# **ОСИ 2 ВОПРОСЫ**

1. **Что такое фреймворк OS?**

Фреймворк – набор библиотек и правил работы с ними.

Набор функций для взаимодействия с ОС.

1. **Что такое POSIX?**

Portable Operating System Interface – переносимый интерфейс ОС – набор стандартов взаимодействия между ОС и прикладной программой.

Используется для совместимости UNIX-подобных систем, но может быть использован и на не-UNIX системах.

Благодаря этим стандартам можно запускать код на разных UNIX-подобных системах.

1. **Что такое аппаратное прерывание?**

*Вообще, прерывание – это запрос на остановку текущего выполняемого кода, чтобы выполнить определенное действие, описанное в обработчике* *прерывания*

Аппаратные – реакция микропроцессора на физический сигнал от некоторого устройства (клавиатура, часы, клавиатура и т.д.).

Являются асинхронными, т.е. происходят в случайные моменты времени.

1. **Что такое программное прерывание?**

Программные – вызываются искусственно с помощью команды из соответствующей программы (например, int)

Предназначены для выполнения некоторых действий операционной системы.

Являются синхронными.

1. **Что такое системный вызов?**

Механизм вызова прикладной программой функция ядра ОС методом системного прерывания. Функции ядра ОС кстати работают в привилегированном режиме супервизора (или режиме гипервизора, если такой возможен). В этом режиме доступна вся память и возможно выполнение всех команд процессора.

На RISC/x86 для этого использовался int, на новых системах:

Intel x86\_64 – SYSENTER/SYSEXIT, AMD – SYSCALL/SYSRET

1. **Что такое процесс OS?**

Единица работы ОС. Объект ядра ОС + адресное пространство.

1. **Что такое контекст процесса OS?**

В контексте сохраняется вся информация, необходимая для продолжения процесса после его приостановки.

Включает в себя:

* пользовательский контекст (сегменты программного кода, данных, стека, содержимое виртуального адресного пространства)
* регистровый контекст
* контекст системного уровня: *статическая* (PID, PPID, состояние, приоритет) и *динамическая часть* (один или несколько стеков для выполнения в режиме пользователя и режиме супервизора)

1. **Что такое адресное пространство процесса?**

Адресное пространство процесса – диапазон адресов, которые выделены процессу. В этом диапазоне выделяются адреса, которые процесс может так или иначе использовать.

Каждому процессу выделяется "плоское" (состоит из одного подряд идущего диапазона адресов) 32- или 64-битовое адресное пространство.

Некоторые ОС предоставляют сегментированное адресное пространство —состоящее больше чем из одного диапазона адресов, т.е. из сегментов.

Современные ОС обычно предоставляют плоское адресное пространство.

1. **Перечислите области памяти адресного пространства процесса и поясните их назначение.**

*Адреса возрастают от CODE до STACK.*

CODE – инструкции ЦП скомпилированной программы, создаются автоматически. Read-only память

STATIC – сохраняется на протяжении всей жизни программы, используется, например, для глобальных переменных. Часто имеет размер 4 байта. Хранится в отделах .bss и .data

HEAP – тут хранятся основные данные. Динамическая обширная память

DATA – содержит глобальные и статические переменные

STACK – контекст функции: аргументы, возвращаемое значение, адрес возврата. Тут хранятся локальные переменные. Эта память автоматически выделяется ЦП. Часто имеет размер 8 MB.

1. **Что такое стандартные потоки процесса?**

Это потоки процесса, имеющие зарезервированный номер (дескриптор) для выполнения некоторых стандартных функций:

0 – поток ввода stdin

1 – поток вывода stdout

2 – поток вывода ошибок stderr

[в DOS были потоки с номерами 4 – stdprn и 5 – stdaux]

1. **Перечислите системные вызовы Windows для создания процесса?**

CreateProcess()

CreateProcessAsUser()

CreateProcessWithTokenW()

CreateProcessWithLogonW()

1. **Перечислите системные вызовы Linux для создания процесса?**

fork() – создает клон текущего процесса (дочерний процесс)

exec() – заменяет текущий процесс новым процессом, запускающим другую программу

1. **С помощью каких утилит можно увидеть перечень процессов в Windows?**

tasklist – отображение списка процессов

taskkill – управление процессами и их завершение

1. **С помощью каких утилит можно увидеть перечень процессов в Linux?**

ps – запущенные процессы

top – процессы в режиме реального времени

lsof – список открытых файлов – также выводит процессы

1. **Перечислите свойства процесса OS.**

* Создается ядром по системному вызову
* Может создавать дочерние процессы (с помощью системного вызова)
* При создании выделяется адресное пространство
* Соответствует исполняемому файлу (exe)
* Процессы изолированы друг от друга
* При приостановке или переключении процесса вся необходимая для продолжения информация сохраняется в контексте
* Имеет PID и PPID
* Для обмена данными между процессами используется IPC – механизм межпроцессного взаимодействия
* Процессу автоматически отводится три потока (0, 1, 2)
* При запуске OS некоторые процессы загружаются и стартуют автоматически (Windows – сервисы, UNIX –демоны)
* Есть процесс инициализации – родительский для всех процессов