**Раздел "Введение"**

1. Операционная система (определение + 2 точки зрения на ОС)
2. Жесткие диски (преимущества перед перфокартами и перфолентами, spooling)
3. Операционная среда (определение + условия выполнения программ)
4. Многозадачность (определение + преимущество). ОС разделения времени (описание).
5. Классификация ОС по назначению.
6. Классификация ОС по режиму обработки задач.
7. Классификация ОС по организации пользовательской работы.
8. Расширяемость, Переносимость, Совместимость, Безопасность.
9. Монолитная архитектура ОС.
10. Микроядерная архитектура ОС.
11. Многоуровневая (слоеная) архитектура ОС.
12. Архитектура ОС, основанная на использовании виртуальных машин. Паравиртуализация.
13. Недетализированные примеры архитектур – UNIX.
14. Недетализированные примеры архитектур – Windows NT.

**Раздел "Процессы и потоки"**

[15] Поток (определение).  
[16] Процесс (определение).  
[17] Многопоточные процессы.  
[18] Контекст процесса. Контекст потока. Переключение контекста потока.  
[19] Диаграмма состояний потока в многозадачной среде.  
[20] Диаграмма состояний потока в однозадачной среде.  
[21] Дескриптор процесса.  
[22] Дескриптор потока.  
[23] Создание процесса, завершение процесса (Windows-style).  
[24] Создание потока, завершение потока (Windows-style).  
[25] Системный вызов fork(). Системный вызов exec(). Системный вызов exit(). Системный вызов wait().

**Раздел "Недетализированные модели аппаратного уровня"**

1. Линейная модель адресации.
2. Сегментная модель адресации.
3. Страничная модель адресации.
4. Сегментно-страничная модель адресации.
5. Численные характеристики моделей.  
   Мой комментарий:  
   Нет отдельного пункта к лекциях. Просто где-то считали
6. Уровни привилегий ЦП.
7. Шлюз вызова.
8. Шлюз задачи.
9. Классификация прерываний.
10. Обработка прерываний контроллером прерываний.
11. Обработка прерывания центральным процессором.

**Раздел "Планирование"**

1. Критерии оценки алгоритмов планирования.
2. Невытесняющее и вытесняющее планирование.
3. Алгоритмы FCFS, SJF, SRT, RR.
4. Приоритетное планирование.
5. Планирование с использованием многоуровневых очередей и обратной связи.
6. Алгоритм планирования Windows.
7. Оригинальный алгоритм планирования UNIX.
8. Алгоритм планирования Linux.

**Раздел "Синхронизация"**

1. Необходимость синхронизации.
2. Постановка задачи взаимного исключения.
3. Проверка корректности работы алгоритмов синхронизации.  
   Мой комментарий:  
   По всей видимости нет отдельного пункта, нужно самому сформулировать
4. Алгоритм Петерсона.
5. Алгоритм булочной.
6. Использование аппаратной поддержки (test&set).
7. Использование аппаратной поддержки (swap).
8. Активное ожидание.
9. Семафор, типы семафоров.
10. Мьютекс.
11. Монитор, условные переменные. Мониторы Хоара и Меса.
12. Решение задачи "Производитель-потребитель" с использованием семафоров.
13. Решение задачи "Производитель-потребитель" с использованием мониторов.
14. Решение задачи "Читатели-писатели" с использованием семафоров.
15. Взаимоблокировка (определение).
16. Необходимые условия возникновения тупика.
17. Предотвращение тупиков.
18. Избегание тупиков. Алгоритм банкира.
19. Граф процесс-ресурс. Редукция графа.
20. Устранение тупиков.

**Раздел "Управление памятью"**

1. Схема с разделами фиксированного размера.
2. Схема с разделами переменного размера.
3. ВАП на основе страничного преобразования. Обработка страничного сбоя (блок-схема).
4. Алгоритмы замещения страниц Biledy, Random, FIFO, LRU, NFU, Second Chance, Clock, Linux, основанный на оценке размеров рабочих множеств.

**Раздел "Подсистема управления файлами"**

1. Архитектура подсистемы управления файлами.
2. Структура жесткого диска, адресация.
3. Типы объектов ФС.
4. Концепция i-node.
5. Структура UNIX FS. Атрибуты суперблока. Атрибуты i-node.
6. Хранение информации о размещении данных файла в UNIX FS.
7. Структура каталога в UNIX FS.
8. Структуры, используемые ядром для обеспечения доступа процессов к файлам. Атрибуты записей. Операции open() и close().
9. Fast File System.
10. Файловая система с протоколированием.

Задачи.

- Задача по теме "Планирование".

- Задача по теме "Синхронизация":

- корректность кода (псевдокод);

- возможные результаты выполнения кода;

- задача на избегание тупика;

- для экзамена - задача на написание псевдокода.

- Задача по теме "Управление памятью".

- Задача по теме "Файловые системы".

- Задача на понимание кода на языке C (по темам лабораторных работ, для экзамена).