

## 1. Что такое поток управления ОС?

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.

## 2. С помощью каких системных вызовов создаются потоки в Windows и Linux?

CreateThread (Windows)  
pthread\_create (Linux)

## 3. 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.

## 4. 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.

## 5. Чем отличаются приоритетная многопоточность от кооперативной многопоточности?

**Приоритетная многопоточность:** В приоритетной многопоточности планировщик операционной системы принимает решение о переключении между потоками на основе их приоритета. Каждому потоку присваивается определенный приоритет, и планировщик определяет, какой поток будет выполняться следующим, исходя из их относительных приоритетов.

**Кооперативная многопоточность:** В кооперативной многопоточности контроль переключения между потоками полностью лежит на самих потоках. Каждый поток должен явно передать управление другому потоку, чтобы он мог выполняться. Потоки сотрудничают (кооперируют) друг с другом, чтобы

определить моменты, когда они готовы передать управление другому потоку. Если поток неявно или явно не передает управление другому потоку, выполнение программы может блокироваться или зависать.

## 6. Что такое диспетчеризация потоков управления OS?

// для диспетчеризации одинаково применимо как слово процесс, так и слово поток  
В общем, при работе нескольких потоков или процессов нужно определить, какой из них будет работать и какое количество времени. Именно этим и занимается диспетчеризация. Она позволяет эффективно загрузить процессор. **Диспетчеризация** — это процесс переключения процессора с одного потока на другой соответственно плану. Работает по принципу FIFO, причем каждый процесс получает ограниченное процессорное время.

Есть 2 основных типа:

1. **С вытеснением** – квантовый генератор генерирует сигнал, что время процесса вышло. Система сама вносит правки в то, какой процесс сейчас будет работать
2. **Без вытеснения** – программа завершится, если:
  - 1) она закончилась
  - 2) она перешла в ожидающий режим
  - 3) она сама решила отдать процесс другой программе

## 7. Что такое контекст потока и для чего он нужен?

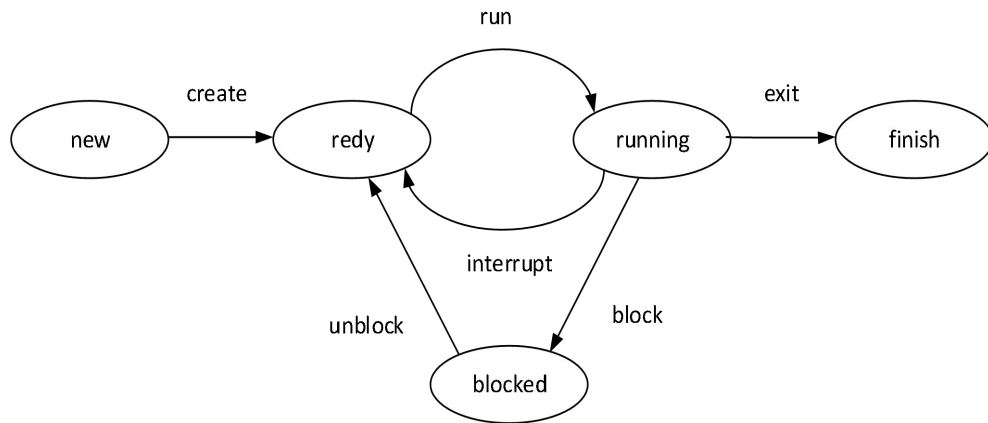
Контекст потока – данные, необходимые для возобновления работы потока при его приостановке, а именно:

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

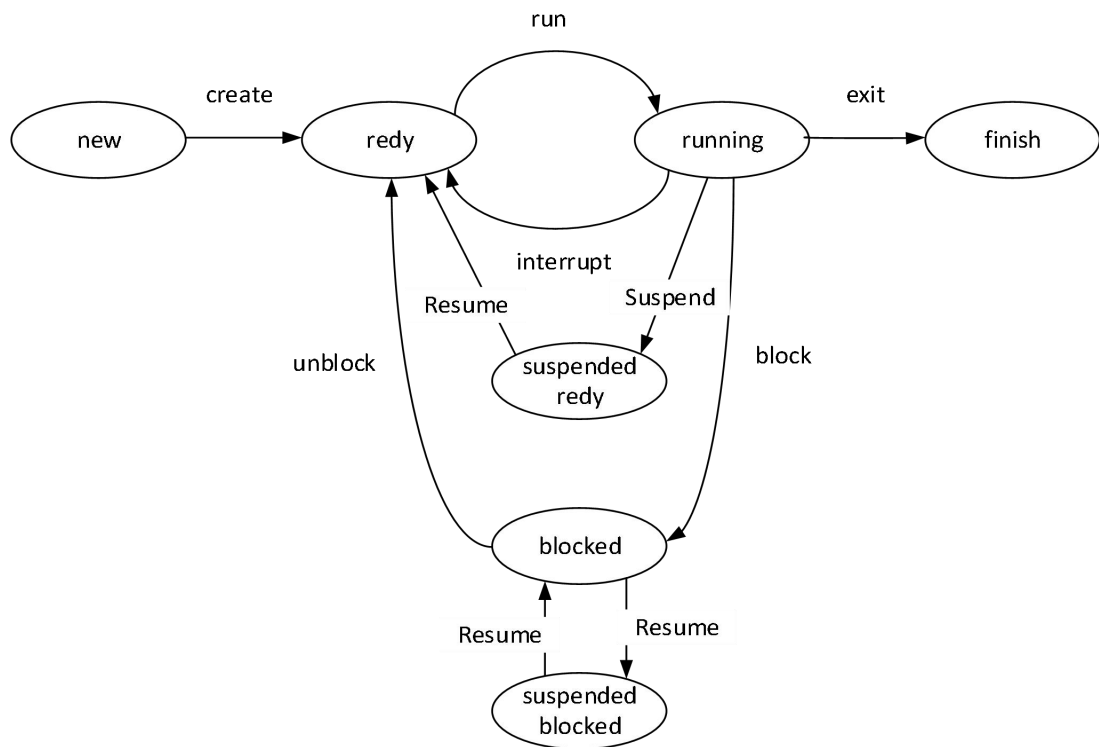
## 8. Перечислите состояния в которых может быть поток и поясните их назначение.

- [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

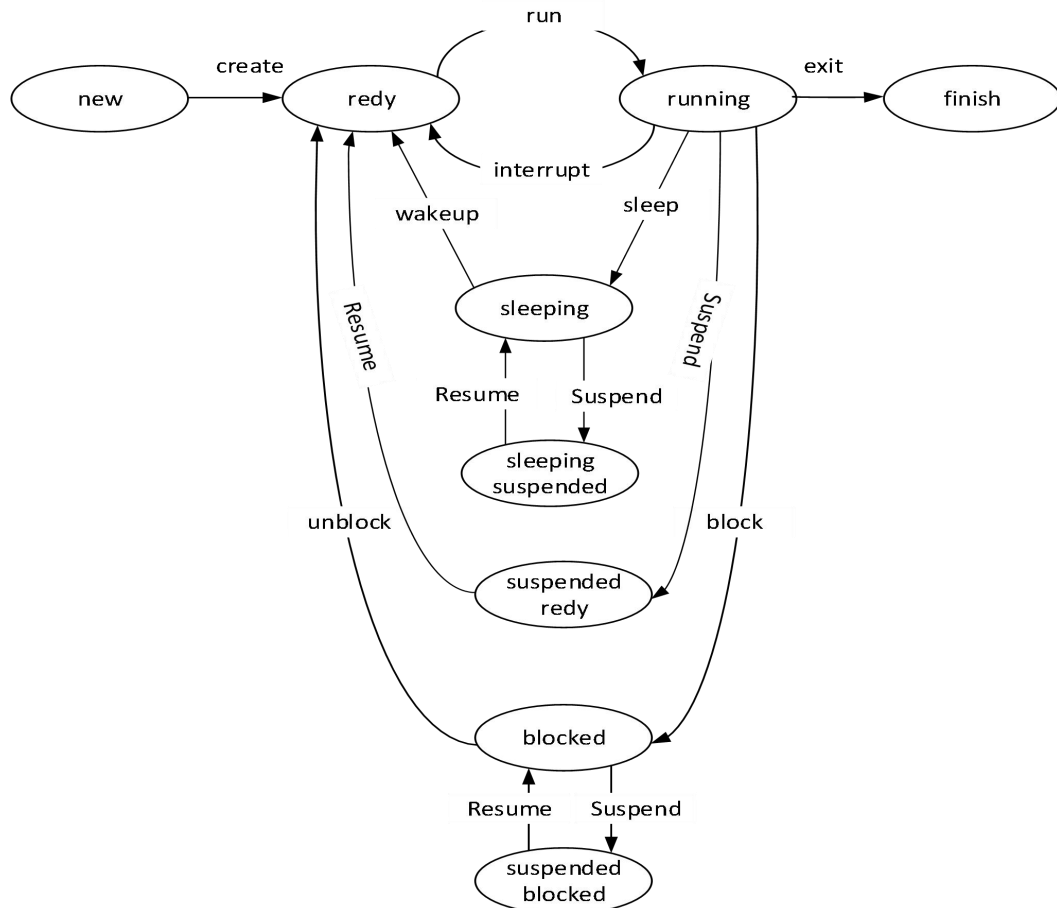
### Модель 5 состояний:



### Модель 7 состояний:



### Модель 12 состояний:



## 9. Что такое LWP?

LWP (light-weight process) – процесс, поддерживающий работу потока пользовательского пространства. Средство достижения многозадачности.

Несколько пользовательских потоков могут быть размещены в одном или нескольких легковесных процессах, что даёт многозадачность на уровне пользователя, которая может иметь некоторые преимущества в производительности.

## 10. Что такое потокобезопасность программного кода?

Свойство программного кода (программы) корректно работать в нескольких потоках одновременно. Гарантирует, что при исполнении нескольких потоков код будет правильно себя вести.

Имеет **два** основных принципа (из лекции Смелова):

1. Код не должен сам себя менять
2. Не должно быть статической области памяти (общей для двух потоков)

## 11. Что такое реентерабельность кода?

Компьютерная программа в целом или её отдельная процедура называется **реентерабельной**, если она разработана таким образом, что одна и та же копия инструкций программы в памяти может быть совместно использована несколькими пользователями или процессами. При этом второй пользователь может вызвать реентерабельный код до того, как с ним завершит работу первый пользователь и это не должно привести к ошибке или потере данных. **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.

## 12. 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.

## 13. Дайте развернутое определение потока OS.

### 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 завершения 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.