**Планировщик EDF**

**Техническое задание**

Санкт - Петербург

2017

**Техническое задание: Разработка и встраивание планировщиков в ОС Linux**

В общем смысле термина, операционная система является посредником между приложениями и ресурсами. К ресурсам обычно относят память и физические устройства. Но центральный процессор (ЦП) можно также считать ресурсом, который планировщик на некоторое время (измеряемое в отрезках) выделяет задаче. Планировщик обеспечивает параллельное выполнение нескольких программ, распределяя ресурсы ЦП между различными задачами различных пользователей.

Важной целью планировщика является эффективное распределение отрезков процессорного времени при условии обеспечения пользователю времени ожидания на приемлемом уровне. Помимо этого, перед планировщиком могут стоять противоречащие друг другу цели, такие, как минимизация времени ожидания при выполнении критически важных задач реального времени и максимальное использование ресурсов ЦП

Существует три уровня планирования заданий:

* Планирование на верхнем уровне или планирование заданий — это средства, которые определяют, каким заданиям будет разрешено активно конкурировать за захват ресурсов системы. Вошедшие в систему задания становятся процессами или группами процессов.
* Планирование на промежуточном уровне. Средства этого уровня определяют, каким процессам будет разрешено конкурировать за захват ЦП. Планировщик этого уровня ёопределяет, какие процессы приостанавливаются, а какие возбуждаются для обеспечения равномерной загрузки системы.
* Планирование на нижнем уровне. Средства этого уровня выполняют диспетчерские функции, определяя, какому из готовых к выполнению процессов будут предоставлены ресурсы ЦП.

Дисциплина планирования должна быть:

• справедливой;

• обеспечивать максимальную пропускную способность системы;

• приемлемые времена ответа для максимального количества пользователей, работающих в интерактивном режиме;

• предсказуемость;

• минимальные накладные расходы;

• сбалансированное использование ресурсов;

• сбалансированность времени ответа и коэффициента использования ресурсов;

• должна исключать бесконечное откладывание процессов;

• учитывать приоритеты;

• оказывать предпочтение тем процессам, которые занимают ключевые ресурсы;

• предусматривать улучшенное обслуживание для процессов, отличающихся «примерным поведением».

Заметим, что многие из этих целей противоречат друг другу, что делает планирование достаточно сложной проблемой.

Факторы, учитываемые при планировании

Для реализации перечисленных выше целей механизмы планирования должны учитывать следующие факторы:

• лимитируется ли процесс вводом-выводом или ЦП;

• является ли процесс пакетным или диалоговым;

• обязательно ли малое время ответа;

• приоритет каждого процесса;

• частоту переключений с низкоприоритетных процессов, ожидающих освобождения уже занятых ресурсов;

• длительность периода времени, в течение которого ожидает каждый процесс;

• суммарное время работы каждого процесса и оценочное время, необходимое каждому процессу для завершения.

Планирование без переключения предусматривает, что после предоставления ресурсов ЦП какому-либо процессу, отобрать ЦП у этого процесса нельзя. Если же ресурсы ЦП можно отобрать, то говорят о планировании с переключением.

Планирование с переключением необходимо в системах, где процессы высокого приоритета требуют немедленного внимания, например в интерактивных системах разделения времени этот способ планирования позволяет гарантировать приемлемые времена ответа.

Статические приоритеты не изменяются, такой механизм установки приоритетов достаточно прост и не сопряжен с большими издержками. Однако следует учитывать, что такой механизм недостаточно гибок, т.к. не реагирует на изменение окружающей ситуации.

Динамические приоритеты позволяют повысить реактивность системы, т.к. реагируют на изменения ситуации, и начальное значение приоритета процесса может быть изменено на новое, более подходящее значение.

Покупаемые приоритеты дают возможность пользователю повысить приоритет задания и получить более высокий уровень обслуживания за "дополнительную плату" (например, уменьшение кванта времени).

В задании необходимо сначала встроить планировщик типа early deadline first, реализующий алгоритм планирования по ближайшему сроку завершения (EDF), был добавлен в ядро Linux, начиная с версии 3.14, выпущенной 30 марта 2014. При наступлении события планирования (задача завершена, установлена новая задача и т. д.) очередь ищет ближайшую к крайнему времени её выполнения задачу, и этот процесс назначается для выполнения. Планирование по ближайшему сроку завершения оптимально когда существует возможность выполнить набор независимых задач, каждая из которых характеризуется временем наступления, требованием выполнения и крайним сроком завершения, так, чтобы гарантировать выполнение всех задач к крайнему сроку, этот алгоритм назначит задачи к выполнению так, чтобы все они были к крайнему сроку завершены. На рисунке 1 приведен пример планирования для 3 задач.

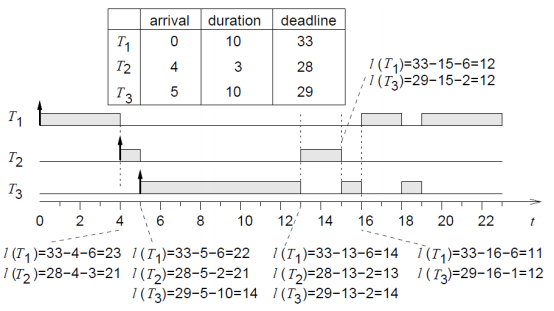


Рис.1. Пример планирования для трех процессов

Так как данный планировщик входит в ядро, его включение состоит в его выборе в файле /sys/block/sda/queue/scheduler :

user@debian:~$ uname -r

3.16.0-4-586

user@debian:~$ dmesg | grep scheduler

[ 0.642934] io scheduler noop registered

[ 0.642937] io scheduler deadline registered

[ 0.642959] io scheduler cfq registered (default)

user@debian:~$ cat /sys/block/sda/queue/scheduler

noop deadline [cfq]

root@debian:echo deadline >/sys/block/sda/queue/scheduler

user@debian:~$ cat /sys/block/sda/queue/scheduler

noop [deadline] cfq

Для того чтобы включить собственный планировщик в ядро, необходимо изменить файл shed.c, скомпилировать ядро и встроить его в систему.

В таблице 1 приведены основные функции реализуемого планировщика:

Таблица 1. Основные функции реализуемого планировщика

