Лекция 2.
Процессы. Планирование процессов.

Процесс

- некоторая деятельность, связанная с исполнением программы на процессоре.
- представляет собой просто динамическое описание кода исполняемого файла, данных и выделенных для них ресурсов.
- характеризует некоторую совокупность набора исполняющихся команд, ассоциированных с ним ресурсов (выделенная для исполнения память или адресное пространство, стеки, используемые файлы и устройства ввода-вывода и т. д.) и текущего момента его выполнения (значения регистров, программного счетчика, состояние стека и значения переменных), находящуюся под управлением операционной системы

Состояния процесса

- *порождение* — подготавливаются условия для первого исполнения на процессоре
- *активное состояние*, или состояние “Счет” — программа исполняется на процессоре
- *ожидание* — программа не исполняется на процессоре по причине занятости какого-либо требуемого ресурса
- *готовность* — программа не исполняется, но для исполнения предоставлены все необходимые в текущий момент ресурсы, кроме центрального процессора
- *окончание* — нормальное или аварийное окончание исполнения программы, после которого процессор и другие ресурсы ей не предоставляются

Простейшая диаграмма состояний процесса

Диаграмма состояний процесса

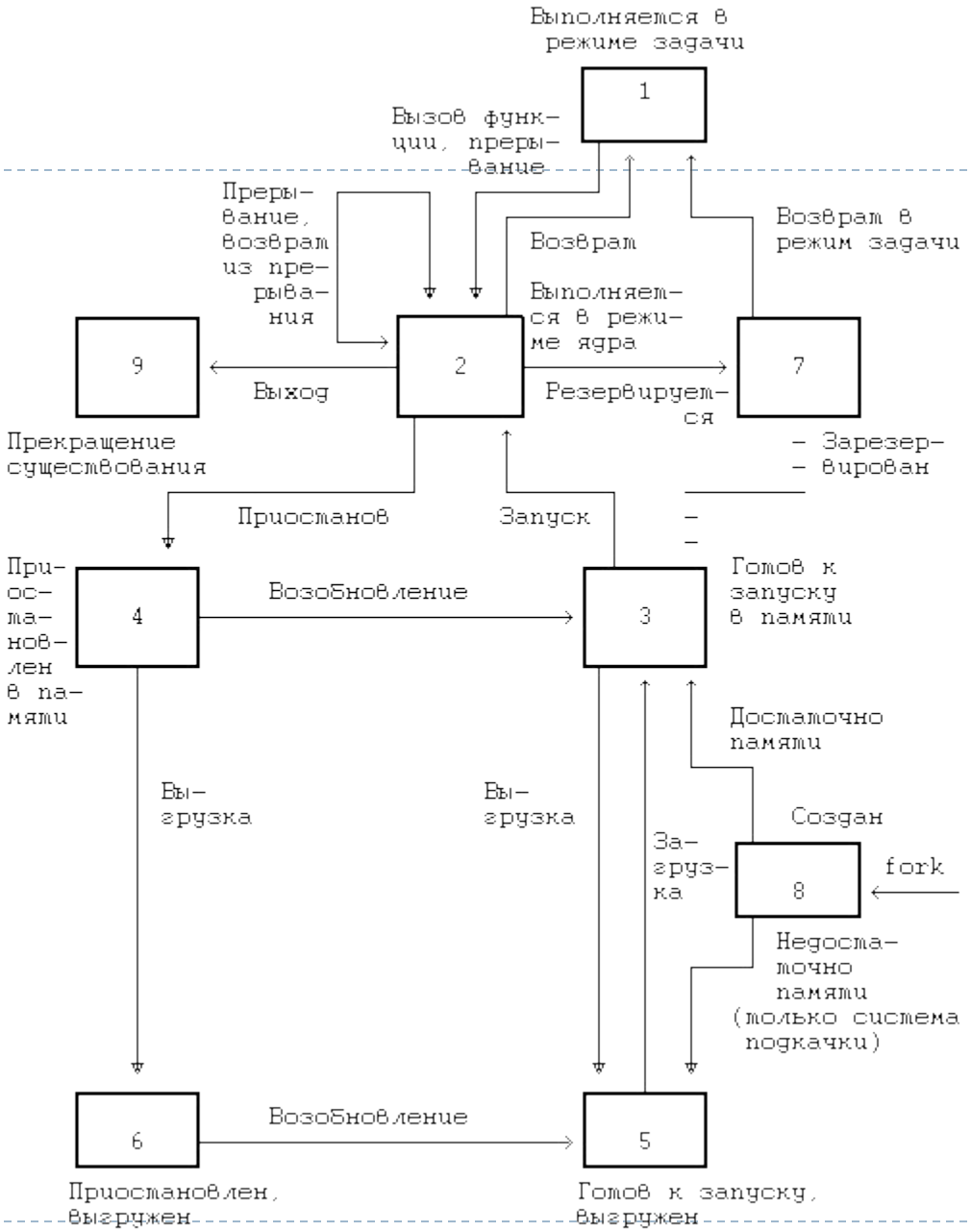
- 15:03:11 up 58 min, 4 users, load average: 0,02, 0,01, 0,00
- 52 processes: 51 sleeping, 1 running, 0 zombie, 0 stopped
- CPU states: 0,8% user, 0,6% system, 0,0% nice, 0,0% iowait, 98,3% idle
- Mem: 127560k av, 124696k used, 2864k free, 0k shrd, 660k buff
- 13460k active, 17580k inactive
- Swap: 152576k av, 8952k used, 143624k free 28892k cached
-
- PID USER PRI NI SIZE RSS SHARE STAT %CPU %MEM TIME COMMAND
- 3097 den 15 0 1128 1128 832 R 2,8 0,8 0:00 top
- 1 root 8 0 120 84 60 S 0,0 0,0 0:04 init
- 2 root 12 0 0 0 0 SW 0,0 0,0 0:00 keventd
- 3 root 19 19 0 0 0 SWN 0,0 0,0 0:00 ksoftirqd_CPU0

Операции над процессами

- *создание процесса – завершение процесса* ;
- *приостановка процесса* (перевод из *состояния* исполнение в *состояние* готовность) – *запуск процесса* (перевод из *состояния* готовность в *состояние* исполнение);
- *блокирование процесса* (перевод из *состояния* исполнение в *состояние* ожидание) – *разблокирование процесса* (перевод из *состояния* ожидание в *состояние* готовность).

Process Control Block и **Контекст процесса**

- *состояние*, в котором находится *процесс* ;
- программный счетчик *процесса* или, другими словами, адрес команды, которая должна быть выполнена для него следующей;
- содержимое регистров процессора;
- данные, необходимые для планирования использования процессора и управления памятью (приоритет *процесса*, размер и расположение адресного пространства и т. д.);
- учетные данные (идентификационный номер *процесса*, какой пользователь инициировал его работу, общее время использования процессора данным *процессом* и т. д.);
- сведения об устройствах ввода-вывода, связанных с *процессом* (например, какие устройства закреплены за *процессом*, таблицу открытых файлов).

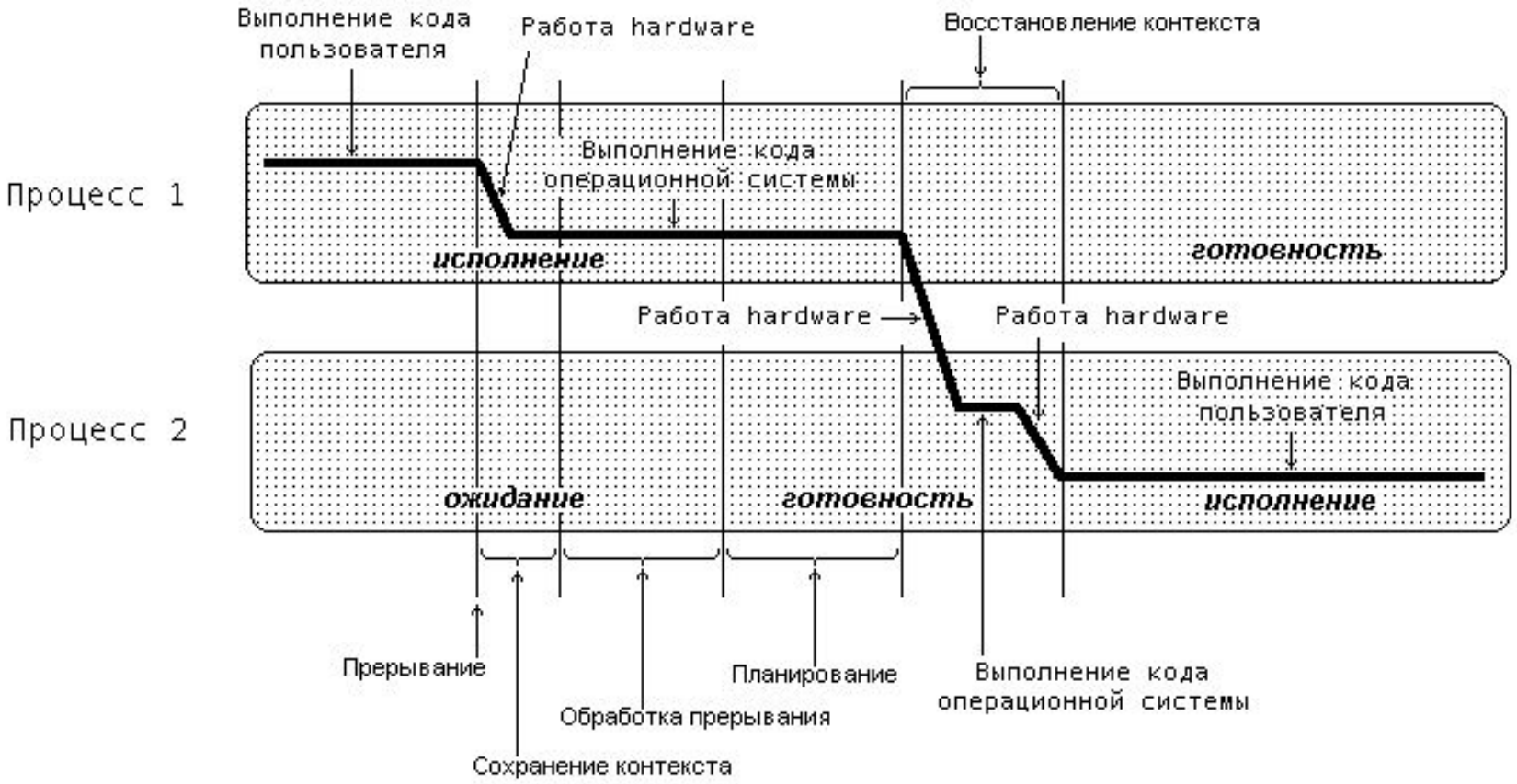

Выполняемся в режиме задачи
1
Вызов функции, прерывание
Прерывание, возврат из прерывания
Возврам в режим задачи
Возврам
Выполняемся в режиме ядра
9
2
7
Выход
Резервируется
Прекращение существования
– Зарезервирован
Приосманов
Запуск
Приостановлен в памяти
4
Возобновление
3
Готов к запуску в памяти
Достамочно памяти
Выгрузка
Выгрузка
Создан
Загрузка
8
fork
Недостамочно памяти (молько система подкачки)
6
Возобновление
5
Приосмановлен, выгружен
Готов к запуску, выгружен

Многоразовые операции

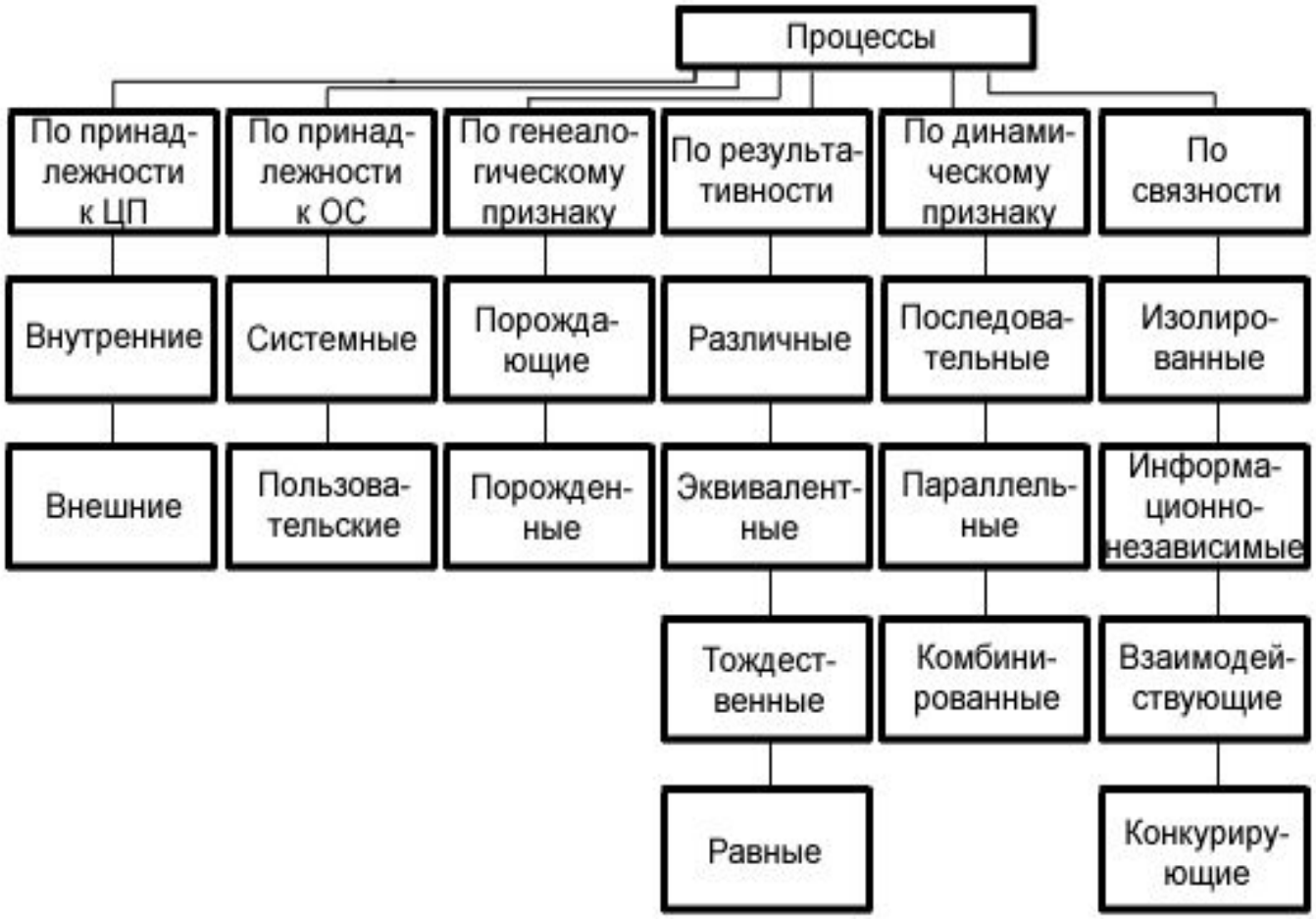
- ***Запуск процесса***. Из числа *процессов*, находящихся в *состоянии* готовность, операционная система выбирает один *процесс* для последующего исполнения. Для избранного *процесса* операционная система обеспечивает наличие в оперативной памяти информации, необходимой для его дальнейшего выполнения. Далее *состояние процесса* изменяется на *исполнение* , восстанавливаются значения регистров для данного *процесса* и управление передается команде, на которую указывает счетчик команд *процесса*. Все данные, необходимые для восстановления *контекста*, извлекаются из *PCB процесса*, над которым совершается *операция*.
- ***Приостановка процесса***. Работа *процесса*, находящегося в *состоянии* исполнение, приостанавливается в результате какого-либо прерывания. Процессор автоматически сохраняет счетчик команд и, возможно, один или несколько регистров в стеке исполняемого *процесса*, а затем передает управление по специальному адресу обработки данного прерывания. На этом деятельность hardware по обработке прерывания завершается. По указанному адресу обычно располагается одна из частей операционной системы. Она сохраняет динамическую часть *системного* и *регистрового контекстов процесса* в его *PCB*, переводит *процесс* в *состояние* готовность и приступает к обработке прерывания, то есть к выполнению определенных действий, связанных с возникшим прерыванием.

- ***Блокирование процесса***. *Процесс* блокируется, когда он не может продолжать работу, не дождавшись возникновения какого-либо события в вычислительной системе. Для этого он обращается к операционной системе с помощью определенного системного вызова. Операционная система обрабатывает системный вызов (инициализирует операцию ввода-вывода, добавляет *процесс* в очередь *процессов*, дожидающихся освобождения устройства или возникновения события, и т. д.) и, при необходимости сохранив нужную часть *контекста процесса* в его *PCB*, переводит *процесс* из *состояния* исполнение в *состояние* ожидание .
Разблокирование процесса. После возникновения в системе какого-либо события операционной системе нужно точно определить, какое именно событие произошло. Затем операционная система проверяет, находился ли некоторый *процесс* в *состоянии* ожидание для данного события, и если находился, переводит его в *состояние* готовность, выполняя необходимые действия, связанные с наступлением события (инициализация *операции* ввода-вывода для очередного ожидающего *процесса* и т. п.).

Переключение контекста



Виды процессов



Критерии планирования и требования к алгоритмам

- *Справедливость*: гарантировать каждому заданию или процессу определенную часть времени использования процессора в компьютерной системе, стараясь не допустить возникновения ситуации, когда процесс одного пользователя постоянно занимает процессор, в то время как процесс другого пользователя фактически не приступал к выполнению.
- *Эффективность*: постараться занять процессор на все 100% рабочего времени, не позволяя ему простаивать в ожидании процессов готовых к исполнению. В реальных вычислительных системах загрузка процессора колеблется от 40 до 90 процентов.
- *Сокращение полного времени выполнения* (turnaround time): обеспечить минимальное время между стартом процесса или постановкой задания в очередь для загрузки и его завершением.
- *Сокращение времени ожидания* (waiting time): минимизировать время, которое проводят процессы в состоянии готовность и задания в очереди для загрузки.
- *Сокращение времени отклика* (response time): минимизировать время, которое требуется процессу в интерактивных системах для ответа на запрос пользователя.

Свойства алгоритмов

- Были предсказуемыми. Одно и то же задание должно выполняться приблизительно за одно и то же время. Применение алгоритма планирования не должно приводить, к примеру, к извлечению корня квадратного из 4 за сотые доли секунды при одном запуске и за несколько суток при втором запуске.
- Имели минимальные накладные расходы, связанные с их работой. Если на каждые 100 миллисекунд, выделенных процессу для использования процессора, будет приходиться 200 миллисекунд на определение того, какой именно процесс получит процессор в свое распоряжение, и на переключение контекста, то такой алгоритм, очевидно, использовать не стоит.
- Равномерно загружали ресурсы вычислительной системы, отдавая предпочтение тем процессам, которые будут занимать малоиспользуемые ресурсы.
- Обладали масштабируемостью, т. е. не сразу теряли работоспособность при увеличении нагрузки. Например, рост количества процессов в системе в два раза не должен приводить к увеличению полного времени выполнения процессов на порядок.

Цели планирования

- справедливость;
- обеспечивать максимальную пропускную способность системы;
- приемлемые времена ответа для максимального количества пользователей, работающих в интерактивном режиме;
- предсказуемость;
- минимальные накладные расходы;
- сбалансированное использование ресурсов;
- сбалансированность времени ответа и коэффициента использования ресурсов;
- должна исключать бесконечное откладывание процессов;
- учитывать приоритеты;
- оказывать предпочтение тем процессам, которые занимают ключевые ресурсы;
- предусматривать улучшенное обслуживание для процессов, отличающихся «примерным поведением».

Факторы, учитываемые при планировании

- лимитируется ли процесс вводом-выводом или ЦП;
- является ли процесс пакетным или диалоговым;
- обязательно ли малое время ответа;
- приоритет каждого процесса;
- частоту переключений с низкоприоритетных процессов, ожидающих освобождения уже занятых ресурсов;
- длительность периода времени, в течение которого ожидает каждый процесс;
- суммарное время работы каждого процесса и оценочное время, необходимое каждому процессу для завершения.

Приоритет

- **Статические** приоритеты не изменяются, такой механизм установки приоритетов достаточно прост и не сопряжен с большими издержками. Однако следует учитывать, что такой механизм недостаточно гибок, т.к. не реагирует на изменение окружающей ситуации.
- **Динамические** приоритеты позволяют повысить реактивность системы, т.к. реагируют на изменения ситуации, и начальное значение приоритета процесса может быть изменено на новое, более подходящее значение.
- **Покупаемые** приоритеты дают возможность пользователю повысить приоритет задания и получить более высокий уровень обслуживания за "дополнительную плату" (например, уменьшение кванта времени).

Параметры планирования (статические)

- Каким пользователем запущен процесс или сформировано задание.
- Насколько важной является поставленная задача, т. е. каков приоритет ее выполнения.
- Сколько процессорного времени запрошено пользователем для решения задачи.
- Каково соотношение процессорного времени и времени, необходимого для осуществления операций ввода-вывода.
- Какие ресурсы вычислительной системы (оперативная память, устройства ввода-вывода, специальные библиотеки и системные программы и т. д.) и в каком количестве необходимы заданию.

Параметры планирования (динамические)

- Сколько времени прошло со времени выгрузки процесса на диск или его загрузки в оперативную память.
- Сколько оперативной памяти занимает процесс.
- Сколько процессорного времени было уже предоставлено процессу

Задачи планирования процессов

- определение момента времени для смены выполняемого процесса;
- выбор процесса на выполнение из очереди готовых процессов;
- переключение контекстов "старого" и "нового" процессов.

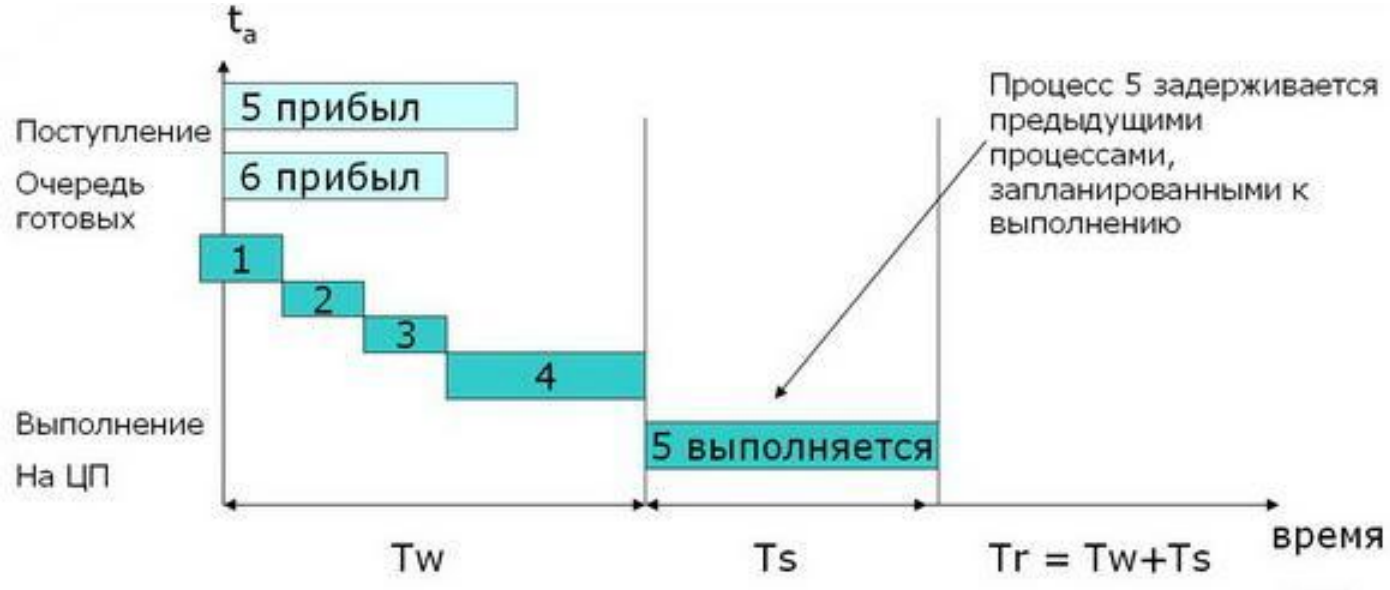
Задачи алгоритмов планирования

- **Для всех систем**
  Справедливость - каждому процессу справедливую долю процессорного времени
  Контроль над выполнением принятой политики
  Баланс - поддержка занятости всех частей системы (например: чтобы были заняты процессор и устройства ввода/вывода)
- **Системы пакетной обработки**
  Пропускная способность - количество задач в час
  Оборотное время - минимизация времени на ожидание обслуживания и обработку задач.
  Использование процесса - чтобы процессор всегда был занят.
- **Интерактивные системы**
  Время отклика - быстрая реакция на запросы
  Соразмерность - выполнение ожиданий пользователя (например: пользователь не готов к долгой загрузке системы)
- **Системы реального времени**
  Окончание работы к сроку - предотвращение потери данных
  Предсказуемость - предотвращение деградации качества в мультимедийных системах (например: потерь качества звука должно быть меньше чем видео)

Алгоритмы планирования

- Планирование по принципу FIFO (first-in-first-out)
- Циклическое планирование round robin (RR)
- Многоуровневые очереди (Multilevel Queue)
- Shortest-Job-First (SJF)
- First-Come, First-Served (FCFS)
- Многоуровневые очереди с обратными связями (Multilevel Feedback Queue)

Метрики планирования



- **ta**— время поступления процесса (когда процесс становится готовым к выполнению);
- **Tw** – время ожидания (которое тратит процесс в очереди на выполнение);
- **Ts** – время выполнения ЦП;
- **Tr** – время оборота (общее время на ожидание и выполнение).

Категории средств обмена информацией

- Сигнальные. Передается минимальное количество информации – один бит, "да" или "нет". Используются, как правило, для извещения процесса о наступлении какого-либо события. Степень воздействия на поведение процесса, получившего информацию, минимальна. Все зависит от того, знает ли он, что означает полученный сигнал, надо ли на него реагировать и каким образом. Неправильная реакция на сигнал или его игнорирование могут привести к трагическим последствиям.
- Канальные. "Общение" процессов происходит через линии связи, предоставленные операционной системой, и напоминает общение людей по телефону, с помощью записок, писем или объявлений. Объем передаваемой информации в единицу времени ограничен пропускной способностью линий связи. С увеличением количества информации возрастает и возможность влияния на поведение другого процесса.
- Разделяемая память. Два или более процессов могут совместно использовать некоторую область адресного пространства. Созданием разделяемой памяти занимается операционная система (если, конечно, ее об этом попросят).

Надежность средств связи

- Не происходит потери информации.
- Не происходит повреждения информации.
- Не появляется лишней информации.
- Не нарушается порядок данных в процессе обмена.