

# Отчет по Курсовому проекту

## по курсу "Операционные системы"

Студент группы М8О-208Б-23 Соловьева Надежда Сергеевна

Работа выполнена

Преподаватель: Егор Живалев

1. **Тема:** Аллокаторы памяти

2. **Цель работы:**

- Приобретение практических навыков в использовании знаний, полученных в течении курса
- Проведение исследования в выбранной предметной области

3. **Задание:** Исследование 2 аллокаторов памяти: необходимо реализовать два алгоритма аллокации памяти и сравнить их по следующим характеристикам:

- Фактор использования
- Скорость выделения блоков
- Скорость освобождения блоков
- Простота использования аллокатора

Вариант 13: "Необходимо сравнить два алгоритма аллокации: списки свободных блоков (первое подходящее) и блоки по 2 в степени n"

Каждый аллокатор памяти должен иметь функции аналогичные стандартным функциям free и malloc (realloc, опционально). Перед работой каждый аллокатор инициализируется свободными страницами памяти, выделенными стандартными средствами ядра. Необходимо самостоятельно разработать стратегию тестирования для определения ключевых характеристик аллокаторов памяти. При тестировании нужно свести к минимуму потери точности из-за накладных расходов при измерении ключевых характеристик, описанных выше.

В отчете необходимо отобразить следующее:

- Подробное описание каждого из исследуемых алгоритмов
- Процесс тестирования
- Обоснование подхода тестирования
- Результаты тестирования
- Заключение по проведенной работе

4. **Код решения:** [best\\_fit\\_allocator \(src/best\\_fit\\_allocator.c\)](#), [power2\\_allocator \(src/power2\\_allocator.c\)](#)

5. **Реализация тестирования:** [main \(src/main.c\)](#)

6. **Анализ:**

**Описание алгоритмов:**

I . "Best-fit аллокатор"

Подробнее в [bestfit \(bestfit.md\)](#)

Алгоритм основан на поиске наиболее подходящего свободного блока памяти для запроса.

- Использует связный список для отслеживания свободных блоков
- При выделении памяти ищет блок с минимальной разницей между запрошенным и доступным размером
- Поддерживает разделение блоков для минимизации внутренней фрагментации
- Объединяет смежные свободные блоки при освобождении памяти

## "II. Power-of-2 аллокатор"

Подробнее в [power2 \(power2.md\)](#)

Алгоритм использует фиксированные размеры блоков, округленные до степени двойки.

- Поддерживает отдельные списки для каждого размера блока (степени двойки)
- Быстрое выделение памяти благодаря отсутствию поиска
- Разделяет большие блоки на меньшие при необходимости
- Потенциально больший расход памяти из-за округления размеров

## Тестирование

Реализовано тестирование следующих характеристик:

- 1). Скорость выделения памяти
- 2). Скорость освобождения памяти
- 3). Коэффициент использования памяти
- 4). Накладные расходы на управление памятью

## Обоснование подхода тестирования

- Размер памяти (1MB) выбран для моделирования реальных условий использования
- 1000 итераций обеспечивают статистически значимые результаты
- Случайные размеры блоков (1-1024 байт) имитируют реальные сценарии использования
- Измерение времени выполняется с использованием `clock()` для точности
- Отслеживание как запрошенной, так и фактически использованной памяти

## Результаты тестирования (усредненные после нескольких экспериментов)

Характеристика	Best-fit аллокатор	Power-of-2 аллокатор
Время аллокации	0.001549 секунд	0.000278 секунд
Время освобождения	0.000006 секунд	0.000012 секунд
Общая память	1048544 байт	1048288 байт
Запрошено памяти	509255 байт	498242 байт
Использовано памяти	533279 байт	706720 байт
Коэффициент использования	50.86%	67.42%
Накладные расходы	~24 байт	0 байт

## Сравнение и заключение

### I. Скорость работы:

- Power-of-2 аллокатор примерно в 5.5 раз быстрее при выделении памяти

- Best-fit аллокатор в 2 раза быстрее при освобождении памяти

## II. Эффективность использования памяти:

- Best-fit более эффективен по использованию памяти (меньше фрагментация)
- Power-of-2 имеет overhead около 42% из-за округления размеров

## III. Рекомендации по применению:

- Best-fit предпочтителен для систем с ограниченной памятью
- Power-of-2 лучше подходит для систем, где критична скорость аллокации

## IV. Особенности реализации:

- Best-fit требует более сложной реализации для управления памятью
- Power-of-2 проще в реализации благодаря фиксированным размерам блоков

7. **Вывод:** В процессе работы над курсовым проектом мне удалось реализовать 2 алгоритма аллокаторов, протестировать их и сравнить характеристики. Я смогла успешно применить на практике знания, полученные на курсе в этом семестре. Как говорится: *"Вот и все! Неплохая получилась история: интересная, весёлая, порой немного грустная, а главное поучительная. Она научила быть нас смелыми и не бояться вызовов, которые готовит нам жизнь. Помогала нам добавиться поставленных целей несмотря ни на что... Но, самое главное, что у этой истории счастливый конец"*. Личная оценка: 8/10.