

Операционные системы

Лекция № 5

Планирование

(1 час)

Понятие планирования

Одна из задач ОС – распределение ресурсов между множеством процессов с потенциально конкурирующими требованиями.

При распределении времени выполнения (процессора) распределение именуется **планированием (scheduling)**

Понятия планировщика и алгоритма планирования

Планировщиком (scheduler) называется часть ОС, отвечающая за выбор очередного процесса для исполнения.

Алгоритм планирования – алгоритм, используемый для выбора очередного процесса для исполнения.

Виды планировщиков

- **Долгосрочный (верхнего уровня) планировщик** – выбирает набор процессов для эффективного использования ресурсов системы;
- **Краткосрочный (нижнего уровня) (динамический, диспетчер (dispatcher))** – занимается выбором очередного процесса из очереди готовых для передачи на исполнение процессору;

Приоритет

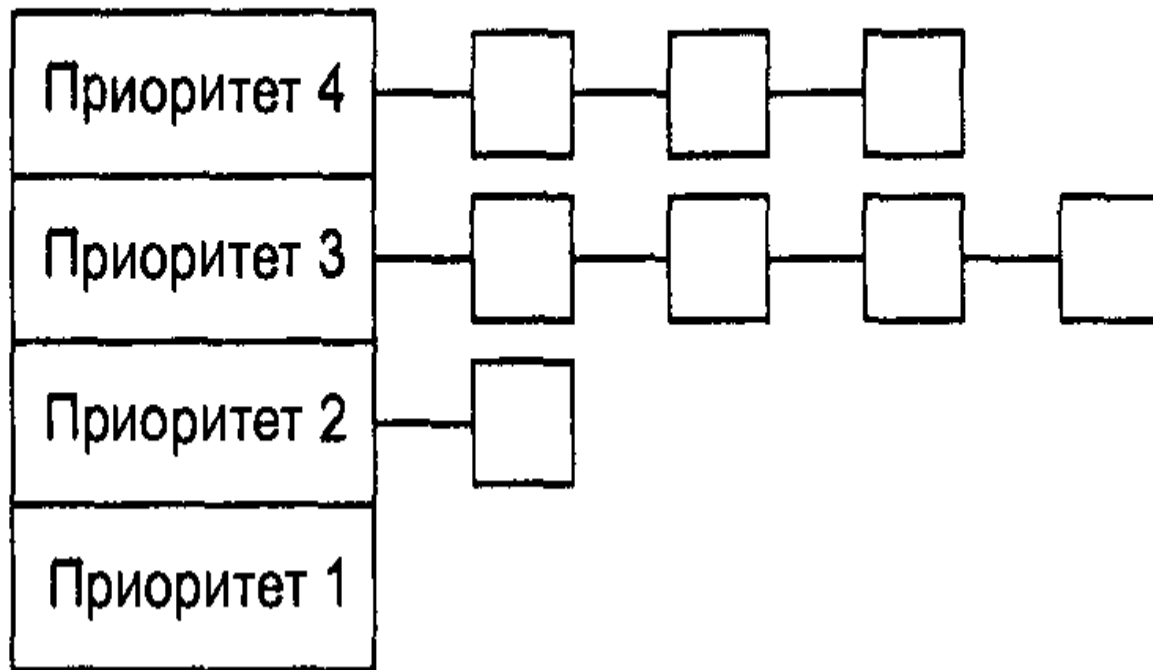
Для задания степени важности процесса ему присваивается **приоритет (priority)**– некоторое значение, согласно которому диспетчер должен выбирать тот процесс, приоритет которого наибольший.

Приоритеты:

- **Статические** (неизменяемые со временем);
- **Динамические** (изменяемые во время выполнения процесса).

Заголовки
очередей

Процессы,
готовые к запуску

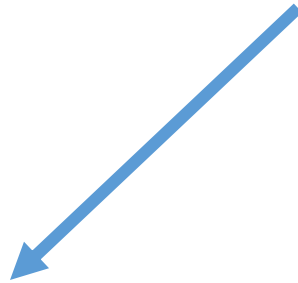


(Самый высокий приоритет)

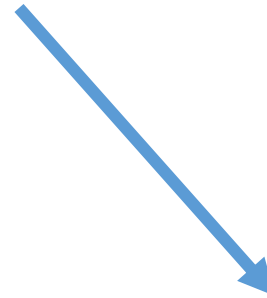
(Самый низкий приоритет)

Категории алгоритмов планирования

Алгоритмы планирования



Невытесняющие
(non-preemptive)



Вытесняющие
(preemptive)

Невытесняющие алгоритмы

(алгоритмы планирования без переключений) – алгоритмы планирования, при которых выполняющийся процесс продолжает выполняться до тех пор, пока не завершится или не окажется в заблокированном состоянии.

Вытесняющие алгоритмы

(алгоритмы планирования **с переключениями**)

Выполняющийся процесс может быть прерван и переведен ОС в состояние готовности к выполнению. Решение о вытеснении может приниматься:

- при запуске нового процесса,
- по прерыванию, которое переводит заблокированный процесс в состояние готовности к выполнению
- периодически – по интервальному таймеру.

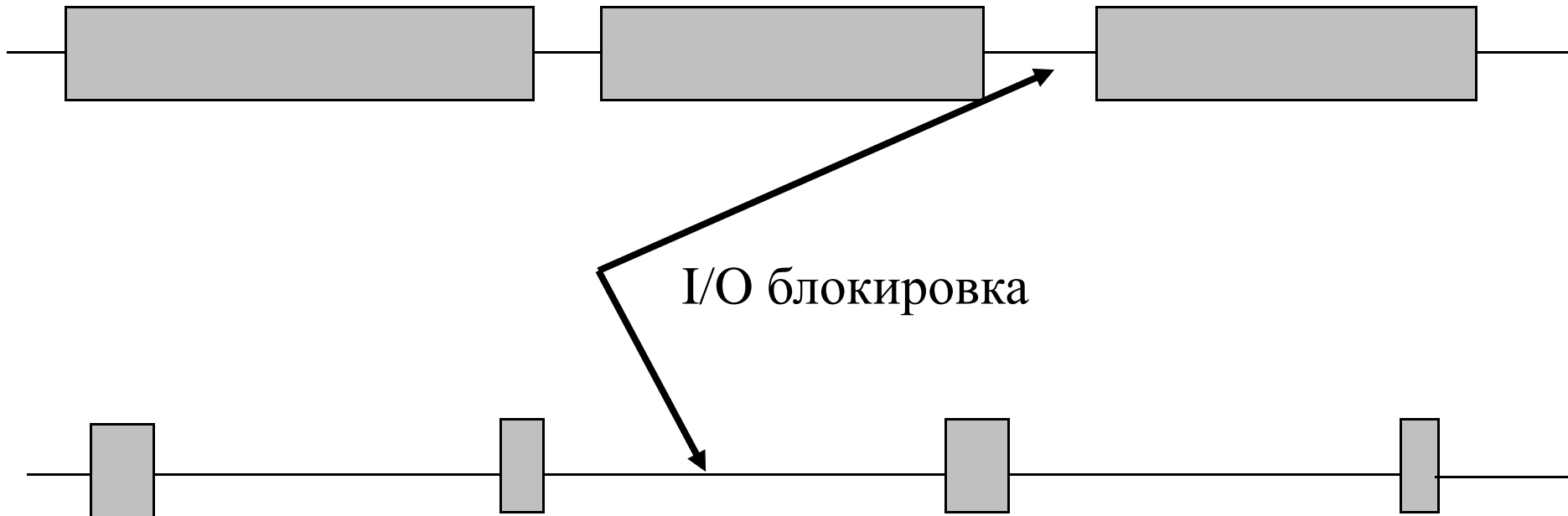
Поведение процессов

Процессы, большую часть времени исполнения занятые вычислениями называются **процессами ограниченными возможностями процессора (processor-bound)**

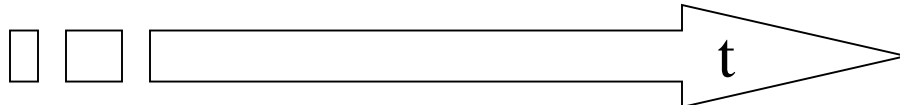
Процессы, большую часть времени ожидающие ввода-вывода называются **процессами ограниченными возможностями устройств ввода-вывода (I/O-bound)**

Поведение процессов

Процесс, ограниченный возможностями CPU



Процесс, ограниченный возможностями ввода-вывода



Ситуации планирования

- При прерывании от таймера;
- При прерывании I/O;
- Вызовы ОС;
- При завершении или создании процесса;
- При блокировании процесса (семафором или I/O).

Категории алгоритмов планирования

По типу ОС:

- Системы пакетной обработки данных;
- Интерактивные системы;
- Системы реального времени.

Задачи алгоритмов планирования (все системы)

- **Справедливость** — предоставление каждому процессу справедливой доли процессорного времени;
- **Предсказуемость** — процесс должен выполняться приблизительно за одно и то же время при приблизительно одинаковой нагрузке на систему;
- **Баланс** — поддержка занятости всех частей системы;
- **Масштабируемость** — механизм планирования не должен терять работоспособность под большой нагрузкой

Задачи алгоритмов планирования

- **Максимальная пропускная способность** — максимальное количество процессов в единицу времени;
- **Гарантированность приемлемого времени отклика** — для максимального количества пользователей, работающих в интерактивном режиме;
- **Максимальное использование ресурсов;**
- **Минимизация накладных расходов;**

Задачи алгоритмов планирования (системы реального времени)

- **Окончание работы к сроку** — предотвращение потери данных

Алгоритмы планирования

1.FCFS - «Первым пришел – первым обслужен»

(first-come-first-served – FCFS)

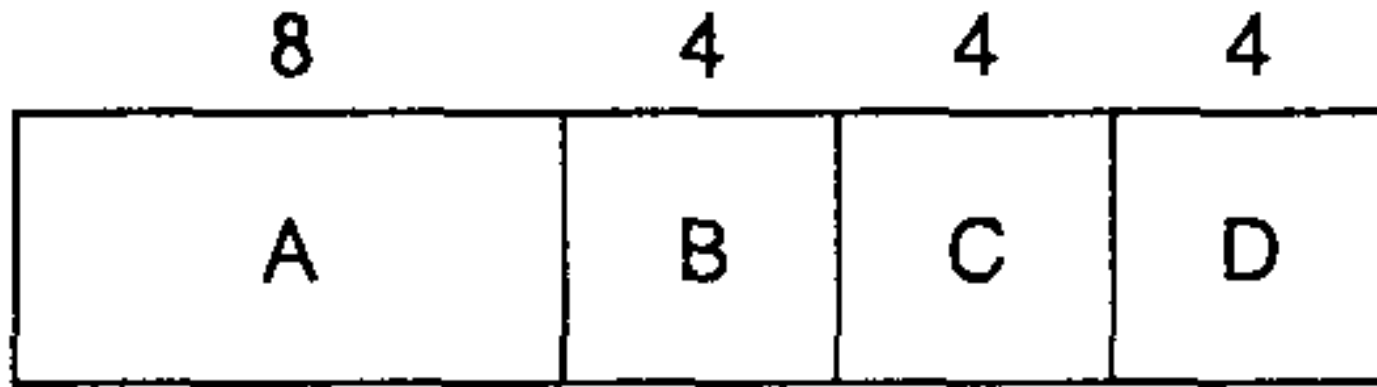
- + : легко понять, легко программировать
- : преимущество процессов, ориентированных на работу с процессором над процессами, ориентированными на ввод-вывод

2. «Кратчайшая задача – первая»

SPN – (стратегия выбора самого короткого процесса (shortest process next – SPN или shortest job next - SJN))

+ : хорошее время отклика для коротких процессов;

- : плохая производительность длительных процессов, низкое время отклика.

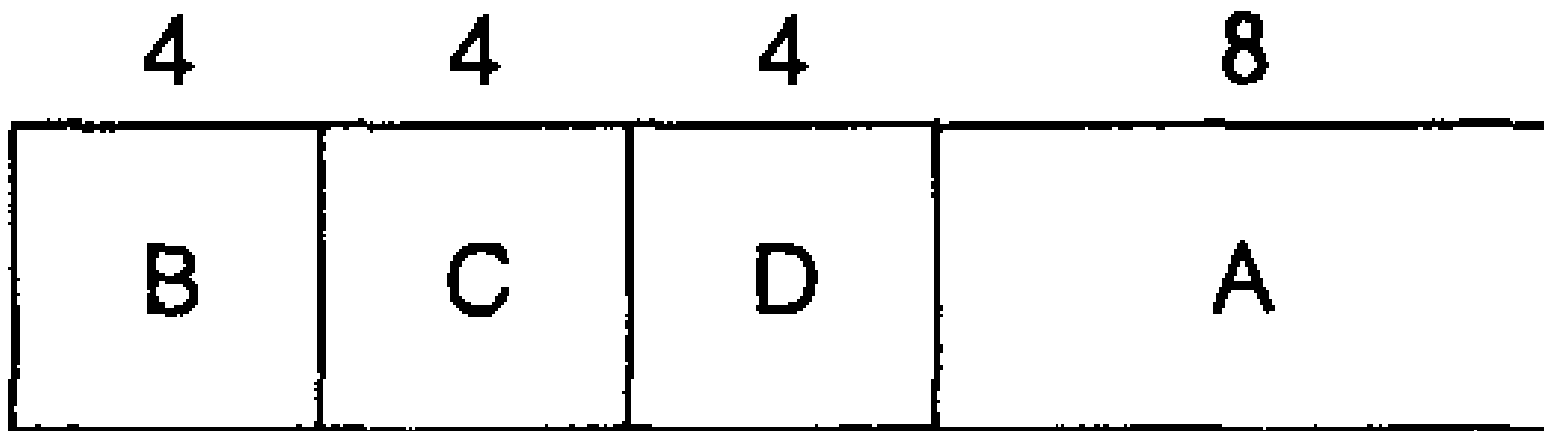


a

В FCFS:

Оборотное время задач: A – 8, B – 12, C – 16, D – 20

Среднее – 14 минут



б

В SPN:

Оборотное время задач: В – 4, С – 8, D – 12, А – 20

Среднее – 11 минут

Общая схема

Даны 4 процесса со временем выполнения a, b, c, d.

Тогда среднее оборотное время

$$(4a+3b+2c+d)/4$$

Вклад a в результирующее время - максимальный

3. Наименьшее остающееся время выполнения

SRT – (shortest remaining time)

Вытесняющая версия алгоритма SPN.

Планировщик выбирает процесс с наименьшим оставшимся временем выполнения. При поступлении новой задачи вызывается перепланирование.

+ : хорошее время отклика;

- : низкая производительность длительных процессов;

Планирование в интерактивных системах

1. RR – Циклическое планирование

(круговое или карусельное планирование, **round robin** – RR)

+ : минимальные накладные расходы; хорошее время отклика; беспристрастность к длине процессов;

- : низкая пропускная способность при малом кванте времени; процесс, ограниченный I/O получает меньше времени.

Текущий
процесс

Следующий
процесс



a

Текущий
процесс



б

Каждому процессу предоставляется некоторый интервал времени процессора - **квант** времени. По его истечению таймер генерирует прерывание. Если к концу кванта времени процесс все еще работает, он приостанавливается, а управление передается другому процессу.

Эта методика известна также как **квантование времени** (time slicing).

Модификации RR

Основной недостаток RR: процесс, ограниченный устройствами I/O получает меньше времени, чем процесс ограниченный возможностями CPU.

Одно из решений: **VRR** (virtual round robin) – наличие вспомогательной очереди, в которую переносятся процессы после окончания I/O с преимуществом над очередью готовых процессов.

Планирование в Windows

- Единица планирования – поток;
- Каждый поток имеет собственный приоритет;
- Существует **32 приоритета**: от 0 (min) до 31 (max). (0 приоритет имеет только 1 поток – поток обнуления страниц);
- все потоки с одинаковым приоритетом – равноправны;
- ОС выделяет квант времени по алгоритму RR потоку с наивысшим приоритетом;

Планирование в Windows

- Если готовых потоков с наивысшим приоритетом нет, то ОС выделяет квант времени потоку, следующему за наивысшим по приоритету (и т.д.);
- Если более приоритетный поток переходит в состояние готовности, то ОС прекращает выполнение текущего потока (не дожидаясь окончания его кванта времени) и запускает на полный квант времени высокоприоритетный.

Планирование в Windows

Приоритет любого потока определяется двумя критериями:

- **Классом** приоритета его процесса;
- **Уровнем** приоритета потока внутри класса приоритета его процесса.

Класс приоритета и уровень приоритета задают базовый приоритет потока.

Планирование в Windows

Существует 6 классов приоритетов:

- `IDLE_PRIORITY_CLASS`
- `BELOW_NORMAL_PRIORITY_CLASS`
- `NORMAL_PRIORITY_CLASS` (по умолчанию)
- `ABOVE_NORMAL_PRIORITY_CLASS`
- `HIGH_PRIORITY_CLASS`
- `REALTIME_PRIORITY_CLASS`

Планирование в Windows

Класс приоритета можно задать:

- При создании процесса `CreateProcess`;
- После создания процесса `SetPriorityClass`;

```
BOOL SetPriorityClass(HANDLE hProcess,  
                     DWORD dwPriorityClass);
```

Получить текущий класс приоритета можно с помощью:

```
DWORD GetPriorityClass(HANDLE  
                      hProcess);
```

Планирование в Windows

- В рамках каждого класса есть следующие уровни:

THREAD_PRIORITY_IDLE

THREAD_PRIORITY_LOWEST

THREAD_PRIORITY_BELOW_NORMAL

THREAD_PRIORITY_NORMAL (по умолчанию)

THREAD_PRIORITY_ABOVE_NORMAL

THREAD_PRIORITY_HIGHEST

THREAD_PRIORITY_TIME_CRITICAL

Планирование в Windows

Изменение уровня **приоритета потока**
относительно других потоков процесса:

```
BOOL SetThreadPriority(HANDLE hThread,  
    int nPriority);
```

(дескриптор должен иметь права
THREAD_SET_INFORMATION).

Получение уровня приоритета:

```
int GetThreadPriority(HANDLE hThread);
```

Планирование в Windows

Для избегания проблемы **голодания** потоков ОС использует динамическое повышение приоритета.

Повышение приоритета выполняется только для потоков с базовым приоритетом от 0 до 15.

Основные причины динамического повышения:

1. Для процесса, окно которого активно в данный момент.
2. Когда окно получает ввод.
3. Когда условие ожидания для заблокированного потока выполняется.

Действующий в данный момент приоритет называется **текущим**.

Планирование в системах реального времени

Если в систему поступает m периодических событий, событие с номером i поступает с периодом P_i и на его обработку уходит C_i секунд работы процессора, все потоки могут быть своевременно обработаны только при выполнении условия

$$\sum_{i=1}^m \frac{C_i}{P_i} \leq 1.$$

Системы реального времени, удовлетворяющие этому условию, называются **поддающимися планированию** или **планируемыми**.

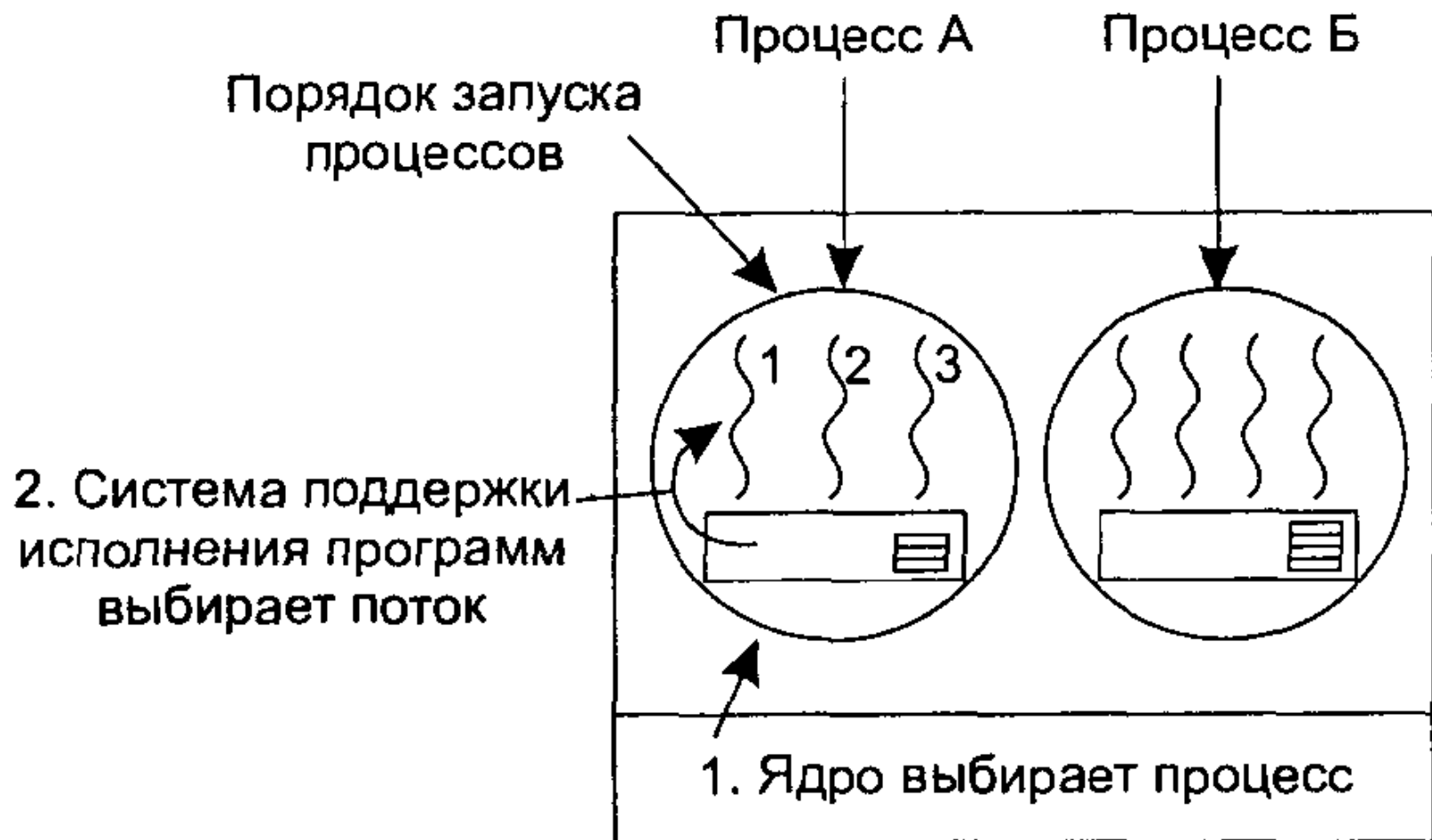
Алгоритмы планирования для систем реального времени

- ***Статические*** – все решения планирования принимаются заранее, до запуска системы.
- ***Динамические.***

Политика и механизм планирования

Используется параметризованный алгоритм планирования, параметры которого могут быть изменены с помощью специального системного вызова процессом пользовательского уровня.

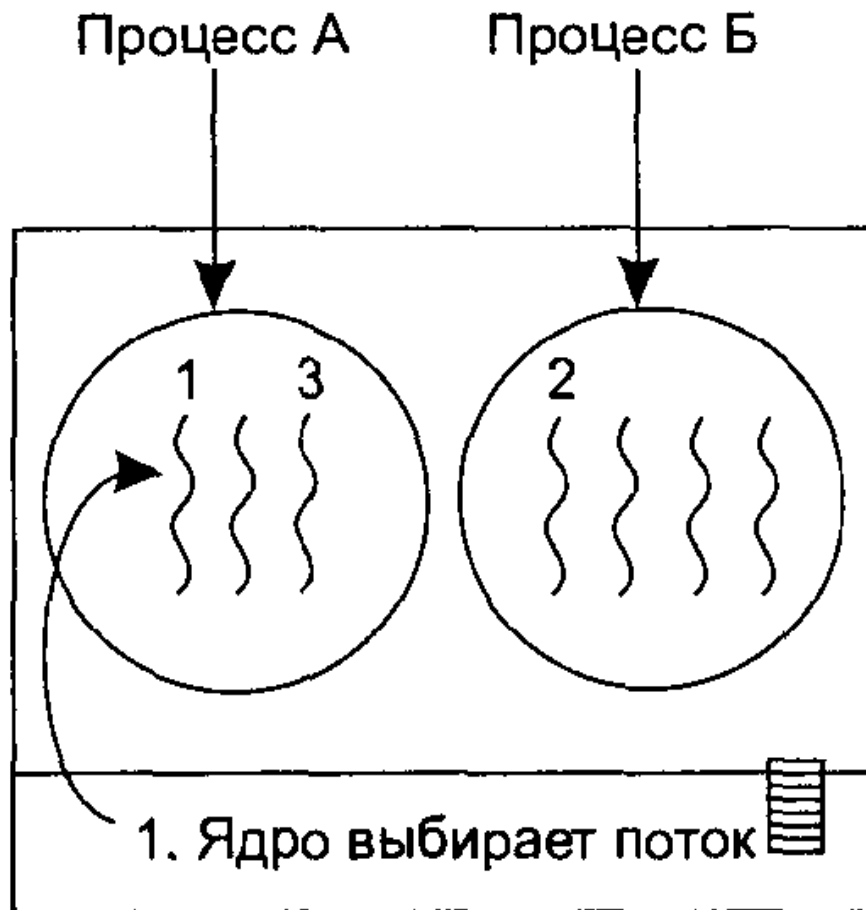
Планирование потоков (ULT)



Возможно: A1, A2, A3, A1, A2, A3

Невозможно: A1, B1, A2, B2, A3, B3

Планирование потоков (KLT)



Возможно: A1, A2, A3, A1, A2, A3

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

Планирование в многопроцессорных системах

- Назначение процессов процессорам;
 - а) ведущий/ведомый процессоры;
 - б) равноправные процессоры;
- Использование многозадачности на отдельном процессоре;
- Диспетчеризация процесса;

Планирование потоков в многопроцессорной системе

- **Разделение загрузки.** Поддерживается глобальная очередь готовых к выполнению потоков, и каждый процессор выбирает поток из этой очереди
- **Бригадное планирование (gang scheduling или групповое (group)).** Множество связанных потоков распределяются для одновременной работы среди множества процессоров, по одному потоку на процессор

Планирование потоков в многопроцессорной системе

- **Назначение процессоров.** Назначение потокам процессоров. Каждой программе на время ее выполнения выделяется количество процессоров, равное количеству потоков программы. По окончании работы программы, процессоры возвращаются в общий пул для назначения другим программам