

# Экзамен по ОСям

---

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 приходиться со сданной практикой. Сданная практика - допуск к зачету.

## 1. В чем заключаются две основные функции операционной системы? Пояснить

---

1. Предоставление прикладным программистам (и прикладным программам, естественно) вполне понятный абстрактный набор ресурсов взамен неупорядоченного набора аппаратного обеспечения
2. Управление аппаратными ресурсами

## 2. В чем разница между системами с разделением времени и многозадачными системами?

---

Разделение времени — вариант многозадачности, при котором у каждого пользователя есть свой диалоговый терминал. ЦП по очереди предоставляется нескольким пользователям, желающим работать на машине.

## 3. В чем разница между режимом ядра и пользовательским режимом? Объясните, как сочетание двух отдельных режимов помогает в проектировании операционных систем.

---

Большинство компьютеров имеют два режима работы: режим ядра и режим пользователя. Операционная система — наиболее фундаментальная часть программного обеспечения, работающая в режиме ядра (этот режим называют еще режимом супервизора). В этом режиме она имеет полный доступ ко всему аппаратному обеспечению и может задействовать любую инструкцию, которую машина в состоянии выполнить. Вся остальная часть программного обеспечения работает в режиме пользователя, в котором доступно лишь подмножество инструкций машины. В частности, программам, работающим в режиме пользователя, запрещено использование инструкций, управляющих машиной или осуществляющих операции ввода-вывода (Input/Output — I/O).

## 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.

---

- Многозадачность противоречит надежности. Чем больше программ работают одновременно, тем чаще между ними могут происходить конфликты, приводящие к "зависанию" компьютера.
- Дружественность противоречит простоте, поскольку достижение настоящей дружественности - это не простая, а сверхсложная задача
- Работоспособность в сети противоречит безопасности. С одной стороны мы хотим, чтобы наш компьютер мог запросто получать информацию со всего мира и общаться с другими компьютерами, а с другой стороны, боимся, что вместе с информацией он может получить вирус.

## 5. Задача про время выполнения программ.

---

*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.*

Выполнение программ может занять 20, 25, 30 или 35 мсек в зависимости от того как операционная система назначит их выполнение. Если P0 и P1 назначены на одном и том же CPU, а P2 на другом, то программы выполнятся за 20 мсек. Если P0 и P2 назначены на один CPU, а P1 на другом, то 25 мсек. Если P1 и P2 назначены на одном цп, а P0 на другом, то 30 мсек. Если все три программы будут назначены на один цп то 35 мсек

## 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?

---

Таблица процессов содержит информацию о каждом процессе: открытые файлы, состояния регистров, и т.д. В системах с одним пользователем все равно куча одновременных процессов, и таблица процессов нужна.

## 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?

---

Системный вызов приводит к прерыванию текущего процесса, переходу в режим ядра, выполнению вызова и возврату к выполнению процесса. Это долго, и если производительность важна, то лучше избегать системных вызовов.

## **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.**

---

### **Недостатки:**

- Не все процессоры поддерживают
- Медленно

Виртуализатор или совсем не эмулирует(имитирует) реальную машину, её архитектуру и процессор или делает это в минимальном варианте для отдельных ресурсов.

Эмулятор - полностью или почти полностью реализует для исполнения кода отдельную машину со своей архитектурой и своими ресурсами. Вплоть до того, что может быть процессор совершенно другой архитектуры.

## **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.**

---

Устройства ввода-вывода это аппаратные ресурсы машины. Ими управляет ядро.

### **Причины:**

- безопасность ресурсов компьютера
- скорость обработки
- предоставление программистам нормального API для работы с вводом-выводом.

## **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.**

---

Аппаратное переключение процесса производится путем записи в регистры ЦП значений, образующих аппаратный контекст процесса. Так как в состав этих регистров входят и адресные регистры (например, сегментные регистры), то замена аппаратного контекста процесса приводит к замене полного контекста процесса, выполняемого на ЦП.

## **11. Когда в результате прерывания или системного вызова управление передается операционной системе, используется, как правило, область стека ядра, отделенная от стека прерываемого процесса. Почему?**

1. Ядро не может доверять пользовательскому стеку, там может быть что угодно.
2. Ядро может оставлять чувствительные данные в стеке, доступ к которым пользовательской программе должен быть закрыт.

## **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?**

Степень загрузки ЦП  $= 1 - p^n = 1 - 0.5^5 = 0.96875$

## **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?**

Степень загрузки ЦП  $= 1 - p^n = 1 - 0.4^6 = 0.995904$

## **14. Может ли поток быть приостановлен таймерным прерыванием? Если да, то при каких обстоятельствах, а если нет, то почему?**

Может. Если прерывания не отключены (если программа в критической области, например)

## **15. В чем заключается самое большое преимущество от реализации потоков в пользовательском пространстве? А в чем заключается самый серьезный недостаток?**

---

Можно реализовать в операционной системе, которая не поддерживает потоки (на данный момент - все). Потоки реализуются с помощью библиотеки.

Недостаток - реализация блокирующих системных вызовов.

**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?**

---

Нет. За время обработки видео-вызова 11мс гарантированно придет аудио-вызов, который вызывается с периодичностью 5 мс.

Или да, ведь  $\frac{1}{5} + \frac{1}{5} + \frac{11}{33} < 1$ , и общее время, необходимое процессу, больше того времени, которое может предоставить процессор.

**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.**

---

$$\frac{35}{50} + \frac{20}{100} + \frac{10}{200} + \frac{x}{250} \leq 1$$

$$\frac{175 + 50 + 12.5 + x}{250} \leq 1$$

$$237.5 + x \leq 250$$

$$x \leq 12.5$$

**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?

---

$$\frac{1}{6} + \frac{1}{6} + \frac{20}{\frac{1000}{25}}$$

$$\frac{1}{3} + \frac{1}{2} < 1$$

Поддается.

**19. Рассмотрите систему, в которой желательно разделить политику и механизм планирования потоков, реализованных на уровне ядра. Предложите средства для достижения этой цели.**

---