



Операционные системы

Лекция 3 Процессы

[Понятие процесса]

Операционная система во время работы выполняет одну или несколько программ, планирует задания (совокупность программы, команд для ее выполнения и входных данных).

Так термины «программа» и «задание» предназначены для описания статических объектов. Программа в процессе выполнения – динамически активный объект!

- Процесс – описывает активные объекты внутри компьютерной системы – абстракция, характеризующая программу во время ее выполнения (не корректно! – процесс – совокупность исполняемых команд, ресурсов, времени выполнения).
- Не существует однозначного соответствия между процессами и программами, обрабатываемыми вычислительной системой (процесс – динамическое описание исполняемого кода и выделенных ресурсов – *не верно даже если процесс обрабатывает одну программу*)
- Процесс управляется ОС (в нем обрабатывается часть кода ядра).

Состояния процесса

Все что выполняется в ОС (в т.ч. части ОС) организовано как набор процессов

Однопроцессорные и мультипрограмные (переключение между процессами)

ОС – 2 состояния процесса

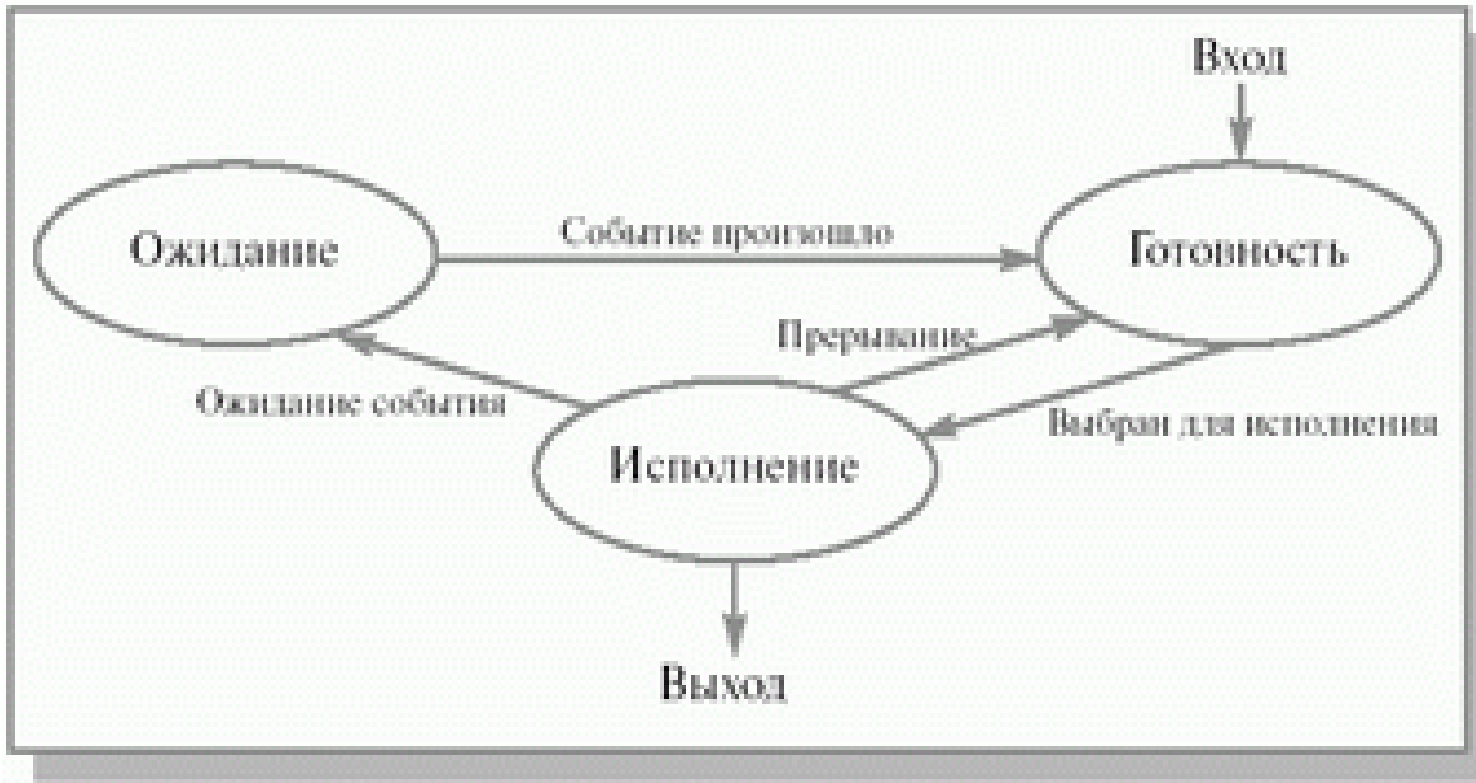


«-» выбор ОС одного из процесс в состоянии «не выполняется»

Новый процесс помещается в состояние «исполняется»

Состояния процесса

Процесс, выбранный для исполнения может ждать события и к выполнению не готов – разделение «не исполняется» на 2 новых состояния



Новый процесс - в состоянии «готовность» - исполнение программного кода процесса

[Состояния процесса]

Выход из состояния «исполнение»:

- ОС прекращает его деятельность
- Нет возможности продолжить выполнение без наступления события (перевод в состояние «ожидание»)
- Возникновение прерывания в вычислительной системе (прерывание таймера) -(возврат в состояние готовность)

« + » -хорошо описывает поведение процессов во время существования

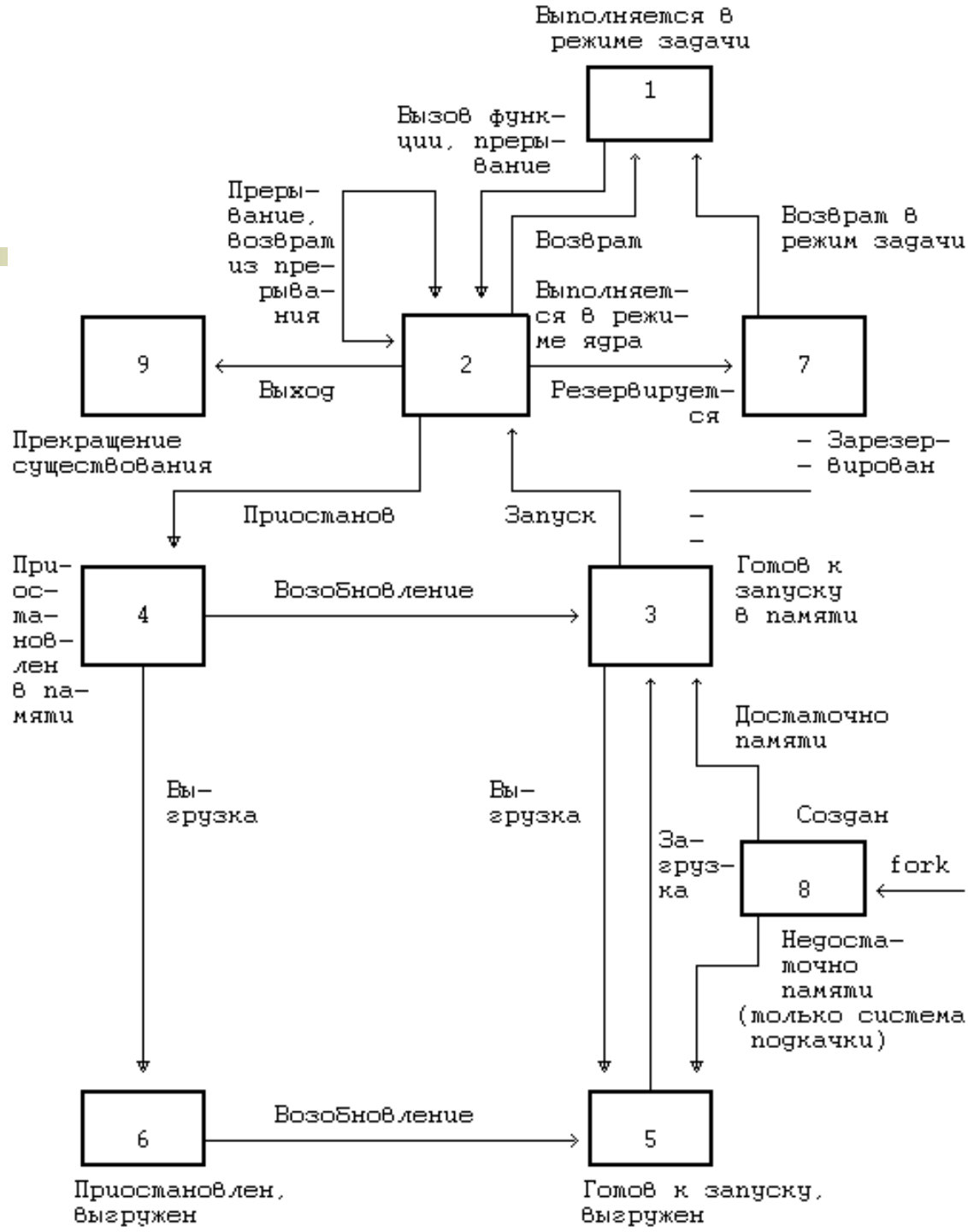
« - » - не акцентирует внимание на появлении процесса в системе и его исчезновении – ввод 2-х дополнительных состояний

Состояния процесса



Состояния процесса (UNIX)

В различных ОС состояния процесса еще более детализированы, однако все ОС подчиняются изложенной модели



Операции над процессами

Процесс не может самостоятельно изменить свое состояние – привилегия ОС (совершая операции над процессами)

3 вида парных операций над процессами

- создание процесса – завершение процесса;
- приостановка процесса– запуск процесса (исполнение - готовность);
- блокирование процесса– разблокирование процесса (исполнение – ожидание, ожидание - готовность).

Непарная операция - изменение приоритета процесса

Process Control Block

Для того, чтоб ОС могла выполнять операции над процессами, каждый процесс представляется некоторой структурой данных, которая содержит спец. Информацию о нем (зависит от ОС):

- состояние;
- программный счетчик процесса (адрес следующей команды);
- содержимое регистров процессора;
- данные, необходимые для планирования использования процессора и управления памятью;
- учетные данные;
- сведения об устройствах ввода-вывода, связанных с процессом (устройства, закрепленные за процессом, таблица открытых файлов).

Контекст процесса

- Во многих ОС информация, характеризующая процесс, хранится не в одной, а в нескольких связанных структурах данных. Эти структуры могут иметь различные наименования, содержать дополнительную информацию.
- Для любого процесса, находящегося в вычислительной системе, вся информация, необходимая для совершения операций над ним, доступна операционной системе.

Управляющая структура данных - PCB (Process Control Block) или блок управления процессом

- Любая операция, производимая операционной системой над процессом, вызывает определенные изменения в PCB.
- В рамках принятой модели состояний процессов содержимое PCB между операциями остается постоянным.

Контекст процесса

Информацию, для хранения которой предназначен блок управления процессом, можно условно разделить на две части:

- содержимое всех регистров процессора (включая значение программного счетчика) - регистровый контекст процесса,
- остальное – системный контекст процесса.

Знания регистрового и системного контекстов процесса достаточно для того, чтобы управлять его работой в операционной системе, совершая над ним операции. Однако недостаточно для того, чтобы полностью охарактеризовать процесс.

Контекст процесса

С точки зрения пользователя (в отличие от ОС), наибольший интерес представляет содержимое адресного пространства процесса, возможно, наряду с регистровым контекстом определяющее последовательность преобразования данных и полученные результаты – **пользовательский контекст** (код и данные, находящиеся в адресном пространстве процесса)

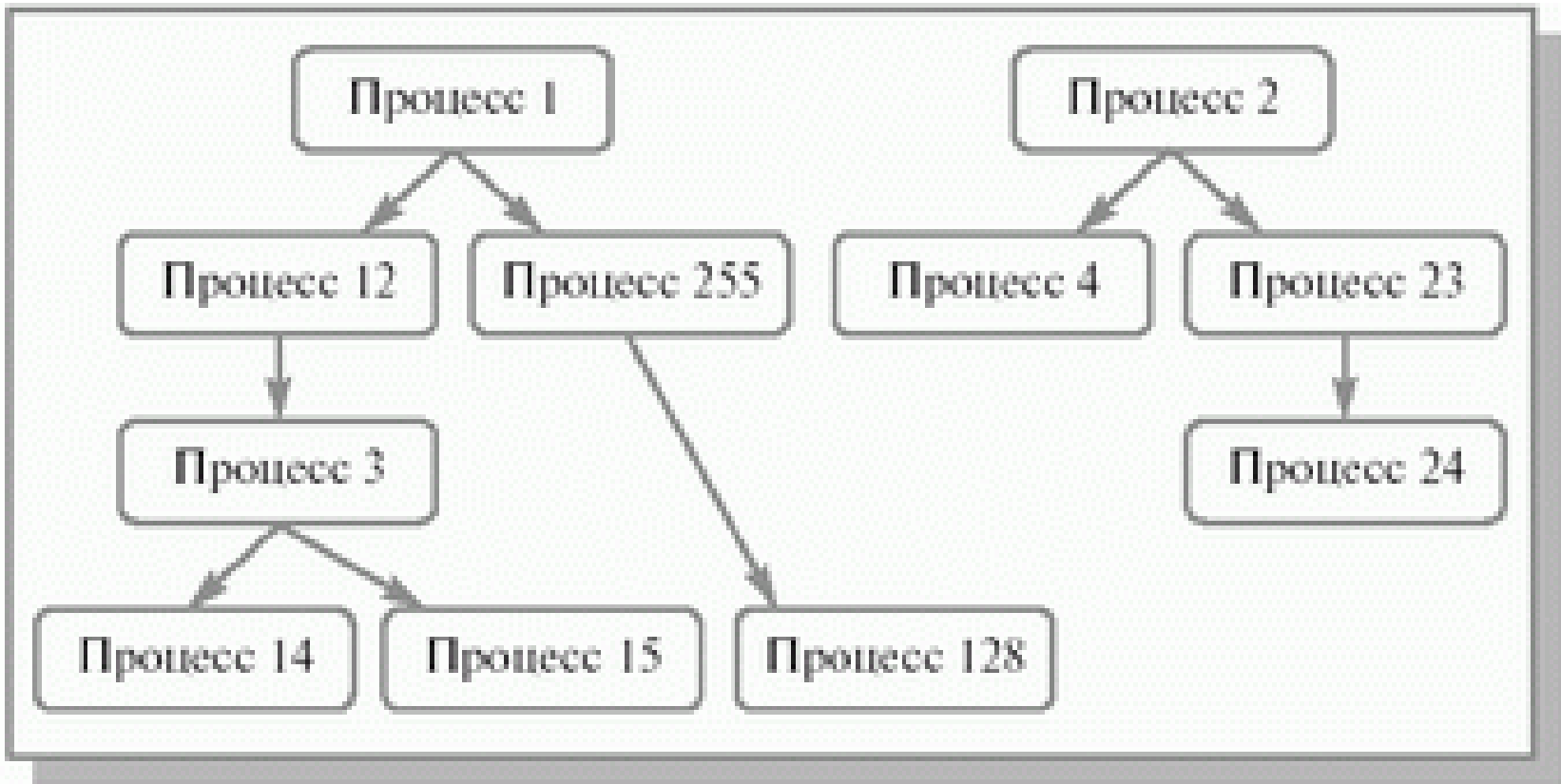
Совокупность регистрового, системного и пользовательского контекстов процесса для краткости принято называть просто контекстом процесса.

В любой момент времени процесс полностью характеризуется своим контекстом.

[Дерево процессов]

- Жизненный путь процесса в компьютере начинается с его рождения.
- Любая операционная система, поддерживающая концепцию процессов, должна обладать средствами для их создания. *В очень простых системах (например, в системах, спроектированных для работы только одного конкретного приложения) все процессы могут быть порождены на этапе старта системы.*
- Сложные операционные системы создают процессы динамически, по мере необходимости.
- Инициатором рождения нового процесса после старта операционной системы может выступить либо процесс пользователя, совершивший специальный системный вызов, либо сама операционная система.

Дерево процессов



[Дерево процессов]

- При рождении процесса система заводит новый РСВ с состоянием процесса рождение и начинает его заполнять.
- Новый процесс получает собственный уникальный идентификационный номер.
- После завершения какого-либо процесса его освободившийся идентификационный номер может быть повторно использован для другого процесса.

[Дерево процессов]

Существует два подхода к выделению ресурсов для процесса (ресурсов: памяти, файлов, устройств ввода-вывода и т. д.):

- новый процесс может получить в свое распоряжение некоторую часть родительских ресурсов, возможно разделяя с процессом-родителем и другими процессами-детьми права на них
- Новый процесс может получить свои ресурсы непосредственно от операционной системы.

Информация о выделенных ресурсах заносится в PCB.

[Дерево процессов]

После наделения процесса-потомка ресурсами необходимо занести в его адресное пространство программный код, значения данных, установить программный счетчик. Здесь также возможны два решения:

- Процесс-потомок становится дубликатом процесса-родителя по регистровому и пользовательскому контекстам, при этом должен существовать способ определения, кто для кого из процессов-двойников является родителем.
- Процесс-потомок загружается новой программой из какого-либо файла.

[Дерево процессов]

Порождение нового процесса как дубликата процесса-родителя приводит к возможности существования программ (т. е. исполняемых файлов), для работы которых организуется более одного процесса. Возможность замены пользовательского контекста процесса по ходу его работы (т. е. загрузки для исполнения новой программы) приводит к тому, что в рамках одного и того же процесса может последовательно выполняться несколько различных программ.

После того как процесс наделен содержанием, в PCB дописывается оставшаяся информация, и состояние нового процесса изменяется на готовность.

Процесс-родитель может продолжать свое выполнение одновременно с выполнением процесса-потомка, а может ожидать завершения работы некоторых или всех своих «потомков».

«Осиротевшие» процессы "усыновляются" одним из системных процессов, который порождается при старте операционной системы и функционирует все время, пока она работает.

Многообразные операции

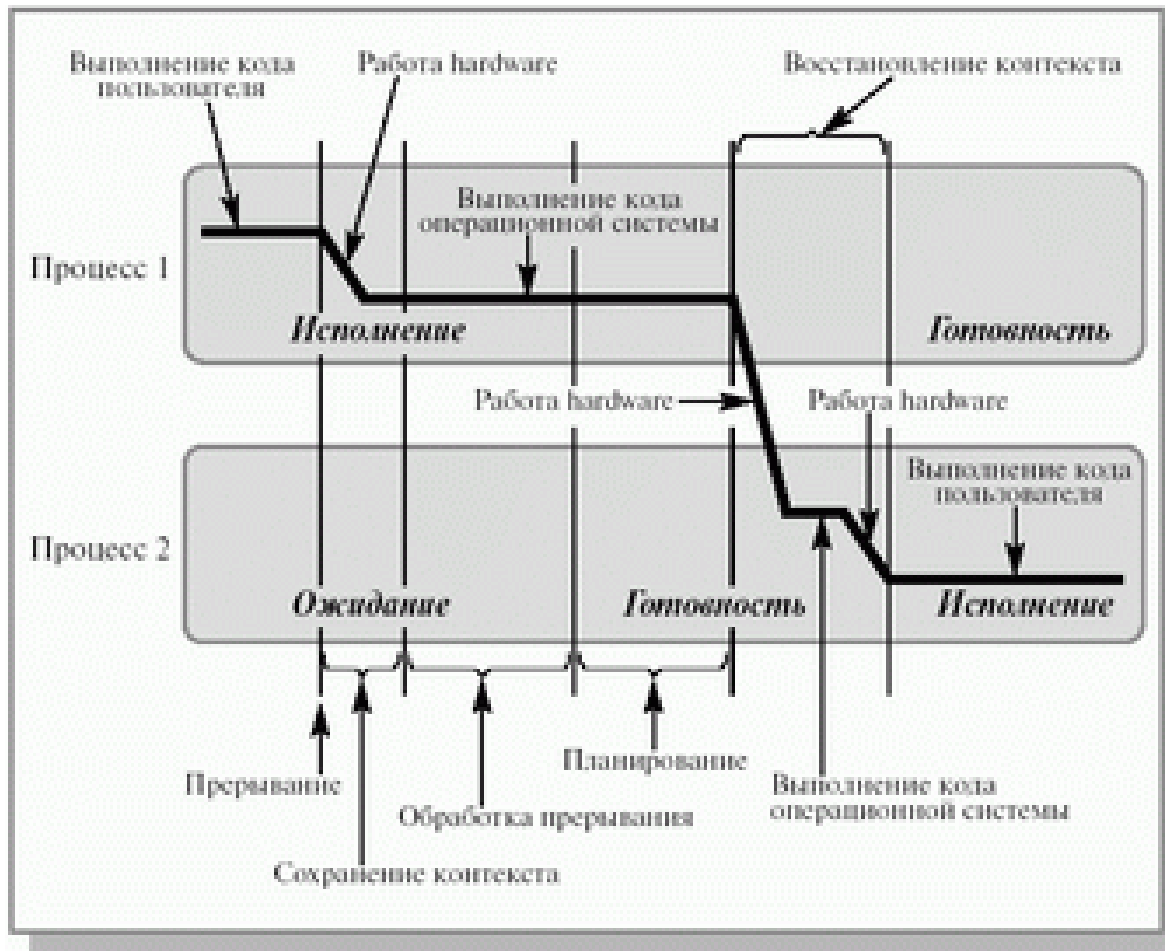
- Одноразовые операции приводят к изменению количества процессов, находящихся под управлением операционной системы, и всегда связаны с выделением или освобождением определенных ресурсов.
- Многообразные операции, напротив, не приводят к изменению количества процессов в операционной системе и не обязаны быть связанными с выделением или освобождением ресурсов.

Многообразные операции

Можно выделить основные действия, которые производит операционная система при выполнении многообразных операций над процессами:

- Запуск процесса
- Приостановка процесса
- Блокирование процесса
- Разблокирование процесса

Смена процесса



[Уровни планирования]

планировании заданий – процедура выбора задания для загрузки в ВС (изменяя порядок загрузки заданий в вычислительную систему, можно повысить эффективность ее использования)

планировании использования процессора (мультипрограмные ВС – выбор процесса для исполнения из состояния «готовность»)

Уровни планирования

- **Долгосрочное** – решение о выборе для запуска того или иного процесса оказывает влияние на функционирование вычислительной системы на протяжении достаточно длительного времени

планирование заданий используется в качестве долгосрочного планирования процессов (отвечает за порождение новых процессов в системе, определяя ее степень мультипрограммирования, т. е. количество процессов, одновременно находящихся в ней) – *используется редко*

- «-» - постоянная степень мультипрограммирования системы

- **Среднесрочное** - swapping (удаление частично выполненного процесса)
- **Краткосрочное** – выбор нового процесса для исполнения оказывает влияние на функционирование системы до наступления очередного аналогичного события, т. е. в течение короткого промежутка времени

для него используется планирование использования процессора. Оно проводится при обращении исполняющегося процесса к устройствам или по завершении определенного интервала времени

Критерии планирования

Для каждого уровня планирования можно разработать различные алгоритмы, которые определяются классами задач ВС и **целями ВС**:

- *Справедливость* (гарантирует каждому заданию или процессу определенную часть времени использования процессора)
- *Эффективность* (необходимость занять процессор на все 100% рабочего времени, не позволяя ему простаивать в ожидании процессов, готовых к исполнению)

Критерии планирования

- *Сокращение полного времени выполнения (turnaround time)* (обеспечение минимального времени между стартом процесса или постановкой задания в очередь для загрузки и его завершением)
- *Сокращение времени ожидания (waiting time)*
- *Сокращение времени отклика (response time)* (минимизировать время, которое требуется процессу в интерактивных системах для ответа на запрос пользователя)

Свойства алгоритмов планирования

Независимо от поставленных целей планирования желательно также, чтобы алгоритмы обладали следующими свойствами:

- *Предсказуемость* (одно и то же задание должно выполняться приблизительно за одно и то же время)
- *Минимальные накладные расходы*
- *Равномерная загрузка ресурсов* (отдача предпочтения тем процессам, которые будут занимать малоиспользуемые ресурсы)
- *Масштабируемость* (чтобы не сразу теряли работоспособность при увеличении нагрузки)

Параметры планирования

Для осуществления поставленных целей разумные алгоритмы планирования должны опираться на какие-либо характеристики процессов в системе, заданий в очереди на загрузку, состояния самой вычислительной системы. Все параметры планирования можно разбить на две большие группы:

■ Статические

- размер оперативной памяти
- максимальное количество памяти на диске для осуществления свопинга
- количество подключенных устройств ввода-вывода

■ Динамические - описывают количество свободных ресурсов на данный момент

Статические параметры процесса

К статическим параметрам процессов относятся характеристики, присущие заданиям уже на этапе загрузки:

- 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

Динамические параметры процессов на этапе загрузки заданий еще не известны

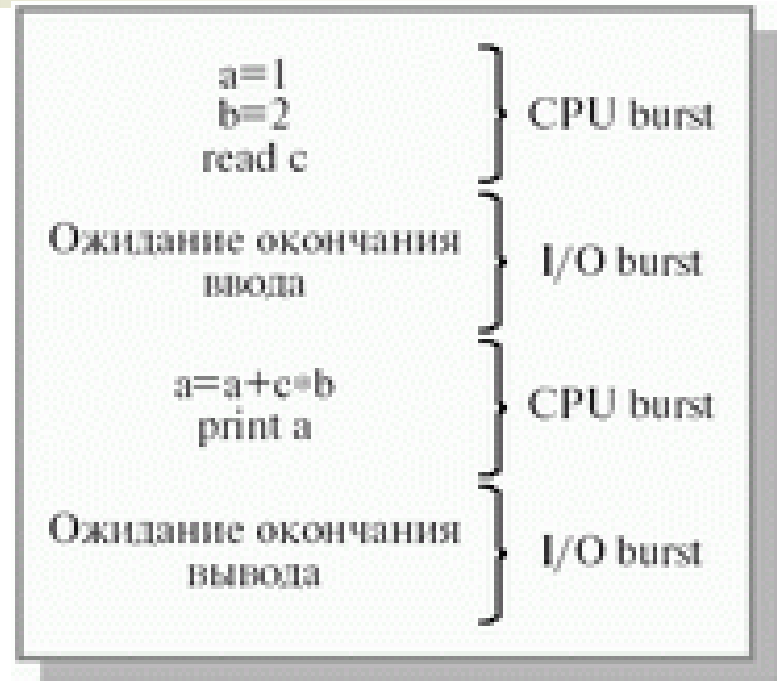
Алгоритмы планирования

- Алгоритмы долгосрочного планирования используют в своей работе статические и динамические параметры вычислительной системы и статические параметры процессов
- Для среднесрочного планирования дополнительно учитываются и динамические характеристики процессов:
 - сколько времени прошло с момента выгрузки процесса на диск или его загрузки в оперативную память;
 - сколько оперативной памяти занимает процесс;
 - сколько процессорного времени уже предоставлено процессу.

[Работа процесса]

Для краткосрочного планирования вводятся еще два динамических параметра:

- Промежуток времени непрерывного использования процессора носит название **CPU burst**
- Промежуток времени непрерывного ожидания ввода-вывода – **I/O burst**



Деятельность любого процесса можно представить как последовательность циклов использования процессора и ожидания завершения операций ввода-вывода

Время диспетчеризации

Процесс планирования осуществляется частью операционной системы, называемой планировщиком. Планировщик принимает решения в следующих случаях:

- Когда процесс переводится из состояния исполнение в состояние закончил исполнение.
- Когда процесс переводится из состояния исполнение в состояние ожидание.
- Когда процесс переводится из состояния исполнение в состояние готовность (например, после прерывания от таймера).
- Когда процесс переводится из состояния ожидание в состояние готовность (завершилась операция ввода-вывода или произошло другое событие).

Планирование

- невытесняющее (nonpreemptive) - если в операционной системе планирование осуществляется только в вынужденных ситуациях
 - MS Windows 3.1 и ОС Apple Macintosh
- + просто реализуем и достаточно эффективен, так как позволяет выделить большую часть процессорного времени для работы самих процессов и до минимума сократить затраты на переключение контекста
- проблема возможности полного захвата процессора одним процессом
- вытесняющее (preemptive) - если планировщик принимает и вынужденные, и невынужденные решения
 - MS Windows NT, UNIX

Планирование

- вытесняющее (preemptive) - если планировщик принимает и вынужденные, и невынужденные решения
 - MS Windows NT, UNIX
- + планирования процесс может быть приостановлен в любой момент исполнения.
- + временные прерывания помогают гарантировать приемлемое время отклика процессов для пользователей, работающих в диалоговом режиме, и предотвращают "зависание" компьютерной системы из-за зацикливания какой-либо программы.

[Алгоритмы планирования]

- First-Come, First-Served (FCFS)

[FCFS]

Процесс	p0	p1	p2	
Продолжительность очередного CPU burst	13	4	1	

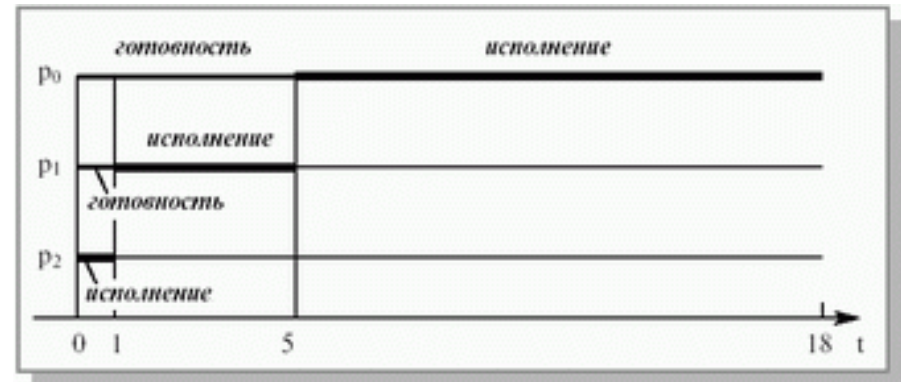
Алгоритмы планирования

- First-Come, First-Served (FCFS)

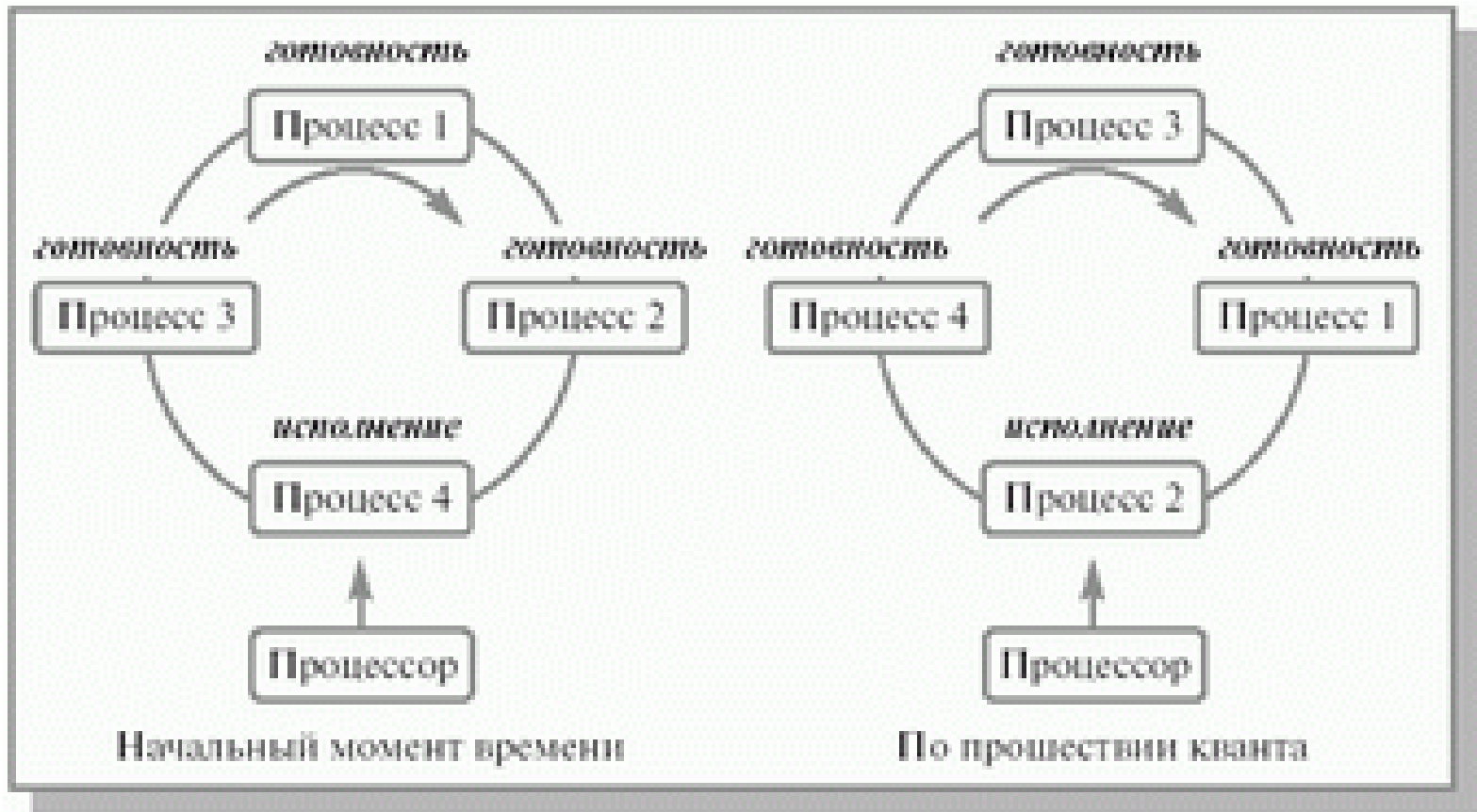


p0,p1,p2

p2, p1, p0



Round Robin (RR)



[RR



T=4

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
p0	И	И	И	И	Г	Г	Г	Г	Г	И	И	И	И	И	И	И	И	И	
p1	Г	Г	Г	Г	И	И	И	И											
p2	Г	Г	Г	Г	Г	Г	Г	Г	И										

T=1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
p0	И	Г	Г	И	Г	И	Г	И	Г	И	И	И	И	И	И	И	И	И	
p1	Г	И	Г	Г	И	Г	И	Г	И										
p2	Г	Г	И																

Shortest-Job-First (SJF)

Процесс	p0	p1	p2	p3	
Продолжительность очередного CPU burst	5	3	7	1	

Shortest-Job-First (SJF)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
p0	Г	Г	Г	Г	И	И	И	И	И								
p1	Г	И	И	И													
p2	Г	Г	Г	Г	Г	Г	Г	Г	Г	И	И	И	И	И	И	И	
p3	И																

Shortest-Job-First (SJF)

Процесс	Время появления в очереди очередного CPU burst	Продолжительно сть	
p0	0	6	
p1	2	2	
p2	6	7	
p3	0	5	

Shortest-Job-First (SJF)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
p0	Г	Г	Г	Г	Г	Г	Г	И	И	И	И	И									
p1			И	И																	
p2							Г	Г	Г	Г	Г	Г	Г	И	И	И	И	И	И	И	
p3	И	И	Г	Г	И	И	И														

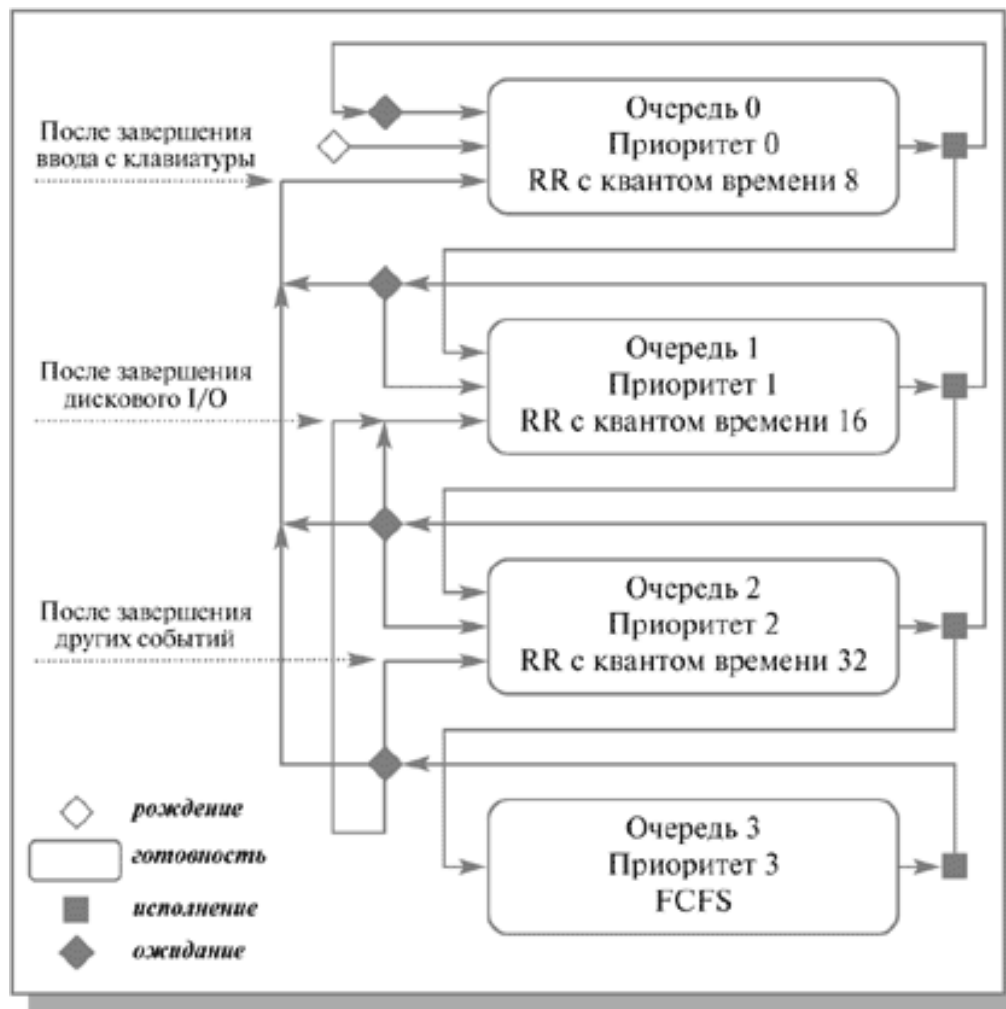
Гарантированное планирование

- $1/N$

Многоуровневые очереди (Multilevel Queue)



Многоуровневые очереди с обратной связью (Multilevel Feedback Queue)



Приоритет выполнения в режиме ядра Уровни приоритетов Процесс

