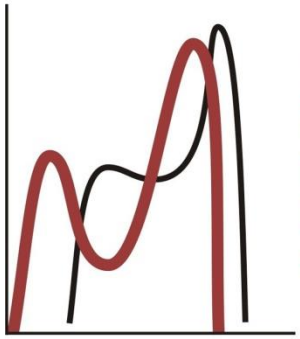


ИНСТИТУТ
МАТЕМАТИКИ
МЕХАНИКИ
КОМПЬЮТЕРНЫХ
НАУК

имени И.И. Воровича —

Архитектура компьютера и операционные системы

Лекция 16. Планирование.

Андреева Евгения Михайловна
доцент кафедры информатики и вычислительного эксперимента

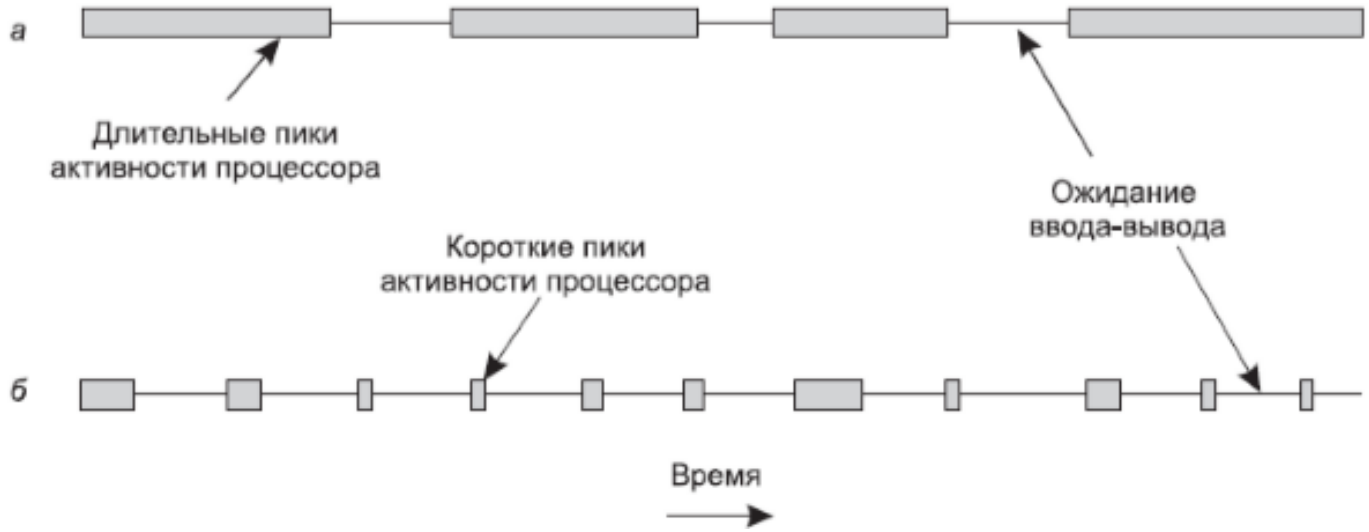
План лекции

- Алгоритмы планирования
- Алгоритмы синхронизации
- Механизмы синхронизации

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Планирование процессов

- Максимально загрузить дорогие устройства (ЦП)
- Минимизировать переключение процессов
- Учет специфики планирования
- Учет среды планирования
- Учет поведения процессов



ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Специфика планирования

- Многопользовательские
  - разумное сочетание пакетных задач с задачами в режиме разделения времени;
- ПК
  - ранее преимущество активного процесса;
  - "дефицит" входной информации при высокоскоростном процессоре;
- Смартфоны
  - учет энергосбережения при планировании;

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Среда планирования

- пакетная
  - неприоритетные алгоритмы планирования (АП) или приоритетные с большим квантом
- интерактивная
  - приоритетные АП
- реального времени
  - выполнение определенной прикладной задачи, особого планирования не требуется.

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Критерии планирования: пользовательские

- Время оборота: (turnaround time, TT) интервал между запуском и завершением процесса.
- Время ожидания: (waiting time, WT) время нахождения процесса в состоянии готовность.
- Время отклика: (response time, RT) интервал между подачей запроса и началом выдачи ответа.
- Предсказуемость: выполнение задания примерно за одинаковое время при разных запусках.

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Критерии планирования: системные

- Пропускная способность: (throughput, TP) количество процессов, завершающихся за единицу времени.
- Использование процессорных ресурсов: средняя доля времени, в течение которого занято процессорное ядро.
- Беспристрастность: (fairness) все процессы должны рассматриваться как равноправные при отсутствии дополнительных указаний.
- Использование приоритетов: предпочтение в обслуживании процессам с более высоким приоритетом.
- Баланс ресурсов: предпочтение в обслуживании процессам с малым использованием системных ресурсов.

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Вытесняющее и невытесняющее планирование

- Когда процесс переводится из состояния исполнение в состояние закончил исполнение.
- Когда процесс переводится из состояния исполнение в состояние ожидание.

невытесняющее планирование

- Когда процесс переводится из состояния исполнение в состояние готовность
- Когда процесс переводится из состояния ожидание в состояние готовность

вытесняющее планирование

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

"Точки" запуска планирования

- Создание процесса ("родительский" или "дочерний")
- Завершение процесса
- Блокировка процесса
- Прерывание ввода-вывода
- По таймеру

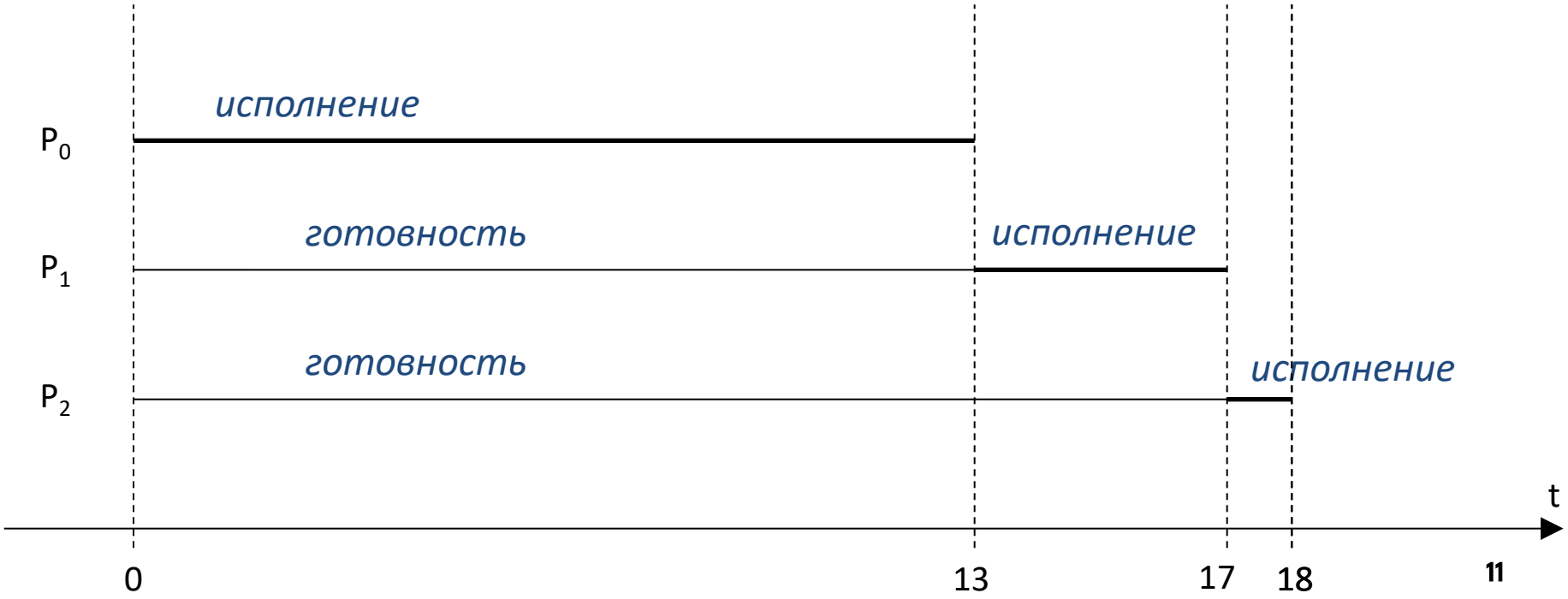
FCFS (First Come – First Served)

- Используется в пакетных системах
- Процессы, находящиеся в состоянии готовность, выстроены в очередь.
- Когда процесс переходит в состояние готовность, он помещается в конец этой очереди. Выбор нового процесса для исполнения осуществляется из начала очереди.
- Алгоритм осуществляет *невытесняющее планирование.*

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

FCFS (First Come – First Served)

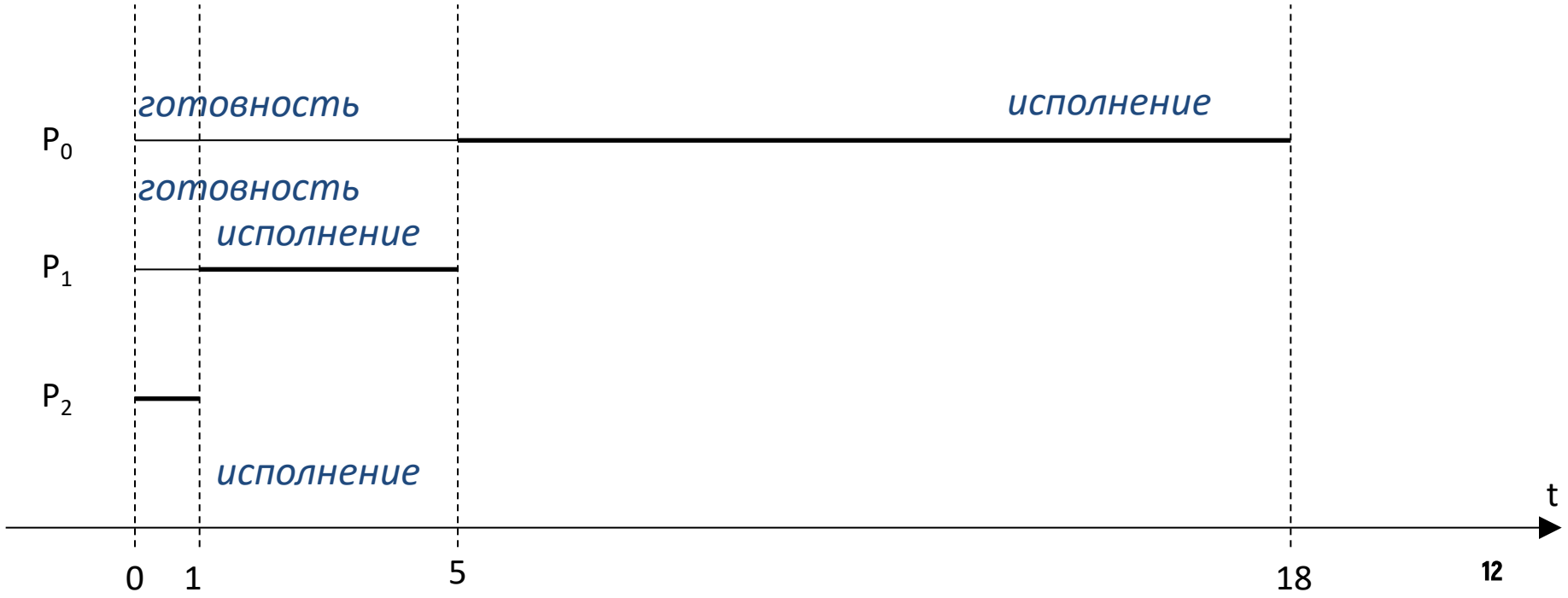
Процессы	P_0	P_1	P_2
Продолжительность	13	4	1



ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

FCFS (First Come – First Served)

Процессы	P_2	P_1	P_0
Продолжительность	1	4	13



ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Пример

- Пример вычисления среднего значения времени оборота (turnaround time, TT) и среднего значения времени ожидания (waiting time, WT)
- Случай 1:
  - TT=(13+17+18)/3=16
  - WT=(0+13+17)/3=10
- Случай 2:
  - TT=(1+5+18)/3=8
  - WT=(0+1+5)/3=2

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

RR (Round Robin)

- Используется в интерактивных системах
- Процессы, находящиеся в состоянии готовность, выстроены в очередь.
- Планировщик выбирает для очередного исполнения процесс, расположенный в начале очереди, и устанавливает таймер для генерации прерывания по истечении определенного *кванта времени*.
- Алгоритм осуществляет *вытесняющее планирование*.

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

RR (Round Robin)

- Остаток времени работы процесса <= кванта времени:
  - процесс освобождает процессор до истечения кванта;
  - на исполнение выбираем новый процесс из начала очереди готовых.
- Остаток времени работы процесса >= кванта времени:
  - по окончании кванта процесс помещается в конец очереди готовых к исполнению процессов;
  - на исполнение выбираем новый процесс из начала очереди готовых.

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

RR (Round Robin)

Процессы	P_0	P_1	P_2
Продолжительность	13	4	1

Величина кванта времени – 4

время	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
P_0	И	И	И	И	Г	Г	Г	Г	Г	И	И	И	И	И	И	И	И	И
P_1	Г	Г	Г	Г	И	И	И	И										
P_2	Г	Г	Г	Г	Г	Г	Г	Г	И									

Исполнение

P_0

Очередь готовых

P_0	P_0	P_0

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

RR (Round Robin)

Процессы	P_0	P_1	P_2
Продолжительность	13	4	1

Величина кванта времени – 1

время	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
P_0	И	Г	Г	И	Г	И	Г	И	Г	И	И	И	И	И	И	И	И	И
P_1	Г	И	Г	Г	И	Г	И	Г	И									
P_2	Г	Г	И															

Исполнение

P_0

Очередь готовых

P_0	P_0	P_0

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

SJF (Shortest Job First)

- Используется в пакетных системах
- Из процессов, находящихся в состоянии готовность, для исполнения выбирается процесс с минимальной длительностью.
- Квантование времени при этом не применяется.
- При *невытесняющем SJF - планировании* процессор предоставляется избранному процессу на все необходимое ему время.
- При *вытесняющем SJF - планировании* учитывается появление новых процессов в очереди готовых к исполнению во время работы выбранного процесса. Если продолжительность работы нового процесса меньше, чем остаток исполняющегося, то исполняющийся процесс вытесняется новым.

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

SJF (Shortest Job First) невытесняющий

Процессы	P_0	P_1	P_2	P_3
Продолжительность	5	3	7	1

время	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
P_0	Г	Г	Г	Г	И	И	И	И	И							
P_1	Г	И	И	И												
P_2	Г	Г	Г	Г	Г	Г	Г	Г	Г	И	И	И	И	И	И	И
P_3	И															

Исполнение

P_0

Очередь готовых

P_0	P_1	P_2	P_3

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

SJF (Shortest Job First) вытесняющий

Процессы	P_0	P_1	P_2	P_3
Продолжительность	6	2	5	5
Момент появления в очереди	0	2	6	0

время	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
P_0	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	И	И	И	И	И	И
P_1			И	И														
P_2							Г	И	И	И	И	И						
P_3	И	И	Г	Г	И	И	И											

Исполнение

P_0

Очередь готовых

P_0	P_1	P_2	P_3

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

SJF (Shortest Job First) приближение

- $\tau(n)$ – продолжительность работы n-го процесса
- $T(n+1)$ – предсказание для n+1-го
- α – параметр от 0 до 1
- $T(n+1)= \alpha\, \tau(n) + (1 - \alpha)T(n)$,
- $T(0)$ – произвольно
- Если $\alpha = 0$, то $T(n+1) = T(n) = \ldots = T(0)$,
  нет учета последнего поведения
- Если $\alpha = 1$, то $T(n+1) = \tau(n)$,
  нет учета предыстории

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Гарантированное планирование

- Используется в интерактивных системах
- В системе разделения времени N пользователей
- Ti – время нахождения i-го пользователя в системе
- τi – суммарное процессорное время процессов i-го пользователя

  τi ‹‹ Ti /N – пользователь обделен

  τi ›› Ti /N– пользователю благоволят

  (τi N) / Ti – коэффициент справедливости
- На исполнение выбираются готовые процессы пользователя с наименьшим коэффициентом справедливости

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Приоритетное планирование

- Используется в интерактивных системах
- Каждому процессу процессор выделяется в соответствии с приписанным к нему числовым значением - приоритетом
- Параметры для назначения приоритета бывают:
  - внешние
  - внутренние
- Политика изменения приоритета:
  - статический приоритет
  - динамический приоритет
- Алгоритм планирования:
  - вытесняющий
  - невытесняющий

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Приоритетное планирование: невытесняющий

Процессы	P_0	P_1	P_2	P_3
Продолжительность	6	2	5	5
Момент появления в очереди	0	2	6	0
Приоритет	4	3	2	1

время	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
P_0	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	И	И	И	И	И	И
P_1			Г	Г	Г	И	И											
P_2							Г	И	И	И	И	И						
P_3	И	И	И	И	И													

Исполнение

P_0

Очередь готовых

P_0	P_1	P_2	P_3

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Приоритетное планирование: вытесняющий

Процессы	P_0	P_1	P_2	P_3
Продолжительность	6	2	5	5
Момент появления в очереди	0	2	6	0
Приоритет	4	3	2	1

время	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
P_0	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	И	И	И	И	И	И
P_1			Г	Г	Г	И	Г	Г	Г	Г	Г	И						
P_2							И	И	И	И	И							
P_3	И	И	И	И	И													

Исполнение

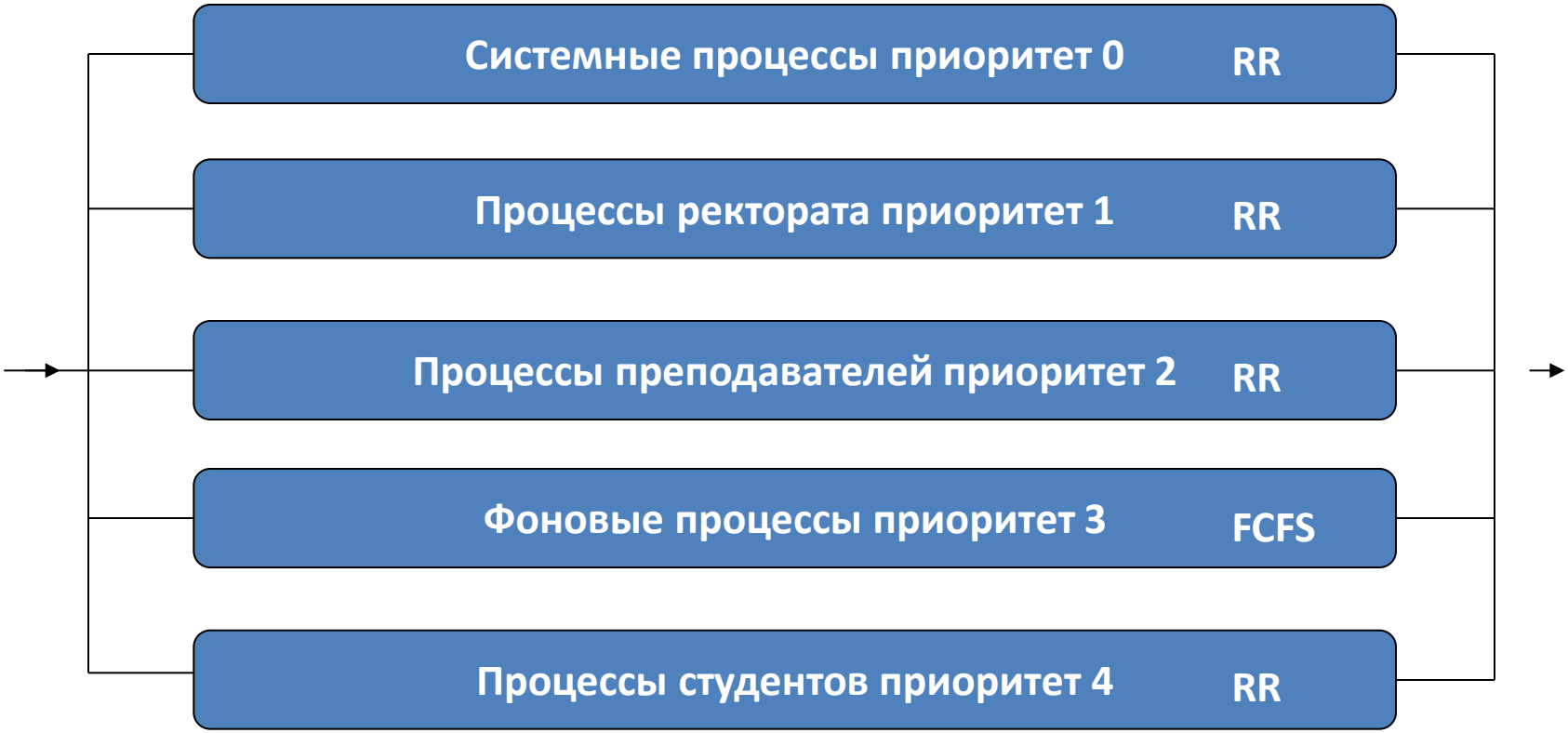
P_0

Очередь готовых

P_0	P_1	P_2	P_3

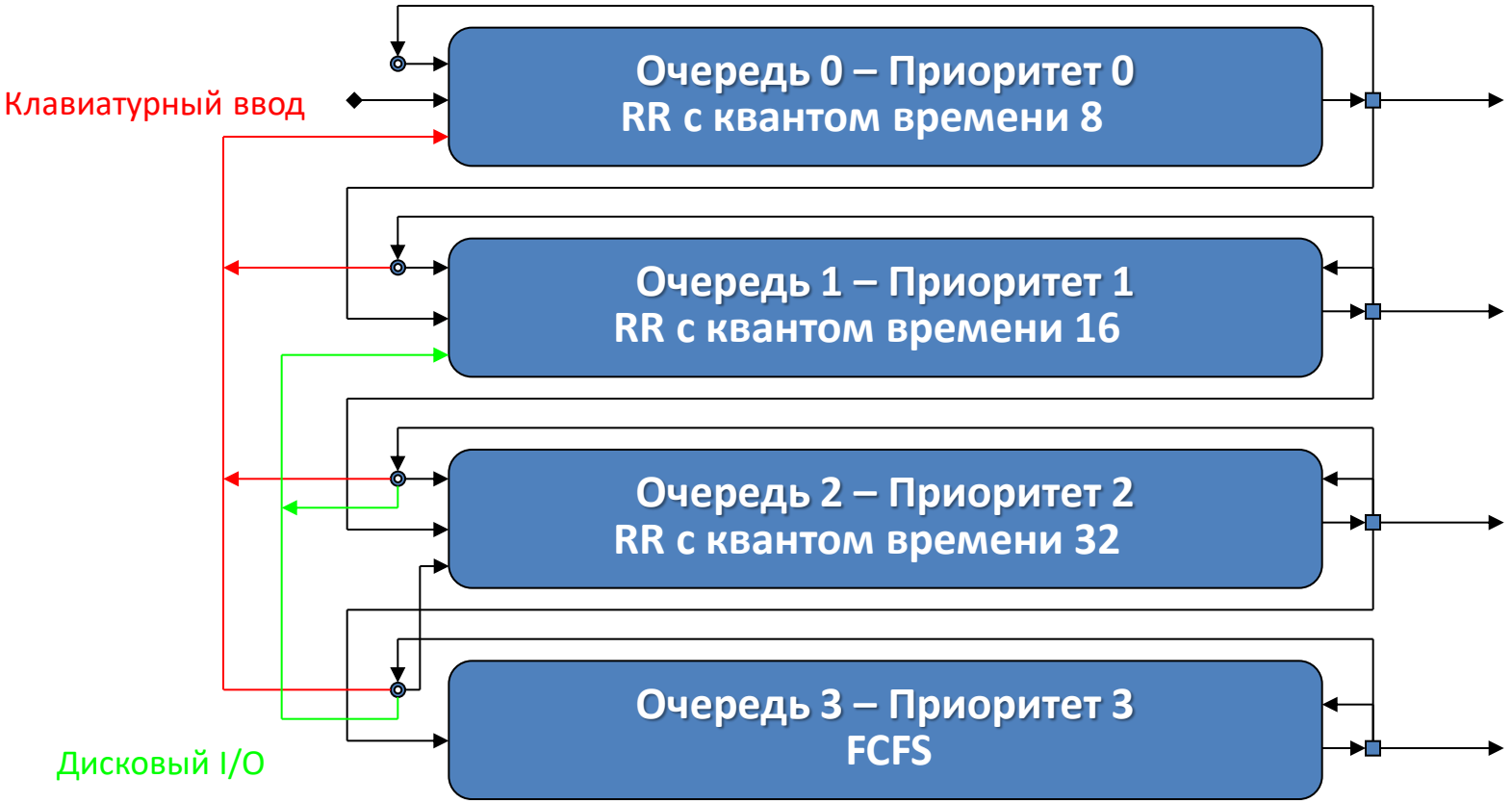
ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Многоуровневые очереди (Multilevel Queue)



ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Многоуровневые очереди с обратной связью (Multilevel Feedback Queue)



ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Многоуровневые очереди с обратной связью (Multilevel Feedback Queue)

- Для полного описания необходимо задать
  - количество очередей в состоянии готовность
  - алгоритм планирования между очередями
  - алгоритмы планирования внутри очередей
  - куда помещается родившийся процесс
  - правила перевода процессов из одной очереди в другую

Проблемы планирования

- Голодная смерть: (starvation) — ситуация невозможности завершения процесса из-за постоянного отказа ему в требуемом ресурсе (процессорное время, . . .) со стороны операционной системы.
- Взаимная блокировка (deadlock) — ситуация в многозадачной среде, при которой несколько процессов находятся в состоянии ожидания ресурсов, занятых друг другом, и ни один из них не может продолжать свое выполнение

Атомарные операции

- Операция в общей области памяти называется **атомарной**, если она завершается в один шаг относительно других потоков, имеющих доступ к этой памяти.
- Во время выполнения такой операции над переменной, ни один поток не может наблюдать изменение наполовину завершенным.
- Атомарная загрузка гарантирует, что переменная будет загружена целиком в один момент времени.
- Атомарность бывает аппаратной (обеспечивается аппаратурой) и программной, когда используются специальные средства межпроцессного взаимодействия

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Interleaving

Последовательность P: a b c

Последовательность Q: d e f

Последовательное выполнение PQ: a b c d e f

Псевдопараллельное выполнение (режим разделения времени) :

a b c d e f

a b d c e f

a b d e c f

a b d e f c

. . .

d e f a b c

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Детерминированные и недетерминированные наборы

P: x=2
y=x-1

Q: x=3
y=x+1

(x, y): (2, 1) (3, 4) (2, 3) (3, 2)

- Недетерминированный набор – при одинаковых начальных данных возможны разные результаты
- Детерминированный набор – при одинаковых начальных данных всегда один результат

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Условия Бернстайна (Bernstain)

P: 1. x=u+v

2. y=x*w

Входные переменные

R1 = {u, v}

R2 = {x, w}

Вход для процесса

R(P)={u, v, x, w}

Выходные переменные

W1 = {x}

W2 = {y}

Выход для процесса

W(P)={x, y}

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Условия Бернстайна

Если для двух процессов P и Q:

1. $W(P) \cap W(Q) = \{\emptyset\}$
2. $W(P) \cap R(Q) = \{\emptyset\}$
3. $R(P) \cap W(Q) = \{\emptyset\}$

то набор процессов {P, Q} является детерминированным

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

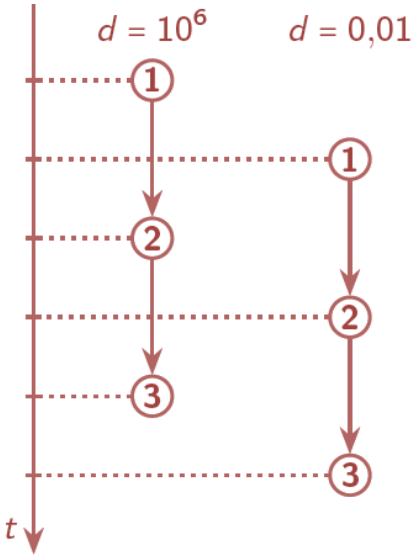
Условие гонки (race condition)

- В недетерминированных наборах встречается состояние гонки
- Условие гонки (race condition) — особенность функционирования системы, при которой её выходной сигнал непредсказуемо зависит от последовательности и/или временных задержек происходящих в ней событий.

- Пример

  входные данные: d

  1. Считать(x)
  2. x ← x + d
  3. Записать(x)



ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Взаимоисключение (mutual exclusion)

- Избежать недетерминированного поведения при неважности очередности доступа можно с помощью взаимоисключения (взаимоблокировки).
- Взаимоисключение – это алгоритм обеспечения неодновременности использования общего ресурса разными потоками.

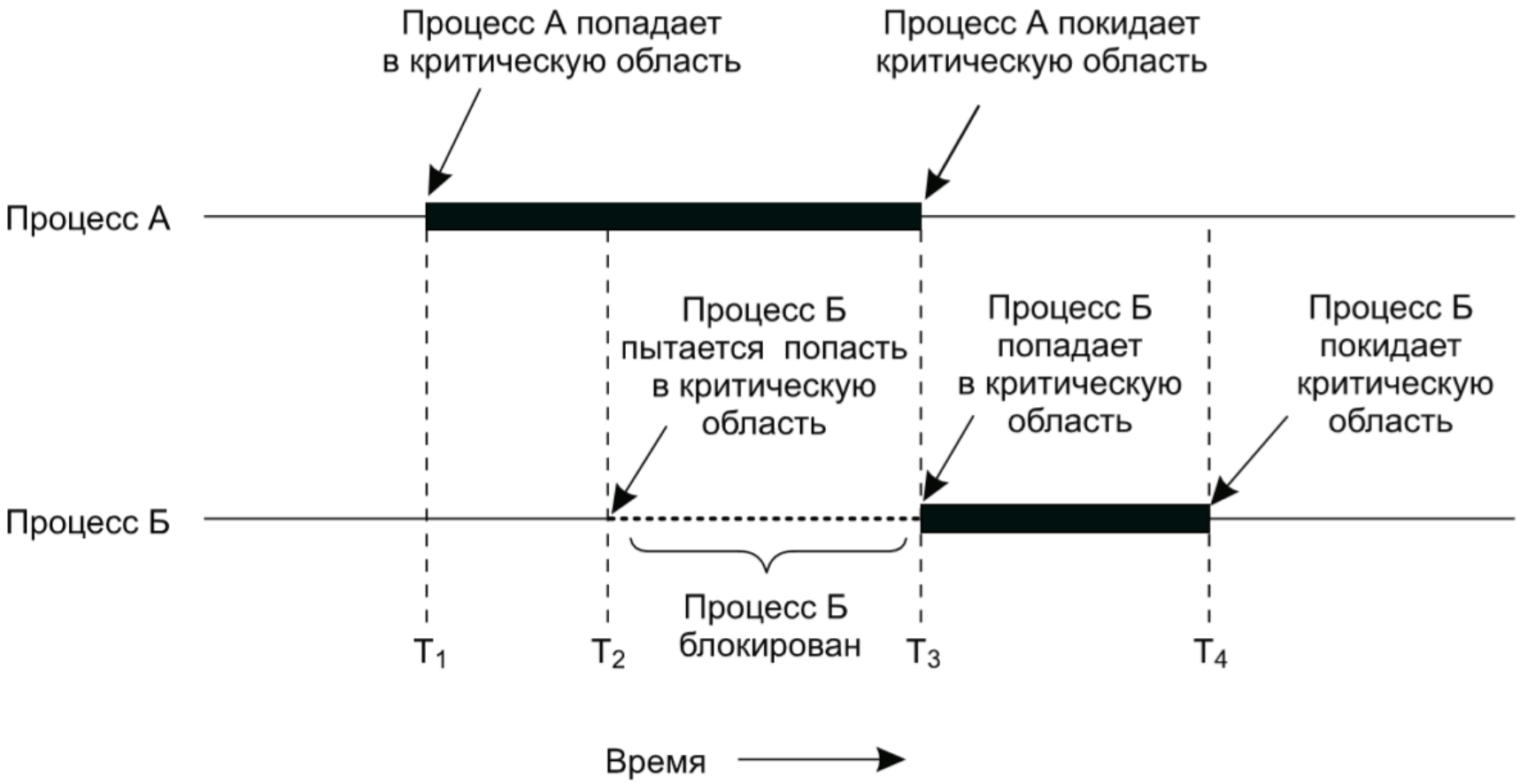
ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Критическая секция

- Критическая секция (critical section) — участок кода, исполняемый не более чем одним потоком из числа имеющих доступ к общему ресурсу.
- Чтобы исключить эффект гонок по отношению к некоторому ресурсу, необходимо организовать работу так, чтобы в каждый момент времени только один процесс мог находиться в своей критической секции, связанной с этим ресурсом.
- Иными словами, необходимо обеспечить реализацию взаимоисключения для критических секций программ.
- Это означает, что по отношению к другим процессам, участвующим во взаимодействии, критическая секция начинает выполняться как атомарная операция.

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Использование критических секций при взаимном исключении



Структура процесса, участвующего во взаимодействии

```
while (some condition) {
        entry section
                critical section
        exit section
                remainder section
}
```

Требования, предъявляемые к алгоритмам

1. Алгоритм должен быть программным
2. Нет предположений об относительных скоростях выполнения и числе процессоров
3. Выполняется условие взаимоисключения (mutual exclusion) для критических участков
4. Выполняется условие прогресса (progress)
   - Процессы, которые находятся вне своих критических участков и не собираются входить в них, не могут препятствовать другим процессам входить в их собственные критические участки.
   - Решение не должно приниматься бесконечно долго.
5. Выполняется условие ограниченного ожидания (bound waiting)

Запрет прерываний

```
while (some condition) {
        запретить все прерывания
            critical section
        разрешить все прерывания
            remainder section
}
```

Обычно используется внутри ОС

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Переменная-замок

Shared int lock = 0;

```
while (some condition) {
      while (lock);   lock = 1;
            critical section
      lock = 0;
            remainder section
}
```

```
while (some condition) {
      while (lock);    lock = 1;
            critical section
      lock = 0;
            remainder section
}
```

Нарушается условие взаимоисключения

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Строгое чередование

Shared int turn = 0;

P_0

```
while (some condition) {
    while (turn != 0);
        critical section
    turn = 1;
        remainder section
}
```

P_1

```
while (some condition) {
    while (turn != 1);
        critical section
    turn = 0;
        remainder section
}
```

Нарушается условие прогресса

Условие взаимоисключения выполняется

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Домашнее задание

- Подготовка у тестированию по материалам лекции
  - Таненбаум Э., Бос Х. Современные операционные системы, стр. 178-198.
- Подготовка к Лаб. 12