ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

"ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту по курсу “Операционные системы”

Тема работы:

“Проектирование гипотетической операционной системы и программная эмуляции отдельных модулей”

Руководители: Чернышова А.В.

Московченко А.В.

(подпись) (дата)

Разработал: Саевский О.В.

ст. гр. ПИ-19а (подпись) (дата)

Донецк – 2021

РЕФЕРАТ

Отчёт по курсовому проекту содержит: 75 страниц, 25 рисунков, 23 таблицы, 4 приложения, 4 источника.

Объект исследования – операционная система, файловая система, планировщик процессов, межпроцессное взаимодействие.

Цель – спроектировать гипотетическую операционную систему, разработать эмуляцию отдельных её модулей.

Результат – проект гипотетической операционной системы, программы эмуляции файловой системы, планировщика процессов и командного интерпретатора.

ОПЕРАЦИОННАЯ СИСТЕМА, ФАЙЛОВАЯ СИСТЕМА, ПЛАНИРОВЩИК ПРОЦЕССОВ, МЕЖПРОЦЕССНОЕ ВЗАИМОДЕЙСТВИЕ, КОМАНДНЫЙ ИНТЕРПРЕТАТОР

СОДЕРЖАНИЕ

# ВВЕДЕНИЕ

Операционная система (ОС) – комплекс [программ](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0), обеспечивающий управление аппаратными средствами [компьютера](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80), организующий работу с [файлами](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB) и выполнение [прикладных программ](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), осуществляющий ввод и вывод данных.

На сегодняшний день, операционная система — это первый и основной набор программ, загружающийся в компьютер. Помимо вышеуказанных функций ОС может осуществлять и другие, например, предоставление общего пользовательского интерфейса. Операционные системы современных компьютеров относятся к числу наиболее сложных систем, созданных когда-либо человеком.

Полученные знания и навыки при проектировании гипотетической ОС являются основой для принятия профессиональных решений при проектировании средств управления в различной операционной обстановке.

# 1 СТРУКТУРА ПРОЕКТИРУЕМОЙ ФАЙЛОВОЙ СИСТЕМЫ

Файловая система – система, которая контролирует и определяют, как будут храниться и именоваться данные на носителе или накопителе информации. От неё зависит способ хранения информации, формат данных и то, как они будут считываться и записываться.

## 1.1 Общая организация файловой системы

Логическая структура диска, которую формирует проектируемая ФС, представлена в таблице 1.1.

Таблица 1.1 – Логическая структура диска

|  |  |  |  |
| --- | --- | --- | --- |
| Служебные области | Суперблок | FAT 1 | |
| FAT 2 | Область данных со списком пользователей | Корневой каталог | Область данных |

Суперблок является начальной точкой файловой системы. Информация, хранимая в суперблоке, используется для организации доступа к остальным данным на диске. Структура суперблока проектируемой ФС представлена в таблице 1.2.

Таблица 1.2 – Структура суперблока

|  |  |
| --- | --- |
| Поле | Размер |
| Тип файловой системы | 5 байт |
| Размер кластера | 4 байта |
| Кол-во FAT таблиц | 2 байта |
| Размер FAT таблицы | 4 байта |

Продолжение таблицы 1.2

|  |  |
| --- | --- |
| Размер области с пользователями | 4 байта |
| Размер корневого каталога | 4 байта |
| Размер области данных | 4 байта |
| Всего: | 27 байт |

Структура одной записи таблицы FAT представлена в таблице 1.3.

Таблица 1.3 – Структура записи таблицы FAT

|  |  |
| --- | --- |
| Поле | Размер |
| Флаг кластера (свободен или завершает файл), либо указатель на следующий кластер файла | 4 байта |

Структура каталога представлена в таблице 1.4.

Таблица 1.4 – Структура каталога

|  |  |
| --- | --- |
| Поле | Размер |
| Имя файла | 10 байт |
| Тип файла | 1 байт |
| Права доступа | 2 байта |
| Дата и время создания | 8 байт |
| Номер начального кластера | 4 байта |
| Размер файла | 4 байта |
| Id владельца | 4 байта |
| Всего: | 33 байта |

Структура поля «Права доступа» представлена в таблице 1.5.

Таблица 1.5 – Структура поля «Атрибуты файла»

|  |  |
| --- | --- |
| Бит | Описание |
| 1 | Чтение файла пользователем-владельцем |
| 2 | Запись файла пользователем-владельцем |
| 3 | Исполнение файла пользователем-владельцем |
| 4 | Чтение файла другими пользователями |
| 5 | Запись файла другими пользователями |
| 6 | Исполнение файла другими пользователями |
| 7 | Зарезервировано |
| 8 | Зарезервировано |
| 9 | Зарезервировано |
| 10 | Зарезервировано |

## 1.2 Виртуальные страницы

Любая программная система имеет логическую модель памяти. Самая простая из них – совпадающая с физической, когда все программы имеют прямой доступ ко всему адресному пространству.  
При таком подходе программы имеют доступ ко всему адресному пространству. Они могут не только мешать друг другу, но и способны привести к сбою работы всей системы. Для этого достаточно, например, затереть кусок памяти, в котором располагается код ОС.

Виртуальная память – это концепция, которая позволяет уйти от использования физических адресов, используя вместо них виртуальные адреса. При этом вся виртуальная память делится на участки памяти постоянного размера, называемые страницами.

## 1.3 Команды для работы с ФС

Для работы с файловой системой используются команды, представленные в таблице 1.6.

Таблица 1.6 – Команды для работы с ФС

|  |  |  |
| --- | --- | --- |
| Имя команды | Аргументы | Описание |
| new.user | Никнейм, Пароль | Добавление нового пользователя |
| sign.in | Никнейм, Пароль | Авторизация пользователя |
| sign.out | - | Выход из сессии |
| read.dir | Название файла | Чтение директории |
| read.file | Название файла | Чтение файла |
| remove.user | Никнейм | Удаление пользователя |
| create.dir | Название файла | Создает каталог с заданным названием |
| create.file | Название файла | Создает файл с заданным названием |
| format.disk | - | Форматирует диск |
| open.dir | Название файла | Открывает каталог с заданным названием |
| write.file | Название файла | Записывает данные в файл. С флагом (-а) выполняет дозапись |
| remove.file | Название файла | Удаляет файл с заданным названием |
| remove.dir | Название файла | Удаляет каталог с заданным названием |
| rename.file | Название файла, Название файла | Изменяет название файла с первого аргумента на второй |
| rename.dir | Название файла, Название файла | Изменяет название каталога с первого аргумента на второй |
| copy.file | Название файла, Название файла | Копирует содержимое файла с именем первого аргумента, в файл с аргументом второго |
| move.file | Название файла, Название файла | Перемещает содержимое файла с именем первого аргумента, в файл с аргументом второго |
| chmod.file | Название файла | Изменяет права доступа файла, с заданным названием |
| chmod.dir | Название файла | Изменяет права доступа каталога, с заданным названием |

## 1.4 Способы организации файлов

Для организации доступа к файлу ФС должна иметь сведения о номерах кластеров, где размещается каждый файл или данные этого файла. В этом ей помогает FAT-таблица (File Allocation Table).

Значение кластера 0 означает, что кластер свободен и в него можно записывать данные, а значение 1 – кластер занят и является концом одного из файлов. В иных случаях последовательность байт показывает номер следующего кластера файла.

Проектируемая файловая система является многоуровневой с поддержкой файлов и каталогов с последовательным и прямым доступом.

## 1.6 Алгоритмы работы некоторых команд ФС.

Алгоритм работы команды create.file (создание файла), изображён на рисунке 1.1.

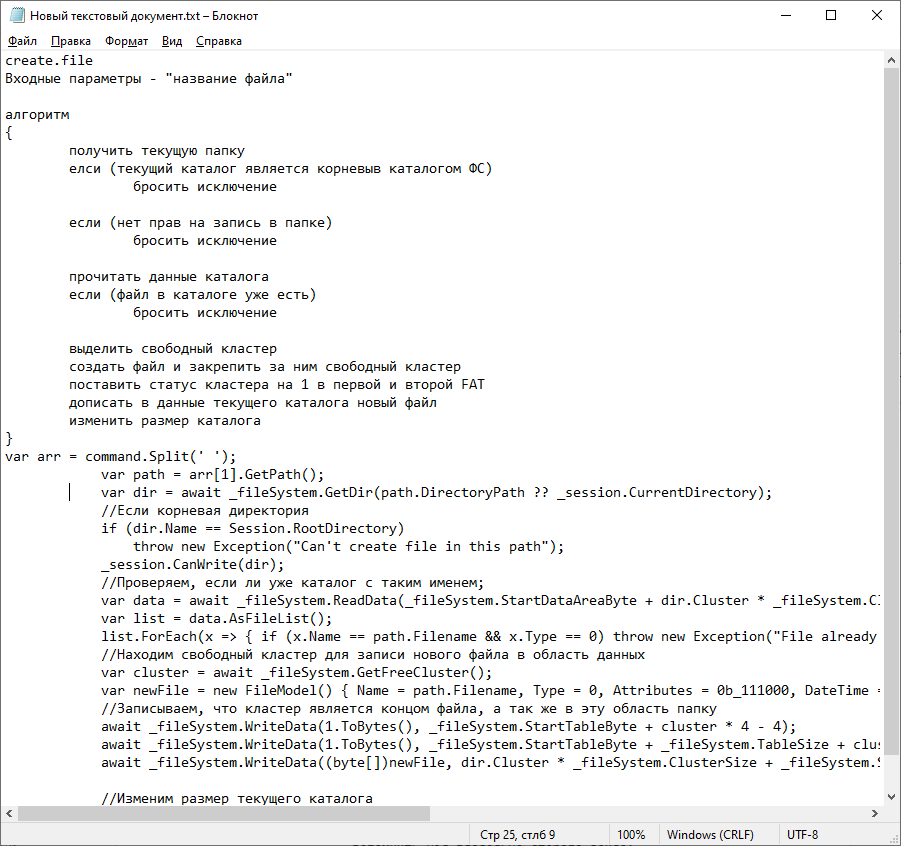


Рисунок 1.1 – Алгоритм работы команды create.file

Алгоритм работы команды write.file (запись в файл), изображён на рисунке 1.2.

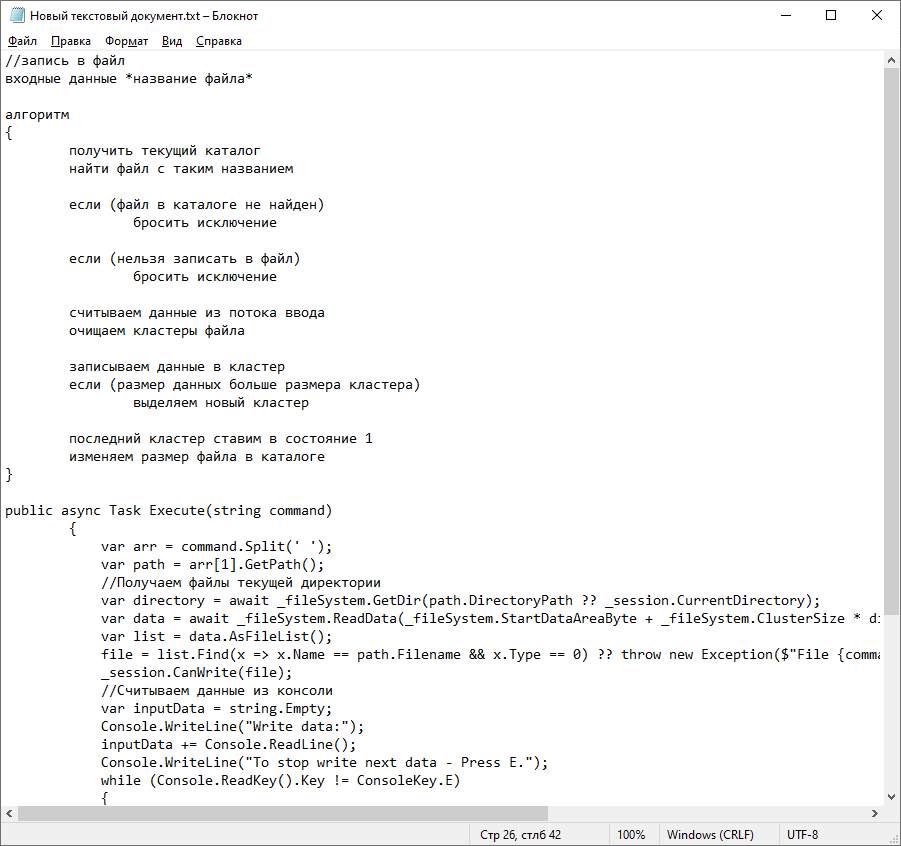


Рисунок 1.2 – Алгоритм работы команды write.file

Алгоритм работы команды remove.file (удаление файла), изображён на рисунке 1.3.

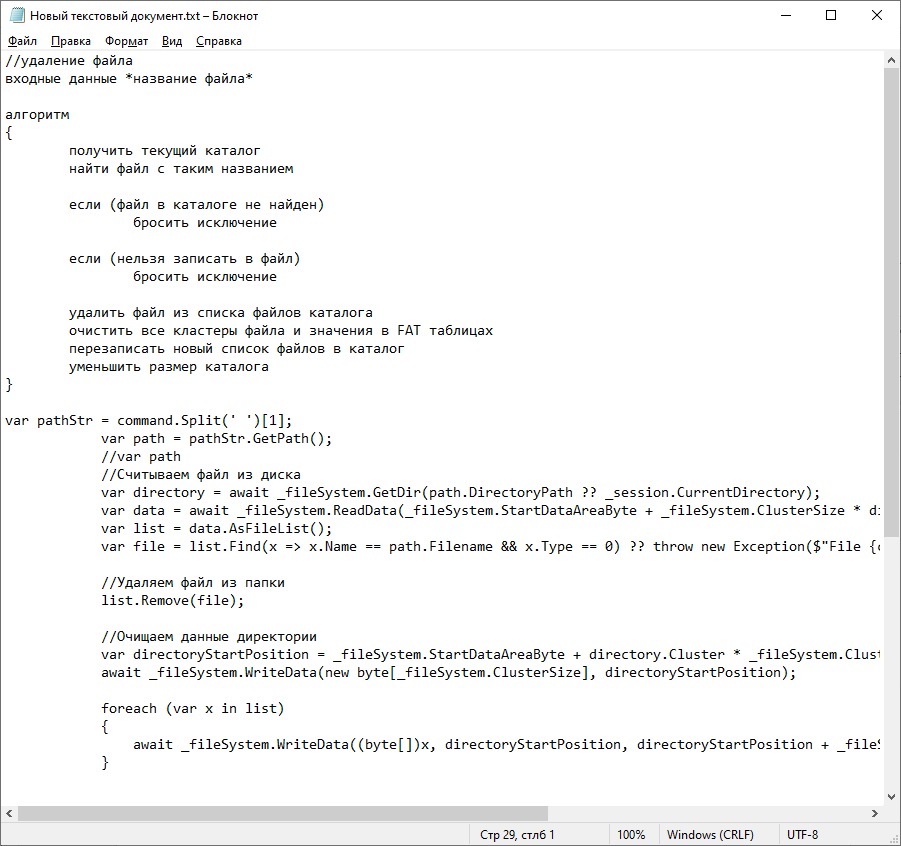


Рисунок 1.3 – Алгоритм работы команды remove.file

# 2 ПРОЦЕССЫ В ОС

Процесс – это программа, которая выполняется в системе. Все процессы в ОС выполняются последовательно. В системе может выполняться сотни и даже тысячи процессов одновременно.

## 2.1 Команды для работы с процессами

Команды для работы с процессами представлены в таблице 2.1.

Таблица 2.1 – Команды для работы с процессами

|  |  |  |
| --- | --- | --- |
| Команда | Аргументы | Описание |
| renice | pid, priority | Изменение приоритета процесса |
| kill | pid | Уничтожение процесса |
| pwd | - | Вывести текущие процессы |
| new.a | cpu\_time, pid | Создание абсолютного процесса |
| new.d | cpu\_time, pid | Создание динамического процесса |

## 2.2 Системные вызовы управления процессами

Системные вызовы для работы с процессами представлены в таблице 2.2.

## 2.3 Диаграмма состояний процесса

Диаграмма состояний процесса изображена на рисунке 2.1.

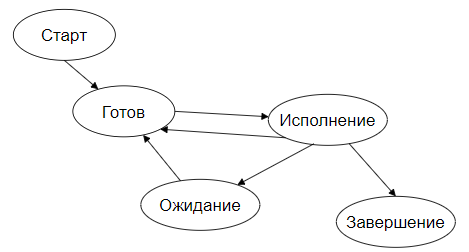


Рисунок 2.1 – Диаграмма состояний процесса

После того, как планировщик добавил процесс в очередь, он находится в состоянии готовности. Из состояния готовности процесс может перейти в состояние выполнения. Если процессу нужно запросить какие-либо данные (например, ввод из клавиатуры), либо же записать данные в файл, он переходит в состояние ожидания. После того, как процесс израсходует CPU burst его работа завершается.

## 2.4 Приоритеты процессов

В проектируемом планировщике процессов используются следующие приоритеты: абсолютные и динамические.

Абсолютный приоритет – это приоритет, который может при отсутствии свободных ресурсов приводит к захвату ресурсов у других процессов.

Динамический приоритет – это приоритет, устанавливаемый и изменяемый операционной системой для обеспечения эффективной работы компьютера.

Основное отличие абсолютных и динамических приоритетов в том, что абсолютные приоритеты система не могут модифицировать процесс, а динамические формируют и модифицируют, исходя из общих критериев эффективности.

## 2.5 Межпроцессное взаимодействие

Межпроцессное взаимодействие – это обмен данными между потоками одного или разных процессов. Реализуется посредством механизмов, предоставляемых ядром ОС или процессом, использующим механизмы ОС и реализующим новые возможности IPC. Может осуществляться как на одном компьютере, так и между несколькими компьютерами сети.

Для проектируемой системы были использованы следующие средства межпроцессного взаимодействия: семафоры и разделяемая память.

В многопоточных приложениях часто требуется, чтобы отдельные участки кода, называемые критическими секциями, не могли работать параллельно, например, при доступе к какому-либо неразделяемому ресурсу или при изменении общих ячеек памяти. Для защиты таких участков можно использовать двоичный семафор или мьютекс. Алгоритм работы критической секции на основе семафор представлен на рисунке 2.2.

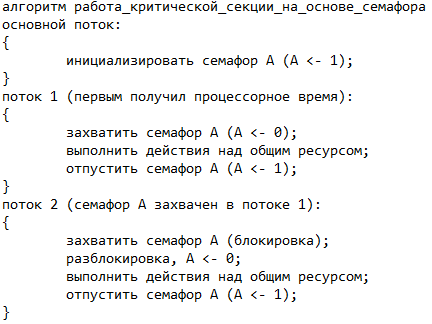


Рисунок 2.2 – Алгоритм работы критической секции на основе семафор

Разделяемая память является самым быстрым средством обмена данными между процессами. После создания разделяемого сегмента памяти любой из пользовательских процессов может подсоединить его к своему собственному виртуальному пространству и работать с ним, как с обычным сегментом памяти. Недостатком такого обмена информацией является отсутствие каких бы то ни было средств синхронизации, однако для преодоления этого недостатка можно использовать технику семафоров. Алгоритм работы программы с разделяемой памятью изображён на рисунке 2.3.

## 

Рисунок 2.3 – Алгоритм работы программы с разделяемой памятью

## 2.6 Выбор дисциплины обслуживания планировщика процессов. Алгоритм работы планировщика процессов в соответствии с выбранной дисциплиной обслуживания

Планировщик процессов определяет, какой из процессов, когда и на какое время получит управление процессором. Для проектируемого планировщика была выбрана дисциплина равномерного циклического квантования, т.е. длительность обслуживания каждого процесса ограничена величиной стандартного кванта времени.

Планировщик использует многоуровневые очереди. Процессы с приоритетом 0 попадают в очередь 0, с приоритетом 1 – в очередь 1, с приоритетом 2 – в очередь 2. Планировщик работает до тех пор, пока в системе есть процессы. Процессы в очереди 1 не могут исполняться, если в очереди 0 есть хотя бы один процесс. Процессы в очереди 2 не будут выбраны для выполнения, пока есть хоть один процесс в очередях 0 и 1.

Планирование процессов внутри очередей 0 и 1 осуществляется с использованием алгоритма Round Robin (RR), а внутри очереди 2 – First Come First Serve (FCFS).

Алгоритм FCFS осуществляет не вытесняющее планирование. Процесс, получивший в свое распоряжение процессор, занимает его до истечения текущего CPU burst. После этого для выполнения выбирается следующий процесс в очереди.

Модификацией алгоритма FCFS является алгоритм RR. Во многом 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. В разрабатываемом планировщике у очереди 0 квант времени равен 4 тактам, а у очереди 1 – 2 тактам.

Алгоритм работы планировщика процессов изображён на рисунке 2.4.

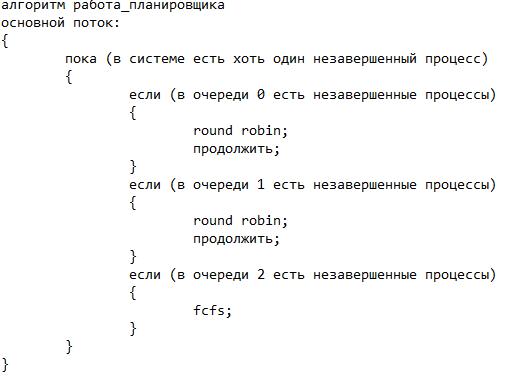


Рисунок 2.4 – Алгоритм работы планировщика процессов

## 2.7 Свопинг процессов

Свопинг процессов – это вытеснение готового к исполнению процесса из оперативной памяти во внешнюю. При этом оперативная память, заполненная процессом, полностью освобождается. Свопинг процессов может использоваться как средство реализации абсолютных приоритетов при отсутствии виртуальной памяти либо инициироваться операцией на внешнем устройстве, требующей значительных временных затрат. Например, в некоторых системах выполняется свопинг процессов на время терминального ввода. Операция ввода с терминала может выполняться до нескольких десятков секунд и освобождение оперативной памяти на это время обеспечивает значительное повышение эффективности работы системы.

# 3 РЕЖИМЫ РАБОТЫ ПРОЕКТИРУЕМОЙ ОС

## 3.1 Мультипрограммный режим работы ОС

Мультипрограммный режим работы – это режим функционирования ОС, при котором несколько программ (процессов) одновременно находятся на стадии выполнения. При наличии единственного центрального процессора параллелизм достигается за счет асинхронной работы устройств. В то время как одна из программ выполняет счет на процессоре, другие программы могут выполнять операции на внешних устройствах. Таким образом, выигрыш эффективности по сравнению с однопрограммным режимом достигается за счет параллельной работы устройств, входящих в состав компьютера.

## 3.2 Многопользовательская защита

Многопользовательская защита – средства ОС, обеспечивающие идентификацию пользователей и различные уровни их привилегий, а также установку и защиту прав собственности на информационные ресурсы, создаваемые в среде ОС.

## 3.3 Интерактивный режим работы ОС

Интерактивный режим – режим работы ОС, при котором основными источниками команд являются пользователи, оперативно взаимодействующие с компьютером посредством терминалов. В интерактивном режиме необходимо обеспечить приемлемое для пользователя время исполнения команды (как правило, в пределах нескольких секунд).

# 4 СТРУКТУРА ОПЕРАЦИОННОЙ СИСТЕМЫ

## 4.1 Общая структура проектируемой ОС

Общая структура операционной системы изображена на рисунке 4.1.

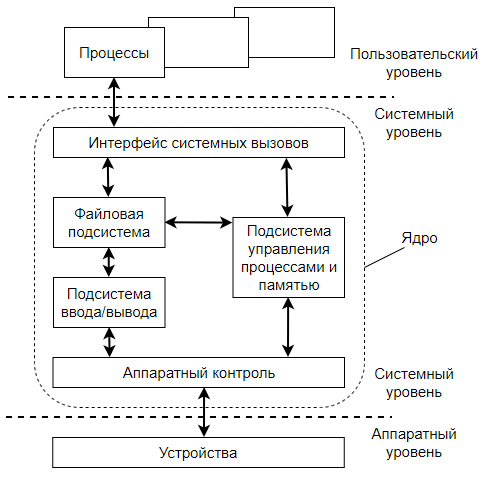


Рисунок 4.1 – Общая структура операционной системы

Ядро ОС обеспечивает базовую функциональность ОС, создает процессы и управляет ими, распределяет память, обеспечивает доступ к файлам и периферийным устройствам. Взаимодействие прикладных задач с ядром происходит посредством стандартного интерфейса системных вызовов.

Интерфейс системных вызовов представляет собой набор услуг ядра и определяет формат запросов на услуги.

Процесс запрашивает услугу посредством системного вызова определенной процедуры ядра. Ядро от имени процесса выполняет запрос и возвращает процессу необходимые данные.

## 4.2 Структура ядра проектируемой ОС. Основные функции и назначение файловой подсистемы, подсистемы управления памятью и процессами, подсистемы управления устройствами

Ядро операционной системы состоит из трёх основных модулей:

– файловая подсистема;

– подсистема управления процессами и памятью;

– подсистема ввода-вывода.

Файловая подсистема обеспечивает унифицированный интерфейс доступа к данным, расположенным на дисках, и к периферийным устройствам. Она контролирует права доступа к файлу, выполняет операции размещения и удаления файла, а также выполняет запись или чтение данных файла. Права доступа к файлам определяют привилегии пользователя в системе. Файловая система обеспечивает перенаправление запросов, адресованных периферийным устройствам, соответствующим модулям подсистемы ввода-вывода.

Запущенная на выполнение программа порождает в системе один или более процессов. Подсистема управления процессами и памятью контролирует:

– создание и удаление процессов;

– распределение системных ресурсов между процессами;

– синхронизацию процессов;

– межпроцессное взаимодействие.

Подсистема ввода-вывода выполняет запросы файловой подсистемы и подсистемы управления процессами для доступа к периферийным устройствам. Она взаимодействует с драйверами устройств – специальными модулями ядра, непосредственно обслуживающими внешние устройства.

## 4.3 Структура управляющих блоков базы данных ОС

Совокупность данных, используемую ОС для управления процессами и ресурсами, называют базой данных ОС (БД ОС). Основная часть базы данных размещается в ядре системы, что обеспечивает ее надежную защиту. Большая часть элементов базы данных создается и уничтожается динамически; также динамически создаются и модифицируются связи между элементами. Поэтому для реализации БД ОС, как правило, используют многосвязные списочные структуры данных. Основу базы данных составляют управляющие блоки, объединяемые в списки сложной структуры. Заголовки основных списков и некоторые глобальные системные переменные образуют статическую часть базы данных. Динамически создаваемые и уничтожаемые блоки размещаются в куче ядра.

БД ОС содержит следующие управляющие блоки:

– блок управления пользователями;

– блок управления файлами;

– блок управления памятью;

– блок управления каталога страниц памяти;

– блок управления процессами;

– блок управления устройствами.

Структура блока управления пользователями (user control block) представлена в таблице 4.1.

Таблица 4.1 – Структура управления пользователями

|  |  |
| --- | --- |
| Поле | Описание |
| Id | ID пользователя |
| Name | Имя пользователя |
| Password | Пароль пользователя |

Структура блока управления файлами представлена в таблице 4.2.

Таблица 4.2 – Структура блока управления файлами

|  |  |
| --- | --- |
| Поле | Описание |
| ID | ID файла |
| NAME | Имя файла |
| START | Начальный кластер |
| UID | Идентификатор владельца |
| ATTR | Атрибуты файла |
| SIZE | Размер файла |
| COUNT | Количество занятых файлом блоков |
| HTIME | Время изменения |

Структура блока управления памятью (memory control block) представлена в таблице 4.3.

Таблица 4.3 – Структура блока управления памятью

|  |  |
| --- | --- |
| Поле | Описание |
| RCB\_HANDLE | Дескриптор блока |
| RCB\_SIZE | Размер блока |
| RCB\_CADR | Размер блока |

Продолжение таблицы 4.3

|  |  |
| --- | --- |
| Поле | Описание |
| RCB\_RIGHT | Права доступа |

Структура блока управления каталога страниц памяти (page mapping table control block) представлена в таблице 4.4.

Таблица 4.4 – Структура блока управления каталога страниц памяти

|  |  |
| --- | --- |
| Поле | Описание |
| PCB\_TID | ID таблицы страниц |
| PCB\_DESCRIPT | Описание страницы |

Структура блока управления процессами (task control block) представлена в таблице 4.5.

Таблица 4.5 – Структура блока управления процессами

|  |  |
| --- | --- |
| Поле | Описание |
| TCB\_ID | ID процесса |
| TCB\_PRI | Приоритет процесса |
| TCB\_NI | Поправка на приоритет процесса |
| TCB\_STATE | Состояние процесса |
| TCB\_CPU | CPU burst процесса |
| TCB\_IO | IO burst процесса |
| TCB\_QID | Идентификатор очереди, в которой состоит процесс |

Структура блока управления устройствами (device control block) представлена в таблице 4.6.

Таблица 4.6 – Структура блока управления устройствами

|  |  |
| --- | --- |
| Поле | Описание |
| DCB\_ID | ID устройства |
| DCB\_NAME | Имя устройства |
| DCB\_ST | Статус устройства |
| DCB\_INT | Таблица точек входа |
| DCB\_INIT | Инициализация |
| DCB\_WRITE | Флаг записи |
| DCB\_READ | Флаг чтения |
| DCB\_PROP | Параметры устройства |
| DCB\_DRV | Адрес драйвера устройства |
| DCB\_NEXT | Адрес следующего DCB |

# 5 РАЗРАБОТКА ПРОГРАММ ЭМУЛЯЦИИ ОС

## 5.1 Описание программных средств

Для разработки программы эмуляции файловой системы был выбран язык программирования Rust. Для разработки программы эмуляции планировщика процессов был выбран язык Go. Разработка велась c помощью редактора исходного кода Visual Studio Code.

## 5.2 Разработка ФС

Диаграмма классов модуля файловой системы изображена на рисунке 5.1.

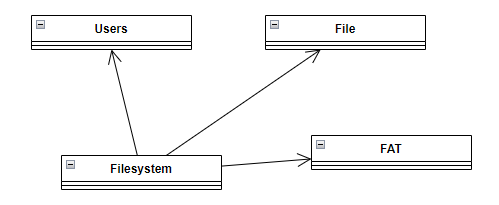


Рисунок 5.1 – Диаграмма классов модуля файловой системы

Методы класса Filesystem перечислены в таблице 5.1.

Таблица 5.1 – Методы класса Filesystem

|  |  |
| --- | --- |
| Имя метода | Описание |
| init | Инициализация файловой системы |
| read\_fs | Чтение файловой системы |
| create\_fs | Создание файловой системы |

Продолжение таблицы 5.1

|  |  |
| --- | --- |
| Имя метода | Описание |
| format\_fs | Форматирование файловой системы |
| write\_superblock | Запись суперблока |
| fill\_to\_end | Заполнение файла нулями |
| set\_next\_free\_block | Установить следующий свободный блок |
| write\_next\_free\_block | Записать следующий свободный блок |
| write\_free\_blocks\_count | Записать количество свободных блоков |

Методы класса FAT перечислены в таблице 5.2.

Таблица 5.2 – Методы класса FAT

|  |  |
| --- | --- |
| Имя метода | Описание |
| fat\_write | Записать значение блока в FAT-таблицу |
| fat\_read | Чтение значения блока FAT-таблицы |
| expand\_file | Расширить файл на 1 блок |
| alloc\_prim\_block | Выделить первичный блок для файла |

Методы класса File перечислены в таблице 5.3.

Таблица 5.3 – Методы класса File

|  |  |
| --- | --- |
| Имя метода | Описание |
| create\_file | Создание файла |
| copy\_file | Копирование файла |
| move\_file | Перемещение файла |
| is\_dir | Проверка на то, является ли файл каталог |
| check\_permissions | Проверка прав доступа |
| set\_permissions | Установка прав доступа |
| attributes\_format | Форматирование атрибутов файла в читаемый вид |

Продолжение таблицы 5.3

|  |  |
| --- | --- |
| Имя метода | Описание |
| file\_exists | Проверка на существование файла |
| rm\_file | Удаление файла |
| file\_write\_data | Запись данных в файл |
| file\_read\_data | Чтение данных из файла |
| find\_file\_offset | Поиск смещения файла |
| filename\_format | Форматирование имени файла |
| file\_write\_name | Запись имени файла |
| file\_write\_prim\_block | Запись первичного блока файла |
| file\_write\_uid | Запись идентификатора владельца файла |
| file\_write\_attrs | Запись атрибутов файла |
| file\_write\_size | Запись размера файла |
| file\_write\_blocks\_count | Запись количества занимаемых блоков файлом |
| file\_write\_time | Запись времени модификации файла |
| file\_read\_name | Чтение имени файла |
| file\_read\_prim\_block | Чтение первичного блока файла |
| file\_read\_uid | Чтение идентификатора владельца файла |
| file\_read\_attrs | Чтение атрибутов файла |
| file\_read\_size | Чтение размера файла |
| file\_read\_blocks\_count | Чтение количества занимаемых блоков файлом |
| file\_read\_time | Чтение времени модификации файла |

Методы класса Users перечислены в таблице 5.4.

Таблица 5.4 – Методы класса Users

|  |  |
| --- | --- |
| Имя метода | Описание |
| add\_user | Добавление пользователя |
| rm\_user | Удаление пользователя |

Продолжение таблицы 5.4

|  |  |
| --- | --- |
| Имя метода | Описание |
| add\_group | Удаление группы |
| user\_and\_group\_names\_format | Форматирование имён пользователей и групп |
| set\_users\_count | Установка количества пользователей |
| set\_groups\_count | Установка количества групп |
| username\_check | Проверка имени пользователя на корректность |
| is\_username\_exist | Проверка на существование имени пользователя |
| is\_group\_name\_exist | Проверка на существование имени группы |
| is\_pass\_valid | Проверка на валидность введенного пароля |
| get\_user\_and\_group\_name | Получить имя пользователя и имя группы по uid |
| get\_username | Получить имя пользователя по uid |
| get\_uid | Получить uid пользователя по имени пользователя |
| get\_guid | Получить guid по имени группы |
| get\_user\_guid | Получить имя группы по guid |
| get\_max\_ids | Получить максимальные значения идентификаторов пользователей и групп |
| login | Авторизоваться под пользователем |

## 5.3 Разработка командного интерпретатора

Для общения с операционной системой пользователю предоставляется текстовая консоль.

Командный интерпретатор – это программа, которая в непрерывном цикле ожидает ввода команд. При вводе команды реализовывается ее синтаксический разбор (существование имени команды, наличие всех обязательных параметров и проверка параметров на их корректность).

Диаграмма классов модуля командного интерпретатора изображена на рисунке 5.2.

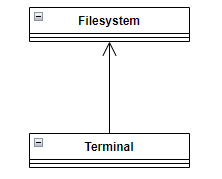


Рисунок 5.2 – Диаграмма классов модуля командного интерпретатора

Методы класса Terminal перечислены в таблице 5.5.

Таблица 5.5 – Методы класса Terminal

|  |  |
| --- | --- |
| Имя метода | Описание |
| prompt | Приглашение на ввод команды |
| commands\_handle | Обработка команды |
| handle\_file\_path | Обработка введенного пути файла |
| append | Обработка команды для записи данных в конец файла |
| cat | Обработка команды для вывода содержимого файла |
| cd | Обработка команды для перемещение в каталог |
| help | Обработка команды для вывода списка доступных команд и их описания |
| ls | Обработка команды для вывода списка файлов в текущем каталоге |
| mv | Обработка команды для перемещения файла |
| rm | Обработка команды для удаления файла |

Продолжение таблицы 5.5

|  |  |
| --- | --- |
| su | Обработка команды для смены пользователя |
| touch | Обработка команды для создания файла |
| useradd | Обработка команды для добавления пользователя |
| userdel | Обработка команды для удаления пользователя |
| users | Обработка команды для вывода списка пользователей |
| write | Обработка команды для перезаписи данных файла |

## 5.4 Эмуляция планирования

Диаграмма классов модуля планировщика процессов изображена на рисунке 5.3.



Рисунок 5.3 – Диаграмма классов модуля планировщика процессов

Методы класса Process перечислены в таблице 5.6.

Таблица 5.6 – Методы класса Process

|  |  |
| --- | --- |
| Имя метода | Описание |
| setState | Установка состояния процессу |
| decCpu | Уменьшение CPU burst процесса на 1 |
| decIo | Уменьшение IO burst процесса на 1 |
| setNi | Установка поправки на приоритет процесса |

Методы класса Sheduler перечислены в таблице 5.7.

Таблица 5.7 – Методы класса Sheduler

|  |  |
| --- | --- |
| Имя метода | Описание |
| newProcess | Добавление процесса |
| remove | Удаление процесса из очереди |
| handleReadyProcesses | Обработка готовых процессов |
| printReadyProcesses | Печать готовых процессов |
| handleQueue0 | Обработка очереди 0 (алгоритм RR) |
| handleQueue1 | Обработка очереди 1 (алгоритм RR) |
| handleQueue2 | Обработка очереди 2 (алгоритм FCFS) |
| handleCommand | Обработка введённых пользователем команд |
| anyReadyProccesses | Проверка на существование процессов для выполнения в очереди |

# 6 ТЕСТИРОВАНИЕ ПРОГРАММЫ. АНАЛИЗ РЕЗУЛЬТАТОВ

Разработанные программные модули для эмуляции операционной системы корректно обрабатывают возможные исключительные ситуации, которые могут возникнуть при неверных действиях пользователя или при некорректном вводе данных.

Реакции системы на возникновения исключительных ситуаций изображены на рисунках 6.1-6.8.

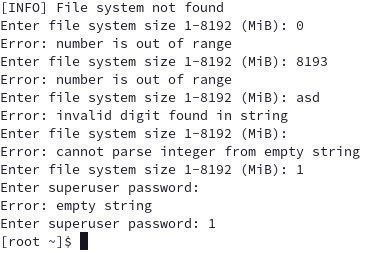


Рисунок 6.1 – Выход за пределы диапазона, пустые строки

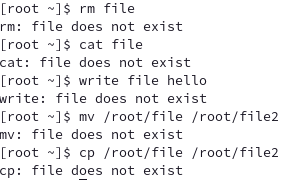


Рисунок 6.2 – Работа с несуществующими файлами

Screenshot_2021-12-13_20-18-26

Рисунок 6.3 – Переключение на несуществующего пользователя

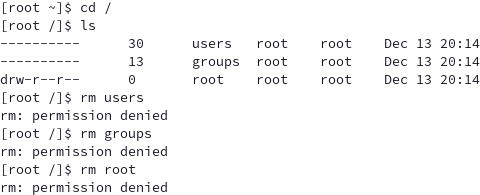


Рисунок 6.4 – Удаление системных файлов

Screenshot_2021-12-13_20-18-56

Рисунок 6.5 – Удаление несуществующего пользователя

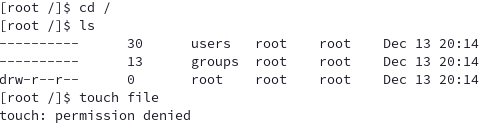


Рисунок 6.6 – Создание файлов в корневом каталоге

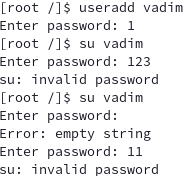


Рисунок 6.7 – Ввод неверного пароля

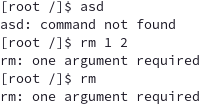


Рисунок 6.8 – Ввод неверной команды

# ВЫВОДЫ

В большинстве вычислительных систем операционная система является основной, наиболее важной (а иногда и единственной) частью системного программного обеспечения. С 1990-х годов наиболее распространёнными операционными системами являются системы семейства Windows и системы класса UNIX (особенно Linux и Mac OS). С 2000-х большое распространение получили мобильные компьютеры (смартфоны и планшеты) и с ними ОС Android и iOS.

ОС является необходимой составляющей ПО ПК, без нее компьютер не может работать в принципе.

В ходе выполнения курсового проекта были изучены и закреплены на практике основные концепции работы операционной системы.

В результате выполнения курсового проекта была спроектирована гипотетическая операционная система, были разработаны отдельные её модули, а именно: файловая система, командный интерпретатор и планировщик процессов.

Разработанная система может быть использована для демонстрации возможностей работы операционной системы.

# ПЕРЕЧЕНЬ ССЫЛОК

1. FAT32 // Википедия. [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/FAT32

2. Организация виртуальной памяти // Хабр. [Электронный ресурс]. – Режим доступа: https://habr.com/ru/company/embox/blog/256191

3. Процесс // Национальная библиотека им. Н. Э. Баумана. [Электронный ресурс]. – Режим доступа: https://ru.bmstu.wiki/Процесс\_(Операционные\_Системы)

4. Межпроцессное взаимодействие // Википедия. [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Межпроцессное\_взаимодействие

# ПРИЛОЖЕНИЕ А

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

# ПРИЛОЖЕНИЕ Б

# ТАБЛИЦА ТРАССИРОВКИ

|  |  |
| --- | --- |
| Очередь готовых процессов:  1) PID: 0, NI: 0, PRI: 0, NI: 0, STATE: Г, CPU: 23  2) PID: 1, NI: 0, PRI: 0, NI: 0, STATE: Г, CPU: 15  3) PID: 2, NI: 0, PRI: 0, NI: 0, STATE: Г, CPU: 21  4) PID: 3, NI: -2, PRI: 1, NI: -2, STATE: Г, CPU: 15  5) PID: 4, NI: 1, PRI: 1, NI: 1, STATE: Г, CPU: 18  6) PID: 5, NI: 5, PRI: 2, NI: 5, STATE: Г, CPU: 4  7) PID: 6, NI: 3, PRI: 2, NI: 3, STATE: Г, CPU: 5  =======================================  Очередь 0 (RR), относительные приоритеты, такт 1:  PID: 0, NI: 0, STATE: И, CPU: 22, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 15, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 21, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 2:  PID: 0, NI: 0, STATE: И, CPU: 21, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 15, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 21, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 3:  PID: 0, NI: 0, STATE: И, CPU: 20, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 15, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 21, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 4:  PID: 0, NI: 0, STATE: И, CPU: 19, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 15, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 21, IO: 3  =======================================  Очередь 0 (RR), относительные приоритеты, такт 1:  PID: 0, NI: 0, STATE: Г, CPU: 19, IO: 3  PID: 1, NI: 0, STATE: И, CPU: 14, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 21, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 2:  PID: 0, NI: 0, STATE: Г, CPU: 19, IO: 3  PID: 1, NI: 0, STATE: И, CPU: 13, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 21, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 3:  PID: 0, NI: 0, STATE: Г, CPU: 19, IO: 3  PID: 1, NI: 0, STATE: И, CPU: 12, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 21, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 4:  PID: 0, NI: 0, STATE: Г, CPU: 19, IO: 3  PID: 1, NI: 0, STATE: И, CPU: 11, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 21, IO: 3  =======================================  Очередь 0 (RR), относительные приоритеты, такт 1:  PID: 0, NI: 0, STATE: Г, CPU: 19, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 11, IO: 3  PID: 2, NI: 0, STATE: И, CPU: 20, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 2:  PID: 0, NI: 0, STATE: Г, CPU: 19, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 11, IO: 3  PID: 2, NI: 0, STATE: И, CPU: 19, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 3:  PID: 0, NI: 0, STATE: Г, CPU: 19, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 11, IO: 3  PID: 2, NI: 0, STATE: И, CPU: 18, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 4:  PID: 0, NI: 0, STATE: Г, CPU: 19, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 11, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 13, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 2:  PID: 0, NI: 0, STATE: И, CPU: 13, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 7, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 13, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 3:  PID: 0, NI: 0, STATE: И, CPU: 12, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 7, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 13, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 4:  PID: 0, NI: 0, STATE: И, CPU: 11, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 7, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 13, IO: 3  =======================================  Очередь 0 (RR), относительные приоритеты, такт 1:  PID: 0, NI: 0, STATE: Г, CPU: 11, IO: 3  PID: 1, NI: 0, STATE: И, CPU: 6, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 13, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 2:  PID: 0, NI: 0, STATE: Г, CPU: 11, IO: 3  PID: 1, NI: 0, STATE: И, CPU: 5, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 13, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 3:  PID: 0, NI: 0, STATE: Г, CPU: 11, IO: 3  PID: 1, NI: 0, STATE: И, CPU: 4, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 13, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 4:  PID: 0, NI: 0, STATE: Г, CPU: 11, IO: 3  PID: 1, NI: 0, STATE: И, CPU: 3, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 13, IO: 3  =======================================  Очередь 0 (RR), относительные приоритеты, такт 1:  PID: 0, NI: 0, STATE: Г, CPU: 11, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 3, IO: 3  PID: 2, NI: 0, STATE: И, CPU: 12, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 2:  PID: 0, NI: 0, STATE: Г, CPU: 11, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 3, IO: 3  PID: 2, NI: 0, STATE: И, CPU: 11, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 3:  PID: 0, NI: 0, STATE: Г, CPU: 11, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 3, IO: 3  PID: 2, NI: 0, STATE: И, CPU: 10, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 4:  PID: 0, NI: 0, STATE: Г, CPU: 11, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 3, IO: 3  PID: 2, NI: 0, STATE: И, CPU: 9, IO: 3  =======================================  Очередь 0 (RR), относительные приоритеты, такт 1:  PID: 0, NI: 0, STATE: И, CPU: 10, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 3, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 9, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 2:  PID: 0, NI: 0, STATE: И, CPU: 9, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 3, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 9, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 3:  PID: 0, NI: 0, STATE: И, CPU: 8, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 3, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 9, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 4:  PID: 0, NI: 0, STATE: И, CPU: 7, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 3, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 3:  PID: 0, NI: 0, STATE: И, CPU: 2, IO: 3  PID: 1, NI: 0, STATE: З, CPU: 0, IO: 0  PID: 2, NI: 0, STATE: Г, CPU: 3, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 4:  PID: 0, NI: 0, STATE: И, CPU: 1, IO: 3  PID: 1, NI: 0, STATE: З, CPU: 0, IO: 0  PID: 2, NI: 0, STATE: Г, CPU: 3, IO: 3  =======================================  Очередь 0 (RR), относительные приоритеты, такт 1:  PID: 0, NI: 0, STATE: Г, CPU: 1, IO: 3  PID: 1, NI: 0, STATE: З, CPU: 0, IO: 0  PID: 2, NI: 0, STATE: И, CPU: 2, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 2:  PID: 0, NI: 0, STATE: Г, CPU: 1, IO: 3  PID: 1, NI: 0, STATE: З, CPU: 0, IO: 0  PID: 2, NI: 0, STATE: И, CPU: 1, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 3:  PID: 0, NI: 0, STATE: Г, CPU: 1, IO: 3  PID: 1, NI: 0, STATE: З, CPU: 0, IO: 0  PID: 2, NI: 0, STATE: О, CPU: 0, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 4:  PID: 0, NI: 0, STATE: О, CPU: 0, IO: 3  PID: 1, NI: 0, STATE: З, CPU: 0, IO: 0  PID: 2, NI: 0, STATE: О, CPU: 0, IO: 2  =======================================  Очередь 0 (RR), относительные приоритеты, такт 1:  PID: 0, NI: 0, STATE: О, CPU: 0, IO: 2  PID: 1, NI: 0, STATE: З, CPU: 0, IO: 0  PID: 2, NI: 0, STATE: О, CPU: 0, IO: 1  Очередь 0 (RR), относительные приоритеты, такт 2:  PID: 0, NI: 0, STATE: О, CPU: 0, IO: 1  PID: 1, NI: 0, STATE: З, CPU: 0, IO: 0  PID: 2, NI: 0, STATE: З, CPU: 0, IO: 0  Очередь 0 (RR), относительные приоритеты, такт 3:  PID: 0, NI: 0, STATE: З, CPU: 0, IO: 0  PID: 1, NI: 0, STATE: З, CPU: 0, IO: 0  PID: 2, NI: 0, STATE: З, CPU: 0, IO: 0  Очередь 0 (RR), относительные приоритеты, такт 4:  PID: 0, NI: 0, STATE: З, CPU: 0, IO: 0  PID: 1, NI: 0, STATE: З, CPU: 0, IO: 0  PID: 2, NI: 0, STATE: З, CPU: 0, IO: 0  =======================================  =======================================  Очередь 1 (RR), динамические приоритеты, такт 1:  PID: 3, NI: -2, STATE: И, CPU: 14  PID: 4, NI: 1, STATE: Г, CPU: 18  Очередь 1 (RR), динамические приоритеты, такт 2:  PID: 3, NI: -2, STATE: И, CPU: 13  PID: 4, NI: 1, STATE: Г, CPU: 18  =======================================  Очередь 1 (RR), динамические приоритеты, такт 1:  PID: 3, NI: -2, STATE: Г, CPU: 13  PID: 4, NI: 1, STATE: И, CPU: 17  Очередь 1 (RR), динамические приоритеты, такт 2:  PID: 3, NI: -2, STATE: Г, CPU: 13  PID: 4, NI: 1, STATE: И, CPU: 16  =======================================  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: И, CPU: 5, IO: 3  PID: 4, NI: 1, STATE: Г, CPU: 9, IO: 3  PID: 6, NI: 3, STATE: Г, CPU: 5, IO: 3  PID: 5, NI: 5, STATE: Г, CPU: 4, IO: 3  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: И, CPU: 4, IO: 3  PID: 4, NI: 1, STATE: Г, CPU: 9, IO: 3  PID: 6, NI: 3, STATE: Г, CPU: 5, IO: 3  PID: 5, NI: 5, STATE: Г, CPU: 4, IO: 3  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: И, CPU: 3, IO: 3  PID: 4, NI: 1, STATE: Г, CPU: 9, IO: 3  PID: 6, NI: 3, STATE: Г, CPU: 5, IO: 3  PID: 5, NI: 5, STATE: Г, CPU: 4, IO: 3  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: И, CPU: 2, IO: 3  PID: 4, NI: 1, STATE: Г, CPU: 9, IO: 3  PID: 6, NI: 3, STATE: Г, CPU: 5, IO: 3  PID: 5, NI: 5, STATE: Г, CPU: 4, IO: 3  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: И, CPU: 1, IO: 3  PID: 4, NI: 1, STATE: Г, CPU: 9, IO: 3  PID: 6, NI: 3, STATE: Г, CPU: 5, IO: 3  PID: 5, NI: 5, STATE: Г, CPU: 4, IO: 3  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: О, CPU: 0, IO: 3  PID: 4, NI: 1, STATE: И, CPU: 8, IO: 3  PID: 6, NI: 3, STATE: Г, CPU: 5, IO: 3  PID: 5, NI: 5, STATE: Г, CPU: 4, IO: 3  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: О, CPU: 0, IO: 2  PID: 4, NI: 1, STATE: И, CPU: 7, IO: 3  PID: 6, NI: 3, STATE: Г, CPU: 5, IO: 3  PID: 5, NI: 5, STATE: Г, CPU: 4, IO: 3  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: О, CPU: 0, IO: 1  PID: 4, NI: 1, STATE: И, CPU: 6, IO: 3  PID: 6, NI: 3, STATE: Г, CPU: 5, IO: 3  PID: 5, NI: 5, STATE: Г, CPU: 4, IO: 3  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 4, NI: 1, STATE: З, CPU: 0, IO: 0  PID: 6, NI: 3, STATE: О, CPU: 0, IO: 3  PID: 5, NI: 5, STATE: И, CPU: 3, IO: 3  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 4, NI: 1, STATE: З, CPU: 0, IO: 0  PID: 6, NI: 3, STATE: О, CPU: 0, IO: 2  PID: 5, NI: 5, STATE: И, CPU: 2, IO: 3  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 4, NI: 1, STATE: З, CPU: 0, IO: 0  PID: 6, NI: 3, STATE: О, CPU: 0, IO: 1  PID: 5, NI: 5, STATE: И, CPU: 1, IO: 3  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 4, NI: 1, STATE: З, CPU: 0, IO: 0  PID: 6, NI: 3, STATE: З, CPU: 0, IO: 0  PID: 5, NI: 5, STATE: О, CPU: 0, IO: 3  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 4, NI: 1, STATE: З, CPU: 0, IO: 0  PID: 6, NI: 3, STATE: З, CPU: 0, IO: 0  PID: 5, NI: 5, STATE: О, CPU: 0, IO: 2  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 4, NI: 1, STATE: З, CPU: 0, IO: 0  PID: 6, NI: 3, STATE: З, CPU: 0, IO: 0  PID: 5, NI: 5, STATE: О, CPU: 0, IO: 1  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 4, NI: 1, STATE: З, CPU: 0, IO: 0  PID: 6, NI: 3, STATE: З, CPU: 0, IO: 0  PID: 5, NI: 5, STATE: З, CPU: 0, IO: 0  =======================================  =======================================  [+] Запущен новый процесс с PID 7 и PRI 1  Очередь 1 (RR), динамические приоритеты, такт 1:  PID: 7, NI: 0, STATE: И, CPU: 19  Очередь 1 (RR), динамические приоритеты, такт 2:  PID: 7, NI: 0, STATE: И, CPU: 18  =======================================  Очередь 1 (RR), динамические приоритеты, такт 1:  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 7, NI: 0, STATE: И, CPU: 4, IO: 3  PID: 4, NI: 1, STATE: З, CPU: 0, IO: 0  PID: 6, NI: 3, STATE: З, CPU: 0, IO: 0  PID: 5, NI: 5, STATE: З, CPU: 0, IO: 0  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 7, NI: 0, STATE: И, CPU: 3, IO: 3  PID: 4, NI: 1, STATE: З, CPU: 0, IO: 0  PID: 6, NI: 3, STATE: З, CPU: 0, IO: 0  PID: 5, NI: 5, STATE: З, CPU: 0, IO: 0  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 7, NI: 0, STATE: И, CPU: 2, IO: 3  PID: 4, NI: 1, STATE: З, CPU: 0, IO: 0  PID: 6, NI: 3, STATE: З, CPU: 0, IO: 0  PID: 5, NI: 5, STATE: З, CPU: 0, IO: 0  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 7, NI: 0, STATE: И, CPU: 1, IO: 3  PID: 4, NI: 1, STATE: З, CPU: 0, IO: 0  PID: 6, NI: 3, STATE: З, CPU: 0, IO: 0  PID: 5, NI: 5, STATE: З, CPU: 0, IO: 0  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 7, NI: 0, STATE: О, CPU: 0, IO: 3  PID: 4, NI: 1, STATE: З, CPU: 0, IO: 0  PID: 6, NI: 3, STATE: З, CPU: 0, IO: 0  PID: 5, NI: 5, STATE: З, CPU: 0, IO: 0  =======================================  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 7, NI: 0, STATE: О, CPU: 0, IO: 2  PID: 4, NI: 1, STATE: З, CPU: 0, IO: 0  PID: 6, NI: 3, STATE: З, CPU: 0, IO: 0  PID: 5, NI: 5, STATE: З, CPU: 0, IO: 0  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 7, NI: 0, STATE: О, CPU: 0, IO: 1  PID: 4, NI: 1, STATE: З, CPU: 0, IO: 0  PID: 6, NI: 3, STATE: З, CPU: 0, IO: 0  PID: 5, NI: 5, STATE: З, CPU: 0, IO: 0  ======================================= | PID: 2, NI: 0, STATE: И, CPU: 17, IO: 3  =======================================  Очередь 0 (RR), относительные приоритеты, такт 1:  PID: 0, NI: 0, STATE: И, CPU: 18, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 11, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 17, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 2:  PID: 0, NI: 0, STATE: И, CPU: 17, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 11, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 17, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 3:  PID: 0, NI: 0, STATE: И, CPU: 16, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 11, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 17, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 4:  PID: 0, NI: 0, STATE: И, CPU: 15, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 11, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 17, IO: 3  =======================================  Очередь 0 (RR), относительные приоритеты, такт 1:  PID: 0, NI: 0, STATE: Г, CPU: 15, IO: 3  PID: 1, NI: 0, STATE: И, CPU: 10, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 17, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 2:  PID: 0, NI: 0, STATE: Г, CPU: 15, IO: 3  PID: 1, NI: 0, STATE: И, CPU: 9, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 17, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 3:  PID: 0, NI: 0, STATE: Г, CPU: 15, IO: 3  PID: 1, NI: 0, STATE: И, CPU: 8, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 17, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 4:  PID: 0, NI: 0, STATE: Г, CPU: 15, IO: 3  PID: 1, NI: 0, STATE: И, CPU: 7, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 17, IO: 3  =======================================  Очередь 0 (RR), относительные приоритеты, такт 1:  PID: 0, NI: 0, STATE: Г, CPU: 15, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 7, IO: 3  PID: 2, NI: 0, STATE: И, CPU: 16, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 2:  PID: 0, NI: 0, STATE: Г, CPU: 15, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 7, IO: 3  PID: 2, NI: 0, STATE: И, CPU: 15, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 3:  PID: 0, NI: 0, STATE: Г, CPU: 15, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 7, IO: 3  PID: 2, NI: 0, STATE: И, CPU: 14, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 4:  PID: 0, NI: 0, STATE: Г, CPU: 15, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 7, IO: 3  PID: 2, NI: 0, STATE: И, CPU: 13, IO: 3  =======================================  Очередь 0 (RR), относительные приоритеты, такт 1:  PID: 0, NI: 0, STATE: И, CPU: 14, IO: 3  PID: 1, NI: 0, STATE: Г, CPU: 7, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 9, IO: 3  =======================================  Очередь 0 (RR), относительные приоритеты, такт 1:  PID: 0, NI: 0, STATE: Г, CPU: 7, IO: 3  PID: 1, NI: 0, STATE: И, CPU: 2, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 9, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 2:  PID: 0, NI: 0, STATE: Г, CPU: 7, IO: 3  PID: 1, NI: 0, STATE: И, CPU: 1, IO: 3  PID: 2, NI: 0, STATE: Г, CPU: 9, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 3:  PID: 0, NI: 0, STATE: Г, CPU: 7, IO: 3  PID: 1, NI: 0, STATE: О, CPU: 0, IO: 3  PID: 2, NI: 0, STATE: И, CPU: 8, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 4:  PID: 0, NI: 0, STATE: Г, CPU: 7, IO: 3  PID: 1, NI: 0, STATE: О, CPU: 0, IO: 2  PID: 2, NI: 0, STATE: Г, CPU: 8, IO: 3  =======================================  Очередь 0 (RR), относительные приоритеты, такт 1:  PID: 0, NI: 0, STATE: Г, CPU: 7, IO: 3  PID: 1, NI: 0, STATE: О, CPU: 0, IO: 1  PID: 2, NI: 0, STATE: И, CPU: 7, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 2:  PID: 0, NI: 0, STATE: Г, CPU: 7, IO: 3  PID: 1, NI: 0, STATE: З, CPU: 0, IO: 0  PID: 2, NI: 0, STATE: Г, CPU: 7, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 3:  PID: 0, NI: 0, STATE: И, CPU: 6, IO: 3  PID: 1, NI: 0, STATE: З, CPU: 0, IO: 0  PID: 2, NI: 0, STATE: Г, CPU: 7, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 4:  PID: 0, NI: 0, STATE: И, CPU: 5, IO: 3  PID: 1, NI: 0, STATE: З, CPU: 0, IO: 0  PID: 2, NI: 0, STATE: Г, CPU: 7, IO: 3  =======================================  Очередь 0 (RR), относительные приоритеты, такт 1:  PID: 0, NI: 0, STATE: Г, CPU: 5, IO: 3  PID: 1, NI: 0, STATE: З, CPU: 0, IO: 0  PID: 2, NI: 0, STATE: И, CPU: 6, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 2:  PID: 0, NI: 0, STATE: Г, CPU: 5, IO: 3  PID: 1, NI: 0, STATE: З, CPU: 0, IO: 0  PID: 2, NI: 0, STATE: И, CPU: 5, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 3:  PID: 0, NI: 0, STATE: Г, CPU: 5, IO: 3  PID: 1, NI: 0, STATE: З, CPU: 0, IO: 0  PID: 2, NI: 0, STATE: И, CPU: 4, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 4:  PID: 0, NI: 0, STATE: Г, CPU: 5, IO: 3  PID: 1, NI: 0, STATE: З, CPU: 0, IO: 0  PID: 2, NI: 0, STATE: И, CPU: 3, IO: 3  =======================================  Очередь 0 (RR), относительные приоритеты, такт 1:  PID: 0, NI: 0, STATE: И, CPU: 4, IO: 3  PID: 1, NI: 0, STATE: З, CPU: 0, IO: 0  PID: 2, NI: 0, STATE: Г, CPU: 3, IO: 3  Очередь 0 (RR), относительные приоритеты, такт 2:  PID: 0, NI: 0, STATE: И, CPU: 3, IO: 3  PID: 1, NI: 0, STATE: З, CPU: 0, IO: 0  PID: 2, NI: 0, STATE: Г, CPU: 3, IO: 3  Очередь 1 (RR), динамические приоритеты, такт 1:  PID: 3, NI: -2, STATE: И, CPU: 12  PID: 4, NI: 1, STATE: Г, CPU: 16  Очередь 1 (RR), динамические приоритеты, такт 2:  PID: 3, NI: -2, STATE: И, CPU: 11  PID: 4, NI: 1, STATE: Г, CPU: 16  =======================================  Очередь 1 (RR), динамические приоритеты, такт 1:  PID: 3, NI: -2, STATE: Г, CPU: 11  PID: 4, NI: 1, STATE: И, CPU: 15  Очередь 1 (RR), динамические приоритеты, такт 2:  PID: 3, NI: -2, STATE: Г, CPU: 11  PID: 4, NI: 1, STATE: И, CPU: 14  =======================================  Очередь 1 (RR), динамические приоритеты, такт 1:  PID: 3, NI: -2, STATE: И, CPU: 10  PID: 4, NI: 1, STATE: Г, CPU: 14  [!] У процесса с PID 3 был изменен приоритет на 2  Очередь 1 (RR), динамические приоритеты, такт 2:  PID: 4, NI: 1, STATE: Г, CPU: 14  =======================================  Очередь 1 (RR), динамические приоритеты, такт 1:  PID: 4, NI: 1, STATE: И, CPU: 13  Очередь 1 (RR), динамические приоритеты, такт 2:  PID: 4, NI: 1, STATE: И, CPU: 12  =======================================  Очередь 1 (RR), динамические приоритеты, такт 1:  PID: 4, NI: 1, STATE: И, CPU: 11  Очередь 1 (RR), динамические приоритеты, такт 2:  PID: 4, NI: 1, STATE: И, CPU: 10  =======================================  [!] У процесса с PID 4 был изменен приоритет на 2  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: И, CPU: 8, IO: 3  PID: 4, NI: 1, STATE: Г, CPU: 9, IO: 3  PID: 6, NI: 3, STATE: Г, CPU: 5, IO: 3  PID: 5, NI: 5, STATE: Г, CPU: 4, IO: 3  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: И, CPU: 7, IO: 3  PID: 4, NI: 1, STATE: Г, CPU: 9, IO: 3  PID: 6, NI: 3, STATE: Г, CPU: 5, IO: 3  PID: 5, NI: 5, STATE: Г, CPU: 4, IO: 3  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: И, CPU: 6, IO: 3  PID: 4, NI: 1, STATE: Г, CPU: 9, IO: 3  PID: 6, NI: 3, STATE: Г, CPU: 5, IO: 3  PID: 5, NI: 5, STATE: Г, CPU: 4, IO: 3  PID: 4, NI: 1, STATE: И, CPU: 5, IO: 3  PID: 6, NI: 3, STATE: Г, CPU: 5, IO: 3  PID: 5, NI: 5, STATE: Г, CPU: 4, IO: 3  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 4, NI: 1, STATE: И, CPU: 4, IO: 3  PID: 6, NI: 3, STATE: Г, CPU: 5, IO: 3  PID: 5, NI: 5, STATE: Г, CPU: 4, IO: 3  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 4, NI: 1, STATE: И, CPU: 3, IO: 3  PID: 6, NI: 3, STATE: Г, CPU: 5, IO: 3  PID: 5, NI: 5, STATE: Г, CPU: 4, IO: 3  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 4, NI: 1, STATE: И, CPU: 2, IO: 3  PID: 6, NI: 3, STATE: Г, CPU: 5, IO: 3  PID: 5, NI: 5, STATE: Г, CPU: 4, IO: 3  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 4, NI: 1, STATE: И, CPU: 1, IO: 3  PID: 6, NI: 3, STATE: Г, CPU: 5, IO: 3  PID: 5, NI: 5, STATE: Г, CPU: 4, IO: 3  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 4, NI: 1, STATE: О, CPU: 0, IO: 3  PID: 6, NI: 3, STATE: И, CPU: 4, IO: 3  PID: 5, NI: 5, STATE: Г, CPU: 4, IO: 3  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 4, NI: 1, STATE: О, CPU: 0, IO: 2  PID: 6, NI: 3, STATE: И, CPU: 3, IO: 3  PID: 5, NI: 5, STATE: Г, CPU: 4, IO: 3  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 4, NI: 1, STATE: О, CPU: 0, IO: 1  PID: 6, NI: 3, STATE: И, CPU: 2, IO: 3  PID: 5, NI: 5, STATE: Г, CPU: 4, IO: 3  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 4, NI: 1, STATE: З, CPU: 0, IO: 0  PID: 6, NI: 3, STATE: И, CPU: 1, IO: 3  PID: 5, NI: 5, STATE: Г, CPU: 4, IO: 3  =======================================  PID: 7, NI: 0, STATE: И, CPU: 17  Очередь 1 (RR), динамические приоритеты, такт 2:  PID: 7, NI: 0, STATE: И, CPU: 16  =======================================  Очередь 1 (RR), динамические приоритеты, такт 1:  PID: 7, NI: 0, STATE: И, CPU: 15  Очередь 1 (RR), динамические приоритеты, такт 2:  PID: 7, NI: 0, STATE: И, CPU: 14  =======================================  Очередь 1 (RR), динамические приоритеты, такт 1:  PID: 7, NI: 0, STATE: И, CPU: 13  Очередь 1 (RR), динамические приоритеты, такт 2:  PID: 7, NI: 0, STATE: И, CPU: 12  =======================================  Очередь 1 (RR), динамические приоритеты, такт 1:  PID: 7, NI: 0, STATE: И, CPU: 11  Очередь 1 (RR), динамические приоритеты, такт 2:  PID: 7, NI: 0, STATE: И, CPU: 10  =======================================  [!] У процесса с PID 7 был изменен приоритет на 2  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 7, NI: 0, STATE: И, CPU: 8, IO: 3  PID: 4, NI: 1, STATE: З, CPU: 0, IO: 0  PID: 6, NI: 3, STATE: З, CPU: 0, IO: 0  PID: 5, NI: 5, STATE: З, CPU: 0, IO: 0  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 7, NI: 0, STATE: И, CPU: 7, IO: 3  PID: 4, NI: 1, STATE: З, CPU: 0, IO: 0  PID: 6, NI: 3, STATE: З, CPU: 0, IO: 0  PID: 5, NI: 5, STATE: З, CPU: 0, IO: 0  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 7, NI: 0, STATE: И, CPU: 6, IO: 3  PID: 4, NI: 1, STATE: З, CPU: 0, IO: 0  PID: 6, NI: 3, STATE: З, CPU: 0, IO: 0  PID: 5, NI: 5, STATE: З, CPU: 0, IO: 0  =======================================  Очередь 2 (FCFS), статические приоритеты:  PID: 3, NI: -2, STATE: З, CPU: 0, IO: 0  PID: 7, NI: 0, STATE: И, CPU: 5, IO: 3  PID: 4, NI: 1, STATE: З, CPU: 0, IO: 0  PID: 6, NI: 3, STATE: З, CPU: 0, IO: 0  PID: 5, NI: 5, STATE: З, CPU: 0, IO: 0  =======================================  PID: 4, NI: 1, STATE: З, CPU: 0, IO: 0  PID: 6, NI: 3, STATE: З, CPU: 0, IO: 0  PID: 5, NI: 5, STATE: З, CPU: 0, IO: 0  =======================================  ======================================= |

# ПРИЛОЖЕНИЕ В

# ЭКРАННЫЕ ФОРМЫ, ОТОБРАЖАЮЩИЕ РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММ ЭМУЛЯЦИИ

Screenshot_2021-12-13_20-37-51

Рисунок В.1 – Создание файла

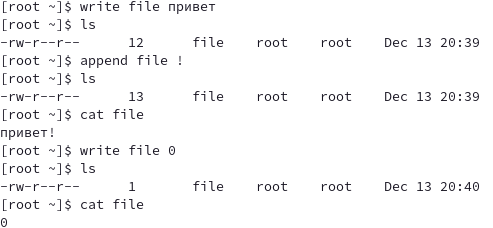


Рисунок В.2 – Запись в файл

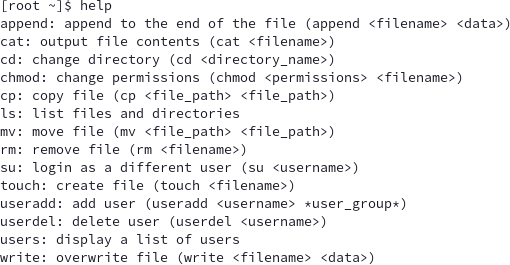


Рисунок В.3 – Команда help

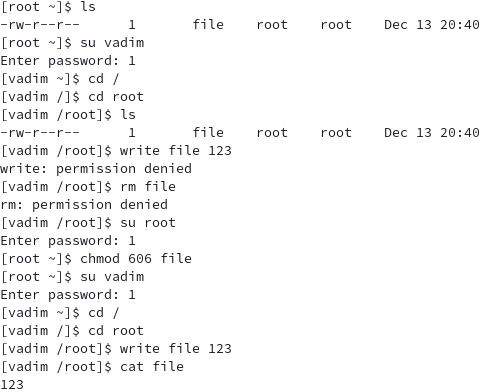


Рисунок В.4 – Смена пользователя, права доступа

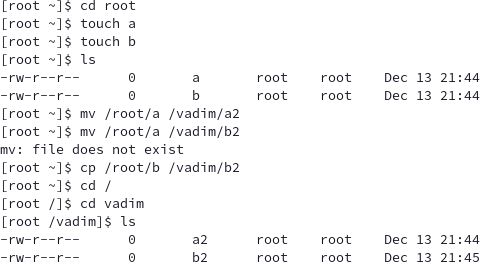


Рисунок В.5 – Перемещение и копирование файлов

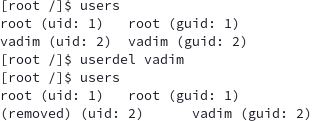


Рисунок В.6 – Удаление пользователя

# ПРИЛОЖЕНИЕ Г

# ЛИСТИНГ ПРОГРАММ