***«Вначале было Слово, и Слово было у Бога, и Слово было Оси».***

**Организационная информация:**

|  |
| --- |
| 1. Теория из первой половинки.- коэффициент 0.3; 2. Теория из второй половинки.- коэффициент 0.3; 3. Задача.- коэффициент 0.4;   За каждый ответ максимум- 100 баллов, умножаем по коэффициентам и складываем.  Будет терпеть нас сидячими(=готовящимися) полтора часа или меньше(30 минут на блок, сдал задачу- терпит час, сдал коллоквиум и задачу- 30 минут). |

**Дополнительная информация:**

|  |
| --- |
| ***Ядро не знает, что внутри процесса несколько потоков.*** |
| ***Каталоги страниц содержат информацию о таблицах страниц, у каждого процесса свой каталог страниц и сколько-то таблиц страниц.*** |
| ***Системные вызовы в микроядрах.***  *Большая часть системных штук вынесена в usermode.*  *+:надежнее(reincarnation service);*  *-:накладные расходы- много смен контекста(в три раза больше, чем в монолитном).* |
| ***Почему не использовать одну большую таблицу страниц(тема: Виртуальная память)?*** *Потому что неудобно хранить, лол. Каталог страниц и так на всё отсылает.* |
| ***Цифровая подпись. Как устанавливается?***  *У нас есть симметричные и асимметричные алгоритмы. Подпись относится ко вторым.*  *Передаём программу из А в Б, то пункт Б должен проверить, модифицирован ли файл(перехватывал ли его злоумышленник и изменял ли). Отправляем приватный ключ, шифруем публичным файл и тоже отправляем. Пункту Б прислали ключ и зашифрованные данные. Получилось расшифровать? Получилась не бессмыслица? Значит, данные были зашифрованы именно публичным ключом(который был только у пункта А)=>данные пришли от доверенного источника.*  *Если Е заменит ключ? У Винды есть предустановленные ключи. И ключ, отправляемый пункту Б, является рекурсивным наследником этих ключей. Если это так, значит, у ключа был сертификат и мы можем верить информации из этого сертификата(там указано, кому он выдан) и мы можем проверить, кому он был выдан.* |
| ***В чём угроза целостности в Белла-Лападулы?*** *Может писать отчёты генералу=> можем перезаписать- нарушение целостности файлов генерала.*  ***Что подразумевается под целостностью в модели Бибы?*** *Тут так сделать нельзя.* |
| ***Проблема инверсии приоритетов.*** *Два процесса(H- высокий приоритет и l- низкий приоритет), процессу H нужны данные из процесса l, он их ждёт и ничего не делает. Мы его останавливаем и вызываем планировщик, который снова передаёт управление H, и мы будем делать так, пока приоритет не понизится.*  *Решается sleep-ом и wakeup-ом.* |

**Информация по билетам(что знать?):**

|  |
| --- |
| 1. **Модель фон Неймана и прерывания.**   Что такое архитектура фон Неймана? +нарисовать. Принципы, +примеры нарушения.  Принцип программного управления- нарушение: прерывания.  Принцип двоичного кодирования- нарушение: троичная.  Принцип адресности(обращается процессор, не программа)- нарушение: защищённый режим.  Принцип однородности памяти- нарушение: Гарвардская архитектура.  Рисуем Гарвардскую архитектуру, узкое место архитектуры фон Неймана.  Рассказываем про прерывания. Контроллер прерываний. Классификация(почему, зачем, что это)+ примеры(1 на класс). Как происходит обработка прерывания? Как АО соединяется с программой в компьютере. |
| 1. **Назначение, состав и функции ОС.**   Определения ОС(программа для управления программами). Рассматриваем ОС как расширенную машину и как менеджер ресурсов(и какие ресурсы менеджерит). Задачи, которые решает ОС(программный интерфейс API, пользовательский интерфейс, создание окружения для программ(загрузка, выделение памяти и т.д.), управление устройствами(драйвера)). Обеспечение безопасности- функции(как обеспечивает, защищённый режим, права доступа и т.д.) и аппаратная поддержка защиты(кольца защиты). Рассказать про многозадачность(вытесняющую). Где работают программы?- уметь объяснить. И как пользовательские программы получают управление над аппаратной частью.  Состав: ядро, службы(демоны), менеджер памяти, загрузчик, файловая система, планировщик |
| 1. **Классификация и примеры современных ОС.**   Классификация по типам ЭВМ(Зоопарк О.С.- Таненбаум(то, что у нас в билете на коллок));  Классификация по принципу и времени выполнения заданий(пакетные, интерактивные, реального времени(бывают разные, с жестким ограничением времени и мягким, критерий существования системы реального времени));  Классификация по типу ядра(4 билет)- что такое и какие +/-, рассказать про различные типы гипервизоров. Примеры на каждое. |
| 1. **Понятие ядра ОС, архитектура ядра.**   Различные типы ядер. Как работает защита в каждом из типов и как проходят системные вызовы? Примеры. |
| 1. **Понятие ядра ОС, системные вызовы.**   Что такое системные вызовы и зачем они нужны? Связь системных вызовов и прерываний(процессор работает в 3 кольце, а ему нужно переключиться в ядро). Тип прерываний- системные вызовы(программное прерывание). Кольца защиты обеспечиваются аппаратной поддержкой, процессор должен в них уметь, должно быть реализовано на железе(набор битиков). Какой бы не был системный вызов, библиотека зовёт команду: syscall(Intel), sysenter(AMD)(программное прерывание), управление передаётся обработчику прерывания и в зависимости от аргументов, переданных dll, определяет, что именно было нужно. Различие системных вызовов в разных архитектурах, рассказ про POSIX и Win32API. Разобрать пример. CreateFile(Win), exec/fork(Unix). Расс казать, что происходит под капотом библиотек(происходит системный вызов). |
| 1. **Управление виртуальной памятью.**   Что такое адресное пространство и зачем это придумали? Сегментная модель(как происходит трансляция адресов), страничная модель(зачем придумали? Как происходит в ней трансляция адресов). Сравнение этих моделей. Swapping(подкачка) и её стратегии. Трансляция адреса в физический при гибридной модели(гибридная трансляция страниц). |
| 1. **Стратегии подкачек и вытеснения страниц памяти.**   Рассказываем про стратегии. Как работает, +,-. Какая требуется аппаратная поддержка(наличие счётчика, поддержка бита R, M, W). Кто занимается трансляцией адресов и подсчётом битов(MMU). Для скорости будут спрашиваться несколько алгоритмов. |
| 1. **Программы, процессы и потоки.** Длинно, шо пздц.   Что такое программа? -Последовательность команд или исполняемый файл, лежащий на диске(2 определения). Процесс компиляции(что, зачем, как). Определение процесса(контейнер, содержащий ресурсы программы), что у процессов разное(модель безопасности процесса(почему нельзя, а если можно, то как и в каких случаях)). Операция смены контекста. Как происходит работа загрузчика? Примеры создания процесса(fork(создание в два этапа, сначала делаем точную копию, потом она начинается различаться одной переменной(возвращаемым значением функции(определение того, в ребёнке мы или в родителе)))), exec). Принцип CopyOnWrite. Сказать про дескрипторы, ProcessID и другие ресурсы, выделяемые на процесс. Службы и демоны. Иерархия процессов в UNIX(зомби). Состояние процессов(выполняется, блокирован, готов). Определение потока(зачем нужны). Примеры приложений, где нам никак не обойтись без потоков(приложения с графическим интерфейсом или менеджеры(ему нужно следить, печатает ли пользователь и проверять входящие данные)). Что общего у потоков(адресное пространство, дескрипторы), что разное(содержимое регистров(выполняют разные инструкции, следовательно, разные регистры)). Три реализации потоков(на уровне пользователя, ядра, гибрида). Проблемы потоков(гонки, взаимоблокировки и т.д.). Рассказываем про exit, join, yield, select. Системные вызовы могут быть блокирующими и неблокирующими. Вопрос со звёздочкой- разобраться со всплывающими потоками. |
| 1. **Кооперативная и вытесняющая многозадачность.**   Определения концепций, плюсы и минусы каждой(когда выгодно применять одну, когда другую). Определение приоритетных и неприоритетных алгоритмов. Критерии алгоритмов планирования(для пакетных, интерактивных и систем реального времени(перечисляем)). Задачи(общие, для пакетных, интерактивных и систем реального времени). За счёт чего возможного реализация вытесняющей многозадачности(таймер). Цепочка прерывания по таймеру(как и кем обрабатывается). Запускает загрузчик, планировщик выбирает следующий. |
| 1. **Планирование задач с приоритетами.**   Начинаем с пакетных систем, потом интерактивное(циклическое, приоритетное(много очередей), самый короткий процесс, гарантированное планирование, лотерейное и справедливое), далее системы реального времени. Как планировщик получает управление, за счёт чего планирует? |
| 1. **Гармонически взаимодействующие процессы.**   Начинаем с определений. Что такое межпроцессорное взаимодействие? Зачем нужно? Почему процессы не могут взять и поменять память в других процессах? А если могут, то когда? Почему называются гармоническими(«большую часть времени процессы работают друг от друга и не подозреваются о существовании друг друга»- Дейкстра)? В точках пересечения- моменты синхронизаций. Подробно рассказываем про проблемы, которые могут в этих точках происходить(гонки, блокировки, взаимоблокировки). Как их концептуально решают? Примитивы синхронизации(просто перечисляем определения и чем друг от друга отличаются(мьютекс, семафоры и т.д.)). Определения- состязательная ситуация и критическая секция. На каждую из проблем по одной задачке. Набор примеров-задач: система бронирования авиабилетов, производитель и потребитель, философы. |
| 1. **Механизмы межзадачного взаимодействия.**   Что такое IPC(межпроцессорное взаимодействие)? Какие проблемы? Определение состязательной ситуации и критической секции. Как решать?- Активное ожидание(в чем заключается, +, -), запрещение всех прерываний(в чем заключается, +, -), аппаратная поддержка специальных команд(XCHG, TCL- спин-блокировка), алгоритм Петерсона(решаем ту же задачу без задействования АО, но медленнее), приостановка и активизация(*инверсия приоритетов!!!*). |
| 1. **Основные примитивы синхронизации потоков.**   А зачем их синхронизировать(проблемы те же, что и с потоками, не будем решать- всё заблокируется и не будет решаться)? Семафоры, мьютексы, мониторы, передача сообщений, барьеры. Понимать, где всё это находится(пользовательские процессы- мьютекс из библиотеки, можем сами написать; ядерные потоки- ядерные примитивы синхронизации(API)). Где хранится мьютекс? |
| 1. **Проблемы тупиков, deadlock и способы борьбы.**   Что такое deadlock-и? Пример задачки с deadlock-ами(философы). Условие существования взаимоблокировок(4). Атаки на каждое из условий. И разбираем это на примере философов(приводим классические примеры решения проблемы(объясняем суть, связь с атаками)).На каждую из проблем по одной задачке. Набор примеров-задач: система бронирования авиабилетов, производитель и потребитель, философы. |
| 1. **Виды атак на ОС.**   Вредоносное ПО, классификация, алгоритмы работы. Эксплуатация уязвимостей в ПО. Атаки на веб-сервисы и сетевые каналы связи. Нарушение принципов фон Неймана при защите от них(Пример: стэк неисполняемый(но память же должна быть однородной), код только в куче). Фишинг и другие методы социальной инженерии. Человек по середине(перехват по сети(Man in the middle)). Алгоритм Диффи- Хелмана(лучше формально, а не на сундучках). Технологи ОС для защиты(защищённый режим, разделение памяти процессов, аутентификация, права доступов к файлам, проверка паролей). Антивирусное ПО(структура и используемые технологии). Межсетевые экраны(firewall). Основы криптографии(симметричные, асимметричные; примеры, плюсы и минусы). Вопрос со звёздочкой- цифровые подписи. Атаки на пароли(при их вводе, хранении), как их правильно хранить. |
| 1. **Многоуровневые системы безопасности и иерархия классов безопасных систем.**   Рассказ по оранжевой книге класс(классы C, B, A+ примеры, чем характеризуется(каким требованиям должна удовлетворять)). Рассказ про модель Белла-Лападулы. Модель Биба. |
| 1. **Идентификация пользователей и права доступа.**   Кто такие пользователи? Что такое группы пользователей и зачем это нужно? UID и GID- рассказать. Матрицы управления доступом. AccessControlList. Роль пользователя(пара пользователь- группа; может иметь несколько групп, но залогинен он в один момент времени в одной группе). Использование UID и GID в парах. Про маски(whildeCard, звёздочка на одной из или обеих позициях). Вводим три понятия: аутентификация, идентификация и авторизация(по одному примеру). |
| 1. **Классификация внешних устройств и их драйверы.**   Общий рассказ про устройства ввода-вывода. Какие бывают(классификация(блочные, символьные и другие+ примеры))? Вспомнить про понятие шины. Контроллеры устройств. Пространство портов ввода-вывода(отдельно, внутри ОП, и там и там(различия, плюсы и минусы)). DMA-DirectMemoryAccess. Уровни аппаратного обеспечения ввода-вывода. +табл. Буферизация. Зачем нужна и какие типы бывают(обычная(одинарная), двойная и кольцевая)? |
| 1. **Файлы, каталоги и файловые системы.**   Что такое файл? Что такое каталог(папка, директория) и зачем он нужен? Что такое файловая система? Различия между файловыми системами(что такое путь? Пути бывают относительными и абсолютными). Рассказываем, как с путями к файлам работают различные файловые системы(влияние регистра, максимальная длина пути к файлу). Понятие расширения файла. Кодировки(тоже различие). Максимальный размер, максимальная длина пути, максимальное длина названия- сравнительный анализ файловых систем. Последовательный и произвольный доступ к файлам. SEEK. Атрибуты файлов. Стандартный набор операций с файлами. Стандартный набор операций с директориями. А что быстрее открыть файл или закрыть? Закрыть. Когда открываем файл нужно проверять права, обращаться к диску и т.д. Жесткие ссылки(*счётчик показывает количество копий файла*, хранится в таблице файловой системы) и символические ссылки. |
| 1. **Основные структуры файловых систем.**   Что такое файловая система? MBR, VBR и рассказать их роль в загрузке компьютера. Разбираем файловую систему. Ограничения файловых система(сравнение размеров названия файла, пути к нему, максимальный размер). Операция форматирования. Как могут размещаться блоки файлов? Вопрос со звёздочкой- что тако i-узлы(i-node)? Разные типы устройств хранения(HDD vs SSD). Какие особенности? Почему может различаться размер блока? *Самый часто используемый файл- файл подкачки.* В HDD нельзя идеально вымерить 4кб, поэтому они больше, чем у SSD, которому пофиг. Как возникает фрагментация и что это такое? Дефрагментация? На каких устройствах следует применять, а на каких нет и почему? Что такое журналирование, зачем оно нужно, каким свойством должны обладать все записи журнала? Как осуществляется работа с журналом? |