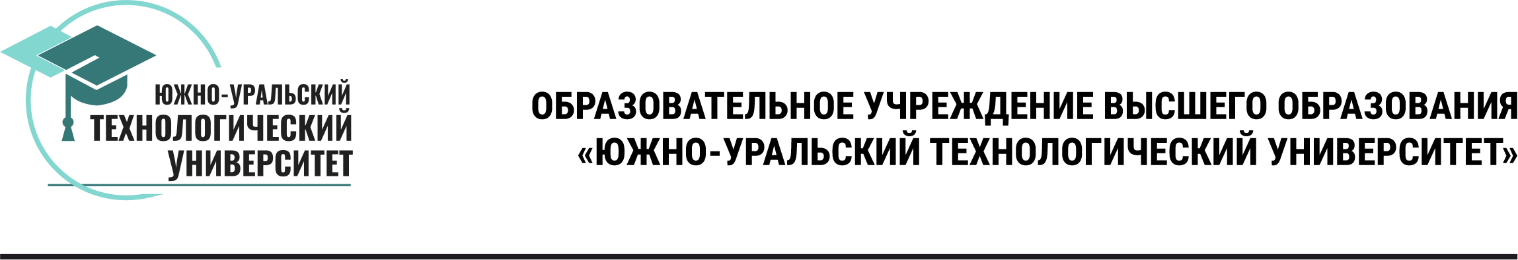
**

Кафедра «Техника и технологии»

Отчёт

по дисциплине:

«Операционные системы»

Направление подготовки/специальность 09.03.02 Информационные системы и технологии

(код, наименование)

Обучающийся Бикматов Кирилл Олегович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ФИО полностью)

Группа \_\_\_\_\_\_И-307\_\_\_\_\_\_ Подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(номер группы)

Форма обучения \_\_Очная\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверил \_\_\_Ефимов Матвей Александрович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия И.О. преподавателя)

Должность Заместитель заведующего кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Челябинск, 20\_ г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

Образовательное учреждение высшего образования

«Южно-Уральский государственный технологический университет»

Кафедра Техники и технологии

ЗАДАНИЕ

на выполнение курсовой работы

по дисциплине «Операционные системы»

студенту группы И-307

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия Имя Отчество студента)

обучающемуся по направлению 09.03.01 «Информационные системы и технологии»

1. Срок сдачи студентом законченной работы: 30.05.2025 г.
2. Исходные данные к работе
3. Лекции по дисциплине «Операционные системы»
4. Практические работы и методические рекомендации по дисциплине «Операционные системы»
5. Учебники и обучающие ресурсы по программированию на Python, находящиеся в свободном доступе и в научном библиотеке университета.
6. Перечень подлежащих разработке вопросов
   1. Осуществить постановку задачи
   2. Спроектировать необходимые структуры данных и функции обработки их
   3. Реализовать функции обработки структур данных, согласно варианту
   4. Реализовать обработку ошибок
7. Дата выдачи задания: «30» мая 2025 г.

Научный руководитель М.А.Ефимов

Задание принял к исполнению К.О.Бикматов

Оглавление

Оглавление

[Введение 4](#_Toc199524819)

[1. Понятие процесса. Состояния процесса 5](#_Toc199524820)

[2. Планирование процессов в системах пакетной обработки 6](#_Toc199524821)

[3. Планирование процессов в интерактивных системах 7](#_Toc199524822)

[4. Планирование процессов в системах реального времени 8](#_Toc199524823)

[5. Операции над процессами 9](#_Toc199524824)

[6. Приоритеты процессов 10](#_Toc199524825)

[7. Понятие потока. Ресурсы и потоки в операционной системе 11](#_Toc199524826)

[8. Понятие ресурса. Классификация ресурсов 12](#_Toc199524827)

[9. Дисциплины распределения ресурсов на основе очередей 13](#_Toc199524828)

[10. Основные элементы графических интерфейсов 14](#_Toc199524829)

[11. Интерфейс прикладного программирования (API) 15](#_Toc199524830)

[Заключение 16](#_Toc199524831)

[Список использованных источников 17](#_Toc199524832)

# Введение

Операционная система является ключевым компонентом любого компьютера, управляющим аппаратными ресурсами и обеспечивающим взаимодействие между пользователем и программным обеспечением. Одной из основных задач операционной системы является управление процессами и ресурсами.

Данный отчёт посвящён рассмотрению таких аспектов, как понятие процесса и потока, их состояния, планирование, операции над ними, а также классификация и распределение ресурсов. Также рассматриваются графические интерфейсы и принципы работы API.

Целью отчёта является углублённое изучение базовых механизмов управления процессами и ресурсами в современных операционных системах, а также анализ особенностей реализации в различных типах систем: пакетной обработки, интерактивных и реального времени.

# 1. Понятие процесса. Состояния процесса

Процесс представляет собой экземпляр выполняемой программы, включающий:

- адресное пространство памяти;

- регистры процессора;

- стек выполнения;

- открытые файлы и другие ресурсы.

Процесс отличается от программы тем, что он находится в активном состоянии выполнения и имеет своё состояние в текущий момент времени.

Состояния процесса:

| Состояние | Описание |

|------------------|----------|

| Новый (new) | Процесс создан, но ещё не передан планировщику |

| Готовый (ready) | Ожидает получения процессорного времени |

| Выполняющийся (running) | Использует процессор для выполнения |

| Ожидание (waiting) | Ждёт завершения внешнего события (например, ввод/вывод) |

| Завершённый (terminated) | Выполнение процесса завершено |

Переходы между состояниями происходят под управлением планировщика операционной системы

# 2. Планирование процессов в системах пакетной обработки

Системы пакетной обработки предназначены для автоматического выполнения задач без прямого взаимодействия пользователя.

Основной целью планирования в таких системах является:

- максимизация пропускной способности;

- эффективное использование центрального процессора.

Наиболее распространённые алгоритмы планирования:

- \*\*FCFS (First-Come, First-Served)\*\* – первый пришёл — первый выполнился.

- \*\*SJF (Shortest Job First)\*\* – сначала самая короткая задача.

- \*\*Приоритетное планирование\*\* – задачи выполняются на основе назначенных приоритетов.

Эти алгоритмы применяются в зависимости от условий использования и требований к производительности.

# 3. Планирование процессов в интерактивных системах

Интерактивные системы характеризуются частым взаимодействием пользователя с приложениями. Ответ должен быть быстрым, поэтому важнейшей характеристикой является время отклика.

Цель планирования в таких системах:

- минимизация времени отклика;

- обеспечение справедливости между процессами;

- предотвращение голодания.

Применяются следующие алгоритмы:

- \*\*Round Robin (RR)\*\* – вытесняющее планирование с квантованием времени.

- \*\*Многоуровневые очереди\*\* – разделение задач на категории.

- \*\*Обратная связь с несколькими очередями\*\* – динамическое изменение приоритета.

Эти методы позволяют эффективно управлять множеством одновременно работающих процессов.

# 4. Планирование процессов в системах реального времени

Системы реального времени используются в областях, где требуется строгое соблюдение временных ограничений: авиация, медицина, автоматизация и т. д.

Типы систем реального времени:

- \*\*Жёсткие (hard real-time):\*\* просрочка недопустима.

- \*\*Мягкие (soft real-time):\*\* небольшая задержка допустима.

Основные алгоритмы планирования:

- \*\*RMS (Rate-Monotonic Scheduling)\*\* – приоритет зависит от частоты запуска задачи.

- \*\*EDF (Earliest Deadline First)\*\* – выбирается задача с ближайшим дедлайном.

Эти подходы обеспечивают предсказуемое поведение и своевременное выполнение задач.

# 5. Операции над процессами

Процессы могут создаваться, уничтожаться, приостанавливаться и возобновляться. Эти операции управляются ядром операционной системы.

Основные операции:

- \*\*Создание процесса:\*\* `fork()` создаёт копию текущего процесса; `exec()` заменяет образ процесса новой программой.

- \*\*Уничтожение процесса:\*\* `exit()` — нормальное завершение; `kill()` — принудительное завершение.

- \*\*Приостановка и возобновление:\*\* `suspend()`, `resume()`.

- \*\*Коммуникация между процессами (IPC):\*\* каналы, очереди сообщений, разделяемая память.

Эти операции обеспечивают жизненный цикл процесса и взаимодействие между различными задачами.

# 6. Приоритеты процессов

Приоритет процесса — числовое значение, определяющее его право на получение процессорного времени.

Виды приоритетов:

- \*\*Статические:\*\* устанавливаются на момент создания.

- \*\*Динамические:\*\* меняются во время выполнения (например, для предотвращения голодания).

Высокий приоритет означает больший шанс получить доступ к процессору. Это важно в системах реального времени и многозадачных средах.

# 7. Понятие потока. Ресурсы и потоки в операционной системе

Поток (thread) — минимальная единица выполнения внутри процесса.

Особенности потоков:

- Все потоки одного процесса разделяют одно адресное пространство.

- Переключение между потоками занимает меньше времени, чем между процессами.

- Облегчает параллелизм внутри одного процесса.

Преимущества потоков:

- Экономия ресурсов.

- Упрощённая межпоточная коммуникация.

Потоки широко используются в современных многопоточных приложениях.

# 8. Понятие ресурса. Классификация ресурсов

Ресурс — любой компонент системы, который может быть использован процессом или потоком.

Классификация ресурсов:

- По типу использования:

- Вытесняемые (можно временно забрать)

- Невытесняемые (их нельзя забирать без последствий)

- По виду:

- Процессорное время

- ОЗУ

- Файлы

- Устройства ввода/вывода

- По возможности совместного использования:

- Разделяемые (например, память)

- Неразделяемые (например, принтер)

Правильное распределение ресурсов — одна из ключевых задач операционной системы.

# 9. Дисциплины распределения ресурсов на основе очередей

Очереди используются для управления доступом к ресурсам и организации запросов.

Типы дисциплин обслуживания:

- FIFO (First In – First Out): первый в очереди — первым обслуживается.

- LIFO (Last In – First Out): последний пришёл — первым обслужен.

- Приоритетная очередь: сначала обслуживаются элементы с высоким приоритетом.

- Циклическая очередь (Round Robin): равномерное распределение ресурсов.

Эти модели помогают организовать эффективное и справедливое распределение ресурсов между процессами.

# 10. Основные элементы графических интерфейсов

Графический интерфейс пользователя (GUI) — способ взаимодействия пользователя с системой через визуальные элементы.

Основные элементы GUI:

- Окна

- Кнопки

- Меню

- Диалоговые окна

- Текстовые поля

- Списки

- Иконки

- Панели инструментов

- Контекстные меню

Фреймворки GUI:

- Windows API

- Qt

- GTK+

- JavaFX

- WPF

Графические интерфейсы значительно упрощают взаимодействие с компьютером для конечного пользователя.

# 11. Интерфейс прикладного программирования (API)

API (Application Programming Interface) — набор функций, предоставляемых библиотеками или сервисами, для взаимодействия с другими программами или системами.

Назначение API:

- Упрощение разработки программ.

- Инкапсуляция сложных действий.

- Единый способ взаимодействия.

Примеры API:

- WinAPI — для Windows.

- POSIX — для UNIX-подобных систем.

- OpenGL — для работы с графикой.

- REST API — для взаимодействия с веб-сервисами

# Заключение

В данном отчёте были рассмотрены ключевые аспекты управления процессами и ресурсами в операционных системах. Изучены состояния процессов, алгоритмы планирования в различных типах систем, понятия потока и ресурса, а также способы организации пользовательского и программного взаимодействия.

Эти знания необходимы для понимания внутренних механизмов операционных систем, проектирования многозадачных приложений и систем реального времени, а также оптимизации использования ресурсов компьютера. Полученные навыки могут быть применены при разработке программного обеспечения, администрировании систем и изучении более сложных тем в области информационных технологий.

# Список использованных источников

1. ГОСТ 7.32–2017. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления.

2. Таненбаум А.С., Бос Х. Современные операционные системы. — СПб.: Питер, 2019.

3. Столлингс В. Операционные системы: internals and design principles. — Вильямс, 2020.

4. Microsoft Docs. Windows API. [https://docs.microsoft.com](https://docs.microsoft.com)

5. Qt Documentation. [https://doc.qt.io](https://doc.qt.io)

6. Oracle JavaFX Documentation. [https://openjfx.io](https://openjfx.io)

```