Модель диспетчера задач операционной системы

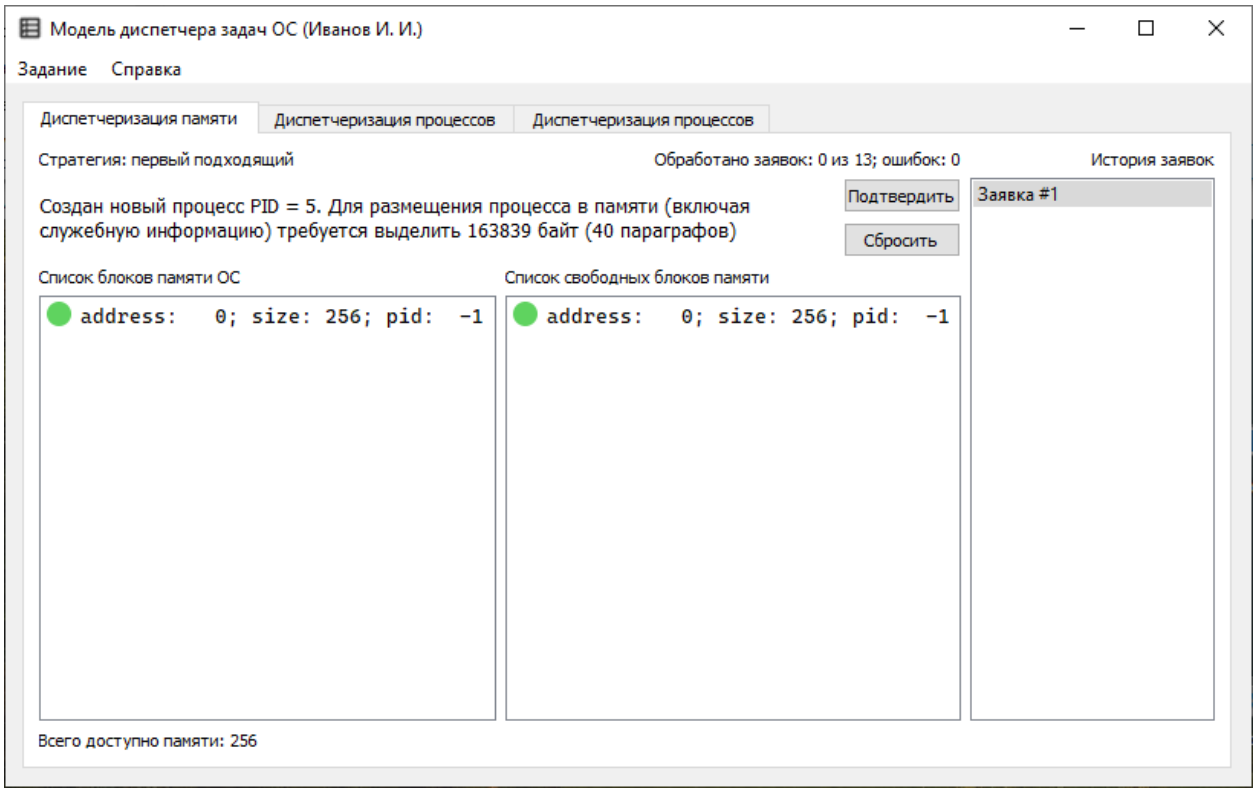
Руководство пользователя

Али Рзаев

Содержание

Интерфейс пользователя

Диспетчер памяти



Вид окна диспетчера памяти

В верхней части окна присутствует описание заявки и количество выполненных заявок. Задача пользователя — корректно обработать поступающие заявки.

В левой части окна находится список блоков памяти. Свободные блоки помечены зеленым кружком, а занятые - красным. Справа от него находится список свободных блоков памяти, который необходимо поддерживать в упорядоченном согласно дисциплине порядке.

Для каждого блока указан его начальный адрес, размер (в параграфах), а также идентификатор процесса, владеющего данным блоком (-1 — нет владельца).

Пользователь имеет возможность выполнить следующие действия:

- Ответить на запрос отказом, сразу нажав на кнопку "Подтвердить".
- Выделить память процессу. Для этого необходимо щелкнуть правой кнопкой мыши на свободном блоке в списке блоков памяти ОС и выбрать пункт «Выделить приложению». После этого в появившемся окне нужно ввести идентификатор процесса, которому выделяется память, а также количество выделяемых параграфов.
- Освободить память. Для этого необходимо щелкнуть правой кнопкой мыши на

занятом блоке в списке блоков памяти ОС и выбрать пункт «Освободить».

- Объединить два свободных блока в один. Для этого необходимо щелкнуть правой кнопкой мыши на свободном блоке в списке блоков памяти ОС и выбрать пункт «Объединить со следующим», Данное действие необходимо выполнять каждый раз, когда образуются соседние свободные блоки.
- Уплотнить (дефрагментировать память). Для этого необходимо щелкнуть правой кнопкой мыши на списке блоков памяти ОС и выбрать пункт «Уплотнение памяти».
- Подтвердить действия, нажав на кнопку "Подтвердить".
- Отменить все действия, произведенные при текущей заявке, нажав на кнопку "Сбросить".

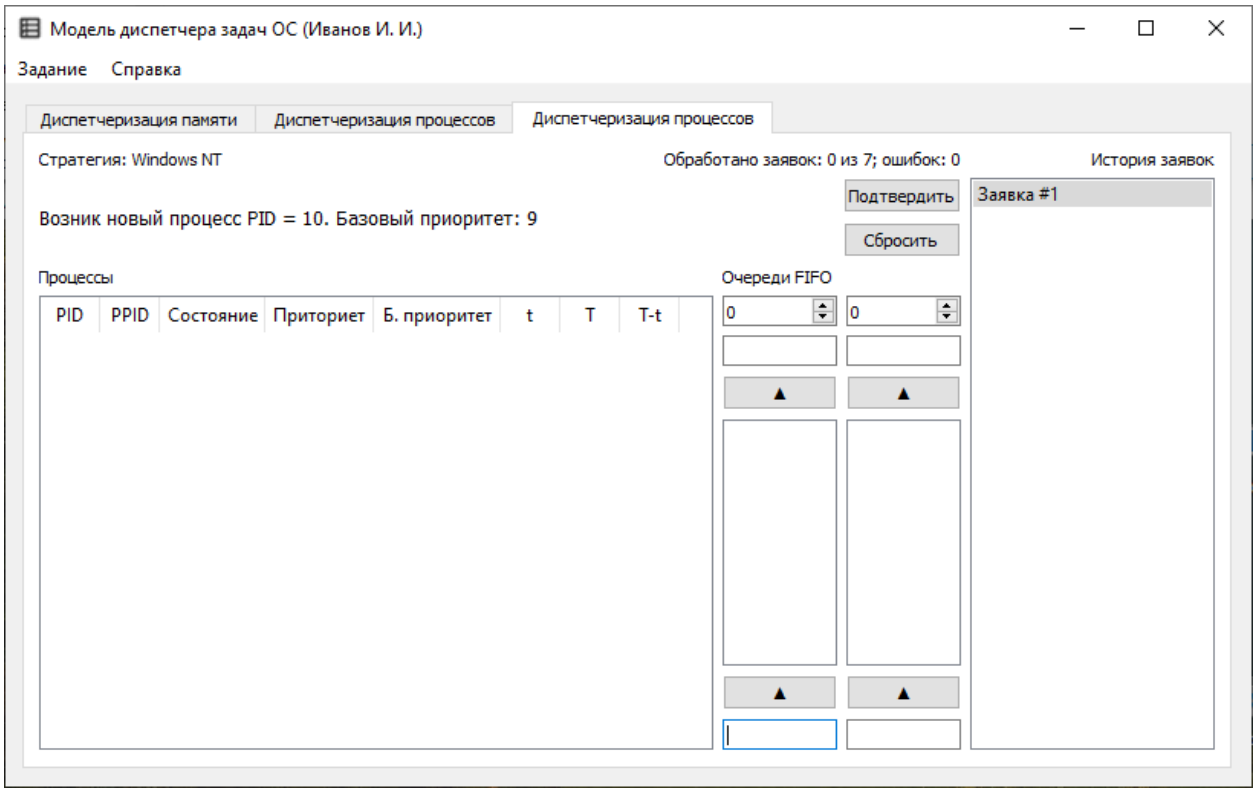
В правой части окна находится список выполненных шагов, который хранит историю действий пользователя. При наведении указателя мыши на элемент списка отображается информация о действиях пользователя на данном шаге, а при щелчке загружается состояние памяти на конец обработки заявки. Изменять состояние памяти в уже обработанных заявках нельзя.

По достижении заданного числа шагов выводится сообщение об успешном выполнении задания.

Сочетания клавиш:

- Кнопка "Подтвердить" - `Alt+Enter`
- Кнопка "Сбросить" - `Ctrl+Z`

Диспетчер процессов



Вид окна диспетчера процессов

Так же, как и для окна диспетчера памяти, в верхней части окна присутствует описание заявки.

В центральной части окна находится список процессов, который содержит следующую информацию:

- Идентификатор процесса — расположен в столбце "PID"
- Идентификатор родительского процесса — расположен в столбце "PPID"
- Состояние процесса; отображается буквой и значком в столбце "Состояние":
  - "E" — процесс выполняется
  - "Q" — процесс готов к выполнению
  - "W" — процесс обратился к устройствам ввода/вывода и ожидает завершения обмена
- Текущий приоритет процесса — расположен в столбце "Приоритет"
- Базовый приоритет процесса (в системе Windows NT) — расположен в столбце "Б. приоритет"
- Время выполнения процесса (инкрементируется только когда процесс выполняется) — находится в столбце "t"
- Предполагаемое время выполнения процесса — находится в столбце "T"
- Разность между предполагаемым временем и реальным временем выполнения

процесса (по сути — время, оставшееся до завершения процесса) — находится в столбце "T-t". Отрицательное значение говорит о том, что процесс превысил лимит времени.

Справа от списка процессов находятся два окна очередей. В любом из них можно отобразить любую очередь (с номерами от 0 до 15). Для добавления значения в конец очереди, необходимо ввести его в нижнее окно ввода и нажать на нижнюю кнопку со стрелкой либо нажать клавишу "Enter". Для извлечения значения из очереди необходимо нажать на верхнюю кнопку со стрелкой. Извлеченное значение помещается в окно ввода над кнопкой. В левом окне отображения очереди имеется возможность изменять порядок элементов в очереди путем перетаскивания мышью.

Пользователь может выполнить следующие действия:

- Ответить на запрос отказом, сразу нажав на кнопку "Подтвердить".
- Добавить процесс в список процессов — щелкнуть правой кнопкой мыши на списке процессов и выбрать пункт "Добавить" в контекстном меню, после чего ввести параметры создаваемого процесса: идентификатор, идентификатор родителя (если родителя нет — -1), значение базового и текущего приоритетов, предполагаемое время выполнения, и нажать кнопку "OK". В зависимости от дисциплины некоторые поля будут неактивны.
- Добавить идентификатор процесса в очередь.
- Поменять порядок элементов в очереди.
- Прочитать значение из начала очереди.
- Удалить процесс из списка процессов и из очереди — щелкнуть правой кнопкой мыши на нужном элементе в списке процессов и в проявившемся меню выбрать пункт "Удалить".
- Переключить процесс в состояние ожидания — щелкнуть правой кнопкой мыши на нужном элементе в списке процессов и в проявившемся меню выбрать пункт "Переключить в состояние ожидания".
- Переключить процесс в состояние готовности — щелкнуть правой кнопкой мыши на нужном элементе в списке процессов и в проявившемся меню выбрать пункт "Переключить в состояние готовности".
- Выбрать процесс для выполнения — щелкнуть правой кнопкой мыши на нужном элементе в списке процессов и в проявившемся меню выбрать пункт "Переключиться".
- Подтвердить действия, нажав на кнопку "Подтвердить".
- Отменить все действия, произведенные при текущей заявке, нажав на кнопку "Сбросить".

В правой части окна находится список выполненных шагов, который хранит историю действий пользователя. При наведении указателя мыши на элемент списка отображается информация о действиях пользователя на данном шаге, а при щелчке

загружается состояние процессов на конец обработки заявки. Изменять состояние процессов в уже обработанных заявках нельзя.

По достижении заданного числа шагов выводится сообщение об успешном выполнении задания.

Сочетания клавиш:

- Кнопка "Подтвердить" - `Alt+Enter`
- Кнопка "Сбросить" - `Ctrl+Z`
- Добавление процесса в очередь - `Enter` в поле ввода

Задания

Диспетчер памяти

События

События, которые должен обработать диспетчер памяти, и порядок их обработки.

После обработки каждого события необходимо упорядочить свободные блоки согласно заданной дисциплине.

Создание нового процесса

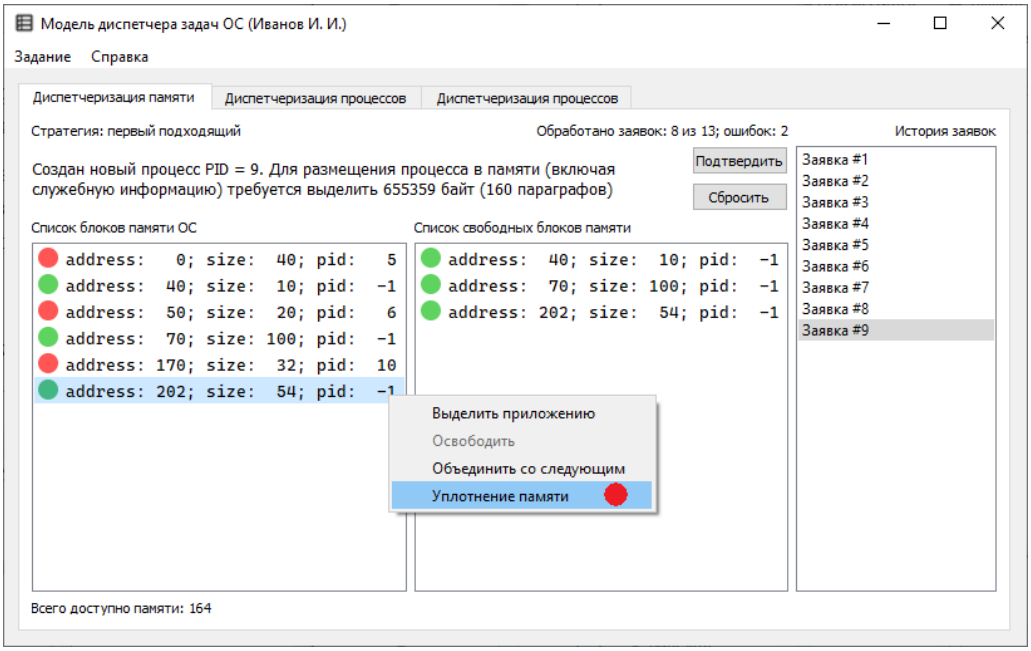
Ограничения:

- Процесс с таким PID не должен существовать (иными словами, процессу не выделен ни один блок памяти)
- Свободной памяти должно быть не меньше, чем требуется процессу

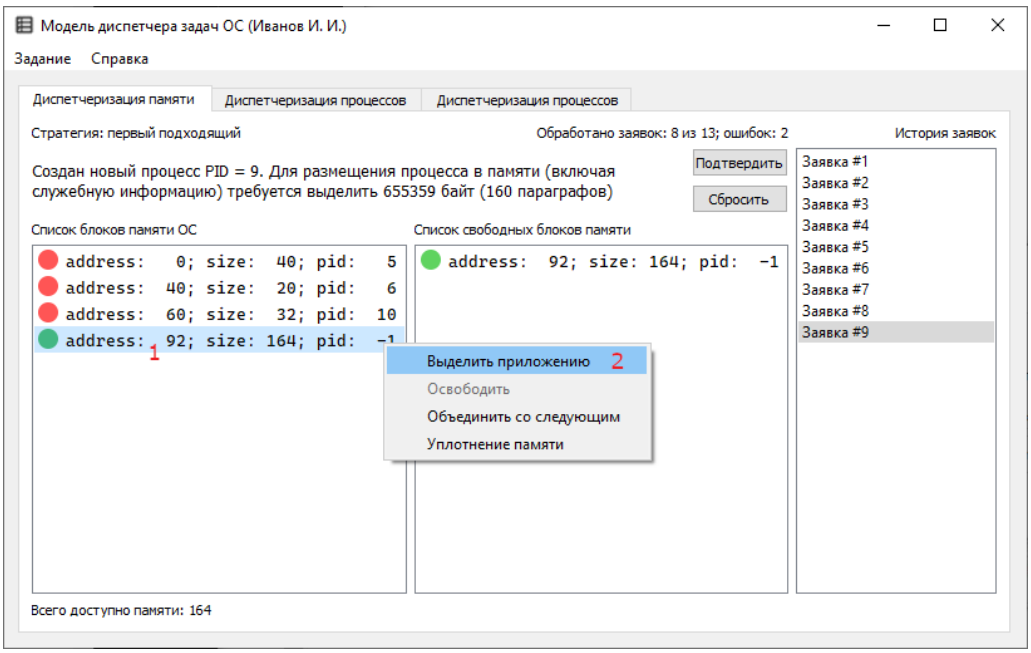
Алгоритм обработки события:

1. Проверить, есть ли свободный блок подходящего размера. Если есть, то выделить процессу память в этом блоке и завершить алгоритм
2. Если свободного блока подходящего размера нет, но суммарно памяти достаточно, то выполнить дефрагментацию, в новом свободном блоке выделить память процессу и завершить алгоритм
3. В противном случае пропустить заявку

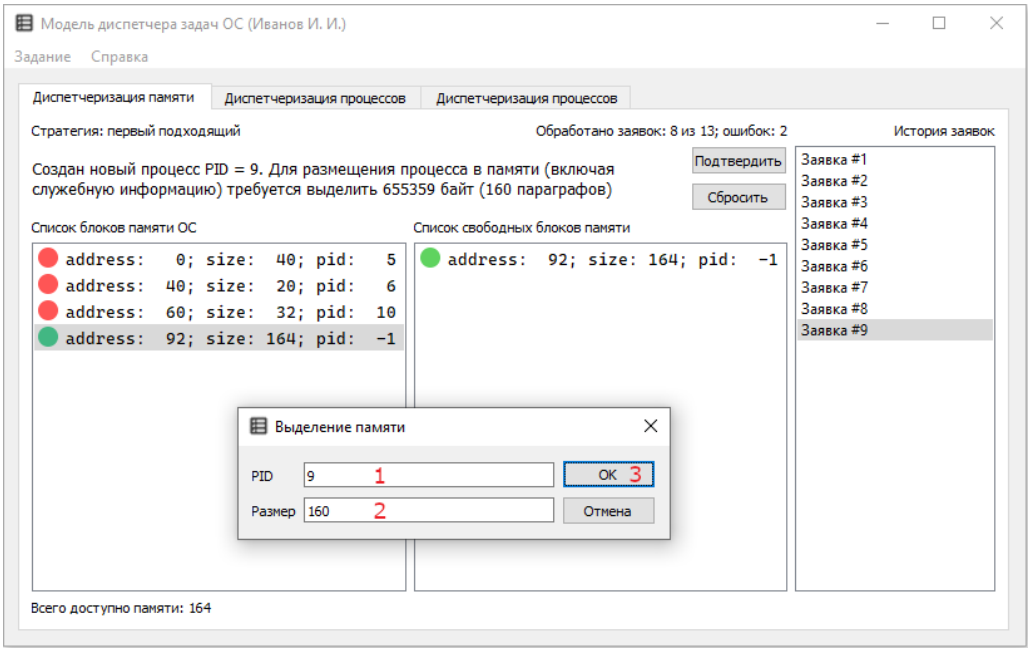
Пошаговая обработка заявки представлена на рисунках ниже.



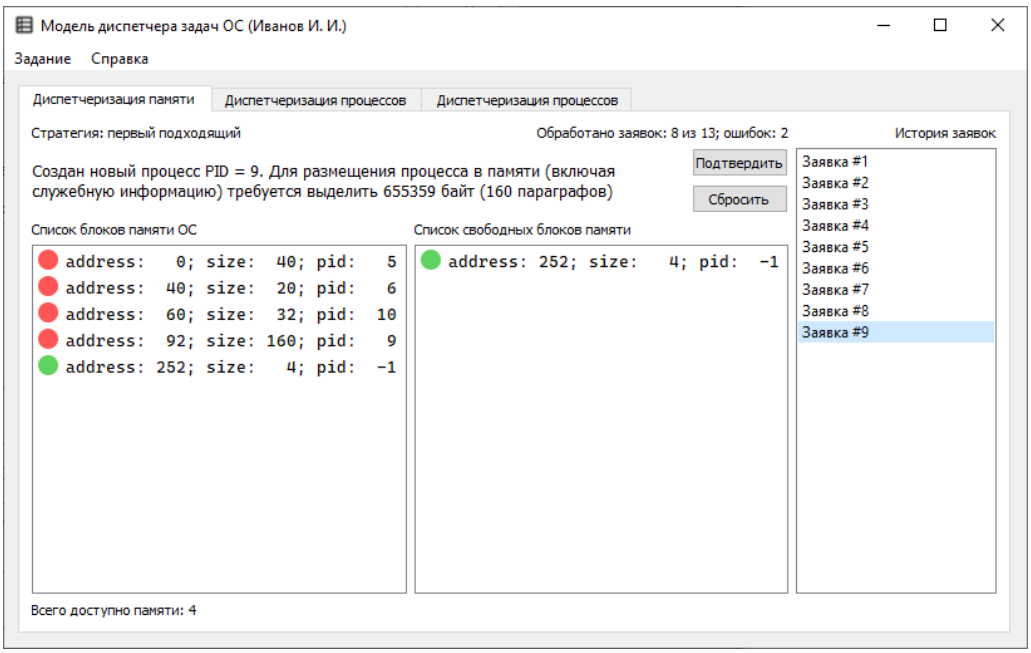
Создание нового процесса - Дефрагментация



Создание нового процесса - Выделение блока



Создание нового процесса - Ввод данных



Создание нового процесса - Конечный результат

Завершение процесса

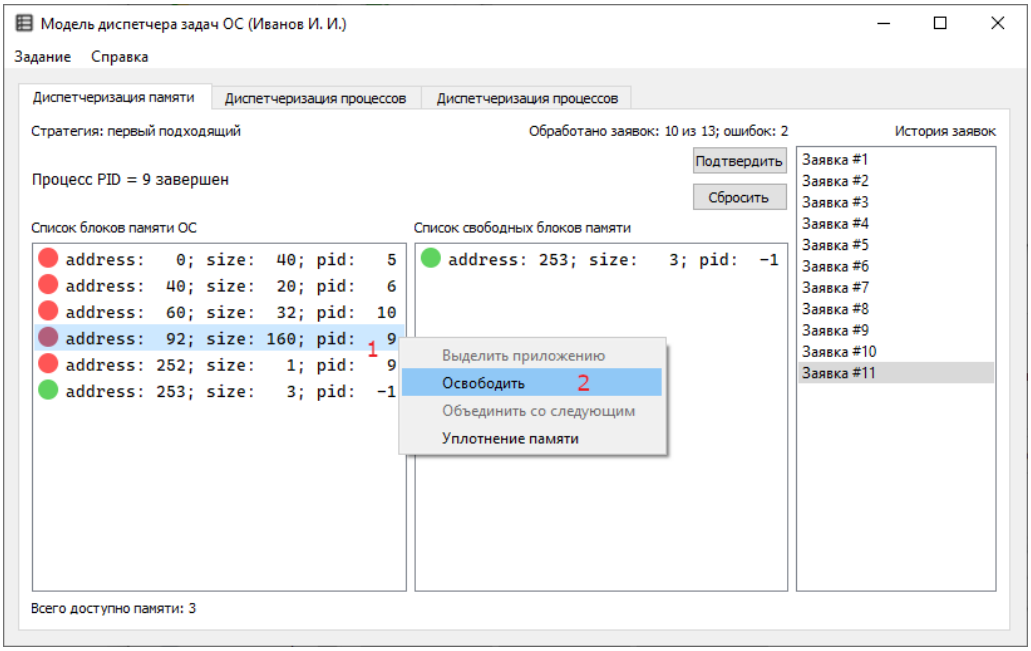
Ограничения:

- Процесс с таким PID должен существовать (иными словами, процессу выделен хотя бы один блок памяти)

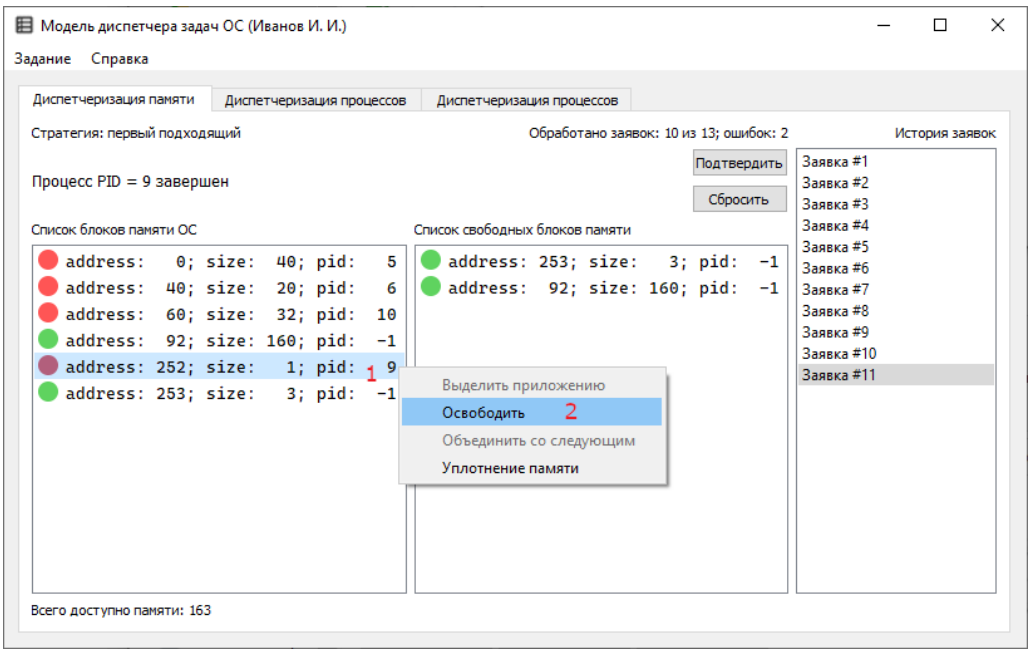
Алгоритм обработки события:

1. Найти все блоки памяти, выделенные данному процессу
2. Освободить эти блоки
3. Объединить, если такие есть, соседние свободные блоки

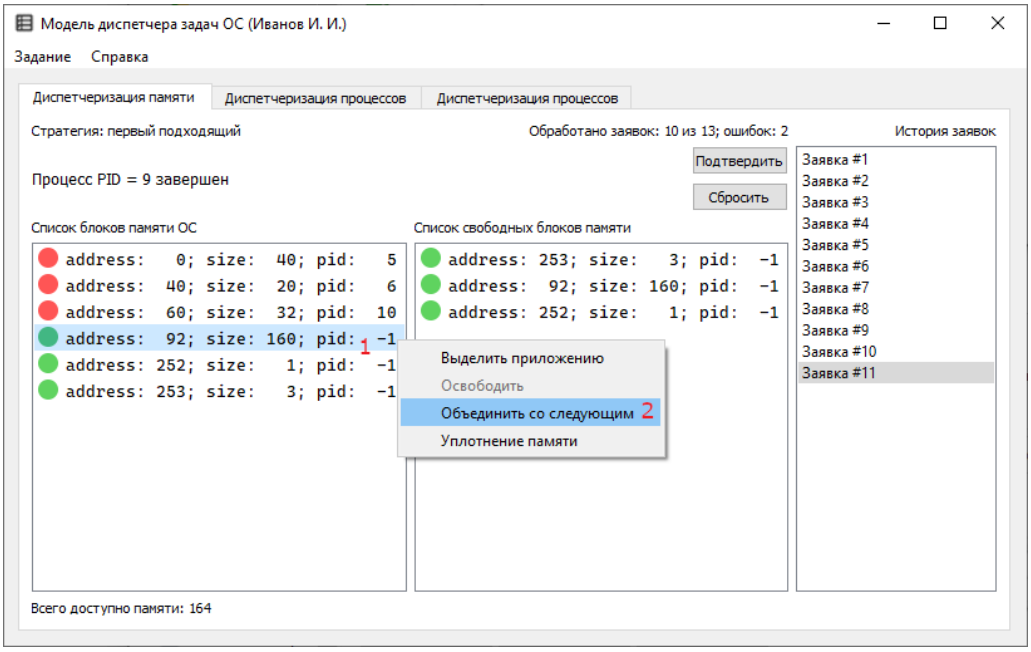
Пошаговая обработка заявки представлена на рисунках ниже.



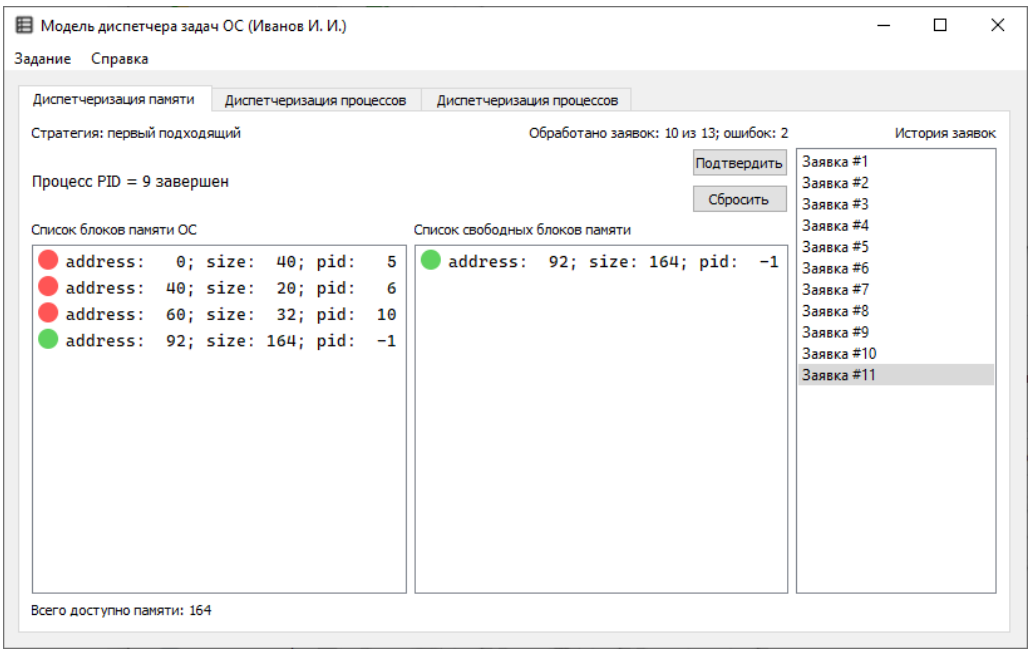
Завершение процесса - Освобождение блока



Завершение процесса - Освобождение блока



Завершение процесса - Объединение блоков



Завершение процесса - Конечный результат

Выделение памяти процессу

Ограничения:

- Процесс с таким PID должен существовать (иными словами, процессу выделен хотя бы один блок памяти)
- Свободной памяти должно быть не меньше, чем требуется процессу

Алгоритм обработки события: см. Создание нового процесса.

Освобождение памяти

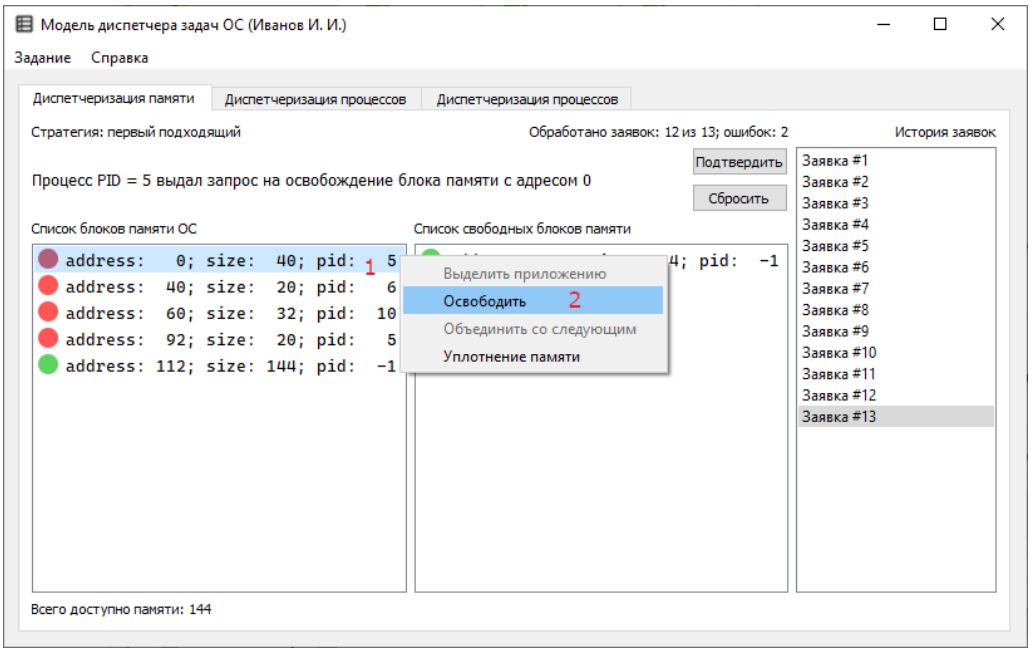
Ограничения:

- Процесс с таким PID должен существовать (иными словами, процессу выделен хотя бы один блок памяти)
- Блок по указанному адресу должен принадлежать данному процессу

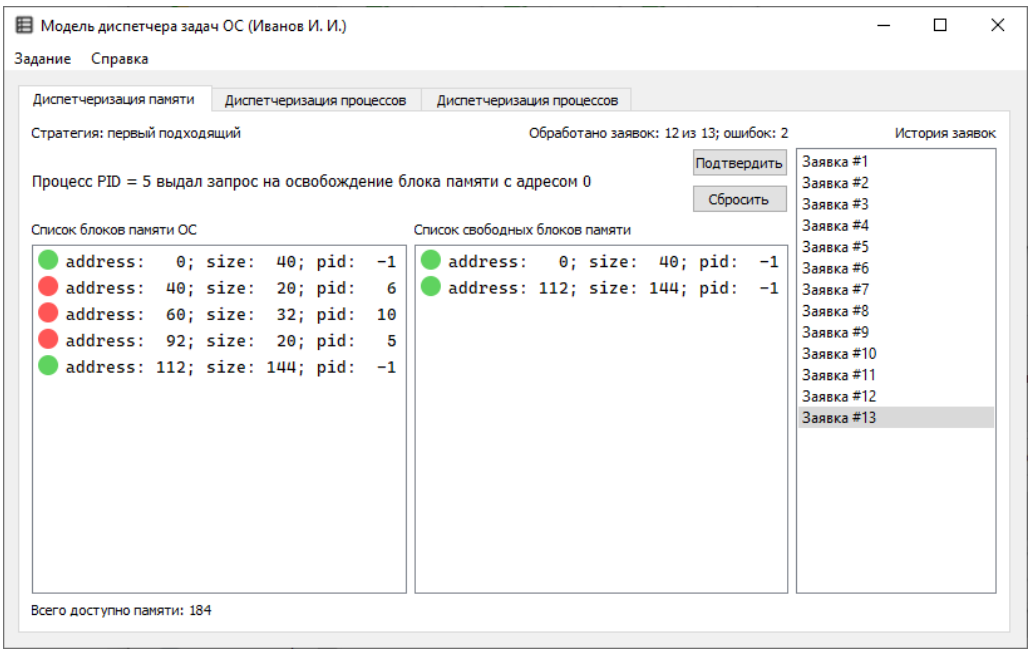
Алгоритм обработки события:

1. Освободить блок памяти
2. Объединить, если такие есть, соседние свободные блоки

Пошаговая обработка заявки представлена на рисунках ниже.



Освобождение памяти - Освобождение блока



Освобождение памяти - Конечный результат

Дисциплины

Дисциплины выделения памяти процессам

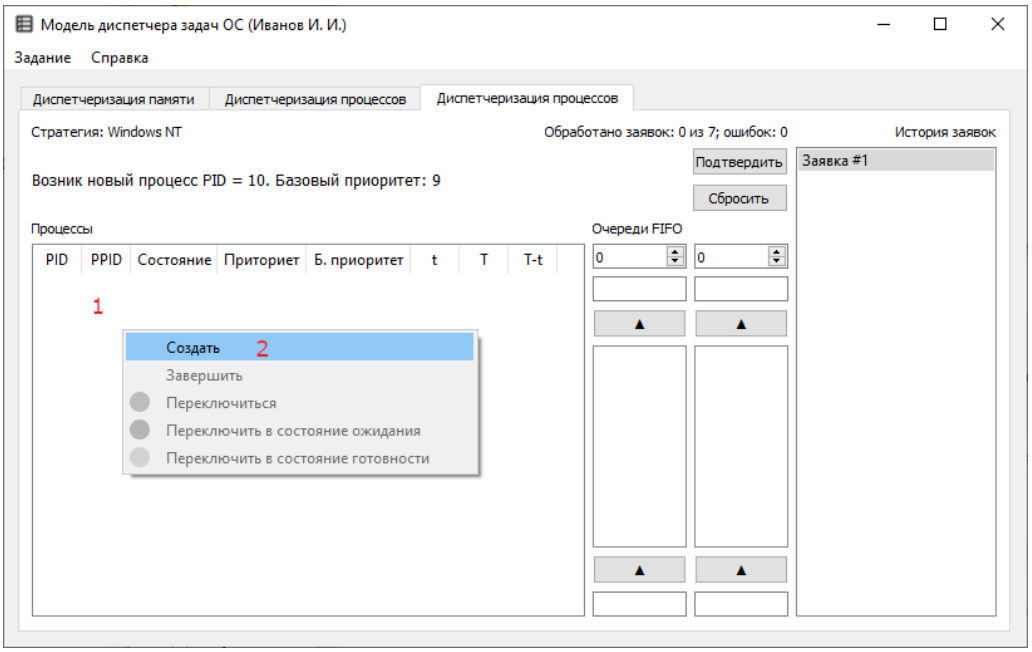
- Первый подходящий. Свободные блоки памяти сортируются в порядке **увеличения** начальных адресов
- Наиболее подходящий. Свободные блоки памяти сортируются по их размерам в порядке **возрастания**, а затем в порядке **увеличения** начальных адресов
- Наименее подходящий. Свободные блоки памяти сортируются по их размерам в порядке **убывания**, а затем в порядке **увеличения** начальных адресов

Диспетчер процессов

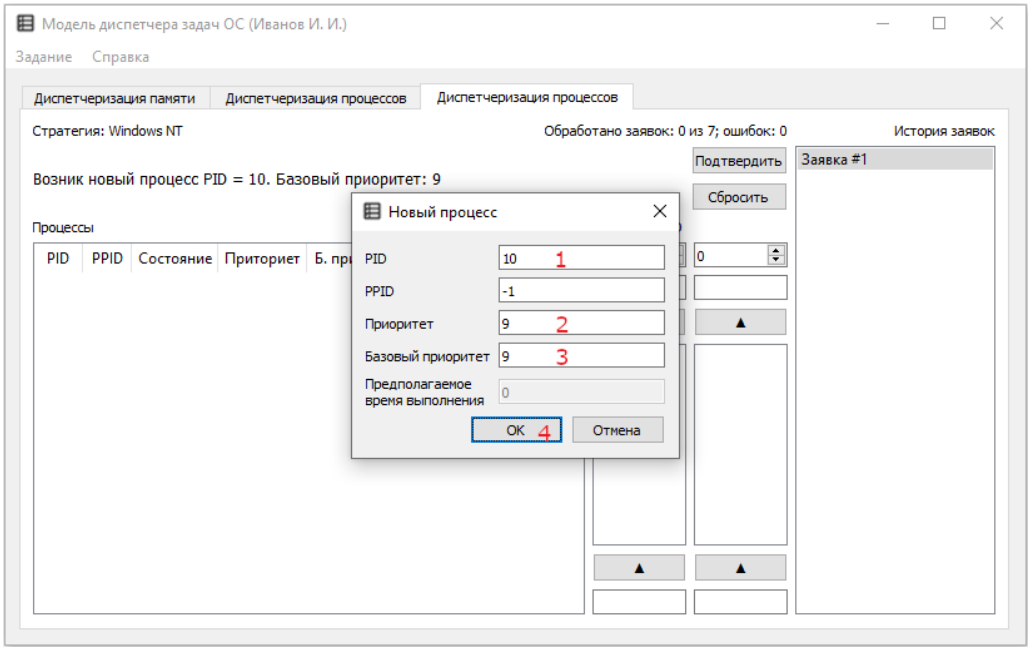
Выполнение типовых действий

На рисунках ниже представлено пошаговое выполнение типовых действий в лабораторной установке.

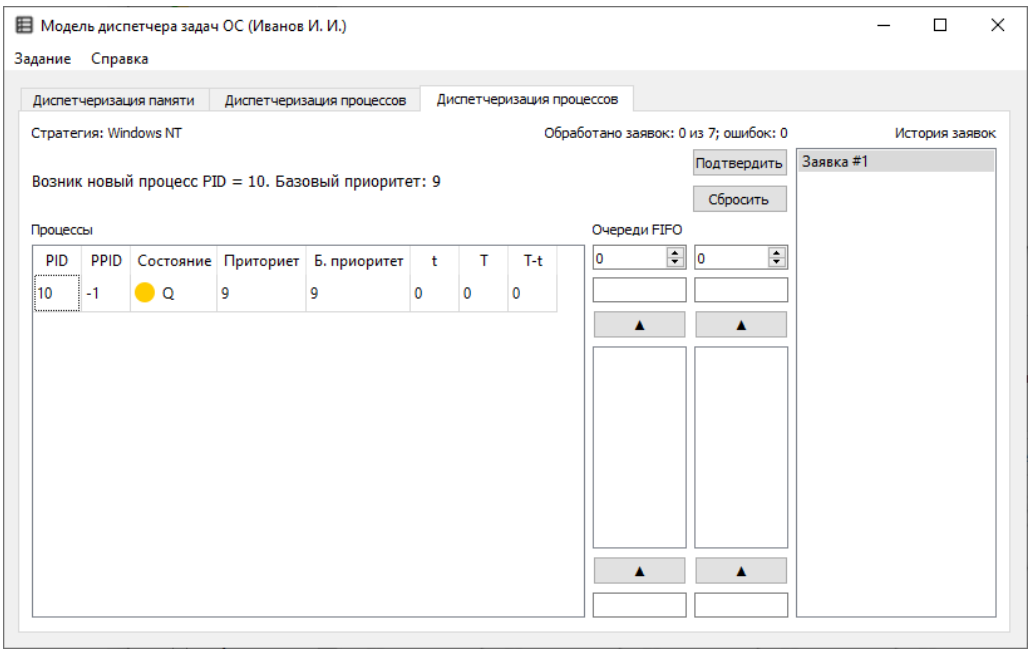
Создание нового процесса



Создание нового процесса - Вызов контекстного меню

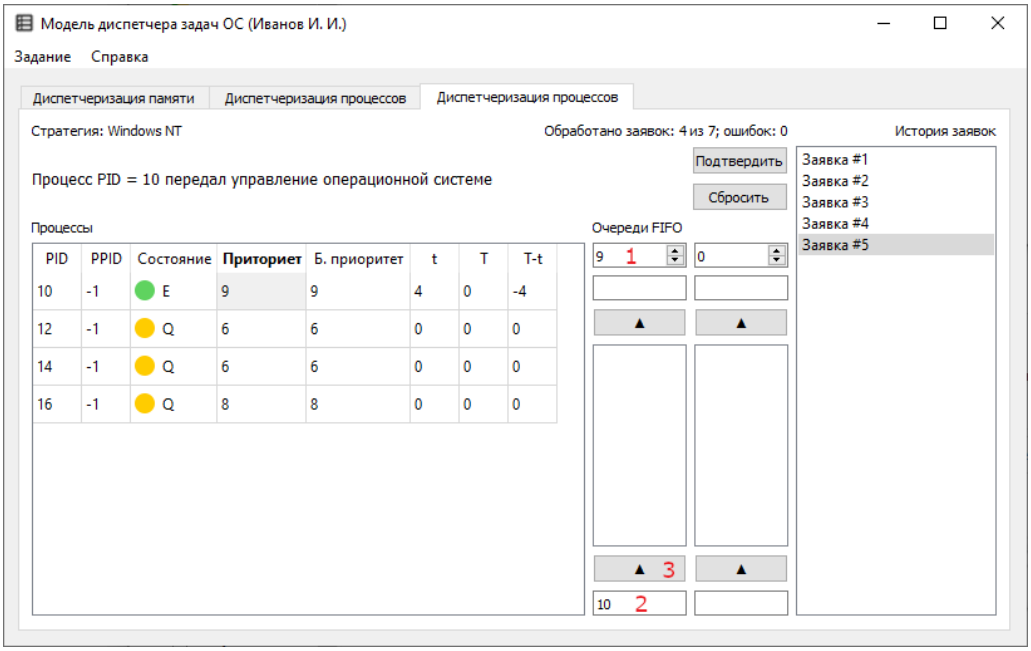


Создание нового процесса - Ввод данных

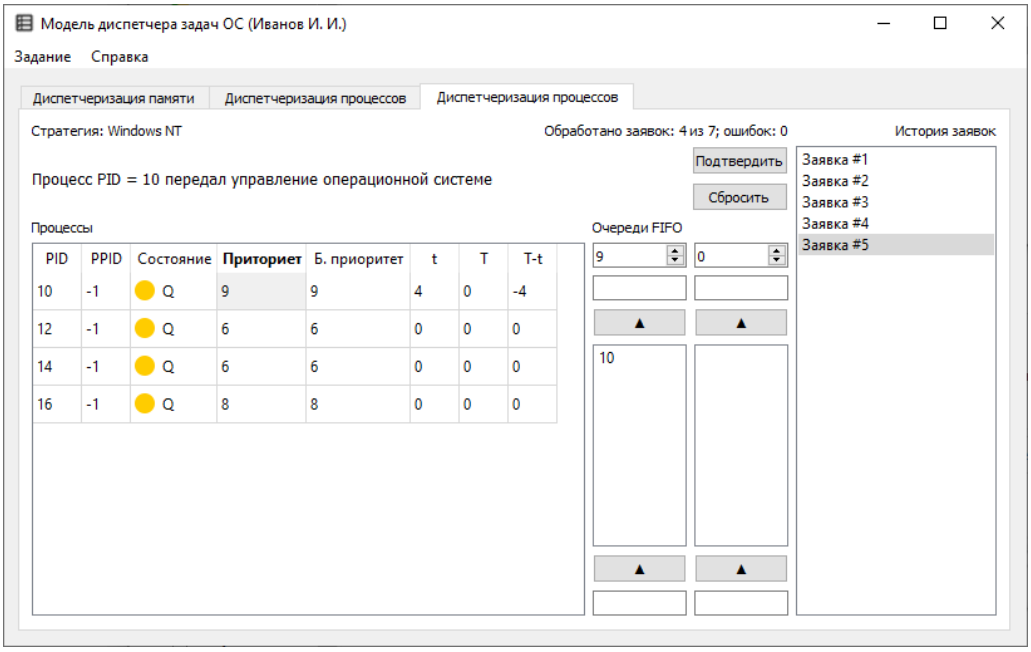


Создание нового процесса - Конечный результат

Добавление процесса в очередь

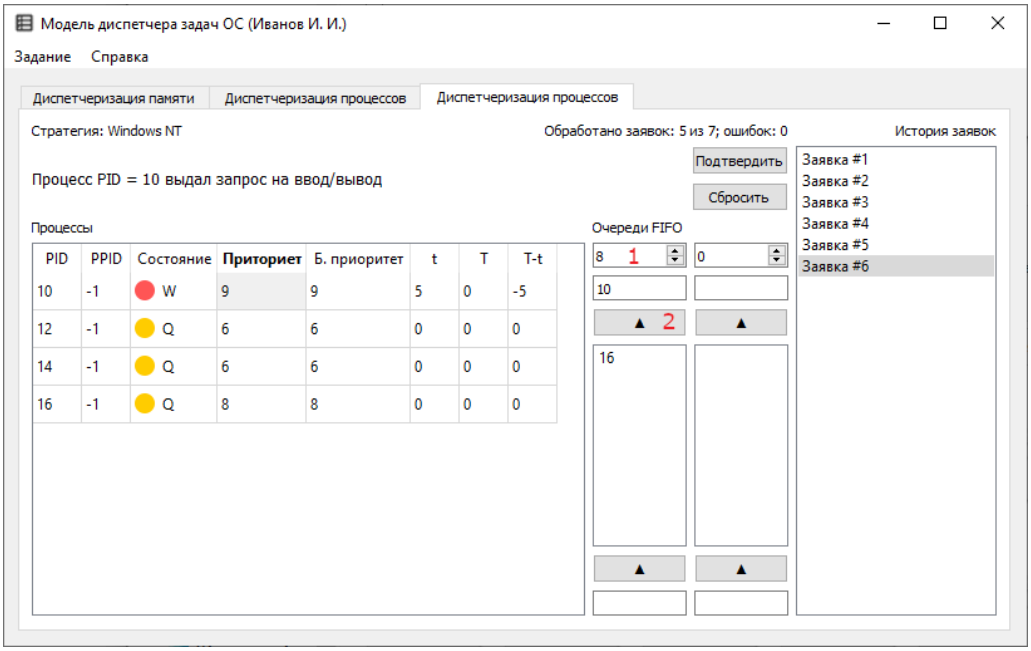


Добавление процесса в очередь - Ввод данных

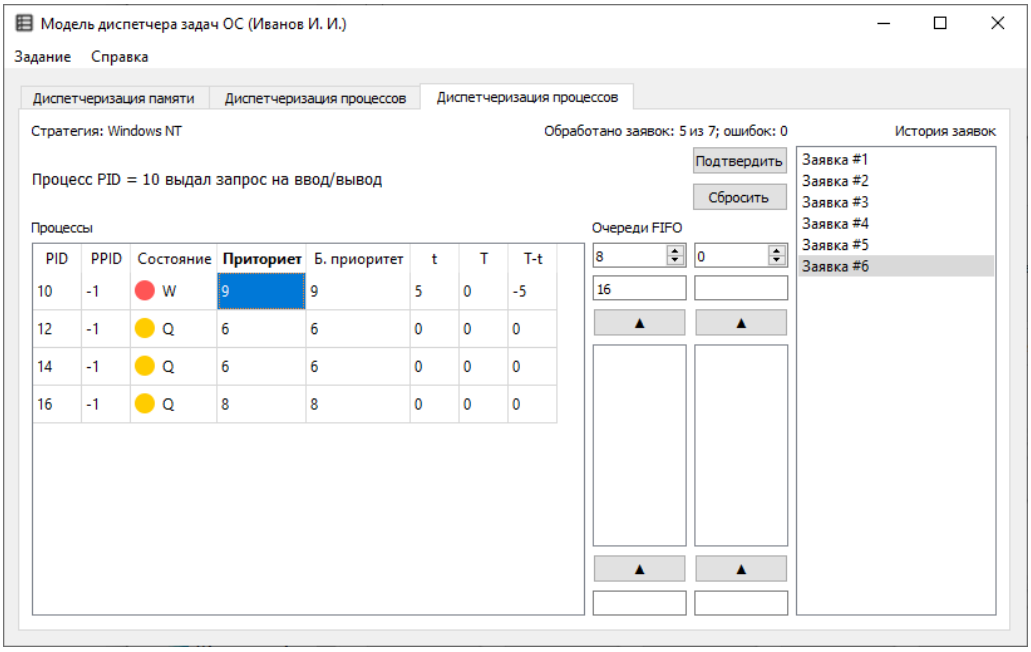


Добавление процесса в очередь - Конечный результат

Извлечение процесса из очереди

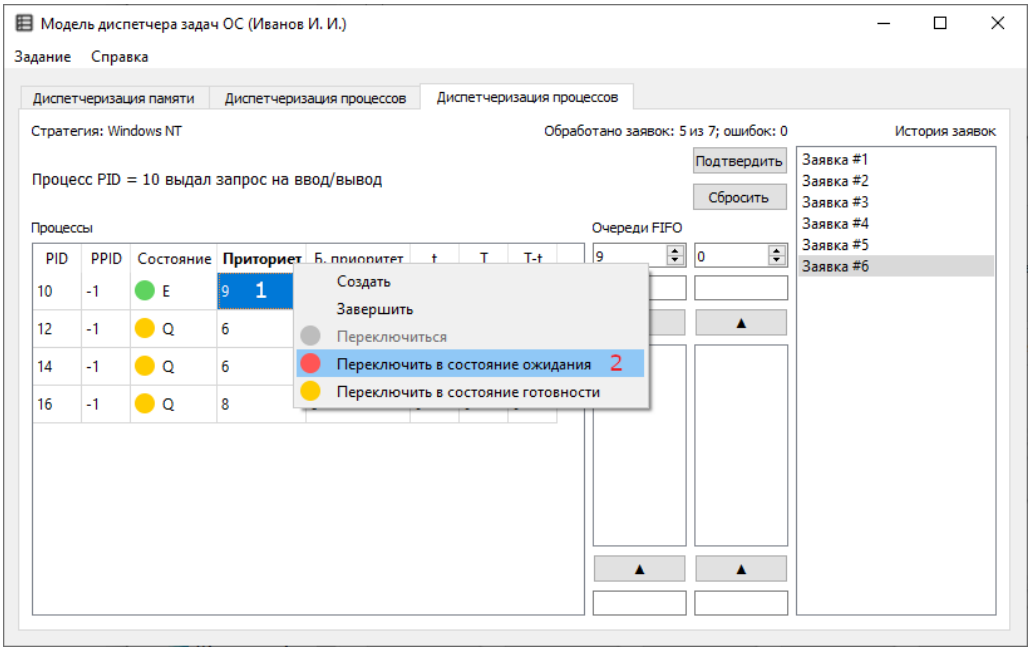


Извлечение процесса из очереди - Выбор очереди

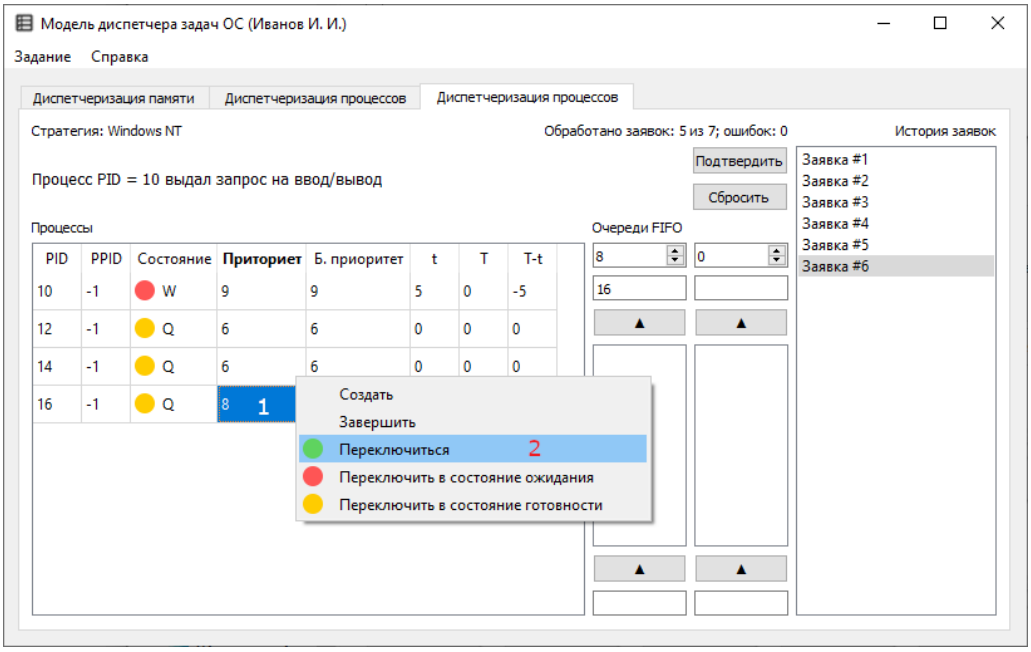


Извлечение процесса из очереди - Конечный результат

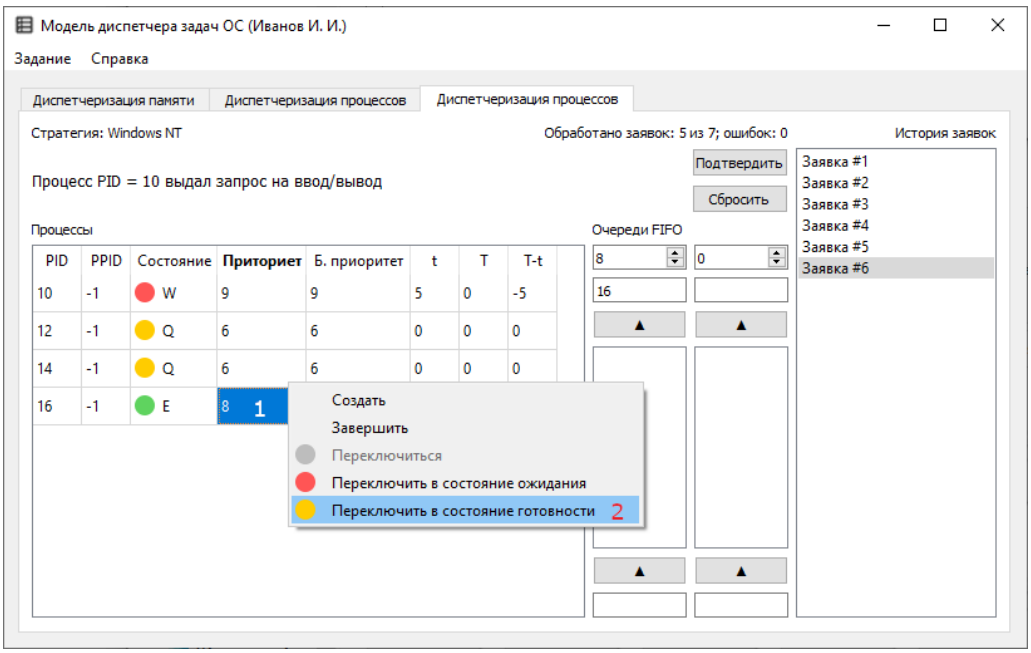
Изменение состояния процесса



Переключение в состояние "Ожидание"

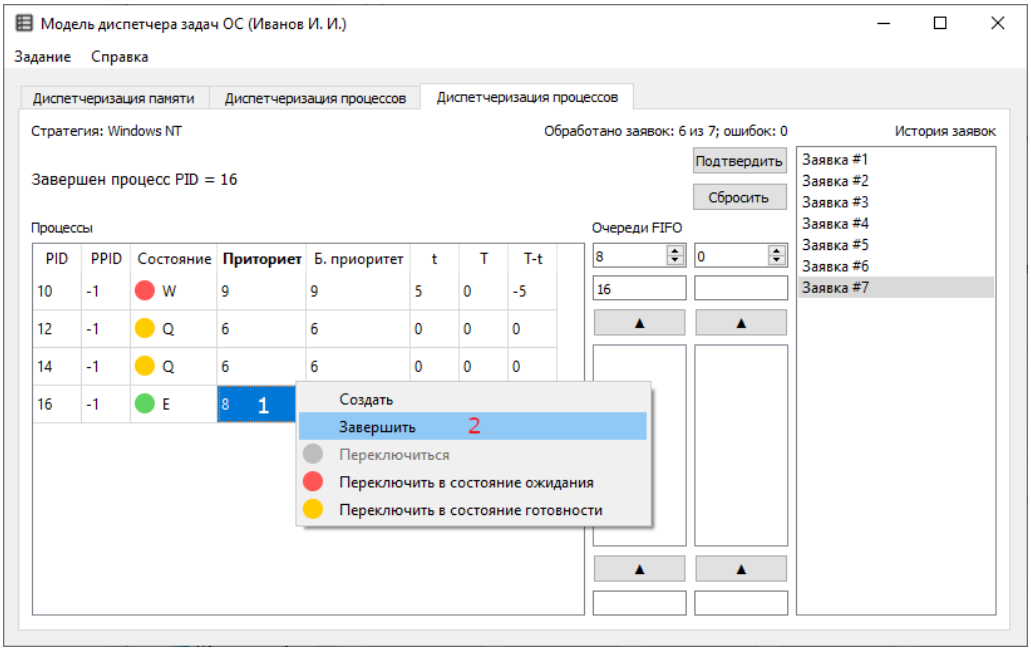


Переключение в состояние "Исполняется"

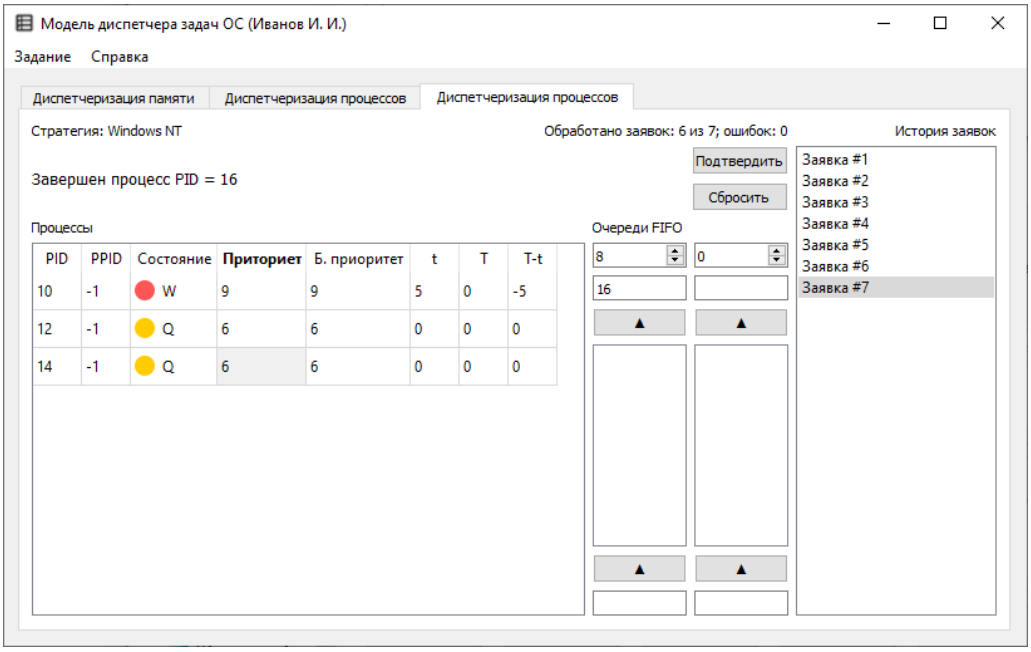


Переключение в состояние "Готов к исполнению"

Завершение процесса



Завершение процесса - Вызов контекстного меню



Завершение процесса - Конечный результат

События

События, которые должен обработать диспетчер процессов, и порядок их обработки.

Создание нового процесса

Ограничения:

- Процесс с таким PID не должен существовать

Алгоритм обработки события:

1. Создать процесс с данными значениями, состояние — `Готов к исполнению`
2. Добавить созданный процесс в соответствующую очередь
3. Если у созданного процесса приоритет ниже, чем у исполняющегося на данный момент, то ничего не делать (алгоритм завершен)
4. Если на данный момент некоторый процесс исполняется, то добавить его в соответствующую очередь
5. Выбрать процесс для исполнения (согласно дисциплине), извлечь его из соответствующей очереди
6. Переключить его в состояние `Исполняется`

Создание дочернего процесса

Ограничения:

- Процесс с таким PID не должен существовать
- Дочерний процесс может быть порожден только процессом, выполняющимся в данный момент

Алгоритм обработки события: см. Создание нового процесса.

Завершение процесса

Ограничения:

- Процесс с таким PID должен существовать
- При завершении процесса удаляются и его дочерние процессы

Алгоритм обработки события:

1. Удалить данный процесс и его дочерние процессы из списка процессов
2. Если данный процесс или один из его дочерних процессов был в состоянии `Исполняется`, то нужно выполнить переключение на новый процесс (см. п. 3)
3. Выбрать процесс для исполнения (согласно дисциплине), извлечь его из соответствующей очереди
4. Переключить его в состояние `Исполняется`

Запрос на ввод/вывод

Ограничения:

- Процесс с таким PID должен существовать
- Процесс должен находится в состоянии `Исполняется`

Алгоритм обработки события:

1. Переключить данный процесс в состояние `Ожидание`
2. Выбрать процесс для исполнения (согласно дисциплине), извлечь его из соответствующей очереди
3. Переключить его в состояние `Исполняется`

Завершение ввода/вывода

Ограничения:

- Процесс с таким PID должен существовать
- Процесс должен находится в состоянии `Ожидание`

Алгоритм обработки события:

1. Добавить данный процесс в соответствующую очередь
2. Переключить его в состояние `Готов к исполнению`
3. Если у данного процесса приоритет ниже, чем у исполняющегося на данный момент, то ничего не делать (алгоритм завершен)
4. Если на данный момент некоторый процесс исполняется, то добавить его в соответствующую очередь
5. Выбрать процесс для исполнения (согласно дисциплине), извлечь его из соответствующей очереди
6. Переключить его в состояние `Исполняется`

Передача управления операционной системе

Ограничения:

- Процесс с таким PID должен существовать
- Процесс должен находится в состоянии `Исполняется`

Алгоритм обработки события:

1. Добавить данный процесс в соответствующую очередь
2. Выбрать процесс для исполнения (согласно дисциплине), извлечь его из соответствующей очереди
3. Переключить его в состояние `Исполняется`

Истечение кванта времени

Алгоритм обработки события:

1. Добавить исполняемый на данный момент процесс в соответствующую очередь
2. Выбрать процесс для исполнения (согласно дисциплине), извлечь его из соответствующей очереди
3. Переключить его в состояние `Исполняется`

Дисциплины

Невытесняющие дисциплины:

- First Come First Serve (FCFS)
- Shortest Job Next (SJN)
- Shortest Remaining Time (SRT)

Вытесняющие дисциплины:

- Round-robin (RR)
- Windows NT
- Unix
- Linux O(1)

Round-robin

Используется только очередь 0, которая заполняется в порядке прибывания. Отсутствуют какие-либо приоритеты.

First Come First Serve

Аналогична RoundRobin, за исключением того, что новые процессы добавляются в очередь 1.

Shortest Job Next

Используется только очередь 0. После помещения процесса в очередь необходимо переместить его на соответствующее место. Для дисциплины SJN вперед помещаются процессы с меньшим заданным временем выполнения. При этом, если время выполнения процесса превышает заданное, ему назначается штраф. При помещении в очередь такой процесс не перемещается, а остальные процессы, даже с большим временем, при помещении в очередь ставятся перед ним.

Shortest Remaining Time

Аналогична SJN, за исключением того, что в очередь вперед помещаются процессы с меньшим оставшимся временем выполнения.

Windows NT

При добавлении процесса в очередь его приоритет уменьшается на единицу, при этом приоритет не может быть меньше базового приоритета.

По завершении операций ввода/вывода ожидающему процессу назначается некоторая добавка к приоритету (он помещается в более приоритетную очередь).

Unix

Чем дольше работает процесс без операций ввода/вывода, тем меньше становится его приоритет (уменьшается каждые 2 такта на 1). По завершении ввода/вывода приоритет увеличивается на 1. Приоритеты 0-7 - динамические, 8-15 - статические.

При создании процесса или при завершении ввода/вывода процесса с более высоким приоритетом, чем у активного, переключение на данный процесс не происходит.

При истечении кванта времени или передаче управления ОС процессорное время передается процессу с высшим приоритетом, стоящим первым в очереди с самым высоким приоритетом.

Linux O(1)

Используются две очереди: 0 - "активная" и 1 - "просроченная". В "активную" очередь добавляются новые процессы, а также процессы, которые завершили ввод-вывод или передали управление ОС. В "просроченную" очередь добавляются процессы, которые не уложились в отведенный квант времени.

При выборе процесса для запуска используют следующий алгоритм:

1. Если "активная" очередь пуста, то перенести в нее все процессы из "просроченной" очереди
2. Извлечь из "активной" очереди процесс и запустить его