

Вопросы к зачету по курсу лекций «Операционные системы и оболочки»

Преподаватель – Одинцов Игорь Олегович (СПбГУ, мат-мех факультет, кафедра информатики)

[Введение в ОС и классификации ОС]

01. Роль ОС и польза от их изучения.
02. Определения ОС.
03. Основные функции ОС.
04. Компоненты ОС.
05. Ресурсы ОС.
06. Компьютерный континуум: роль ОС, эволюционная модель компьютеров
07. Классификации ОС по особенностям аппаратных платформ. Суперкомпьютеры, кластеры, гриды, облака.
08. Большие задачи
09. Приложения будущего
10. Классификации ОС по особенностям областей использования.
11. Классификация ОС по особенностям культуры решения задач.
12. Классификации ОС по типу централизации.
13. Типы параллелизма. Роль параллелизма в будущих приложениях
14. Модели параллельного и распределенного программирования и их аппаратная поддержка
15. Классификация ОС по типу архитектуры ядра системы. Сравнительный анализ ядер
16. Виртуализация в ОС
17. Классификации ОС по особенностям алгоритмов управления ресурсами
18. Классификации ОС по особенностям двоичного (бинарного) интерфейса приложений

[Оболочки ОС]

19. Оболочки операционных систем и их классификации.
20. История развития человеко-компьютерного взаимодействия
21. Интерфейс командной строки. Командные интерпретаторы, их функции и язык.
22. Графический интерфейс. История и эволюция графических оболочек
23. Пользовательский опыт (опыт взаимодействия) и его роль при разработке интерфейсов.
24. Принципы проектирования графических оболочек и проблема удобства использования (юзабилити).
25. Естественно-жестовый интерфейс, области применения и примеры приложений
26. Естественно языковой интерфейс, области применения и примеры приложений
27. Нейрокомпьютерный интерфейс, области применения и примеры приложений

[Процессы, коммуникация и синхронизация]

28. Аппаратная поддержка алгоритмов операционных систем
29. Процессы: определение, состояния, адресное пространство.
30. Простейшая программа на языке C, порождающая процесс в ОС Unix.
31. Поддержка многопоточности процессами. Преимущества многопоточности.
32. Сеть Петри как модель функционирования процессов.
33. Уровни и цели планирования процессов в централизованных ОС.
34. Приоритеты процессов.
35. Невытесняющие алгоритмы планирования.
36. Вытесняющие алгоритмы планирования.
37. Планирование процессов в многопроцессорных ОС.
38. Стратегии и проблемы распределения процессов по процессорам в распределенных ОС.
39. Коммуникация процессов. Классификация механизмов коммуникации.
40. Сигналы ОС Unix как простейшие средства коммуникации процессов.
41. Уровневые протоколы, семиуровневая модель ВОС.
42. Стек TCP/IP. Основные протоколы стека.
43. Обмен данными между коммуникационными узлами.
44. Адресация в сетях TCP/IP.
45. Роль портов. Использование портов стандартными сервисами.
46. Транспортные протоколы.
47. Маршрутизация в сетях TCP/IP.
48. Интернет вещей.
49. Проблемы современных сетей.
50. Удаленный вызов процедур
51. Вызов заданий в грид.
52. Грид: определения, классификации, основные задачи.
53. Области применения грид. Типичные задачи для грид.
54. Основы синхронизации процессов в централизованных ОС. Взаимные исключения и блокировки.
55. Программное решение задачи взаимных исключений. Алгоритм Петерсона.
56. Команда test&set как аппаратная поддержка синхронизации.
57. Блокировка с запретом прерываний.
58. Крутящаяся блокировка как средство синхронизации.
59. Семафоры как средство синхронизации.
60. Мониторы как средство синхронизации.
61. Обзор языковых подходов к программированию для параллельных вычислительных систем.

62. Подход с использованием прагм. Программная модель OpenMP.
63. Подход с использованием коммуникационных библиотек. Программная модель MPI.
64. Решение задачи передачи данных между процессами "читатель" и "писатель" с помощью крутящихся блокировок
65. Решение задачи передачи данных между процессами "читатель" и "писатель" с помощью семафоров.
66. Решение задачи передачи данных между процессами "читатель" и "писатель" с помощью мониторов.
67. Основы синхронизации процессов в сетевых и распределенных ОС. Алгоритм Лампорта.
68. Роль координатора. Алгоритмы выбора координатора.
69. Применение централизованного алгоритма для синхронизации в распределенных ОС.
70. Применение алгоритма с круговым маркером для синхронизации в распределенных ОС.
71. Атомарные транзакции. Двухфазный протокол утверждения.
72. Синхронизации в группах.
73. Описание основных проблем при работе процессов с ресурсами: состязание, голодание, зависание, тупик.
74. Необходимые условия возникновения тупиков. Предотвращение тупиков.
75. Обход тупиков.
76. Обнаружение тупиков.
77. Тупики в распределенных системах.

[Память и виртуальная память]

78. Иерархия классов памяти. Принципы, лежащие в основе иерархии.
79. Основная память. Привязка адресов.
80. Редактор связей.
81. Загрузчик. Оверлеи. Свопинг.
82. Концепция управления виртуальной памятью в централизованных ОС.
83. Страничная организация памяти. Аппаратная поддержка.
84. Стратегии размещения.
85. Стратегии вталкивания.
86. Стратегии вытеснения: принцип оптимальности, случайный выбор, не используемая в последнее время, реже всего используемая.
87. Стратегии вытеснения: первая пришедшая, второй шанс, часы.
88. Стратегии вытеснения: концепция рабочего множества, часы с рабочим множеством.
89. Управление виртуальной памятью в распределенных ОС.

[Файлы и файловые системы, драйверы устройств]

90. Файл. Атрибуты файла. Организация файлов и методы доступа. Каталоги.
91. СУБД. Реляционная модель данных.
92. Файловые системы. Классификация. Примеры концепций.
93. Большие данные.
94. Архитектура распределенных файловых систем.
95. Драйвер. Классы драйверов устройств и алгоритм работы типичного драйвера.
96. Анализ возможных алгоритмов работы с диском.

[Разработка ОС]

97. Отправные точки при разработке ОС.
98. Цели проектирования ОС.
99. Реализация и производительность ОС.

Коллекция вопросов весна-лето 2015