

Оптимизация процесса тестирования OpenJDK для встраиваемых платформ

Антон Остроухов

Группа 16.Б11-мм, Программная инженерия
Кафедра системного программирования, СПбГУ

Научный руководитель:
д.ф.-м.н., профессор А. Н. Терехов

Консультант:
ассистент каф. системного программирования А. П. Козлов

Введение в проблему

- OpenJDK - открытая реализация JavaSE
- Одна из поддерживаемых платформ - ARMv7 + Linux
 - **soft**: FPU нет, **softfp**: FPU опционален
- Среда Java умеет определять наличие FPU на системе
- Необходимо тщательное тестирование **soft** и **softfp** конфигураций
- Использование **softfp** систем в качестве **soft** (FPU отсутствует) позволит динамически распределять нагрузку и ускорить тестирование

Цель и задачи

Цель: реализация возможности отключения и включения математического сопроцессора в ядре Linux для ARMv7.

Задачи:

1. сделать обзор программной модели FPU на процессоре ARM Cortex-A7, реализующем архитектуру ARMv7;
2. рассмотреть возможные способы смены режима работы FPU;
3. реализовать выбранный способ;
4. опробовать решение в существующем процессе тестирования.

Программная модель FPU в ARMv7

- Бит EN в Floating-Point Exception Register отвечает за вкл/выкл FPU
- Изначально FPU выключен
- При старте ОС исполняется код поддержки FPU (если он написан/подключен), который инициализирует и организует работу FPU
 - В ядре Linux включается параметром CONFIG_VFP в конфигурационном файле перед сборкой ядра

Возможные способы решения проблемы

1. Образ ядра Linux без модуля поддержки FPU
 - а. При запуске ядро не инициализирует FPU
2. Изменение состояния FPU между запусками системы
 - а. Конфигурация FPU при запуске ОС
3. Изменение состояния FPU во время выполнения
 - а. Конфигурация FPU во время переключения контекста

Реализация

1. Структура процесса содержит поле, отвечающее за требуемое состояние FPU.
2. Уведомление ядра о желаемом состоянии FPU через **procfs**.
3. При уведомлении ядра или при переключении контекста происходит изменение бита EN регистра FPXCR.
4. Дочерний процесс наследует состояние FPU у родительского и может изменять его только для себя.

Поддержка оптимизации переключения контекста

- В ядре Linux переключение контекста FPU оптимизировано
 - FPU всегда включается при порождении нового процесса
 - FPU всегда выключается при переключении контекста на существующий процесс
 - FPU включается и загружает контекст текущего процесса только при попытке использования
- Эта оптимизация “ломала” изначальную реализацию
 - Оптимизация расширена для поддержки вкл/выкл FPU во время выполнения

Апробирование: до изменений

softfp

soft

RasPi-01 8 ч. 23 мин.

RasPi-02 10 ч. 34 мин.

RasPi-03 6 ч. 1 мин.

RasPi-04 7 ч. 15 мин.

RasPi-05 9 ч. 0 мин.

RasPi-06 6 ч. 14 мин.

RasPi-07 5 ч. 26 мин.

RasPi-08 9 ч. 10 мин.

RasPi-09 3 ч. 4 мин.

RasPi-10 34 ч. 12 мин.

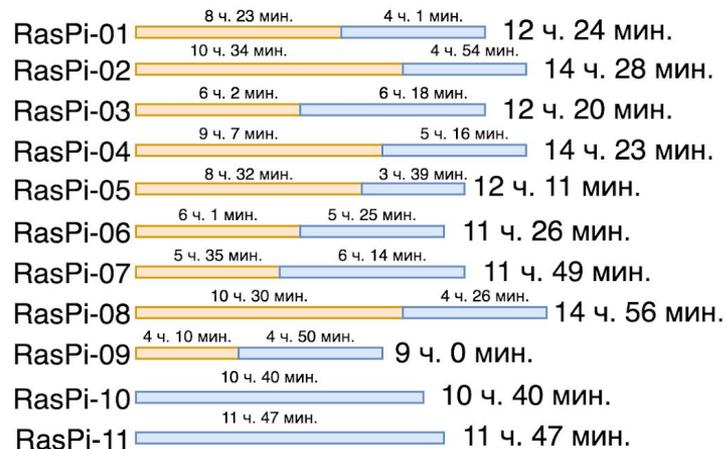
RasPi-11 35 ч. 52 мин.

→ Время тестирования

Апробирование: после изменений

softfp

soft



→ Время тестирования

Результаты

1. Сделан обзор программной модели FPU на процессоре ARM Cortex-A7, реализующем архитектуру ARMv7.
2. Рассмотрены три возможных способа смены режима работы FPU.
3. Реализован способ смены режима работы FPU во время выполнения.
4. Решение опробовано в существующем процессе тестирования. Ускорение составило 58%.