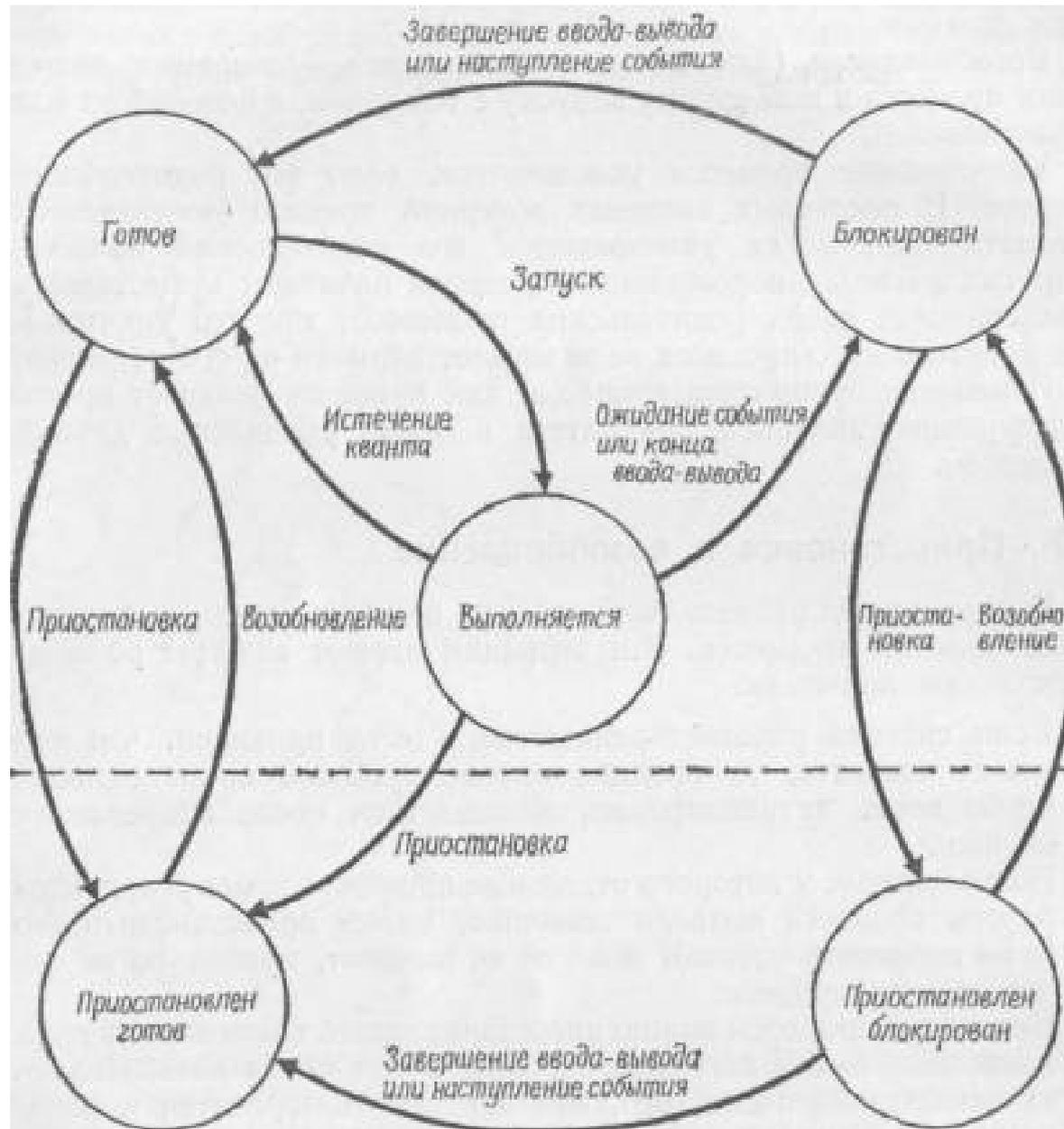


ТВП

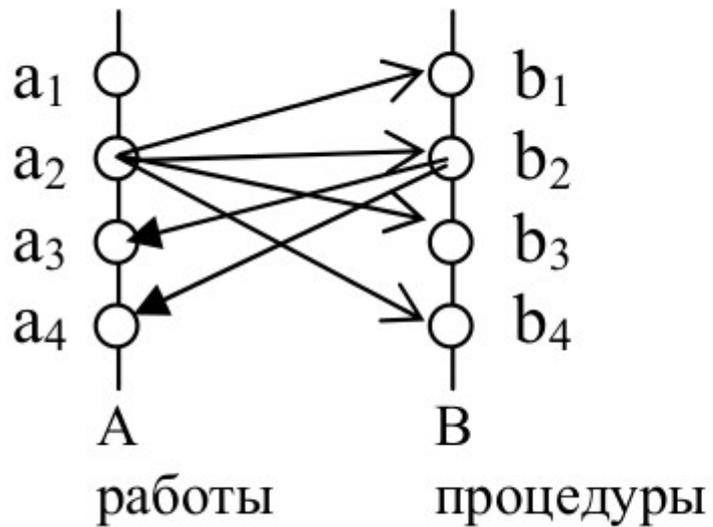
Процессы. Принципы работы

Лекция №4
(версия 1.0)

Приостановка и возобновление



Сложные взаимодействия процессов



$$S \subseteq A \times B$$

$$R(S|a_2) = \{b_1, b_2, b_3, b_4\}$$

$$R(S|b_2) = \{a_3, a_4\}$$

Уровни параллелизма

1. Уровень заданий

- между заданиями
- между фазами задания

2. Программный уровень

- между частями программы
- в пределах цикла ПО

3. Командный уровень

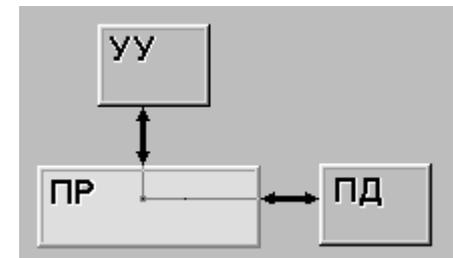
- между фазами выполнения команды
- между отдельными командами

4. Арифметический и разрядный уровень

- между элементами векторной операции
- внутри логических схем арифм. уст-ва.

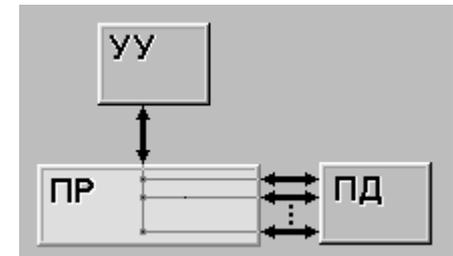
Классификация Флинна. SISD

SISD (single instruction stream / single data stream) - одиночный поток команд и одиночный поток данных. К этому классу относятся, прежде всего, классические последовательные машины, или иначе, машины фон-неймановского типа, например, PDP-11 или VAX 11/780. В таких машинах есть только один поток команд, все команды обрабатываются последовательно друг за другом и каждая команда инициирует одну операцию с одним потоком данных. Не имеет значения тот факт, что для увеличения скорости обработки команд и скорости выполнения арифметических операций может применяться конвейерная обработка - как машина CDC 6600 со скалярными функциональными устройствами, так и CDC 7600 с конвейерными попадают в этот класс.



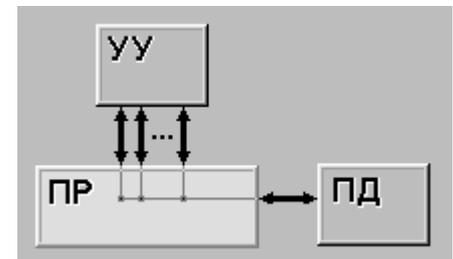
Классификация Флинна. SIMD

SIMD (single instruction stream / multiple data stream) - одиночный поток команд и множественный поток данных. В архитектурах подобного рода сохраняется один поток команд, включающий, в отличие от предыдущего класса, векторные команды. Это позволяет выполнять одну арифметическую операцию сразу над многими данными - элементами вектора. Способ выполнения векторных операций не оговаривается, поэтому обработка элементов вектора может производиться либо процессорной матрицей, как в ILLIAC IV, либо с помощью конвейера, как, например, в машине CRAY-1.



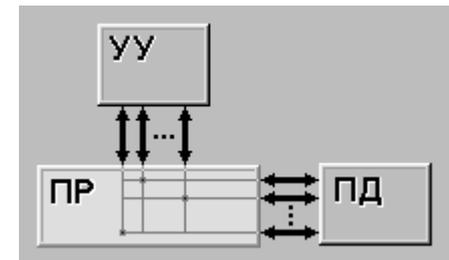
Классификация Флинна. MISD

MISD (multiple instruction stream / single data stream) — множественный поток команд и одиночный поток данных. Определение подразумевает наличие в архитектуре многих процессоров, обрабатывающих один и тот же поток данных. Однако ни Флинн, ни другие специалисты в области архитектуры компьютеров до сих пор не смогли представить убедительный пример реально существующей вычислительной системы, построенной на данном принципе. Ряд исследователей относят конвейерные машины к данному классу, однако это не нашло окончательного признания в научном сообществе.

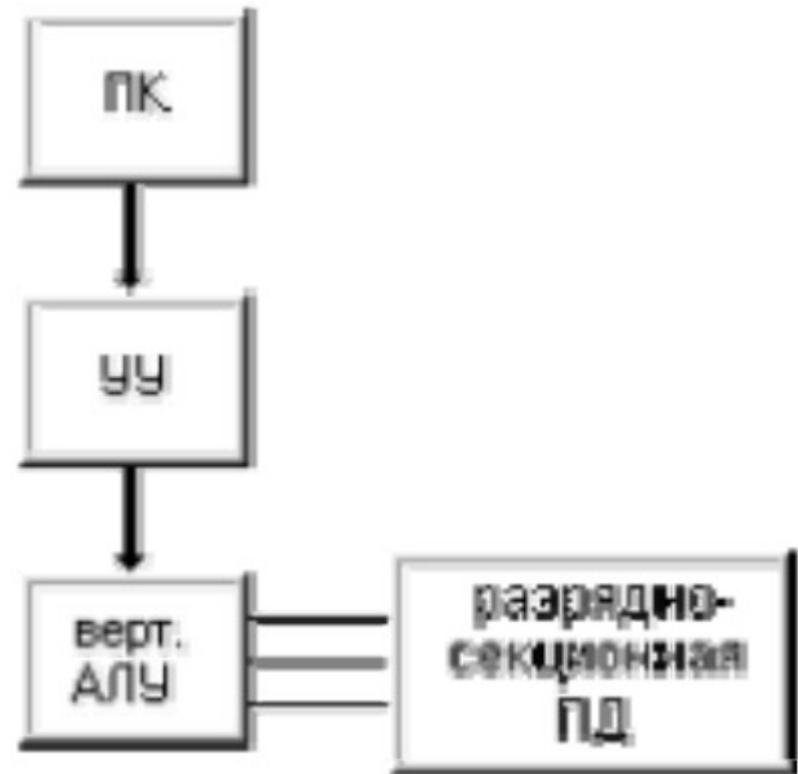


Классификация Флинна. MIMD

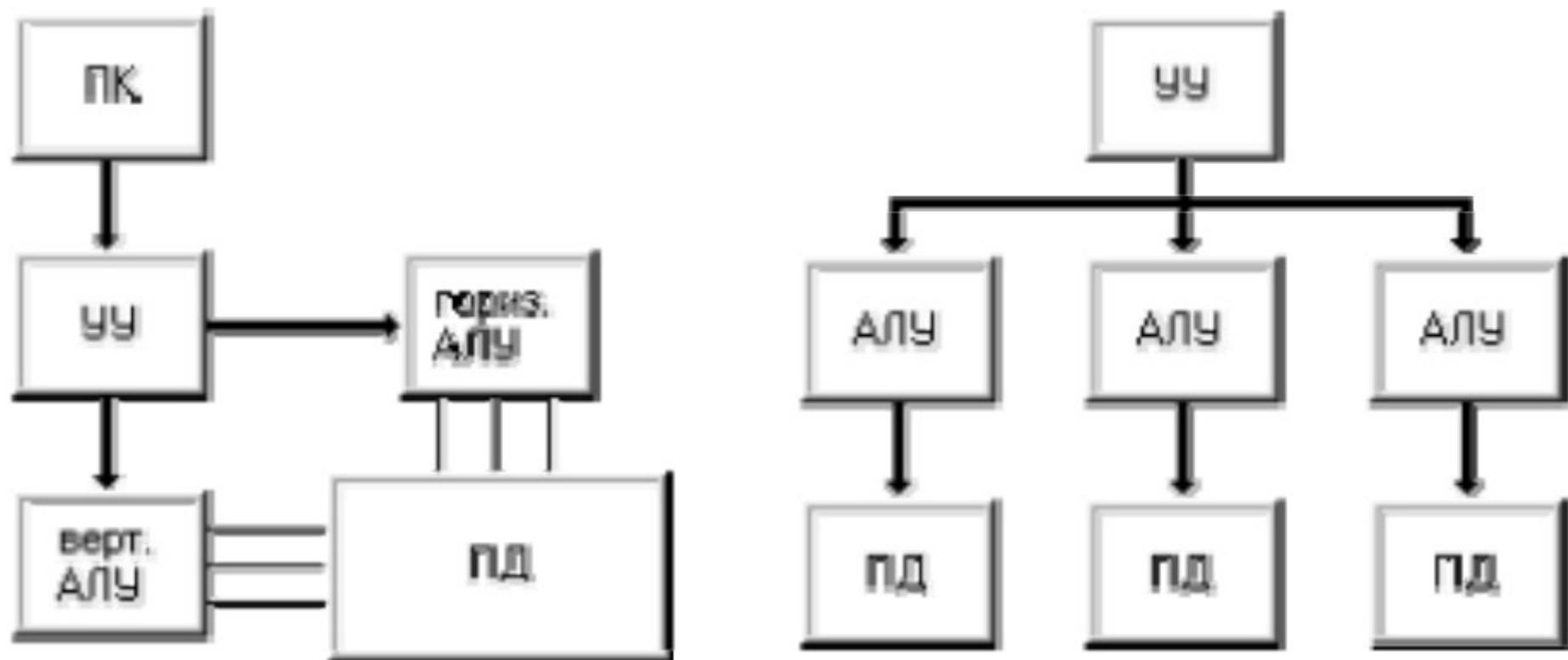
MIMD (multiple instruction stream / multiple data stream) - множественный поток команд и множественный поток данных. Этот класс предполагает, что в вычислительной системе есть несколько устройств обработки команд, объединенных в единый комплекс и работающих каждое со своим потоком команд и данных.



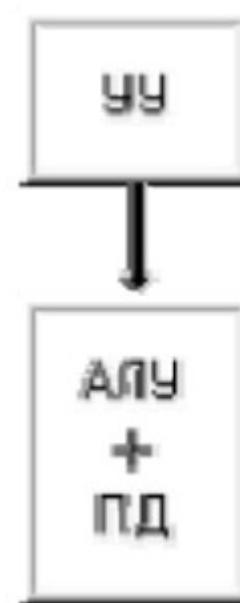
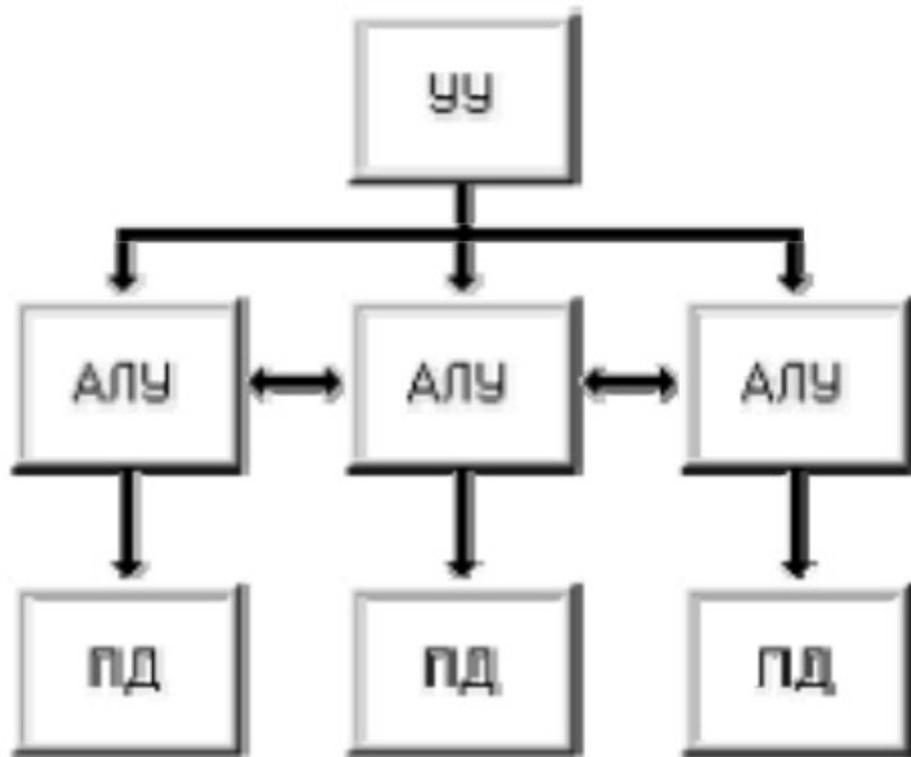
Классификация Шора



Классификация Шора



Классификация Шора

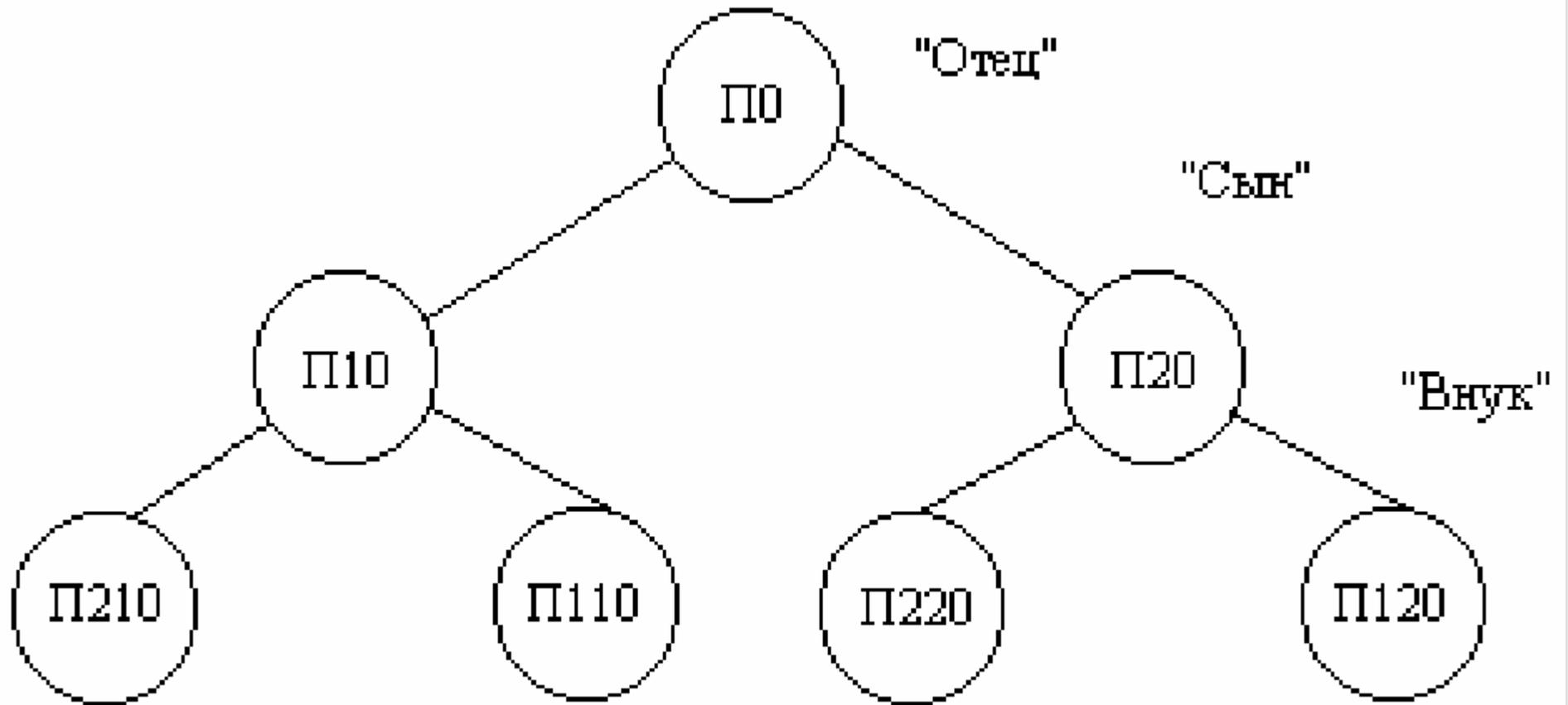


модель Хорнинга-Ренделла

Хорнинг и Ренделл (1973) построили формальное определение понятия процесса. Основные термины модели:

- набор переменных состояния $X = \{x_0, x_1, \dots, x_N, \dots\}$;
- состояние $X = \{x_0 = a, x_1 = b, \dots, x_N = z, \dots\}$;
- пространство состояний;
- действия;
- работа;
- функция действия;
- процесс $P = (R_x, \{f\}, R_0)$, $R_0 \subset R_x$
- начальное состояние.

Свойства процессов



Алгоритмы планирования процессов

Планирование процессов включает в себя решение следующих задач:

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

Вытесняющие и невытесняющие алгоритмы планирования

Существует два основных типа процедур планирования процессов - вытесняющие (preemptive multitasking) и невытесняющие (non-preemptive multitasking).

First-Come, First-Served (FCFS)

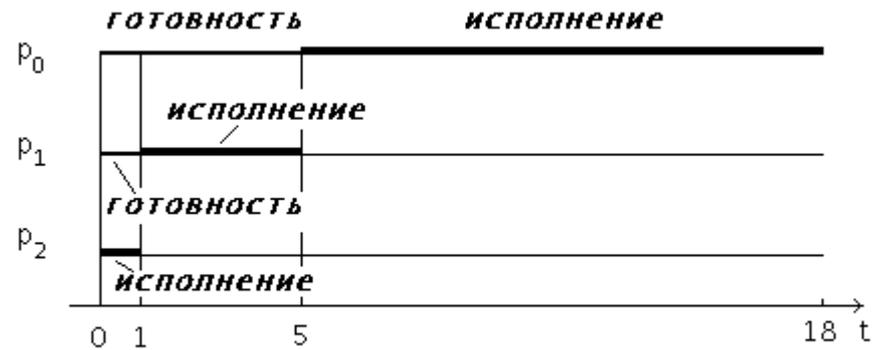
- Простейшим алгоритмом планирования является алгоритм, который принято обозначать аббревиатурой FCFS по первым буквам его английского названия - First Come, First Served (первым пришел, первым обслужен). Представим себе, что процессы, находящиеся в состоянии готовности, организованы в очередь. Когда процесс переходит в состояние готовности, он, а точнее ссылка на его PCB, помещается в конец этой очереди. Выбор нового процесса для исполнения осуществляется из начала очереди с удалением оттуда ссылки на его PCB. Очередь подобного типа имеет в программировании специальное наименование FIFO - сокращение от First In, First Out (первым вошел, первым вышел).

Процесс	p0	p1	p2
Продолжительность очередного CPU burst	13	4	1

First-Come, First-Served (FCFS)



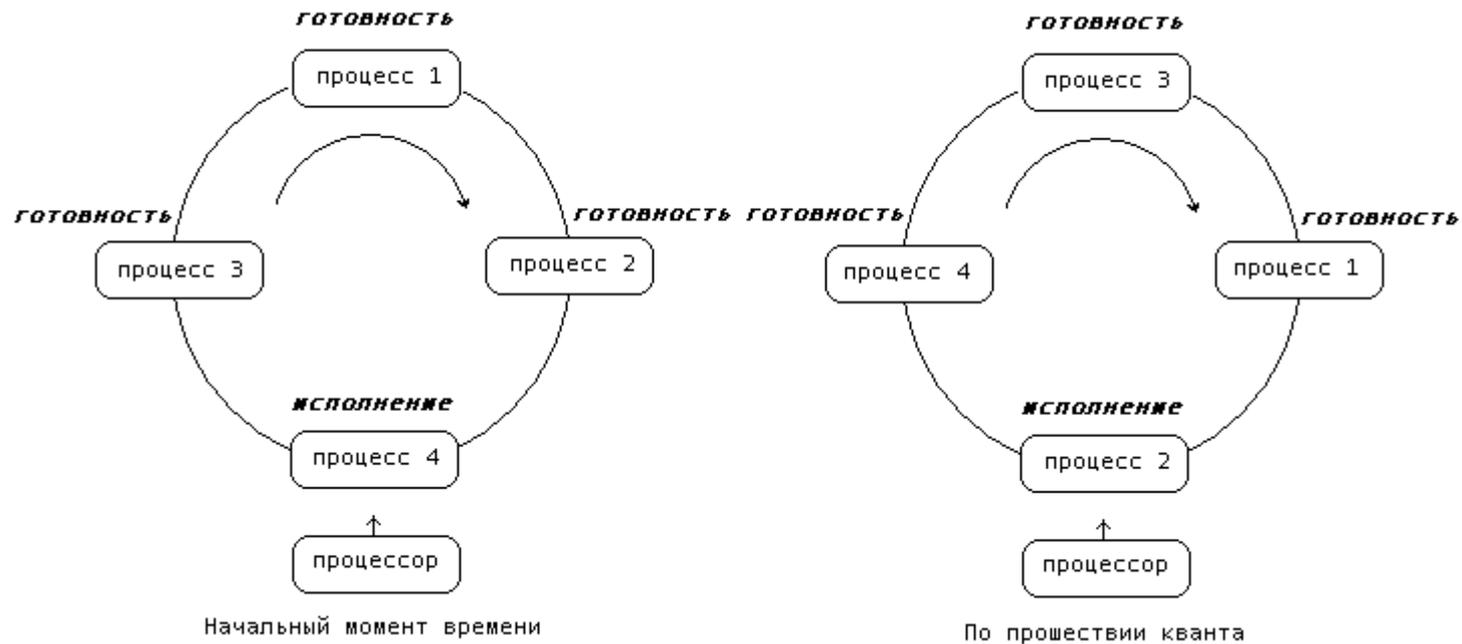
First-Come, First-Served (FCFS)



Round Robin (RR)

- Модификацией алгоритма FCFS является алгоритм, получивший название Round Robin (Round Robin - это вид детской карусели в США) или сокращенно RR. По сути дела это тот же самый алгоритм, только реализованный в режиме вытесняющего планирования. Можно представить себе все множество готовых процессов организованным циклически - процессы сидят на карусели. Карусель вращается так, что каждый процесс находится около процессора небольшой фиксированный квант времени, обычно 10 - 100 миллисекунд (см. рисунок 3.4.). Пока процесс находится рядом с процессором, он получает процессор в свое распоряжение и может исполняться.

Round Robin (RR)



Shortest-Job-First (SJF)

- При рассмотрении алгоритмов FCFS и RR мы видели, насколько существенным для них является порядок расположения процессов в очереди процессов готовых к исполнению. Если короткие задачи расположены в очереди ближе к ее началу, то общая производительность этих алгоритмов значительно возрастает. Если бы мы знали время следующих CPU burst для процессов, находящихся в состоянии готовности, то могли бы выбрать для исполнения не процесс из начала очереди, а процесс с минимальной длительностью CPU burst. Если же таких процессов два или больше, то для выбора одного из них можно использовать уже известный нам алгоритм FCFS. Квантование времени при этом не применяется. Описанный алгоритм получил название "кратчайшая работа первой" или Shortest Job First (SJF).

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

Гарантированное планирование

- При интерактивной работе N пользователей в вычислительной системе можно применить алгоритм планирования, который гарантирует, что каждый из пользователей будет иметь в своем распоряжении $\sim 1/N$ часть процессорного времени. пронумеруем всех пользователей от 1 до N . Для каждого пользователя с номером i введем две величины: T_i - время нахождения пользователя в системе, или, другими словами длительность сеанса его общения с машиной, и t_i - суммарное процессорное время уже выделенное всем его процессам в течение сеанса. Справедливым для пользователя было бы получение T_i/N процессорного времени. Если

Приоритетное планирование

- Алгоритмы SJF и гарантированного планирования представляют собой частные случаи приоритетного планирования. При приоритетном планировании каждому процессу присваивается определенное числовое значение - приоритет, в соответствии с которым ему выделяется процессор. Процессы с одинаковыми приоритетами планируются в порядке FCFS. Для алгоритма SJF в качестве такого приоритета выступает оценка продолжительности следующего CPU burst. Чем меньше значение этой оценки, тем более высокий приоритет имеет процесс. Для алгоритма гарантированного планирования приоритетом служит вычисленный коэффициент справедливости. Чем он меньше, тем больше приоритет у процесса.