



УНИВЕРСИТЕТ ИТМО

Реализация дисциплины обслуживания CBWFQ для ядра Linux

Автор: Куклина Мария Дмитриевна

Научный руководитель: Шинкарук Дмитрий Николаевич

Санкт-Петербург, 2018

Цели и задачи

Цель – реализация дисциплины обслуживания Class-Based Weighted Fair Queueing (CBWFQ) в ядре Linux.

Задачи

- ✓ Проанализировать и сравнить дисциплины обслуживания PQ, CBQ, HTB, HFSC, FWFQ, CBWFQ.
- ✓ Восстановить алгоритмы Class-Based WFQ.
- ✓ Настроить среду для реализации и тестирования.
- ✓ Реализовать модуль ядра CBWFQ в ядре Linux.
- ✓ Реализовать интерфейс утилиты tc для управления модулем.
- ✓ Провести тестирование.



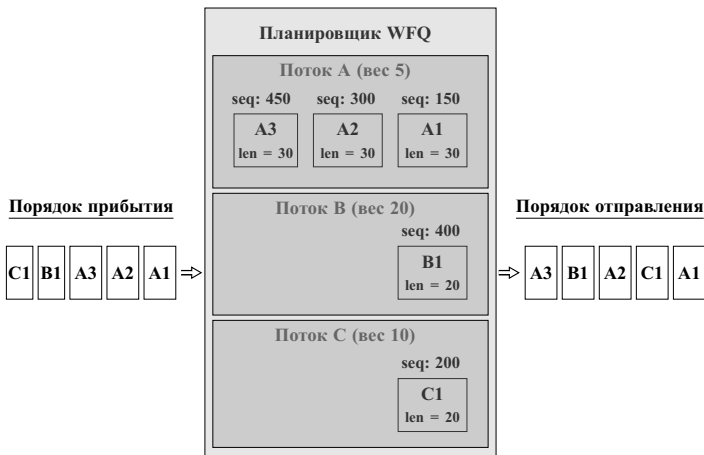
Сравнительная таблица ДО

Свойство	PQ	CBQ	HTB	HFSC	FWFQ	CBWFQ
Метод планирования	RR	WRR	RR	RT/LS	WFQ	WFQ
Отбрасывание	TD	TD	TD	TD	ED/AD	TD/WRED
Честность	-	-	-	-	+	+
Разделение канала	-	+	+	+	-	-
Решение проблемы голодания	-	+	+	+	+	+
Сложность реализации	Низк	Выс	Сред	Выс	Сред	Сред
Сложность конфигурации	Низк	Выс	Сред	Выс	Низк	Низк
Конфигурация классов	-	+	+	+	-	+
Реализация в Linux	+	+	+	+	-	-

Обозначения: (W)RR – (Weighted) Round Robin, RT/LS – Real Time/Link Sharing. TD – Tail Drop, ED/AD – Early Dropping/Aggressive Dropping.



WFQ на основе вычисления порядкового номера пакета



Class-Based Weighted Fair Queueing

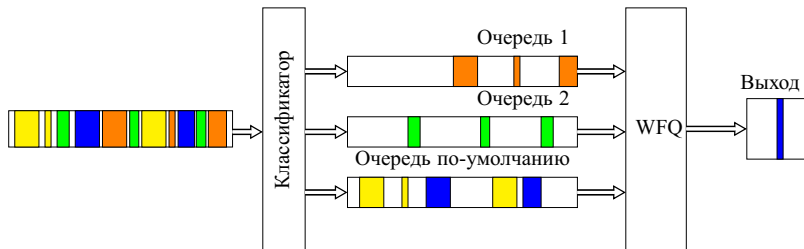


Схема движения пакетов в планировщике CBWFQ

- ✓ Гибкая конфигурация классов.
- ✓ Гарантированное выделение минимальной полосы пропускания.

Качество обслуживания в ядре Linux

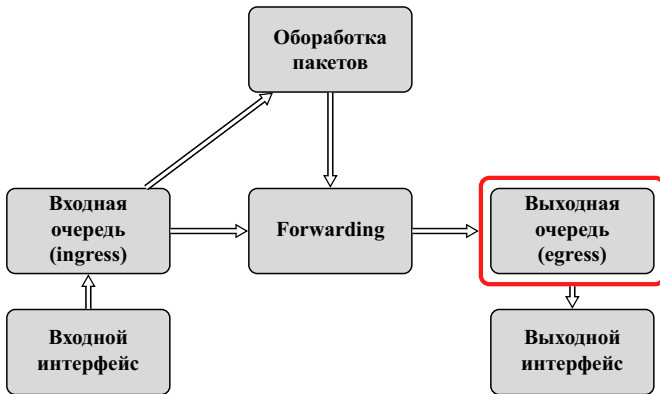


Схема движения пакетов в ядре Linux.

Плагин для утилиты tc

Опции для настройки дисциплины.

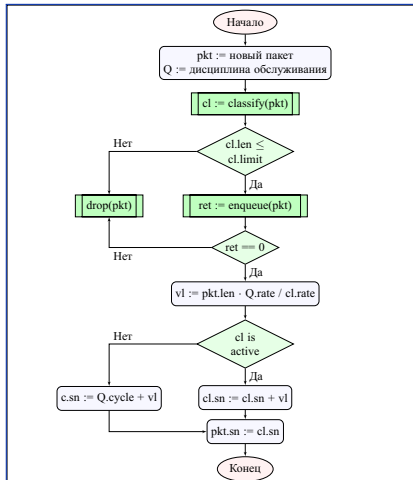
- ✓ “bandwidth” — пропускная способность канала.
- ✓ “default” — ключевое слово, определяющее, что далее пойдут команды для настройки класса по умолчанию.

Опции для настройки классов.

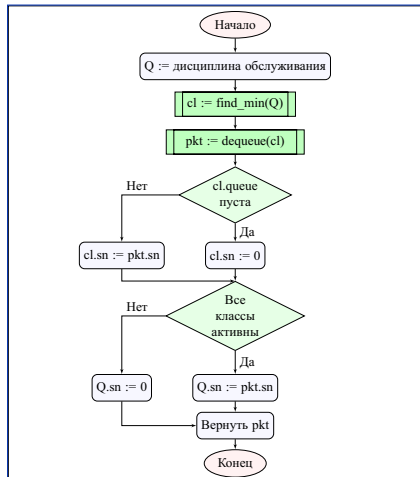
- ✓ “rate” — минимальная пропускная способность для класса.
- ✓ “limit” — максимальное количество пакетов в очереди класса.



Блок-схемы основных алгоритмов CBWFQ

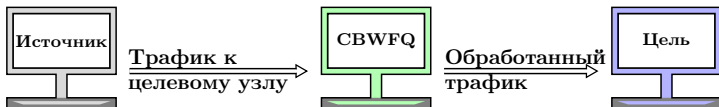


Алгоритм enqueue



Алгоритм dequeue

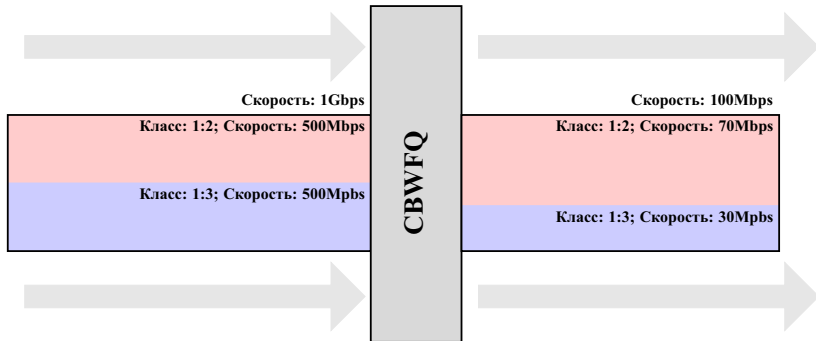
Настройка тестовой среды



```
tc qdisc add dev $IFACE root handle 1: cbwfq bandwidth \
    100Mbps default rate 5Mbps
tc class add dev $IFACE parent 1: classid 1:2 cbwfq \
    rate 25Mbps
tc class add dev $IFACE parent 1: classid 1:3 cbwfq \
    rate 70Mbps
tc filter add dev ens4 parent 1:1 protocol ip u32 match \
    ip dport $TESTPORT1 0xffff flowid 1:2
tc filter add dev ens4 parent 1:0 protocol ip u32 match \
    ip dport $TESTPORT2 0xffff flowid 1:3
```

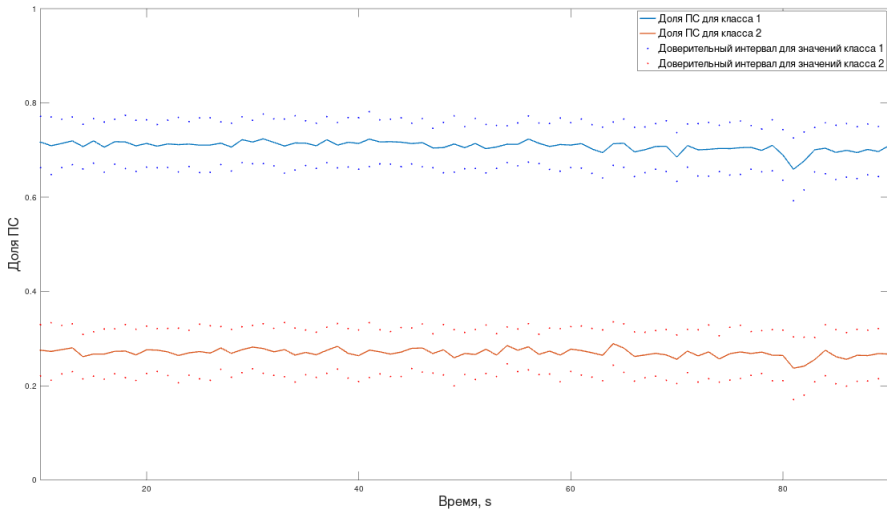


Структура эксперимента №1



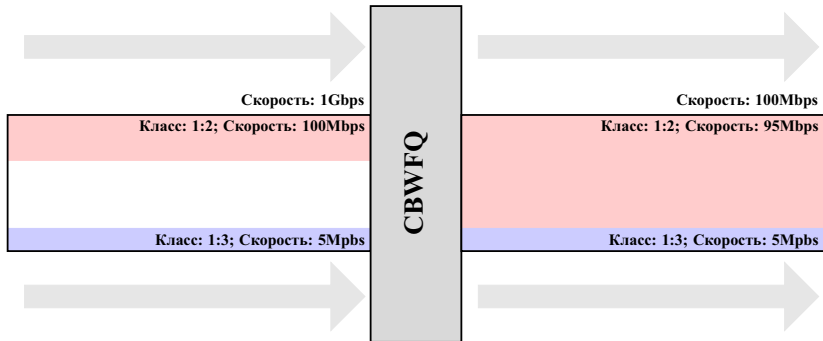


Результат эксперимента №1, БОЛЬШЕ ШРИФТ



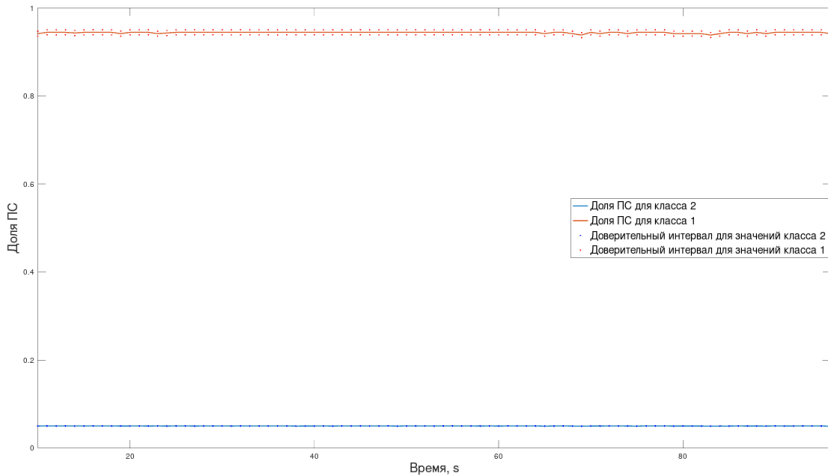


Структура эксперимента №2





Результат эксперимента №2, БОЛЛШЕ ШРИФТ



Вывод

- ✓ Проведён сравнительный анализ классовых дисциплин обслуживания.
- ✓ Проведено исследование модели WFQ.
- ✓ Реализован интерфейс для системы tc.
- ✓ Реализован алгоритм CBWFQ в ядре Linux.
- ✓ Перспектива развития работы: реализация алгоритма WRED и доработка модуля до дисциплины LLQ.



УНИВЕРСИТЕТ ИТМО

Спасибо за внимание!

Санкт-Петербург, 2018