

Общий список вопросов к зачету/ экзамену по дисциплине “Операционные системы”

Будут сформированы “билеты” по 4 вопроса. Первый вопрос будет по практике с требованием построить какую-либо диаграмму по практическому заданию. Остальные три вопроса по теории из каждой аттестации (если у вас сдана та или иная аттестация, то вопрос по этой аттестации Вам задаваться не будет). Те, кто не смог построить диаграмму не смогут продолжить сдачу зачета.

На зачет обязательно приходить со сданной практикой. Сданная практика - допуск к зачету.

Возможность достать практику в новом году:

- Разуваев Денис, 8 группа (2 курс), 9 января, 13 30
- Ногих Александр, 2 группа (3 курс), 10 января 10:00
- Денисенко Андрей, 7 группа (2 курс), 10 января 10:00
- Коротков Владислав, 3 группа (3 курс), 9 января 9:45 и МКН (4 курс), 9 января 11:30
- Полещук Хасан, 4 и 1 группы (3 курс), 9 января 10:00
- Чекмарев Андрей, 5 группа (3 курс) 10 января в 13:25

Автоматы. Будут 2 типа автоматов:

- 1) Те, кто сразу получают автомат и приносят зачетку:)
- 2) Те, кому необходимо подтвердить свой автомат сдачей диаграмм по практическому заданию.

Вопросы:

1. В чем заключаются две основные функции операционной системы? Пояснить.
2. В чем разница между системами с разделением времени и многозадачными системами?
3. В чем разница между режимом ядра и пользовательским режимом? Объясните, как сочетание двух отдельных режимов помогает в проектировании операционных систем.
4. При создании операционных систем одновременно решаются задачи, например, использования ресурсов, своевременности, надежности и т. д. Приведите пример такого рода задач, требования которых могут противоречить друг другу.
5. Рассмотрим систему, имеющую два центральных процессора, у каждого из которых есть два потока (работающих в режиме гипертрейдинга). Предположим, есть три запущенные программы: P0, P1 и P2 со временем работы 5, 10 и 20 мс соответственно. Сколько времени займет полное выполнение этих программ? Следует принять во внимание, что все три программы загружают центральный процессор на 100 %, не осуществляют блокировку во время выполнения и не меняют центральный процессор, назначенный для их выполнения.

6. Почему в системах разделения времени необходима таблица процессов? Нужна ли она в операционных системах персональных компьютеров, работающих под управлением UNIX или Windows при единственном пользователе?
7. С точки зрения программиста, системный вызов похож на вызов любой другой библиотечной процедуры. Важно ли программисту знать, какая из библиотечных процедур в результате приводит к системным вызовам? Если да, то при каких обстоятельствах и почему?
8. Виртуальные машины приобрели высокую популярность по различным причинам. И тем не менее у них имеется ряд недостатков. Назовите их. Поясните разницу между эмуляцией и виртуализацией.
9. Инструкции, касающиеся доступа к устройствам ввода-вывода, обычно относятся к привилегированным инструкциям, то есть они могут выполняться в режиме ядра, но не в пользовательском режиме. Назовите причину привилегированности этих инструкций.
10. Предположим, вам нужно разработать новую компьютерную архитектуру, которая вместо использования прерываний осуществляет аппаратное переключение процессов. Какие сведения необходимы центральному процессору? Опишите возможное устройство аппаратного переключения процессов.
11. Когда в результате прерывания или системного вызова управление передается операционной системе, используется, как правило, область стека ядра, отделенная от стека прерываемого процесса. Почему?
12. У компьютерной системы достаточно места, чтобы хранить в основной памяти пять программ. Половину своего времени эти программы простаивают в ожидании ввода-вывода. Какая доля процессорного времени при этом тратится впустую?
13. Представьте себе мультипрограммную систему со степенью 6 (то есть имеющую в памяти одновременно шесть программ). Предположим, что каждый процесс проводит 40 % своего времени в ожидании ввода-вывода. Каким будет процент использования времени центрального процессора?
14. Может ли поток быть приостановлен таймерным прерыванием? Если да, то при каких обстоятельствах, а если нет, то почему?
15. В чем заключается самое большое преимущество от реализации потоков в пользовательском пространстве? А в чем заключается самый серьезный недостаток?
16. Представьте себе систему реального времени с двумя голосовыми вызовами с периодичностью, равной 5 мс для каждого из них, со временем центрального процессора, затрачиваемого на каждый вызов, равным 1 мс, и с одним видеопотоком с периодичностью, равной 33 мс, со временем центрального процессора, затрачиваемого на каждый вызов, равным 11 мс. Можно ли спланировать работу такой системы?
17. Гибкая система реального времени имеет четыре периодически возникающих события с периодами для каждого, составляющими 50, 100, 200 и 250 мс. Предположим, что эти четыре события требуют 35, 20, 10 мс и x процессорного времени соответственно. Укажите максимальное значение x , при котором система все еще поддается планированию.
18. Системе реального времени необходимо обработать два голосовых телефонных разговора, каждый из которых запускается каждые 6 мс и занимает 1 мс процессорного времени при каждом использовании процессора, и один видеопоток со скоростью 25 кадров в секунду, где каждый кадр требует 20 мс процессорного времени. Поддается ли эта система планированию?

19. Рассмотрите систему, в которой желательно разделить политику и механизм планирования потоков, реализованных на уровне ядра. Предложите средства для достижения этой цели.
20. Процессу, запущенному в системе CTSS, для завершения необходимо 30 квантов времени. Сколько раз он должен быть перекачан на диск, включая самый первый раз (перед тем, как он был запущен)?
21. В состоянии готовности к выполнению находятся пять заданий. Предполагаемое время их выполнения составляет 9, 6, 3, 5 и x . В какой последовательности их нужно запустить, чтобы свести к минимуму среднее время отклика? (Ответ будет зависеть от x .)
22. Как в операционной системе, способной отключать прерывания, можно реализовать семафоры?
23. Могут ли два потока, принадлежащие одному и тому же процессу, быть синхронизированы с помощью семафора, реализованного в ядре, если эти потоки реализованы на уровне ядра? Ответьте на тот же вопрос применительно к потокам, реализованным на уровне пользователя. Предполагается, что к семафору не имеют доступа никакие другие потоки любых других процессов. Обоснуйте свой ответ.
24. Объясните, как значение кванта времени и время переключения контекста влияют друг на друга в алгоритме циклического планирования.
25. Генерация сигналов
26. `sa_flags`
27. `signal` и `sigset`
28. Алгоритм раскрутки стека
29. Сигналы для управления заданиями
30. Типы сигналов
31. Основные этапы разработки ОС с пояснением каждого этапа
32. Стековый кадр
33. `Sxhab` + стандартный отладчик
34. Планирование процессов
35. Управление процессами
36. Процесс (физическое представление)
37. Структура управления процессами в ОС
38. Записи таблицы процессов
39. Модель состояния процесса
40. Этапы создания процесса
41. Создание процесса ОС UNIX
42. Создание процесса ОС Windows
43. Переключение контекста процесса
44. Параллелизм
45. Потоки. Использование объектов потоками
46. Отличие потоков от процессов
47. Потоки на пользовательском уровне, потоки управляемые ядром
48. Иерархия процессов
49. Память без использования абстракций.
Свопинг. Уплотнение памяти.
50. Адресное пространство
51. Этап компиляции загрузки и выполнения
52. Диспетчер памяти

53. Запись таблицы страниц
54. Преобразование виртуального адресного пространства в физическое
55. Как в операционной системе, способной отключать прерывания, можно реализовать семафоры?
56. Могут ли два потока, принадлежащие одному и тому же процессу, быть синхронизированы с помощью семафора, реализованного в ядре, если эти потоки реализованы на уровне ядра? Ответьте на тот же вопрос применительно к потокам, реализованным на уровне пользователя. Предполагается, что к семафору не имеют доступа никакие другие потоки любых других процессов. Обоснуйте свой ответ.
57. Объясните, как значение кванта времени и время переключения контекста влияют друг на друга в алгоритме циклического планирования.
58. Представьте себе мультипрограммную систему со степенью 6 (то есть имеющую в памяти одновременно шесть программ). Предположим, что каждый процесс проводит 40 % своего времени в ожидании ввода-вывода. Каким будет процент использования времени центрального процессора?
59. Может ли поток быть приостановлен таймерным прерыванием? Если да, то при каких обстоятельствах, а если нет, то почему?
60. В чем заключается самое большое преимущество от реализации потоков в пользовательском пространстве? А в чем заключается самый серьезный недостаток?
61. Что такое модульная ОС?
62. Рассказать про разные ассемблеры
63. Какой мем стоит на 22 слайде презентации "разработка ос"
64. Операционной системы: понятие, функции, примеры.
65. Эволюция операционных систем.
66. Операционной системы: понятие, классификации.
67. Схематично опишите решение 4 практической задачи
68. Напишите на ассемблере Hello world
69. Управление памятью с помощью побитовых матриц
70. Генерация сигналов
71. sa_flags
72. signal и sigset
73. Алгоритм раскрутки стека
74. Сигналы для управления заданиям
75. Типы сигналов
76. Стековый кадр
77. Опишите два свойства модели Биба
78. Опишите два свойства модели Белла - Лападулы
79. Средства защиты должны точно выполнять свои функции и быть изолированы от пользователя. Должны ли они выполняться в виде отдельного модуля, и если да, то какого? (изобразить схематично)
80. Опишите принцип минимальных полномочий.
81. Операции с каталогами.
82. Преимущества и недостатки непрерывного размещения файлов.
83. Преимущества и недостатки размещения файлов непрерывным списком.
84. Запись каталога файловой системы MS-DOS
85. Запись каталога файловой системы UNIX.
86. Структура файловой системы
87. signal и sigset

88. Loopback
89. Какие требования предъявляют к долговременному хранению информации?
90. Сколько существует подходов к определению прав доступа и какие?
91. Схематично изобразите файловую структуру
92. Схема структуры двоичных файлов
93. Диаграммы последовательностей, состояний и деятельности для всех практических задач
94. Диаграмма взаимодействия, развертывания для всех практических задач
95. PTHREAD (схемы обязательны)
96. 2 схематичное решение задачи обедающие философы
97. Fuse (схема обязательна)
98. API (схема взаимодействия). API сокетов
99. GAS и NASM, различия
100. Puts
101. Gcc
102. PTHREAD (схемы обязательны)