

## Создание и переключение процесса

### Что происходит при создании процесса

Во-первых, нам нужно присвоить ему какой-то **идентификатор**, нам нужно его назвать – часто используются цифры. Дальше любому процессу **нужна память** (тут может происходить простое предоставление памяти, если она есть или если памяти не хватает, то возможно операционная система запустит какую-то работу по освобождению памяти, может она сделает свопинг, то есть она перенесет некоторые процессы из состояния **заблокирован** в состояние **заблокирован/приостановлен**). Дальше нужно **инициализировать контрольный блок** – это та структура данных, которая представляет процесс в памяти, и **создать или изменить** какие-то **существующие структуры**. Возможно созданный процесс является дочерним процессом другого процесса и тогда нужно изменить структуру. Таблицы, о которых мы говорили, многие из них также нужно обновить (как минимум таблицу процессов – мы создали новый процесс – это означает, что его нужно внести в эту таблицу, записать туда его описание, записать, где в памяти он находится, если он уже там находится, или к какому участку памяти он имеет доступ, и так далее).

### Переключение процесса

При переключении процесса очень важно понять – **какой процесс**, собственно, **стимулировал это переключение** – может это была операционная система, может какой-то процесс напрямую захотел этого; и **что же операционная система должна делать со всеми этими структурами, когда происходит приключение процесса**.

### Что происходит при переключении процесса

- Нужно **сохранить** где-то всё это **состояние процесса (регистры)**, нужно сохранить всё то, что нужно будет для того, чтобы этот процесс возобновить в будущем.
- Нужно **обновить контрольный блок** процесса, который запущен в **данный момент**, который мы собираемся остановить.
- Этот **контрольный блок** нужно **перенести в соответствующую очередь**. Иными словами – мы сменим состояние этого процесса и, в зависимости от того, что же с ним происходит – он может перейти сразу в очередь *ready*, то есть ту очередь из которой он в ближайшее время запустится. Он может перейти в состояние **заблокирован**, если приключение процесса происходит потому, что этот процесс начинает ожидание устройства ввода-вывода или чего-нибудь еще. Или же он может напрямую уходить в состояние *ready/suspend*, то есть **готов**, но **приостановлен**.
- Дальше нужно **выбрать следующий процесс для выполнения** и это, чаще всего, делается в очереди на основании того, какие там есть процесс и какие у них приоритеты, и как, собственно, эта очередь выглядит. Естественно, в первую очередь нужно смотреть на очередь процессов, которые готовы к запуску.
- Дальше нужно **взять** этот процесс, **на которой нам нужно переключиться, и обновить его контрольный блок**.
- Нужно **обновить все структуры данных управления памятью**.
- **Восстановить состояние процесса**. То, что записано в процессе, который ждал своей очереди – нужно выгрузить, нужно взять из памяти и восстановить окружение, восстановить контекст, чтобы этот процесс мог продолжать работу.
- И, как минимум, нужно **восстановить состояние регистров процессора**. В любой момент, когда мы останавливаем процесс – какая-то инструкция завершается, но, когда мы этот процесс восстановим и он должен продолжать работу – будет запущена следующая инструкция и эта инструкция ожидает, что только что завершилась предыдущая, она ожидает, что из-за этого определенные регистры процессора имеют определенное значение и, с точки зрения этой

следующей инструкции, нет никакой разницы: только что это произошло или 25 минут назад. С точки зрения процесса, который продолжается – процессор и всё, что его окружает, и весь компьютер – находятся в том же состоянии, в котором он должен находиться. Поэтому нам нужно, как минимум, восстановить все регистры процессора, чтобы можно было продолжать и не было ошибок.