

Разработка и встраивание планировщика в ОС Linux

Студент: Степанов Даниил

Санкт-Петербургский политехнический университет Петра Великого

31 марта 2017 г.

Планирование

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
- ...

Факторы, учитываемые при планировании

Для реализации перечисленных 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

Переключение

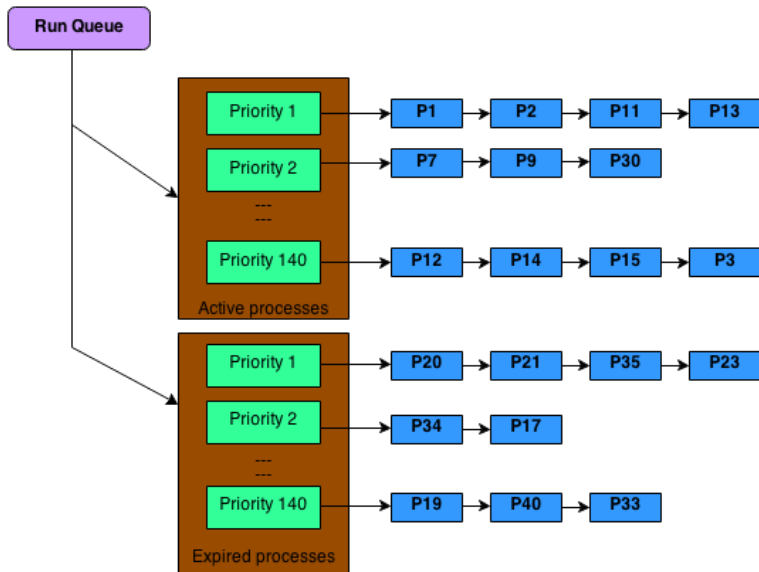
Планирование без переключения предусматривает, что после предоставления ресурсов ЦП какому-либо процессу, отобрать ЦП у этого процесса нельзя.

Если же ресурсы ЦП можно отобрать, то говорят о планировании с переключением.

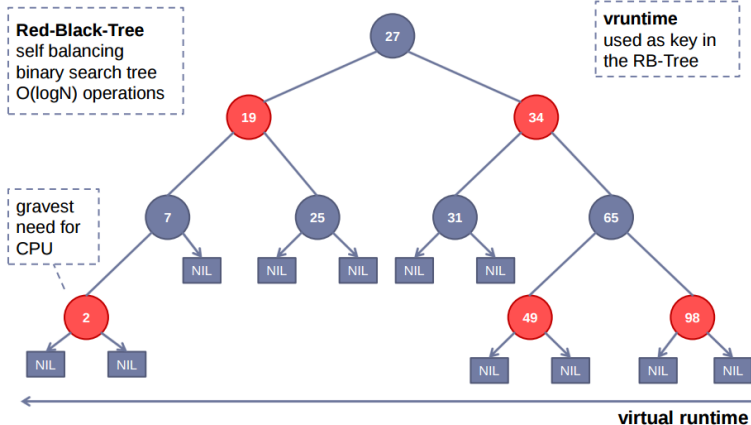
Приоритеты

- Система может присваивать процессам приоритеты автоматически или они могут назначаться извне
- Приоритеты могут быть заслуженными или купленными. Они могут быть статическими или динамическими
- Они могут назначаться по какому-то рациональному принципу или присваиваться в ситуациях, когда системе просто необходимо каким-либо образом различать процессы

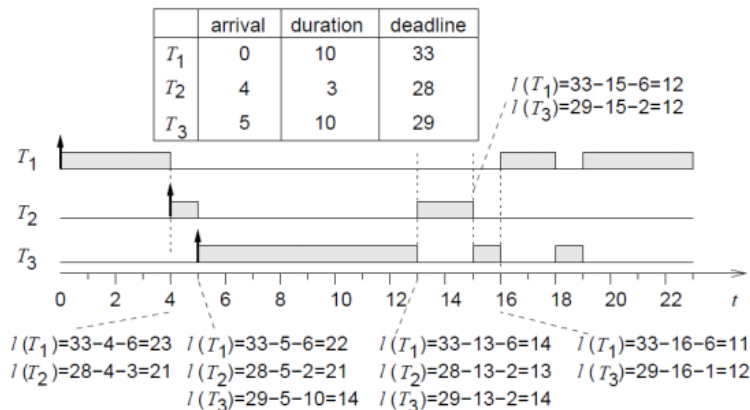
Планировщик $O(1)$



Completely Fair Scheduler



Earliest deadline first




```
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
```

Встраивание своего планировщика

- `/usr/src/linux-*/kernel/sched.c`
- `sudo apt-get install linux-source-2.6.24 kernel-package libncurses5-dev fakeroot`
- `tar xvf linux-source-2.6.24.tar`
- `cp /boot/config-`uname -r` /usr/src/linux/.config`
- `fakeroot make-kpkg --initrd --append-to-version=-custom kernel_image kernel_headers`
- `dpkg -i`
`linux-image-2.6.24.3-custom_2.6.24.3-custom-10.00.Custom_i386.deb`

В текущей версии ядра команды не изменились

Основные функции

Имя функции	Описание функции
<code>schedule</code>	Главная функция планировщика. Планирует выполнение задачи с наивысшим приоритетом.
<code>load_balance</code>	Проверяет, не требуется ли перераспределение нагрузки и, в случае необходимости, предпринимает попытку переноса задач.
<code>effective_prio</code>	Возвращает эффективный приоритет задачи (рассчитанный на основе статического приоритета задачи и всех "наград" и "штрафов").
<code>recalc_task_prio</code>	Вычисляет "награду" или "штраф" для указанной задачи на основе времени простоя.
<code>source_load</code>	Вычисляет консервативную оценку загрузки ЦП-источника (с которого может быть перенесена задача).
<code>target_load</code>	Вычисляет либеральную оценку загрузки целевого ЦП (на который может быть потенциально перенесена задача).
<code>migration_thread</code>	Высокоприоритетный системный поток, осуществляющий миграцию задач между ЦП.

Задания на курсовую работу:

- Разработать планировщик $O(1)$ и встроить его в ядро
- На собственных тестах сравнить производительность встроенных и разработанного планировщиков