программирования параллельных потоков. Взаимоблокировки, ситуации гонок и их избежание. Задача об обедающих философах.
36. Синхронизация потоков. Основные примитивы взаимодействия в Windows и Linux (POSIX), примеры их использования.
37. Алгоритмы блокировок. Алгоритмы Деккера и Петерсона. Алгоритмы с аппаратной реализацией атомарных операций.
38. Межпроцессное взаимодействие в Linux. Разделяемая память, именованные и неименованные каналы.
39. Межпроцессное взаимодействие в Windows. Обмен дескрипторами, совместное использование объектов синхронизации. Разделяемая память. Неименованные каналы.