

Направления подготовки: «Информатика и вычислительная техника» и «Информационные системы и технологии»

Профили образовательных программ:

«Системотехника и автоматизация проектирования в строительстве»

«Системотехника и информационные технологии управления в строительстве»

ОПЕРАЦИОННЫЕ СИСТЕМЫ

Тема 5. Ресурсы и управление ими в операционных системах





Содержание разделов курса

№	Наименование раздела дисциплины	Тема и содержание лекций
2	Управление процессами и ресурсами	Тема 5. Ресурсы в вычислительных системах. Понятие «ресурс», классификация ресурсов вычислительной системы: разделяемые и закрепляемые, потребляемые и восстанавливаемые. Deskрипторы ресурсов. Динамическое и статическое распределение ресурсов.
		Тема 6. Управление процессами и потоками. Понятия «задача», «процесс», «поток». Состояние процесса. Структура контекста процесса. Идентификатор и дескриптор процесса. Иерархия процессов. Организация учета процессов. Параллельно и последовательно используемые программные модули. Системные и пользовательские процессы. Планирование и диспетчеризация в ОС.
		Тема 7. Синхронизация процессов. Межпроцессное взаимодействие. Состязания, критические области, взаимные блокировки процессов. Необходимость и средства синхронизации. Критические секции. Семафоры. Мьютексы. События.

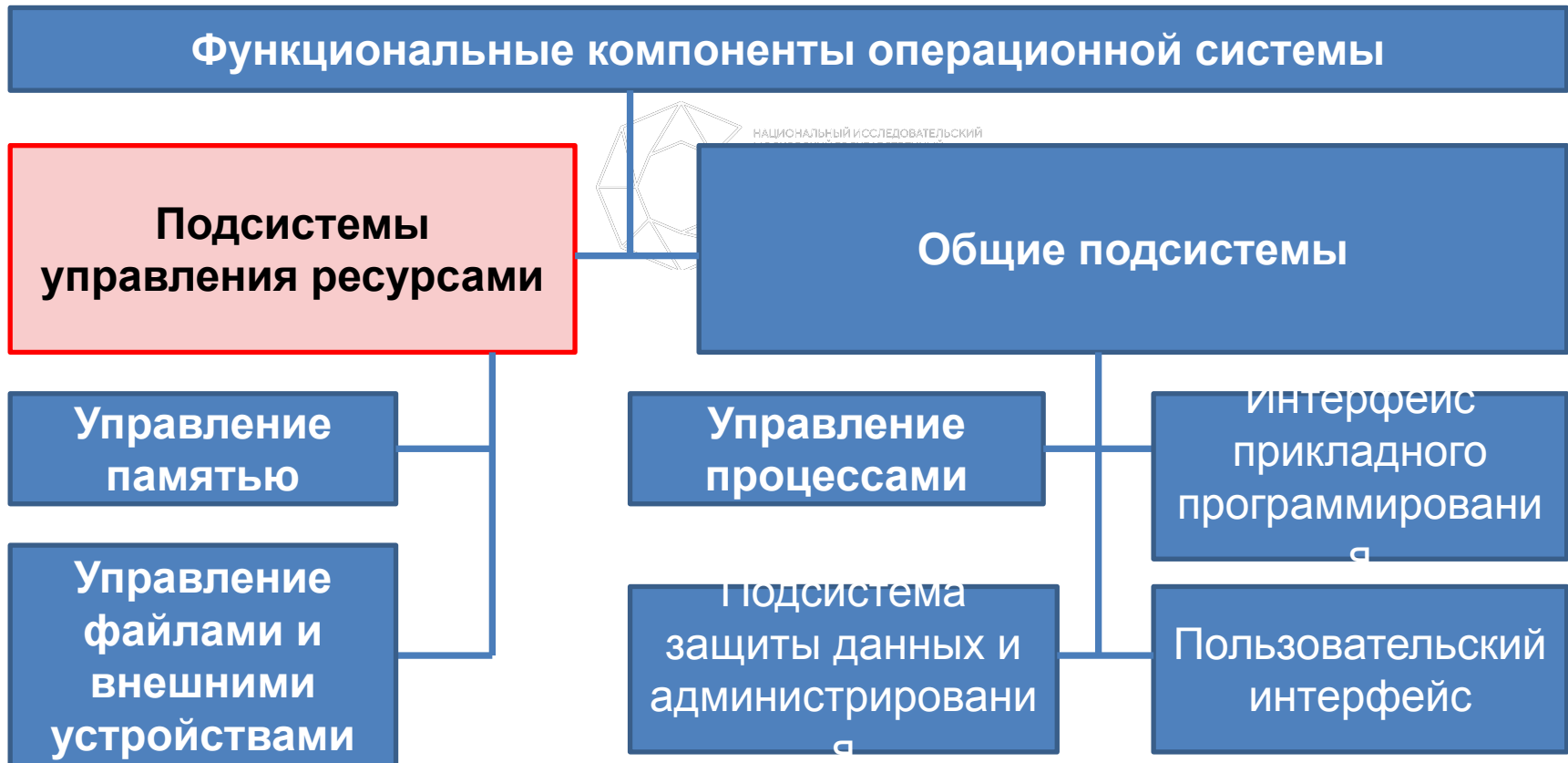
Операционная система (ОС) – комплекс управляющих и обрабатывающих программ, который, с одной стороны, выступает как интерфейс между аппаратурой компьютера и пользователем с его задачами, а с другой – предназначен для **наиболее эффективного использования ресурсов вычислительной системы и организации надежных вычислений.**

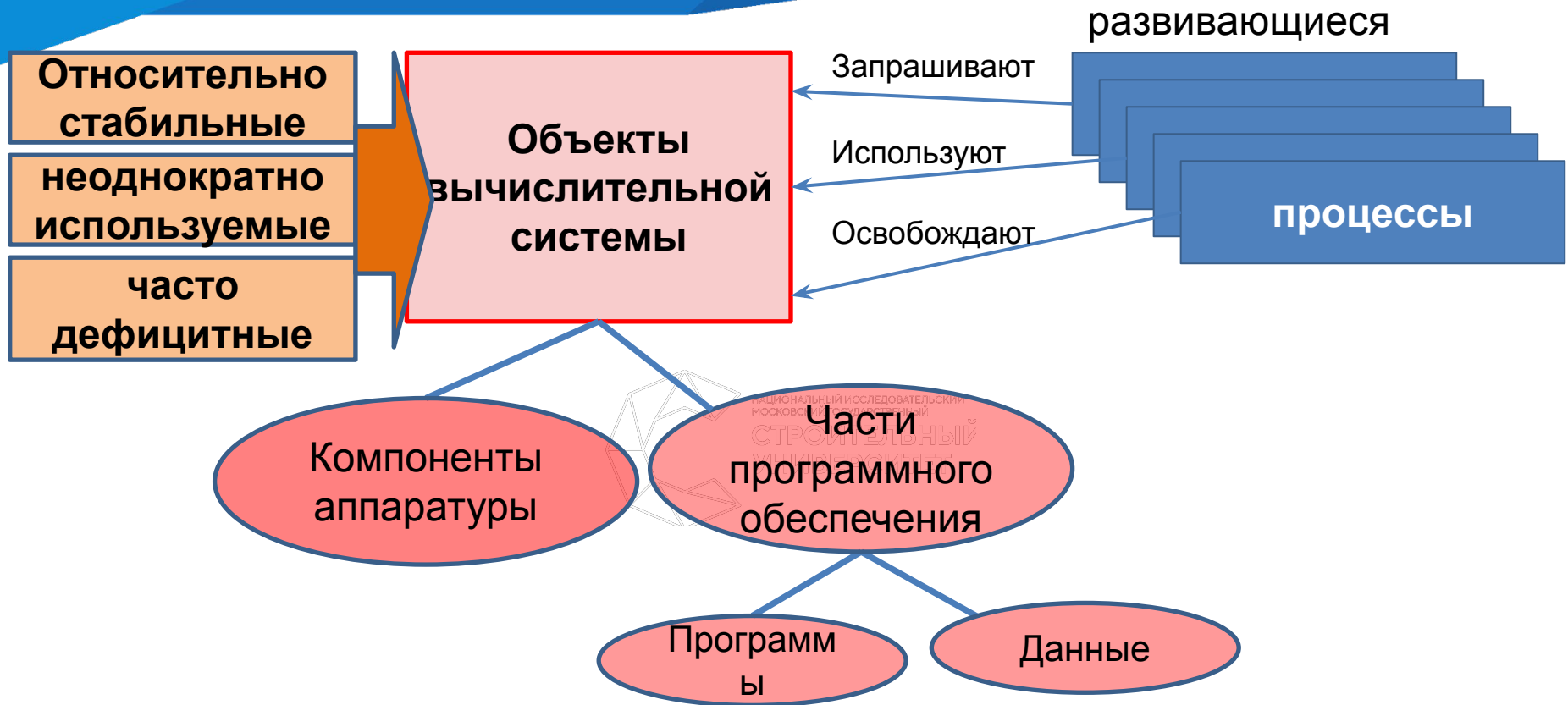


В соответствии с этим определением ОС выполняет **две группы функций:**

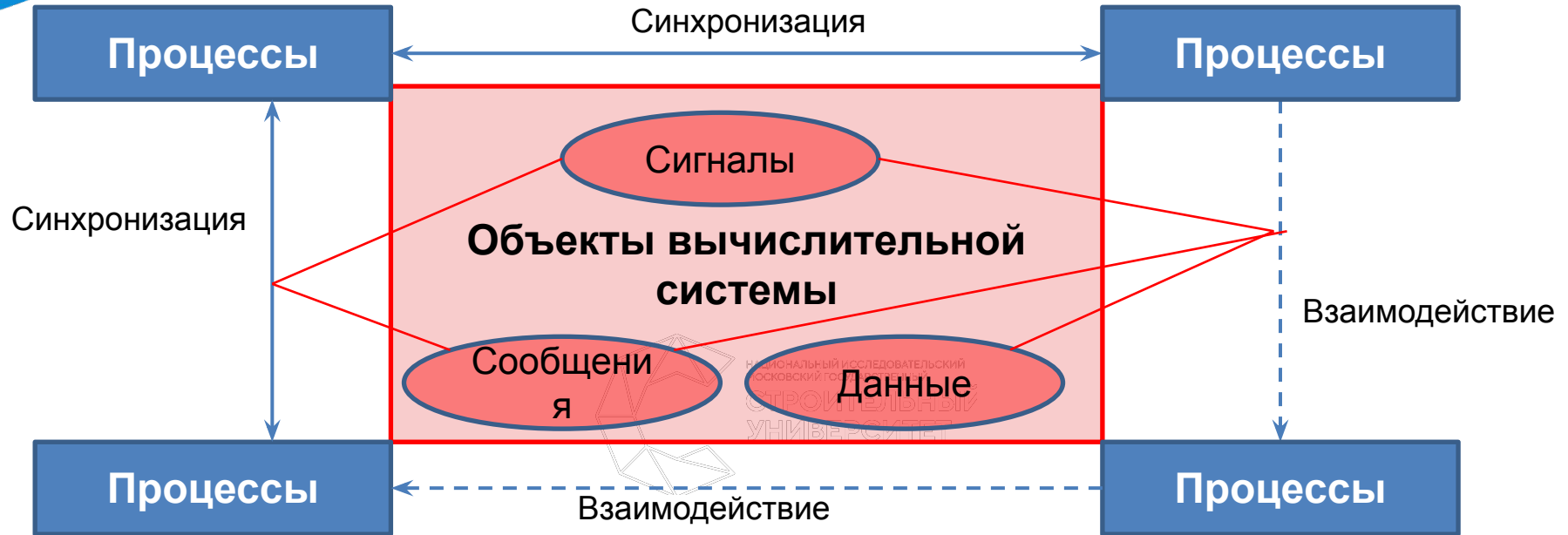
- предоставление пользователю или программисту вместо реальной аппаратуры компьютера абстрактной виртуальной машины, с которой удобней работать и которую легче программировать;
- **повышение эффективности использования компьютера путем рационального управления его ресурсами в соответствии с некоторым критерием.**

Функции операционной системы автономного компьютера обычно группируются либо в соответствии *с типами локальных ресурсов*, которыми управляет ОС, либо в соответствии *со специфическими задачами, применимыми ко всем ресурсам*. Такие группы функций называют *функциональными компонентами* или *подсистемами*.





Повторно используемые ресурсы – относительно стабильные, неоднократно используемые, часто дефицитные объекты вычислительной системы, которые запрашиваются, используются и освобождаются развивающимися процессами.



Потребляемые ресурсы – объекты, появляющиеся в ситуациях **синхронизации** и **связи** взаимодействующих выполняющихся процессов, не возвращающиеся в систему после **однократного** их **использования/потребления**.

Время работы процессора – самый дорогой **потребляемый** ресурс!

Расходные материалы, используемые в работе **повторно используемых** ресурсов, так же относят к **потребляемым ресурсам**: картридж, чернила, бумага.

Процесс будет приостановлен из-за неудовлетворенного запроса на ресурс!

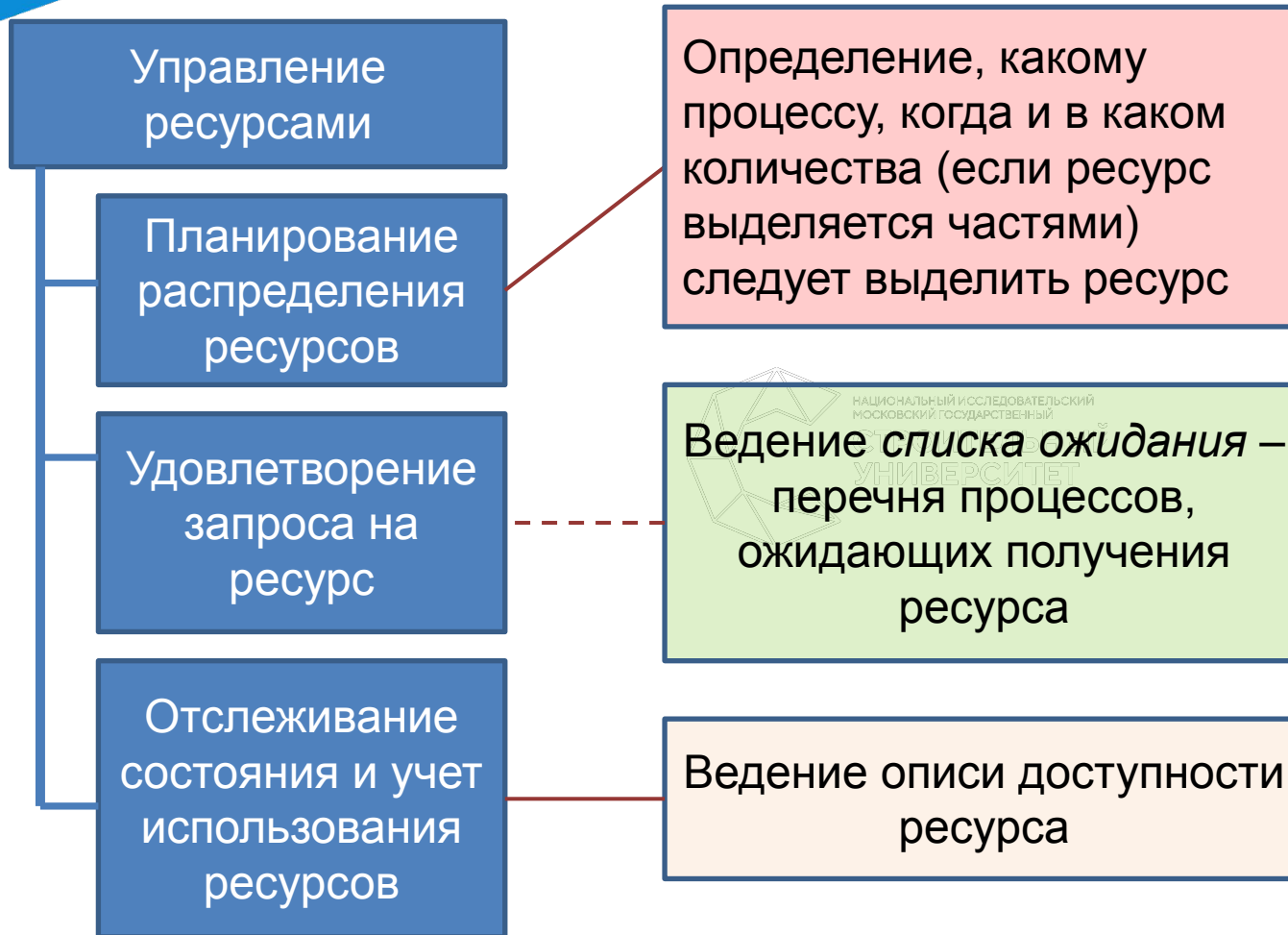
ОБОБЩЕННОЕ ОПРЕДЕЛЕНИЕ РЕСУРСА:

Ресурсом в вычислительной системе является всё то, отсутствие чего может вызвать переход какого-либо процесса в состояние логической блокировки

**Пропуская способность
вычислительной
системы**

Время реакции системы

Большинство функций управления ресурсами
выполняются операционной системой автоматически и
прикладному программисту недоступны.



**Дескриптор (описатель) ресурса –
системная структура данных,
хранящая основные сведения о ресурсе, позволяющие осуществлять
в отношении ресурса функции учёта и распределения**



Минимальный состав дескриптор ресурса

Идентификатор ресурса

Опись доступности ресурса

Список ожидающих процессов

Минимальный состав дескриптор ресурса

Идентификатор ресурса
Id; SR

Опись доступности ресурса

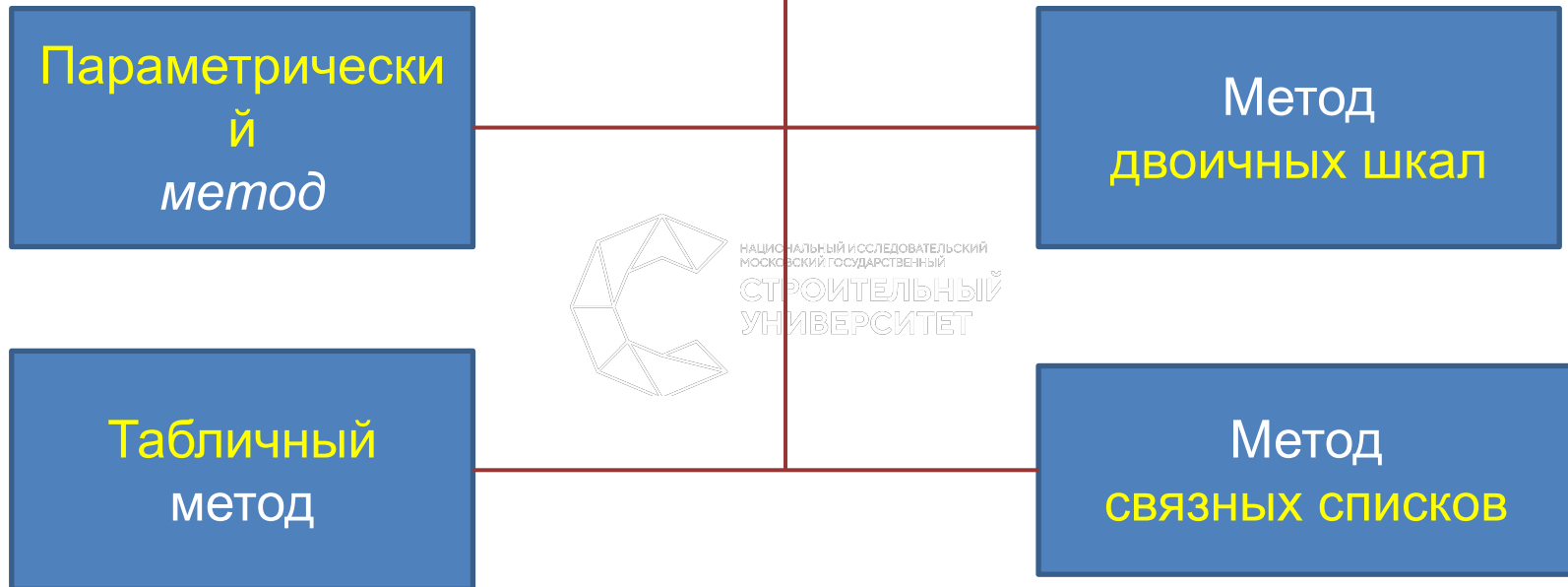
Список ожидающих процессов

Идентификатор ресурса **Id**, присвоенный ресурсу системой, **SR** – логическая метка повторно используемого ресурса

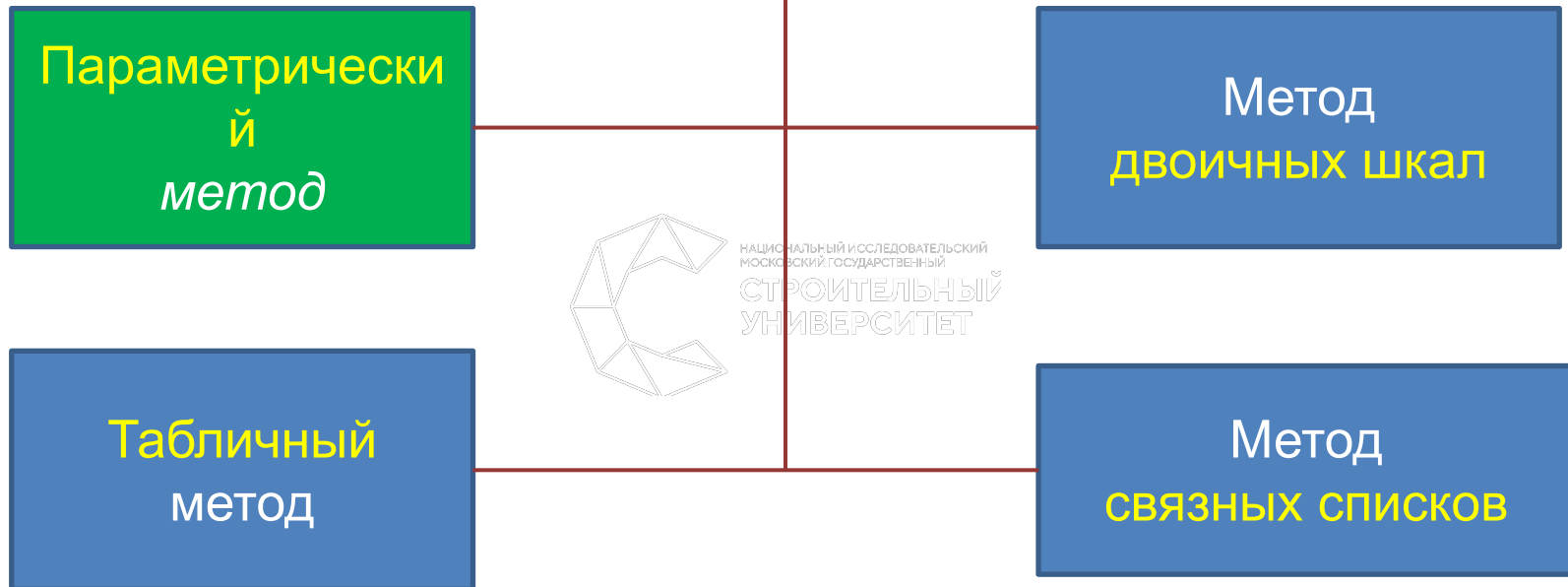
Систему учета ресурса.
Адреса программ вставки в опись новых элементов и удаления из описи распределенных ресурсов

Информация о процессах, заблокированных из-за отсутствия или недостаточности ресурса

Наиболее распространённые методы учёта ресурсов



Наиболее распространенные методы учета ресурсов



При использовании *параметрического* метода учета ресурс описывается либо количественно, либо адресами.

Пример: описание свободной памяти (адрес начала и адрес конца либо адрес начала и размер).

Для хранения данных о ресурсах операционная система создает специальную *таблицу системных указателей*.

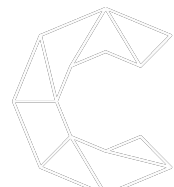
Наиболее распространенные методы учета ресурсов

Табличный
метод

Метод
двоичных шкал

Метод
связных списков

Параметрический
метод



Табличный метод используется для учета **внешних устройств**.
В системе используется набор таблиц 2-х типов:
1 глобальная и
несколько локальных таблиц.

Глобальная таблица

тип ресурса	количество ресурса	адрес локальной таблицы
Ресурс 1	<u>Количество 1</u>	Адрес локальной таблицы 1
...
Ресурс i	Количество i	Адрес локальной таблицы i
...
Ресурс N	Количество N	Адрес локальной таблицы N

N - число строк глобальной таблицы определяется **числом видов учитываемых в системе устройств.**

Каждой строке глобальной таблицы соответствует отдельная **локальная таблица**.

локальная таблица 1

Идентификатор устройства	адрес драйвера управления устройством	Данные о состоянии устройства				
		параметр1	...	параметр J	...	параметр K

локальная таблица i

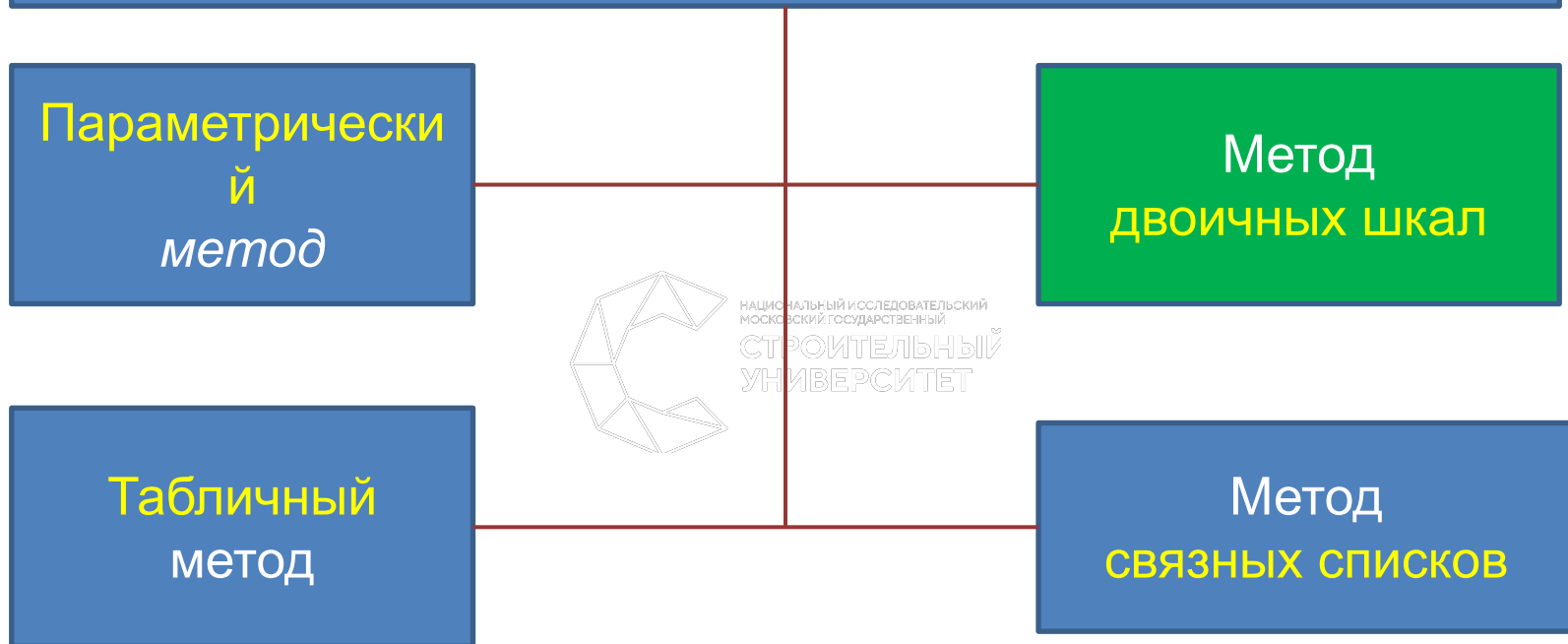
Идентификатор устройства	адрес драйвера управления устройством	Данные о состоянии устройства				
		параметр1	...	параметр J	...	параметр K

локальная таблица N

Идентификатор устройства	адрес драйвера управления устройством	Данные о состоянии устройства				
		параметр1	...	параметр J	...	параметр K

Количество столбцов состояния устройства и их названия зависят от устройства. Количество строк локальной таблицы определяется **числом устройств данного вида**, указанным в глобальной таблице.

Наиболее распространённые методы учёта ресурсов



Метод двоичных шкал используется чаще всего тогда, когда ресурс можно представить совокупностью *равных по размеру* частей (элементов).

Обычно используется при учёте оперативной памяти или дискового пространства.

Используются $(k+1)$ шкала разрядностью N (N – максимальное число единиц ресурса, для которого ведется учет).

K шкал по числу процессов, использующих данный ресурс, плюс 1 шкала – системная.

Разряды системной шкалы принимают значения :

1(единица ресурс **свободна**) или **0** (единица ресурс **занята**).

Разряды шкалы процесса принимают значения :

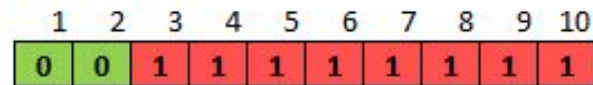
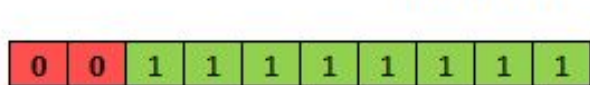
1(единица ресурс **не доступна**) или **0** (единица ресурс **используется процессом**).

При выделении ресурсов разряды системной шкалы просматриваются слева

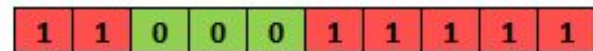
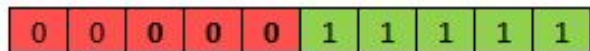


шкалы процессов

Выделение ресурса процессам

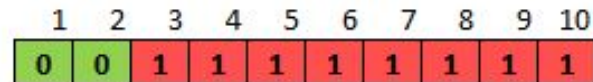
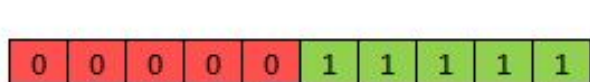


процесс 1; 2 ед. ресурса



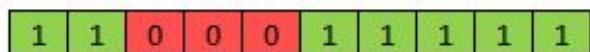
процесс 2; 3 ед. ресурса

Освобождение ресурса после завершения процесса



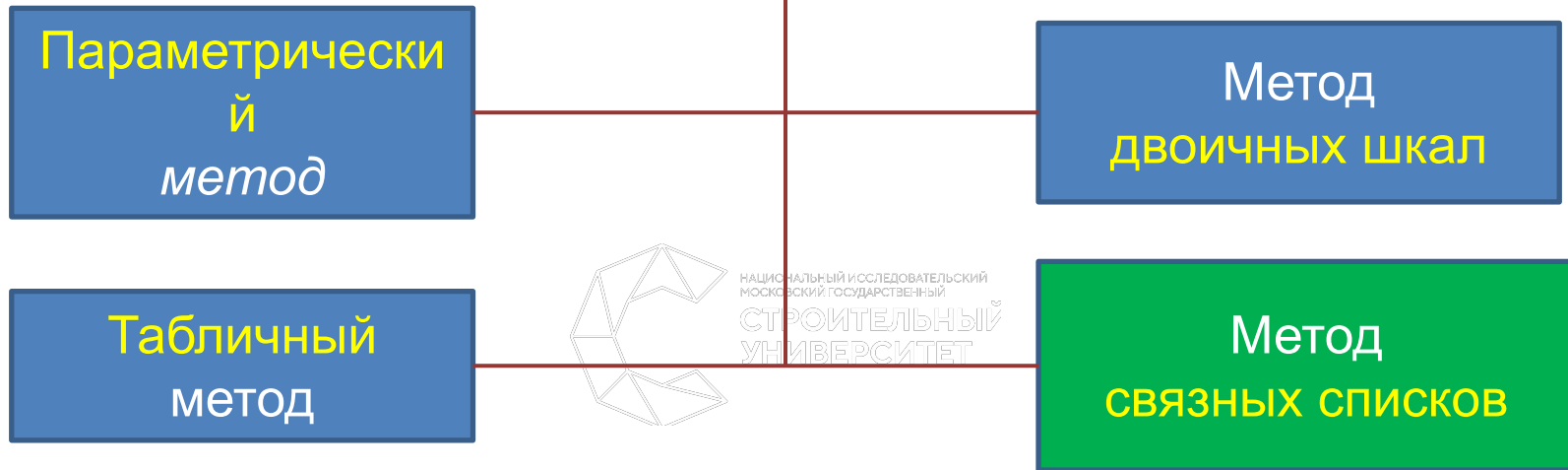
процесс 1; 2 ед. ресурса

Поразрядное сложение



инвертирование шкалы

Наиболее распространённые методы учёта ресурсов



Ресурсы, выделяемые процессам, и свободный ресурс описываются связными списками (иногда двунаправленным), причем **каждому процессу** соответствует свой **отдельный** список.

Свободная часть ресурс описывается **отдельным** списком.

Элемент списка соответствует фрагменту ресурса **произвольного** размера.

Элемент списка в простейшем случае должен содержать указатель на **следующий элемент** списка.

В общем случае элемент списка содержит:

- 1) указатель на то, какому процессу выделен ресурс ,
- 2) в каком количестве процессу выделен ресурс,
- 3) ссылка на следующий элемент,
- 4) ссылка на предыдущий элемент.

Среди системных объектов обязательно хранится указатель на начало списка.



В зависимости от метода организации доступа ресурсы можно разделить на две категории: **разделяемые** и **закрепляемые/выделенные**.

Ресурсы вычислительной системы

разделяемые -

ресурсы, которые могут (условно) **одновременно использоваться несколькими процессами**, например, процессор, память, жесткие диски и т.д.

Закрепляемые или выделенные

- ресурсы, которые должны отдаваться процессу в **монопольное использование**, например CD/DVD дисководы, принтеры и т.д. Для монополизации ресурса он должен закрепляться за процессом средствами ОС.

Способность некоторого ресурса быть разделяемым в большей степени обусловлена **порядком организации работы с этим ресурсом**, а не **его конструктивными особенностями**, т.е. один и тот же ресурс может быть разделяемым в одной ОС и закрепляемым в другой.



Способы управления ресурсами

Статическое управление -

предполагает, что все ресурсы, необходимые для выполнения процесса, **выделяется при его создании** (до начала его выполнения)

+

Процесс никогда не будет заблокирован из-за отсутствия ресурсов

-

Увеличивает стоимость системы из-за необходимости наличия большого количества ресурсов

Динамическое управление

предполагает, что ресурсы **выделяются процессам по мере выполнения процессов и по ходу возникновения запросов на ресурс**

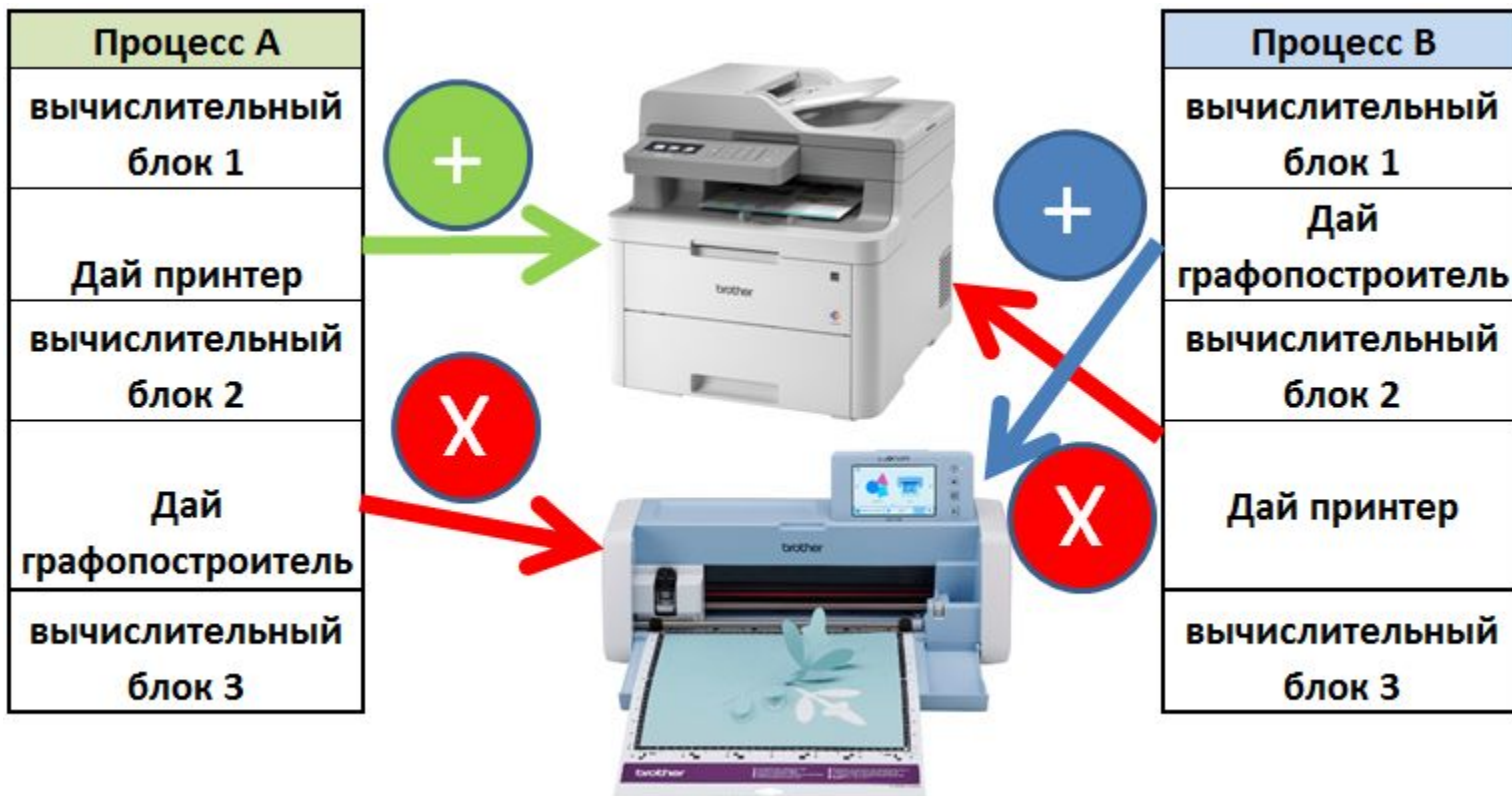
+

Снижение числа ресурсов в системе, следовательно, снижение её стоимости

-

Возможные блокировки процессов и тупиковая ситуация в работе операционной системы

«Тупик» - состояние операционной системы, при котором она *не может нормально развиваться*, то есть выполнять некоторые *процессы*, так как они *заблокированы из-за недоступности необходимого ресурса*.



Решить проблему «тупика» позволяет «алгоритм банкира».



Пусть есть несколько процессов, разделяющих в ходе своего выполнения общий ресурс.

Процессы выдают запросы на ресурс, указывая необходимое число его единиц.

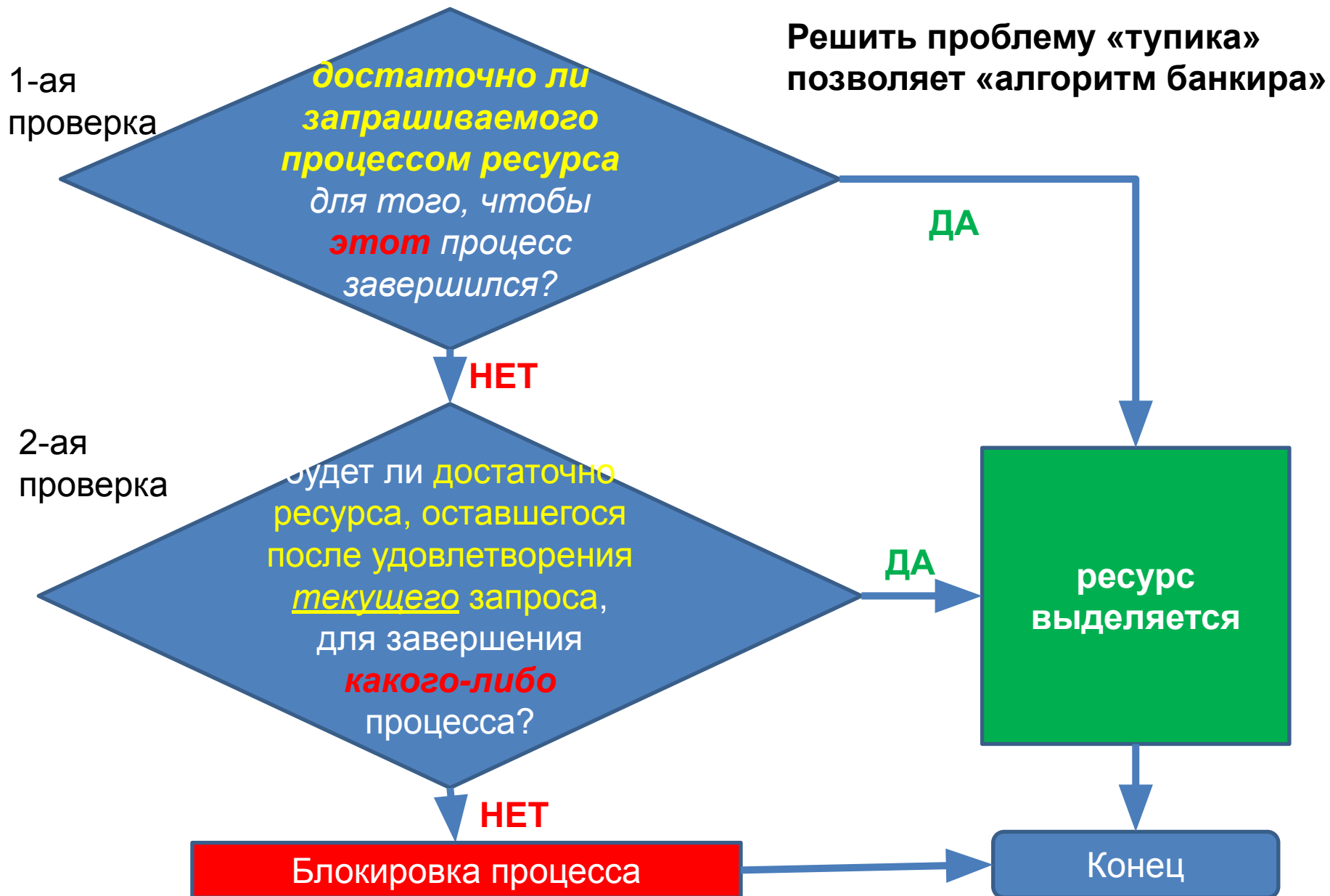
Известны максимальные потребности в ресурсах каждого из процессов и общее число ресурсов в системе.

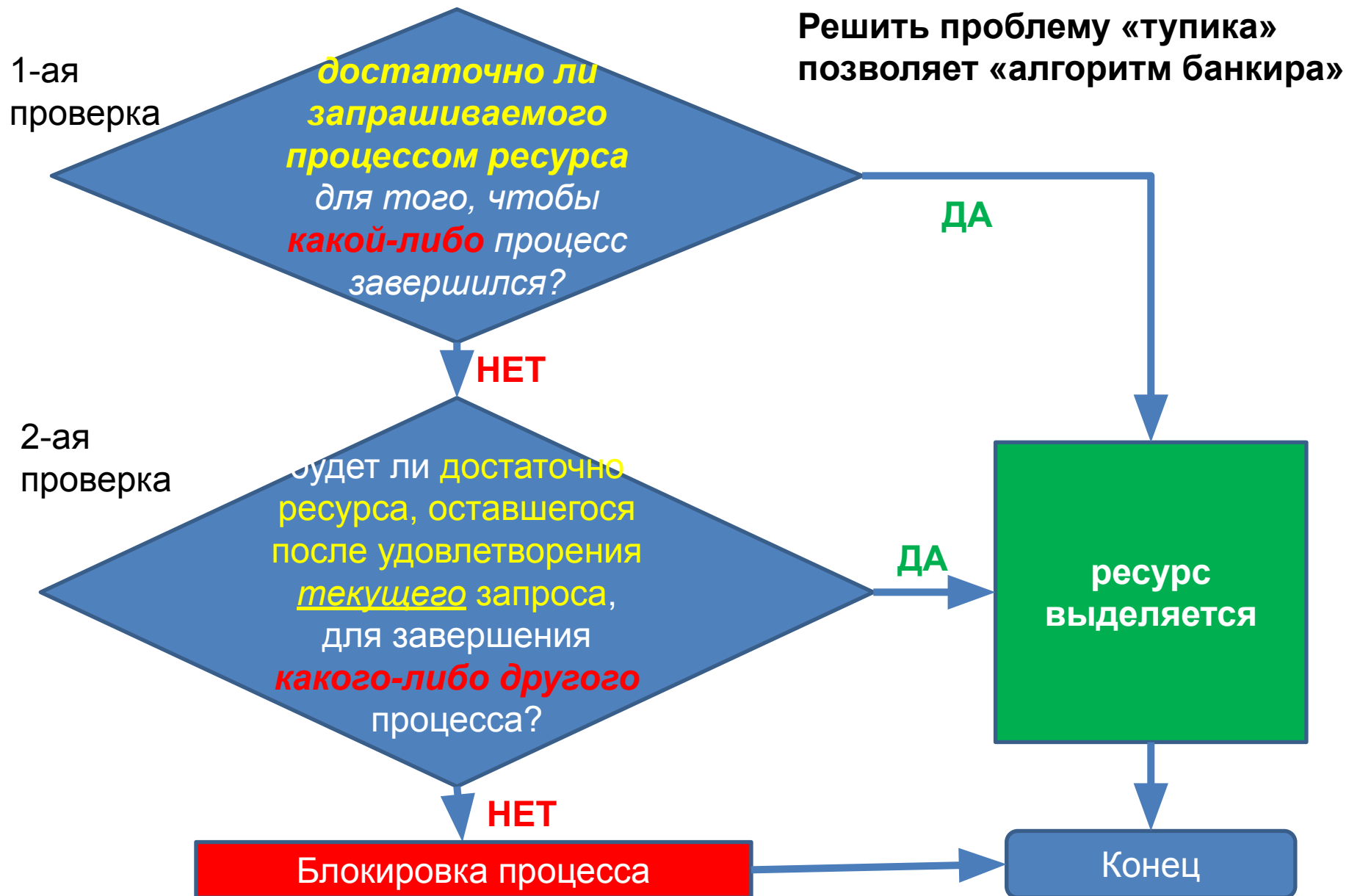
После завершения процесса происходит возврат в систему всего освобождённого ресурса.

Необходимо запрограммировать обслуживание запросов так, чтобы не возникала «тупиковая» ситуация.



«Алгоритм банкира»





1-ая
проверка

**достаточно ли
запрашиваемого
процессом ресурса**
для того, чтобы
какой-либо процесс
завершился?

Решить проблему «тупика»
позволяет «алгоритм банкира»

ДА

Какое отношение имеют
другие процессы к
текущему запросу?

НЕТ

2-ая
проверка

будет ли **достаточно**
ресурса, оставшегося
после удовлетворения
текущего запроса,
для завершения
какого-либо другого
процесса?

ДА

ресурс
выделяется

На каком основании
исключается
рассмотрение
дальнейшего развитие
текущего процесса?

НЕТ

блокировка процесса

Конец

Исходное состояние системы

процесс	имеет	мак
A	0	6
B	0	5
C	0	4
D	0	7
свободно	10	

запрос: A -> 1

процесс	имеет	мак
A	0	6
B	0	5
C	0	4
D	0	7
свободно	10	

процесс	имеет	мак	потребность
A	1	6	5
B	0	5	5
C	0	4	4
D	0	7	7
свободно	9		

запрос: B -> 1

процесс	имеет	мак
A	1	6
B	0	5
C	0	4
D	0	7
свободно	9	

процесс	имеет	мак	потребность
A	1	6	5
B	1	5	4
C	0	4	4
D	0	7	7
свободно	8		

1) достаточно ли запрашиваемого процессом ресурса для того, чтобы **ЭТОТ** процесс завершился? если да, ресурс выделяется, если нет, выполняется вторая проверка;

2) будет ли достаточно ресурса, оставшегося после удовлетворения текущего запроса, для завершения **какого-либо** процесса?

Если **да**, **запрос удовлетворяется** (ресурс выделяется); если **нет**, **процесс**, породивший запрос, **блокируется**.

запрос: C -> 1

процесс	имеет	маж
A	1	6
B	1	5
C	2	4
D	4	7
свободно	2	

процесс	имеет	маж	потребность
A	1	6	5
B	1	5	4
C	3	4	1
D	4	7	3
свободно	1		

Безопасное состояние

запрос: B -> 1

процесс	имеет	маж
A	1	6
B	1	5
C	2	4
D	4	7
свободно	2	

процесс	имеет	маж	потребность
A	1	6	5
B	2	5	3
C	2	4	2
D	4	7	3
свободно	1		

«Тупик»

1) достаточно ли запрашиваемого процессом ресурса для того, чтобы **ЭТОТ** процесс завершился? если да, ресурс выделяется, если нет, выполняется вторая проверка;

2) будет ли достаточно ресурса, оставшегося после удовлетворения текущего запроса, для завершения **какого-либо** процесса?

Если **да**, **запрос** удовлетворяется (ресурс выделяется); если **нет**, **процесс**, породивший запрос, **блокируется**.

Ограничения на широкое практическое применение «алгоритм банкира».

Часто для процессов неизвестно максимальное количество требующихся им ресурсов