

ИСПОЛЬЗОВАНИЕ `IO_URING` ДЛЯ ВЫПОЛНЕНИЯ СИСТЕМНЫХ ВЫЗОВОВ `IOCTL` В LINUX АСИНХРОННО

Сатышев В.И. (ИТМО)

Научный руководитель – инженер факультета прикладной информатики Зубаков А.Н. (ИТМО)

Введение. В последнее время широкое распространение получили идеи использования асинхронного взаимодействия при проектировании различных систем. Одна из возможных областей использования асинхронного взаимодействия – выполнение асинхронных системных вызовов при работе с ядром операционной системы. Так, в версии 5.1 ядра Linux был представлен новый асинхронный интерфейс для выполнения IO операций под названием `io_uring`. Настоящая работа посвящена рассмотрению возможности использования `io_uring` для выполнения системных вызовов `ioctl` асинхронно. С практической точки зрения, отказ от синхронных `ioctl` позволит значительно повысить быстродействие и надежность систем, работа которых в значительной степени зависит от взаимодействия с различными аппаратными компонентами, например, систем хранения данных (СХД). В [1] была предложена модификация `io_uring`, позволяющая вносить реализацию произвольных системных вызовов в `io_uring`. В частности, была продемонстрирована реализация асинхронных `ioctl` системных вызовов для NVMe дисков. Тем не менее, на данный момент отсутствует реализация для протекла SCSI, который остается широко востребован в сфере СХД. В связи с этим, становится актуальным вопрос поддержания асинхронных `ioctl` системных вызовов для SCSI устройств в Linux.

Основная часть. Рассмотрена реализация асинхронных `ioctl` для `sg` и `scsi_ioctl` модулей Linux с использованием функциональности `io_uring_cmd`, представленной в [1]. Приводятся результаты тестирования времени выполнения тестовых программ на языке программирования C как в случае использования синхронных, так и асинхронных `ioctl` с использованием библиотеки `liburing` [2] в контексте сбора SMART метрик с дискового массива.

Выводы. Выполнена реализация асинхронных `ioctl` для `sg` и `scsi_ioctl` модулей Linux. Полученные результаты времени выполнения тестовых программ показывают ускорение в случае использования асинхронного подхода. В контексте практического использования результатов выполнение `ioctl` асинхронно может улучшить быстродействие и надежность СХД, использующих при работе протокол SCSI, за счет уменьшения общего времени, требуемого для взаимодействия с устройствами хранения данных, например, при снятии SMART метрик для проведения диагностики системы.

Список использованных источников:

1. Joshi K., et al. I/O Passthru: Upstreaming a Flexible and Efficient I/O Path in Linux // 22nd USENIX Conference on File and Storage Technologies (FAST 24). – 2024.
2. `liburing` [Электронный ресурс] / Maintainer Jens Axboe. — URL: <https://github.com/axboe/liburing> (дата обращения: 18.01.2025).