

Содержание

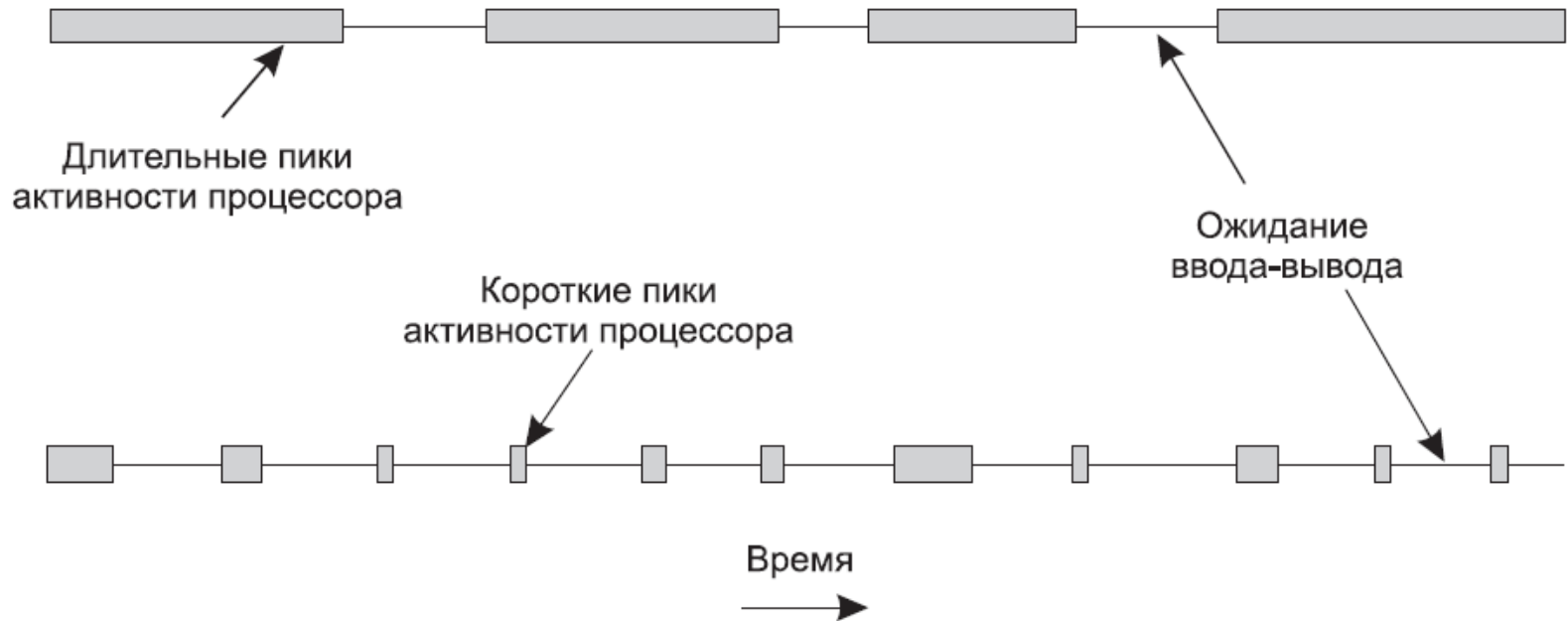
- 1. Основные понятия**
- 2. Планирование в пакетных системах**
- 3. Планирование в интерактивных системах**
- 4. Политика и механизмы планирования**
- 5. Планирование потоков**

1. Основные понятия

Планировщик и алгоритм планирования.

1) Поведение процесса:

- **ограниченная скорость вычислений,**
- **ограниченная скорость работы устройств ввода-вывода.**



2) Момент принятия решения планировщиком (когда планировать?):

- *при создании нового процесса* (какой выполнять, родительский или дочерний);
- *при завершении процесса;*
- *при блокировании процесса;*
- *при возникновении прерывания.*

Алгоритмы по реакции на прерывания по таймеру:

- *неприоритетный,*
- *приоритетный.*

3) Категории алгоритмов планирования по среде работы планировщика:

- *пакетная,*
- *интерактивная,*
- *реального времени.*

4) Задачи алгоритмов планирования:

Все системы:

- **равнодоступность** — предоставление каждому процессу справедливой доли времени процессора;
- **принуждение к определенной политике** — наблюдение за выполнением установленной политики;
- **баланс** — поддержка загрузки всех составных частей системы.

Пакетные системы:

- **производительность** — выполнение максимального количества заданий в час;
- **оборотное время** — минимизация времени между представлением задачи и ее завершением;
- **использование центрального процессора** — поддержка постоянной загрузки процессора.

Интерактивные системы:

- **время отклика** — быстрый ответ на запросы;
- **пропорциональность** — оправдание пользовательских надежд.

Системы реального времени:

- **соблюдение предельных сроков** — предотвращение потери данных;
- **предсказуемость** — предотвращение ухудшения качества в мультимедийных системах.

Определения:

Производительность — это количество заданий, выполненных за один час.

Оборотное время — это среднестатистическое время от момента передачи задания на выполнение до момента завершения его выполнения.

Время отклика — это время между выдачей команды и получением результата.

2. Планирование в пакетных системах

Алгоритмы планирования:

- 1) **«Первым пришел — первым обслужен»**, очередь готовых процессов.
- 2) **«Сначала самое короткое задание»** (если сроки выполнения известны заранее).

Пример 1. Все процессы готовы к выполнению одновременно.

Время выполнения	8	4	4	4	4	4	4	8
Процесс	A	B	C	D	B	C	D	A
Оборотное время	8	12	16	20	4	8	12	20
Среднее время	14,0				11,0			

Пример 2. Если время поступления процессов различное, то алгоритм перестает быть оптимальным.

Время выполнения	2	4	1	1	1	4	1	1	1	2
Процесс	A	B	C	D	E	B	C	D	E	A
Время поступления	0	0	3	3	3	0	3	3	3	0
Оборотное время	2	6	7	8	9	4	5	6	7	9
Среднее время	4,6					4,4				

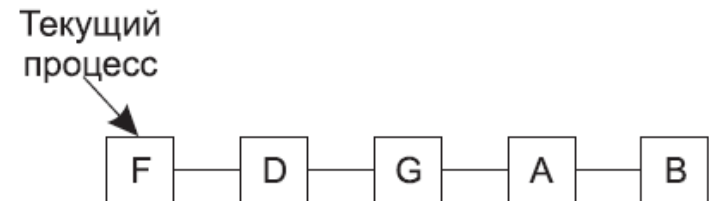
- 3) **«Приоритет наименьшему времени выполнения»** (если сроки выполнения известны заранее).

3. Планирование в интерактивных системах

Алгоритмы планирования:

1) **Циклическое планирование** – каждому процессу назначается одинаковый **квант** времени.

Очередь готовых процессов:



Продолжительность кванта процессорного времени зависит от времени **переключение контекста** и от среднего времени выполнения процессов.

2) **Приоритетное планирование** – запускается тот процесс, который находится в состоянии готовности и имеет наивысший приоритет.

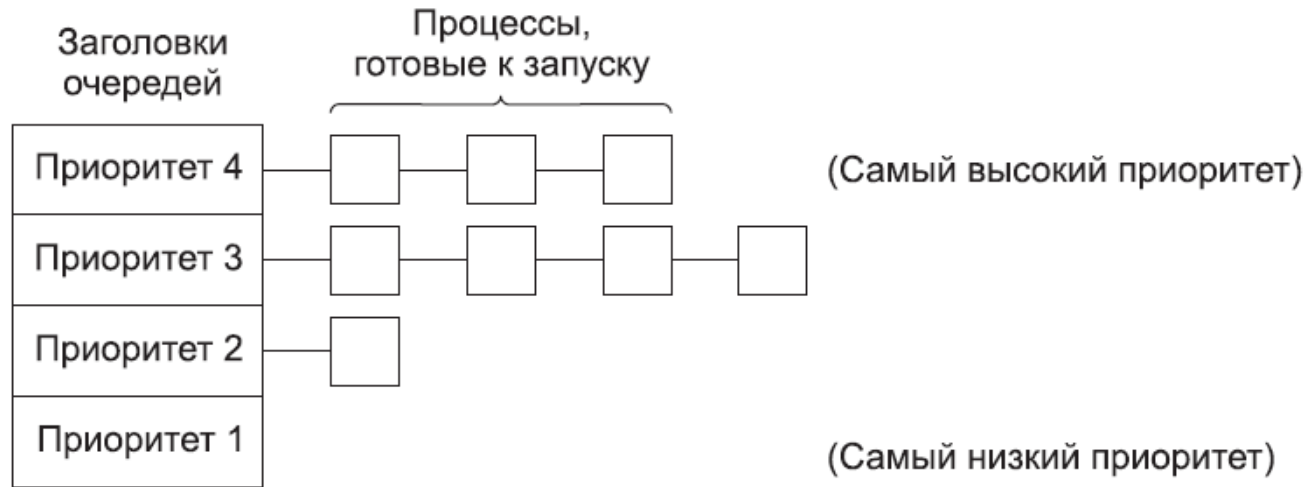
Чтобы предотвратить бесконечное выполнение высокоприоритетных процессов:

- понижать уровень приоритета текущего выполняемого процесса с каждым прерыванием по таймеру;
- выделить каждому процессу максимальный квант допустимого времени выполнения.

Приоритеты могут присваиваться процессам в динамическом режиме. Простой алгоритм успешного обслуживания процессов (ограниченных, например, низкой скоростью работы устройств ввода-вывода) — установка значения приоритета в $1/f$, где f — это часть последнего кванта времени, использованного этим процессом.

3) **Использование нескольких очередей.**

Процессы группируются по классам приоритетности, приоритетное планирование применяется к этим классам, а внутри каждого класса используется циклическое планирование.



4) **Выбор следующим самого короткого процесса.** Оценка предыдущего поведения и запуск процесса с самым коротким вычисленным временем выполнения.

5) **Гарантированное планирование.** Если имеется n работающих процессов, то каждому процессу гарантируется доля $1/n$ от общего процессорного времени.

Планировщик отслеживает, сколько процессорного времени затрачивает каждый процесс, и вычисляет соотношение израсходованного и отпущенного времени. Запускается процесс с самым низким соотношением, который будет работать до тех пор, пока его соотношение не превысит соотношение его ближайшего конкурента.

6) **Лотерейное планирование.** Процессам раздаются «лотерейные билеты» на доступ к процессорному времени.

Планировщик в случайном порядке выбирается лотерейный билет, и время отдается процессу, обладающему этим билетом. Важным процессам, чтобы повысить их шансы на выигрыш, могут выдаваться дополнительные билеты. Взаимодействующие процессы могут по желанию обмениваться билетами.

Лотерейное планирование очень быстро реагирует на изменение обстановки. Оно может быть использовано, когда трудно справиться другими методами.

7) **Справедливое планирование** учитывает, кто является владельцем процесса. Каждому пользователю распределяется некоторая доля процессорного времени.

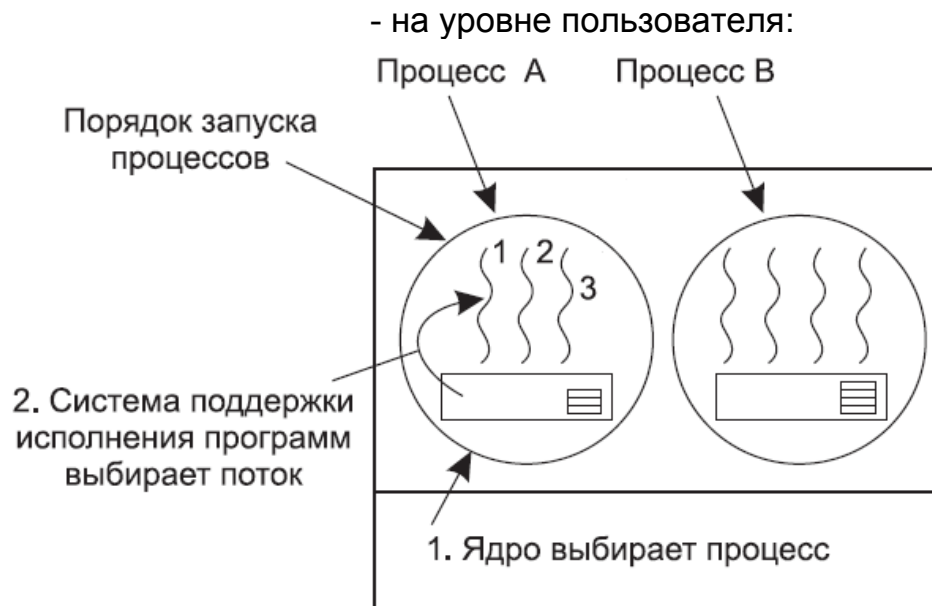
4. Политика и механизмы планирования

Принцип разделения **механизма** и **политики** планирования.

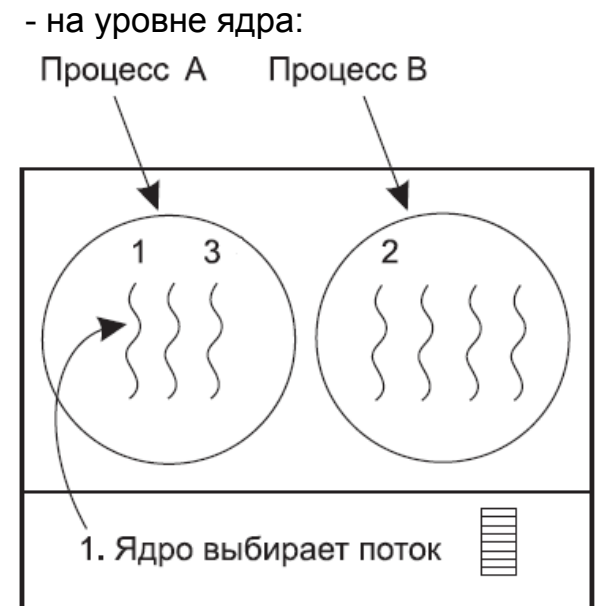
Пример. Предположим, что ядро ОС применяет алгоритм приоритетного планирования, но предоставляет системный вызов, с помощью которого процесс может установить (или изменить) приоритеты своих дочерних процессов. Таким образом родительский процесс может управлять порядком планирования работы дочерних процессов, хотя он сам планированием не занимается. **Механизм** находится в ядре, а **политика** устанавливается пользовательским процессом.

5. Планирование потоков

Планирование потоков



Возможно: A1, A2, A3, A1, A2, A3
Невозможно: A1, B1, A2, B2, A3, B3



Возможно: A1, A2, A3, A1, A2, A3
Также возможно: A1, B1, A2, B2, A3, B3

- На уровне пользователя.

Алгоритм планирования, используемый системой поддержки исполнения программ, – на практике циклическое и приоритетное планирование. Отсутствие таймерного прерывания в отношении потока, выполняемого слишком долго, обычно не вызывает проблем, поскольку потоки взаимодействуют.