

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ**

**ОТЧЕТ О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ**

Шаго Павел Евгеньевич

22.Б07-ПУ, 01.03.02 Прикладная математика и информатика

**Планировщики 2**

Научный руководитель

к.ф.-м.н., доцент Корхов В. В.

Санкт-Петербург

2 мая 2025 г.

# Содержание

<b>Введение</b>	<b>3</b>
<b>1 Earliest eligible virtual deadline first (EEVDF)</b>	<b>4</b>
1.1 Введение . . . . .	4
1.2 Рассмотр причин отказа от CFS . . . . .	4
<b>2 sched_ext</b>	<b>5</b>
2.1 Введение . . . . .	5
<b>Заключение</b>	<b>6</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>7</b>

# Введение

Данная работа продолжает исследования связанные с планировщиками и фокусируется на конкретных реализациях, архитектурных решениях, сравнительном анализе.

# 1 Earliest eligible virtual deadline first (EEVDF)

## 1.1 Введение

**EEVDF** — планировщик потоков в ядре Linux, который в 2023 году (с версии ядра Linux 6.6) вытеснил Completely Fair Scheduler (CFS) в качестве планировщика по умолчанию.

CFS на тот момент использовался в ядре 16 лет, то есть ровно половину всего времени существования ядра Linux. С работой CFS косвенно сталкивался любой человек из-за широкого Это решение поражает, казалось бы CFS за такой срок уже доведен до идеала, поэтому причина замены

Более полное базовое описание дано в главе 6 (раздел 3[6.3]) первого отчета о научно-исследовательской работе[1].

В прошлом отчете данный планировщик потоков был упомянут, но не рассмотрен в настоящей работе он рассматривается подробно.

## 1.2 Рассмотр причин отказа от CFS

## 2 sched\_ext

### 2.1 Введение

**sched\_ext** (scheduler extensions) — инфраструктура ядра Linux позволяющая описывать планировщик набором BPF программ (поэтому sched\_ext также называют BPF scheduler). BPF (Berkeley Packet Filter) — это такой специальный байт-код позволяющий добавлять функциональность в ядро из пользовательского пространства, т.е. исполнять код находящийся в непривилегированном пользовательском пространстве в привилегированных режимах работы ядра. Изначально разработан для добавления функциональности сетевым картам, но с тех пор сильно поменялся, а конкретнее текущий BPF это синоним eBPF (extended BPF), изначальный же вариант сейчас называется cBPF (classic BPF) и практически не используется. BPF предоставляет ограниченный набор функций и спроектирован так, чтобы его можно было легко верифицировать, т.е. чтобы перед исполнением в ядре можно было легко и быстро узнать нет ли в данном коде небезопасных последовательностей инструкций. У BPF есть стандарт, хотя он достаточно условный и он может быть использован для доработки различных систем ядра в числе которых:

## Заключение

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Отчет о научно-исследовательской работе: Планировщики б. и. Павел Е. Шаго 2024
- [2] Joseph Y-T. Leung Handbook of Scheduling: Algorithms, Models, and Performance Analysis CRC Press 2004
- [3] Andrew S. Tanenbaum, Herbert Bos MODERN OPERATING SYSTEMS Pearson Education 2023
- [4] Torvalds L. Linux Kernel [Электронный ресурс] / L. Torvalds. – Режим доступа: <https://github.com/torvalds/linux> – Дата обращения 15.04.2025.
- [5] sched-ext/scx [Электронный ресурс]. – Режим доступа: <https://github.com/sched-ext/scx> – Дата обращения 27.04.2025.
- [6] lkml [Электронный ресурс]. – Режим доступа: <https://lkml.org> – Дата обращения 20.04.2025.
- [7] Liz Rice Learning eBPF O'Reilly Media, Inc 2023
- [8] William Stallings Operating Systems: Internals and Design Principles Pearson Education 2020
- [9] Michael J. Morrison Resource Management and Scheduling in Multitasking Operating Systems CRC Press 2017