

Планировщик микроядра KasperskyOS

Олег Большаков, АО «Лаборатория Касперского», Россия, Москва, 125212 Ленинградское шоссе, д.39А, стр.2 БЦ «Олимпия Парк», Oleg.Bolshakov@kaspersky.com

введение

В операционной системе ядро отвечает за распределение системных ресурсов, одним из которых является центральный процессор. В микроядерной операционной системе KasperskyOS большинство функций самой системы реализовано вне ядра. Однако планировщик остаётся центральным компонентом микроядра, обеспечивая справедливое и предсказуемое использование процессорного времени. В данном докладе рассматриваются архитектура планировщика, модель состояний потоков, поддерживаемые классы планирования и их параметры, приводится практический пример работы планировщика.

Операционная система KasperskyOS изначально разрабатывалась с как встраиваемая с целью обеспечения высокого уровня информационной безопасности и не ориентировалась на использование в системах реального времени. В рамках развития ОС в планировщик были добавлены дисциплины планирования реального времени — карусельная (round-robin) и FIFO, которые будут рассмотрены в докладе.

процессы и потоки

Процесс в KasperskyOS представляет собой контейнер ресурсов — память, дескрипторы, один или несколько потоков. Поток — единица исполнения, содержащая контекст процессора (регистры, указатели стека и кода). Каждый процесс имеет минимум один поток. Потоки совместно используют все ресурсы процесса. Потоки могут приостанавливаться, ожидая завершения операций ввода-вывода, системных вызовов или аппаратных прерываний. Планировщик позволяет большому количеству потоков эффективно работать на ограниченном количестве процессорных ядер.

состояния потоков

Поток KasperskyOS может находиться в одном из семи состояний:

- *ThreadRunning* — выполняется на процессоре;
- *ThreadActive* — готов к исполнению;
- *ThreadSleeping* — заблокирован (спит);
- *ThreadGoingToSleep* — переходит в сон;
- *ThreadYielding* — добровольно освобождает процессор;
- *ThreadTerminating* — завершает работу свою;
- *ThreadSuspended* — исключён из планирования (изменение параметров).

При создании поток переходит из *ThreadSleeping* в *ThreadActive*. Во время работы поток может переходить из одного состояния в другое согласно диаграмме переходов состояний потоков. При завершении своей работы поток переходит в состояние *ThreadTerminating*.

классы планирования

Планировщик KasperskyOS поддерживает три класса планирования:

- *real-time* — дисциплины round robin (RR) и first in, first out (FIFO), приоритет определяет порядок выбора;
- *fair* — класс «честного» планирования (используется по умолчанию), приоритет влияет на долю процессорного времени (примерно 10:3:1 для максимального, среднего и минимального приоритетов);
- *idle* — потоки бездействия, один на каждое ядро.

Планировщик сначала выбирает поток из очереди готовых к исполнению *real-time*, затем из *fair*, а при их отсутствии исполняет *idle*-поток.

параметры планирования

При создании потока задаётся приоритет: от 0 (самый низкий) до 15 (самый высокий) для обычных потоков, до 31 для пользовательских потоков реального времени и до 47 для потоков реального времени ядра. Приоритеты потоков-обработчиков прерываний находятся в диапазоне от 34 до 37. Квант времени работы потока задаётся для дисциплины планирования

RR в пределах 2-100 мс, для *FIFO* — не используется, для *fair* — рассчитывается автоматически.

перепланирование

Планировщик запускается в ответ на события: аппаратные прерывания, таймерные сигналы, межпроцессорные прерывания (IPI), системные вызовы, изменение количества процессорных ядер, создание или удаление потоков, изменение параметров потоков, захват или высвобождение объектов синхронизации. Эти события запускают цикл планирования.

выводы

Планировщик KasperskyOS поддерживает дисциплины планирования реального времени round-robin и FIFO и готов к использованию в системах реального времени. Планировщик предоставляет гибкую настройку режимов работы потоков — различные дисциплины планирования, приоритеты, кванты. Всё это позволяет архитекторам систем точно задавать поведение приложений и оптимизировать использование процессорных ресурсов.

направления развития

- *Защита от инверсии приоритетов* — в текущей реализации есть защита от инверсии на объекте синхронизации мутекс. Ведутся исследования по выявлению других механизмов ОС, которые могут приводить к инверсии приоритетов.
- *Сокращение задержки обработки системного таймера* — ведутся исследования задержек обслуживания прерываний системного таймера.
- *Сокращение задержек обработки прерываний* — ведутся работы по уменьшению времени реакции на прерывания.

ключевые слова

KasperskyOS, микроядро, планировщик, потоки, состояния потоков, классы планирования, реальное время, квант времени, приоритеты.