***Ссылки:*1. Стандарт POSIX** - <https://intuit.ru/studies/courses/47/47/lecture/1397?page=4>   
**2. Средства безопасности OC** – Танненбаум страница 527  
**3. Хранение паролей в ОС** - <https://habr.com/ru/post/114150/> и <https://intsystem.org/security/users-passwords-store/>  
**4. Загрузка ОС** Танненбаум страницы 58, 853 (Linux), 986 (Windows)  
**5. Средства восстановления работоспособности системы** - <https://ru.wikipedia.org/wiki/Восстановление_системы> (windows), для linux – куча дистрибутивов с программами для восстановления - <https://losst.ru/chem-vosstanavlivat-sistemu-linux>   
**6. Средства ОС для управления энергопитанием** – Танненбаум страница 467 **7. Возможности командных оболочек Windows / UNIX.** <https://intuit.ru/studies/courses/1059/225/lecture/27285?page=1> **-** (windows) <https://intuit.ru/studies/courses/23/23/lecture/27151> <http://ftp.altlinux.org/pub/docs/Heap/modules/unix_base_admin.dralex/ch02.html> - (unix)  **8. Гипервизоры (Хen,QEMU, Microsoft Hyper-V... )** <https://habr.com/ru/company/vps_house/blog/349788/> <https://www.sim-networks.com/blog/hypervisors-vmware-kvm-xen-openvz>  **9. Средства виртуализации приложений (Windows services for Lnux** - <https://habr.com/ru/post/176823/> **, Wine** - <https://hackware.ru/?p=13977> **, Docker** - <https://habr.com/ru/post/253877/> **, ...)**  **10. Средства API ОС для работы с процессами/потоками, в т.ч. средства синхронизации** <https://intuit.ru/studies/courses/1592/53/lecture/1571> (Средства синхронизации потоков управления)+ Танненбаум **11. Обзор операционных систем, созданных в России** <https://zen.yandex.ru/media/softbit/rossiiskie-operacionnye-sistemy-5c973de5c5686200b3ca12bd>  **12. Модель потоков m:n от Google** <https://www.opennet.ru/opennews/art.shtml?num=53443>

**13. От MBR до GPT** [**https://habr.com/ru/post/347002/**](https://habr.com/ru/post/347002/)

**14. Эволюция файловых систем Microsoft** <https://support.microsoft.com/ru-ru/help/100108/overview-of-fat-hpfs-and-ntfs-file-systems> <https://intuit.ru/studies/courses/991/216/lecture/5570?page=3> (все сразу) <https://ru.wikipedia.org/wiki/FAT> (fat) <https://ru.bmstu.wiki/HPFS_(High_Performance_File_System)> (hpfs) <https://intuit.ru/studies/professional_skill_improvements/10808/courses/1078/lecture/16586?page=1> (ntfs)

**Что такое условная переменная и какие возможности ей предоставляются (в терминах программного интерфейса)?**

Это примитив синхронизации, обеспечивающий блокирование одного или нескольких потоков до момента поступления сигнала от другого потока о выполнении некоторого условия или до истечения максимального промежутка времени ожидания. Когда процессу требуется войти в свою критическую область, сначала он проверяет значение условной переменной. Если оно равно 0, процесс устанавливает его в 1 и входит в критическую область. Если значение уже равно 1, процесс просто ждет, пока оно не станет равно нулю.

**Для чего предназначен алгоритм банкира и почему его использование на практике ограничено?**

Цель алгоритма - избежать взаимоблокировки, распределяя ресурсы по определенным правилам.  
Стоит сказать о сопутствующем термине: безопасное состояние - состояние, для которого имеется по крайней мере одна последовательность событий, которая не приведет к взаимоблокировке.  
Алгоритм - пусть у системы в наличии n ресурсов и она принимает запрос от процесса пользователя только тогда, когда его максимальная потребность в ресурсах не превышает n. Пользователь гарантирует, что если ОС в состоянии удовлетворить его запрос, то все ресурсы будут возвращены системе в течение конечного времени. Выделение ресурсов возможно только тогда, когда система остаётся в таком состоянии, что она способна обеспечить всем процессам их выполнение в течение конечного времени.

Преимущество алгоритма в том, что при его использовании нет необходимости в перераспределении ресурсов и откате процессов назад.

Недостатки (из-за которых данный алгоритм не используется):  
1) Необходимость постоянства числа пользователей и числа ресурсов (в большинстве систем количество пользователей и ресурсов изменяется динамически).  
2) Число работающих пользователей должно оставаться постоянным.  
3) Клиенты обязаны возвращать ресурсы.

4) Необходимость знать заранее максимальные потребности процессов в ресурсах (эта информация недоступна в большинстве систем).

**Каковы преимущества организации решения задачи на компьютере в варианте приложения с несколькими потоками по сравнению с вариантом с несколькими процессами?**   
1) В приложениях одновременно может происходить несколько действий, часть которых может периодически быть заблокированной. Модель программирования упрощается за счет разделения такого приложения на несколько последовательных потоков, выполняемых в квазипараллельном режиме.  
2) Скорость создания и ликвидации потоков выше (в некоторых системах в 10-100 раз) по сравнению с процессами.

3) Потоки одного процесса имеют возможность совместно использовать ресурсы, доступные процессу.

4) Когда выполняются объёмные вычисления, а также часть времени тратится на ожидание ввода-вывода использование потоков позволяет этим действиям перекрываться по времени, что влечёт ускорение работы программы.

5) Потоки полезны для систем, имеющих несколько центральных процессоров (несколько ядер процессора), где есть реальная возможность параллельных вычислений.

**Предложите не менее двух недостатков алгоритма планирования "первой – самая короткая задача".**  
1) Длительные процессы могут не получить процессорное время.

2) Не всегда возможно точно определить время выполнения конкретного процесса.

3) В пакетных системах количество процессорного времени, необходимое заданию для выполнения, указывает пользователь при формировании задания. Если пользователь укажет больше времени, то он будет ждать результата дольше, если меньшее – то задача может не досчитаться до конца.   
**Какие две структуры данных наиболее широко используются для учета свободного пространства оперативной памяти?**  
1) Связные списки  
2) Битовые массивы

**Что такое мьютекс и какие возможности им предоставляются (в терминах программного интерфейса)?**

Мьютекс - аналог одноместного семафора, НО допускает только один процесс в контролируемом участке. Остальные процессы, пытающиеся получить доступ к этому разделу, засыпают в ожидании выхода первого процесса из этого раздела (т.е. пока мьютекс не будет освобождён).

Мьютекс может быть в одном их двух состояний: открыт и закрыт.  
Основная задача мьютекса - запрет доступа к ресурсу для процессов, отличных от того, который первым завладел данным ресурсом. Мьютекс должен защищать систему от возникновения состояния гонки.

**Предложите не менее пяти вспомогательных свойств страниц при реализации страничной модели управления памятью**

1) Бит присутствия-отсутствия

2) Биты защиты

3) Биты модификации

4) Бит защиты

5) Бит блокирования кеша

**В каких условиях алгоритм планирования по принципу FIFO будет наиболее эффективен с точки зрения среднего времени отклика и почему?**   
В условиях, когда время выполнения процессов примерно одинаково.  
Иначе длительные процессы, выполняемые перед коротковременными, увеличивают время отклика у кратковременных процессов.

**Привести примеры множеств разделяемых ресурсов для процессов и потоков. В чем отличия между ними?**

Процессы разделяют процессорное время, файлы, оперативную память. У процессов адресные пространства изолированы.

Потоки, принадлежащие одному процессу, имеют общее адресное пространство.

**какого механизма реализуется произвольная графовая структура файловой системы?**

sim-limk (символическая ссылка) - специальный файл в файловой системе, в котором вместо пользовательских данных содержится путь к файлу, открываемому при обращении к данной ссылке.

hard-link (жесткая ссылка) - структурная составляющая файла, описывающий его элемент каталога. Жесткая ссылка связывает i-узел с каталогом и даёт ему имя. Если у файла несколько таких ссылок, то он фигурирует в различных каталогах.

**Предложите не менее шести функций файловых систем**1) Хранение информации  
2) Структурирование файлов  
3) Защита данных от несанкционированного доступа

4) Обеспечение прав доступа к данным  
5) Обеспечение надежности и отказоустойчивости  
6) Распределение внешней памяти между файлами  
7) Создание однородной среды для разных носителей информации

8) Обеспечение совместного доступа к файлам.

**Пояснить термин "взаимоблокировка"**

Это ситуация, при которой несколько процессов находятся в состоянии ожидания ресурсов, занятых друг другом, и ни один из них не может продолжать свое выполнение.

**Что такое кооперативная многозадачность?**

Вид многозадачности, при котором следующая задача выполняется только после того, как текущая задача явно объявит себя готовой отдать процессорное время другим задачам.

**Какие ограничения формата описания разделов MBR устраняет формат GPT?**  
1) MBR допускает только четыре основных, а остальные — дополнительные. GPT допускает неограниченное количество основных разделов.  
2) MBR имеет ограничение размера раздела в 2 ТБ. GPT позволяет создавать разделы любого размера.  
3) MBR хранит только одну копию данных раздела в первом секторе жёсткого диска, что может привести к потере всей информации в случае повреждении информации о разделах. GPT хранит копию данных раздела, позволяя восстановить их в случае повреждения основного заголовка.  
4) MBR не имеет способа узнать о повреждении данных. GPT хранит значения контрольной суммы для проверки, что данные не повреждены, и может выполнить необходимое восстановление из других областей диска в случае повреждения.

**не менее пяти задач операционной системы по управлению оперативной памятью.**   
1) Отображение адресного пространства процесса на конкретные области физической памяти.  
2) Контроль доступа к адресным пространствам процессов.  
3) Отслеживание свободной и занятой памяти.  
4) Распределение памяти между конкурирующими процессами.  
5) Выгрузка процессов во внешнюю память при недостатке места в оперативной памяти.

**двух базовых моделей устраняет сегментно-страничная модель управления оперативной памятью?**

В одномерной памяти может возникнуть ситуация, когда адресного пространства может не хватить для хранения всей необходимой приложению информации (например, при наличии разрастающихся таблиц, одна таблица может «упереться» в другую). Сегментно-страничная организация памяти избавляет программиста от необходимости контролировать разрастающееся адресное пространство.

Адресное пространство каждого сегмента может разрастаться со временем. При большом размере сегментов появляются проблемы с их хранением в оперативной памяти. Эти проблемы решаются сегментно-страничной модели управления памяти.

(Например, при наличии нескольких таблиц выстраиваемых в процессе компиляции и разрастающихся со временем в одномерной памяти (к какой относится память со страничной организацией) должны быть выделены последовательные участки виртуального адресного пространства. Если программа содержит намного большее, чем обычно, количество переменных, но вполне обычное количество всех остальных компонентов, то участок адресного пространства, выделенный под таблицу имен, может заполниться  
до отказа, но для других таблиц может остаться большое количество свободного пространства.  
Этот недостаток исправляется введение сегментно-страничной организации памяти - для каждой таблицы можно выделить свой сегмент.)

**Какие недостатки присущи схеме изоляции памяти процесса, использующей два параметра – базу и границу?**   
1) Сегменты одной структуры могут занимать в памяти несмежные участки, а незанятая память в диапазоне между такими сегментами этой структуры не может быть использована другими процессами.  
2) Необходимость применения операций сложения и сравнения к каждой ссылке на ячейку памяти.

3) Необходимость наличия в центральном процессоре двух специальных регистров.

**Что такое семафор и какие возможности им предоставляются (в терминах программного интерфейса)?**   
Это средство синхронизации, которое представляет собой целочисленную переменную для подсчета количества активизаций, отложенных на будущее.

Если её значение равно 0, то сохранённые активизации отсутствуют. Иначе, ожидается не менее одной активизации.

Для семафора определены операции down и up. Когда процесс пытается применить операцию down, то сначала проверяется значение семафора: если оно не равно 0, то значение уменьшается на 1 и процесс продолжает свою работу, иначе процесс приостанавливается, не завершая down. Операция up увеличивает значение, адресуемое семафором, на 1. Если с этим семафором связаны приостановленные процессы, способные завершить ранее начатые операции down, система выбирает один из них и позволяет ему завершить его операцию down. Таким образом, после применения операции up в отношении семафора, с которым были связаны приостановленные процессы, значение семафора так и останется нулевым, но количество приостановленных процессов уменьшится на 1.

Проверка значения, его изменение, приостановка процесса (если необходимо) и активизация одного из процессов осуществляются как единое и неделимое атомарное действие (т.е. ни один из процессов не может быть заблокирован при его выполнении).

**Предложите не менее трех ситуаций, в которых ОС может осуществлять планирование процессов. Каковы необходимые условия для планирования?**   
1) При создании нового процесса необходимо принять решение, какой из процессов выполнять, родительский или дочерний.  
2) При завершении процесса планировщик должен понять, какой процесс выполнять следующим.  
3) При блокировке процесса в ожидании какого либо действия (например в ожидании завершения операции ввода-вывода), планировщику необходимо выбрать другой процесс для выполнения.

4) При возникновении прерывания ввода-вывода. Планировщик должен решить, какой процесс ему запускать: тот, что только что перешел в состояние готовности, тот, который был активен до поступления прерывания, или какой-нибудь другой процесс.  
5) При возникновении прерывания от аппаратного таймера.

**Что такое вытесняющая многозадачность?**   
Это многозадачность, при которой операционная система принимает решения о переключении процессора с выполнения одного процесса на другой по истечении некоего кванта времени.

**Какой аппаратный компонент необходим для реализации вытесняющей многозадачности?**

Аппаратный таймер, обработка прерывания которого позволяет операционной системе осуществлять переключение между задачами.

**Что такое критическая секция?**

Это участок исполняемого кода программы, в котором производится доступ к общему ресурсу, который не должен быть одновременно использован более чем одним потоком выполнения.

**Предложите не менее атрибутов в составе блока управления процессом**

1) регистры  
2) счетчик команд  
3) указатель стека  
4) слово состояния программы  
5) состояние процесса

6) приоритет  
7) идентификатор процесса  
8) время начала процесса  
9) родительский процесс  
10) использованное процессорное время