# **Вопросы к лабораторной работе №4**

1. **Что такое поток управления OS?**

Объект ядра операционной системы, которому ОС выделяет процессорное время, наименьшая единица работы ядра ОС.

Последовательность инструкций, выполняемых процессором в выделенные ОС интервалы времени. При создании процесса в нём есть как минимум один основной поток.

1. **С помощью каких системных вызовов создаются потоки в Windows и Linux?**

CreateThread (Windows)

pthread\_create (Linux)

1. **Что такое системные и пользовательские потоки?**

**Системные потоки** (потоки ядра ОС) – выполняют различные сервисы ОС и запускаются ядром ОС, используются для реализации пользовательских потоков.

**Пользовательские потоки** – потоки, служащие для решения задач пользователя, и запускаемые приложением.

1. **Что такое многопоточность?**

Это способность ЦП, позволяющая выполнять 2 или более процесса/потока с инструкциями независимо друг от друга, используя ресурсы одного процессора и разделяя между ними процессорное время.

1. **Чем отличаются приоритетная многопоточность от кооперативной многопоточности?**

это две разные стратегии управления многопоточными процессами

*1. Приоритетная многопоточность* (Preemptive Multithreading):

потоки могут быть принудительно приостановлены операционной системой (планировщиком) в любой момент, чтобы предоставить возможность выполнения другим потокам с более высоким приоритетом.

Операционные системы, такие как Windows и большинство десктопных Linux-систем, используют приоритетную многозадачность.

*2. Кооперативная многопоточность* (Cooperative Multithreading):

Здесь потоки сами решают, когда они готовы освободить процессор и передать управление другим потокам. Операционная система не вмешивается в переключение контекста между потоками, и каждый поток должен самостоятельно учесть интересы других потоков.

Этот подход часто используется в системах реального времени и в некоторых мобильных операционных системах.

1. **Что такое диспетчеризация потоков управления OS?**

***Диспетчеризация*** — это процесс переключения процессора с одного потока на другой соответственно плану. Работает по принципу FIFO, причем каждый процесс получает ограниченное процессорное время.

Диспетчеризация сводится к следующему:

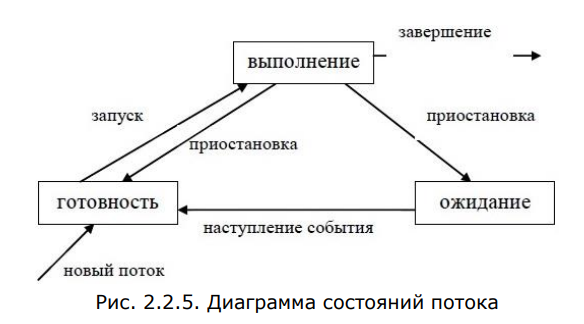
* сохранение контекста текущего потока, который требуется сменить;
* загрузка контекста нового потока, выбранного в результате планирования;
* запуск нового потока на выполнение

1. **Что такое контекст потока и для чего он нужен?**

Контекст потока – данные, необходимые для возобновления работы потока при его приостановке, а именно:

Программный код, набор регистров, стек памяти, оперативная память, стек ядра и маркер доступа.

1. **Перечислите состояния в которых может быть поток и поясните их назначение.**

****

* [5] New – поток создан
* [5] Ready – готов к исполнению
* [5] Running – исполняется
* [12] Sleeping – заснул на некоторое время
* [12] Sleeping suspended – спит и приостановлен до события
* [7] Suspended ready – готов и приостановлен до события
* [5] Blocked – заблокирован извне
* [7] Suspended blocked – заблокирован извне и приостановлен до события
* [5] Finish – поток завершил исполнение

**Модель 5 состояний:**



**Модель 7 состояний:**



**Модель 12 состояний:**



1. **Что такое LWP?**

LWP (light-weight process) – это средство достижения многозадачности, представляющее отдельный слой между потоками ядра и пользовательскими потоками. LWP работает в пользовательском пространстве поверх одного потока выполнения ядра.

Один процесс может содержать несколько LWP, что позволяет им совместно использовать ресурсы процесса.

ОС управляет LWP (распределяет процессорное время между LWP в соответствии с их приоритетами и состоянием)

LWP внутри одного процесса обычно имеют общее адресное пространство, что упрощает обмен данными между ними.

Создание и управление LWP требует меньше системных ресурсов, более эффективное параллельное выполнение задач

1. **Что такое потокобезопасность программного кода?**

Свойство программного кода (программы) корректно работать в нескольких потоках одновременно. Для обеспечения потокобезопасности кода часто используются механизмы синхронизации,

Потокобезопасность относится к возможности множества потоков (thread) безопасно работать с общими данными и ресурсами.

1. **Что такое реентерабельность кода?**

**Реентерабельность** — свойство одной копии программного кода работать в нескольких потоках одновременно. Реентерабельный код всегда потокобезопасен. Реентерабельный код не использует статическую память и не изменяет сам себя, все данные сохраняются в динамической памяти.

Реентерабельность относится к способности функции или участка кода выполняться безопасно и корректно при вызове из разных потоков (или рекурсивно) в одно и то же время, и каждый вызов будет работать независимо и безопасно, не влияя на состояние других вызовов

даже без явной синхронизации.

1. **Что такое Fiber?**

Fiber (волокно) – механизм для ручного планирования выполнения кода в рамках потока. Находится внутри потоков (процессы –> потоки –> волокна) и является особенно легковесным потоком.

1. **Дайте развернутое определение потока OS.**

Объект ядра операционной системы, его наименьшая единица работы, являющаяся средством диспетчеризации доступа к процессорному времени, имеющая свой идентификатор, контекст, состояние.

**Основные свойства потоков:**

* поток – это наименьшая единица работы ядра ОС, последовательность команд процессора
* каждый процесс имеет как минимум один поток (основной, main)
* каждый поток имеет свой идентификатор
* создание потока осуществляется с помощью системного вызова
* процесс – контейнер для потоков
* у потоков есть контекст – данные, необходимые для возобновления работы потока при его приостановке
* поток может создавать дочерние потоки и их завершать
* завершение родительского потока приводит к завершению всех его дочерних (требуется ожидание дочернего завершение потока)
* потоки в рамках одного процесса не изолированы, все ресурсы кроме процессорного времени – общие
* существует парадигма многопоточности (возможность выполнять два и более потоков одновременно)
* желательно программы должны быть потокобезопасными – то есть корректно работать в нескольких потоках одновременно
* реентерабельность программы – свойство одной копии программного кода работать в нескольких потоках одновременно;
* волокно – механизм для ручного планирования выполнения кода в рамках потока.