Виртуализация x86 на ARM

студент: Кринкин М. Ю.
руководитель: к. т. н. Кринкин К. В.

СПбАУ НОЦНТ РАН, 2013

Мотивация

- ~25% на содержание ЦОД (центр обработки данный) составляют затраты на энергопотребление [1]

- Использование энергоэффективных ARM процессоров в ЦОД [2] [3] позволит снизить затраты.

[1] http://www.imena.ua
[2] http://habrahabr.ru/post/135723/
[3] http://www.pcworld.com/article/2013277/amd-to-sell-armbased-server-chips-in-2014.html

Цель

- Разработка технологии кросс-архитектурной миграции процессов Linux с использованием динамической трансляции:
  - разработка механизма создания образа (suspend) процесса Linux из userspace
  - восстановление (resume) процесса из образа на архитектуре ARM с помощью динамического транслятора QEMU

Задачи

- Исследование технологий динамической трансляции и миграции процессов
- Реализация механизма создания образа процесса (suspend)
- Реализация восстановления процесса (resume) в QEMU
- Тестирование производительности динамической трансляции

Технологии миграции

- Linux-cr [1]
- CRIU [2]
- DMTCP [3]
- BLCR [4]

[1] https://ckpt.wiki.kernel.org/index.php/Main_Page
[2] http://criu.org/Main_Page
[3] http://dmtcp.sourceforge.net/index.html
[4] https://ftg.lbl.gov/projects/CheckpointRestart/

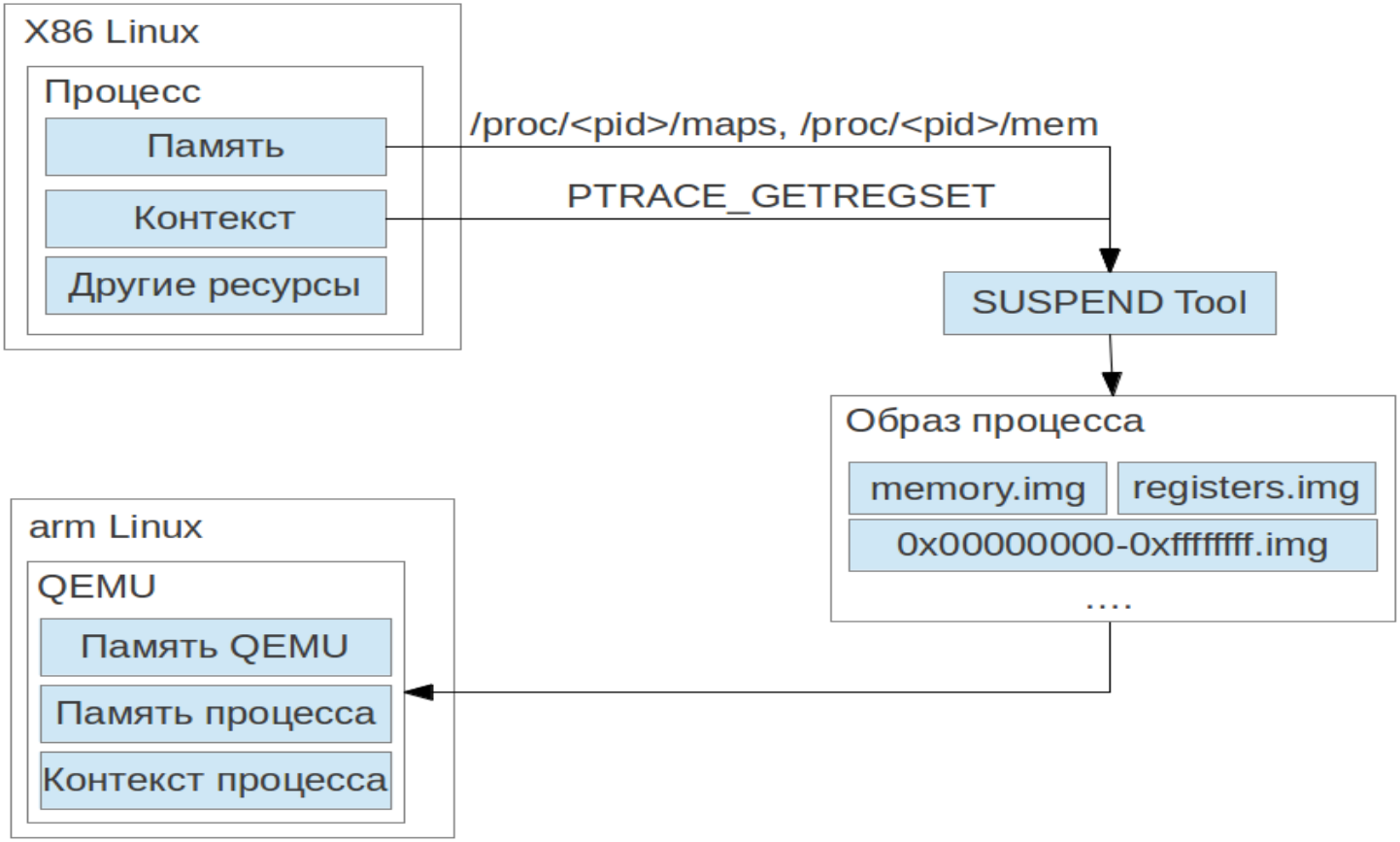
QEMU

- QEMU - Quick EMUlator [1]
- Режимы работы:
  - системный (полная эмуляция)
  - user space (виртуализация одного процесса для Linux и BSD)

- Поддерживаемые платформы:
  - x86 (32 и 64)
  - arm
  - PowerPC
  - Sparc (32 и 64)

[1] http://wiki.qemu.org/Main_Page

Suspend/Resume



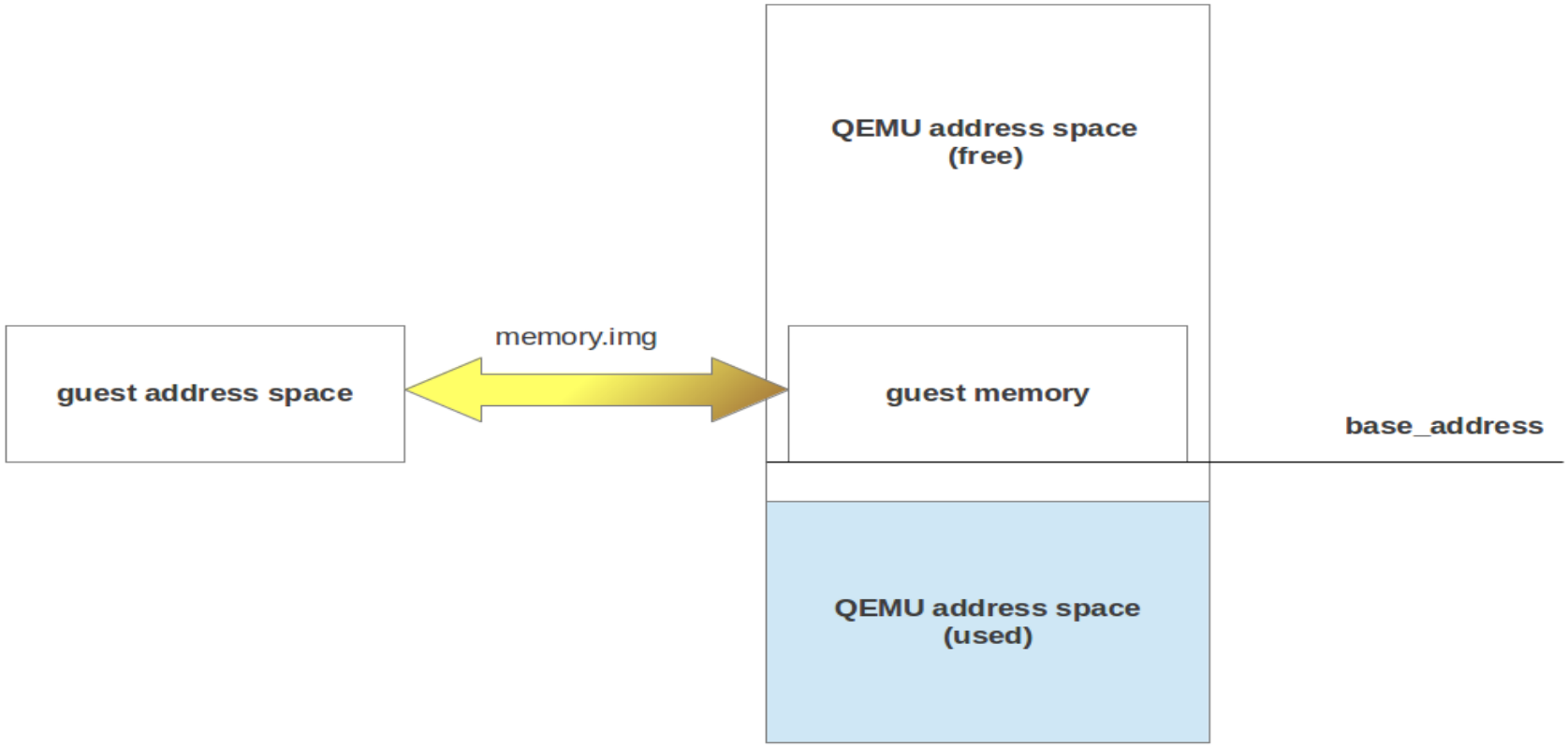
Снимок регистров CPU

- Состояние CPU - регистры процессора, доступны через системный вызов ptrace:
  - Подключение и останов процесса посредством системного вызова Ptrace (команды PTRACE_SEIZE и PTRACE_INTERRUPT)
  - Получение состояния регистров процессора посредством вызова Ptrace (команда PTRACE_GETREGSET)

Снимок памяти процесса

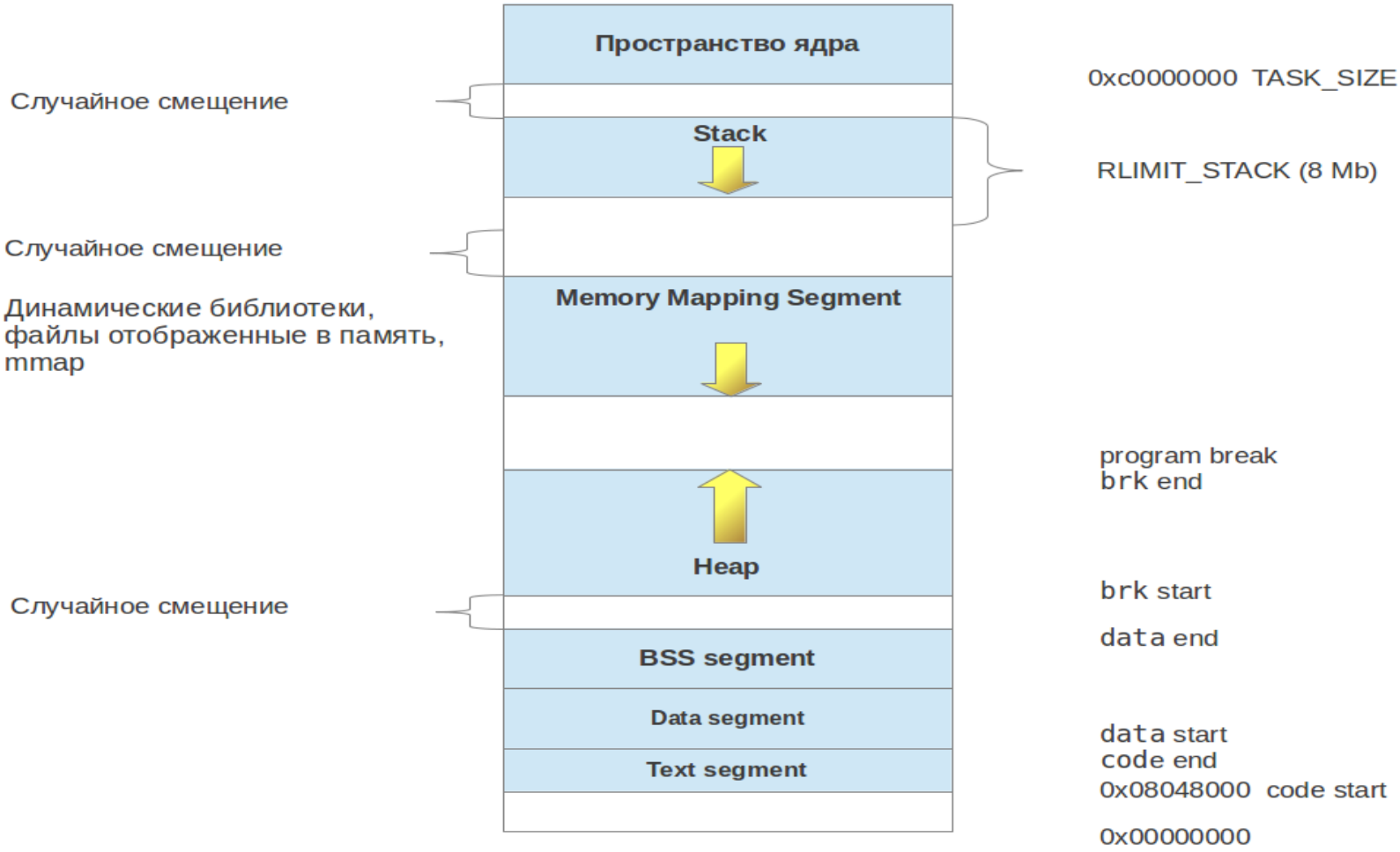
- Описание адресного пространства процесса доступно через файловую систему /proc
  - Список используемых программой регионов адресного пространства доступен в файле /proc/<pid>/maps
  - Содержимое памяти доступно через файл /proc/<pid>/mem

Отображение адресов при динамической трансялции



host_address = guest_address + base_address

Linux memory layout



Результаты тестирования

- Доступ к памяти (чтение/запись):
  - накладные расходы 0.009 мкс/оп
  - нативное выполнение 0.007 мкс/оп
- Арифметика (операция деления):
  - накладные расходы 0.129 мкс/оп
  - нативное выполнение 0.008 мкс/оп
- Ввод/Вывод (без учета дисковых операций):
  - накладные расходы 1.33 мкс/оп
  - нативное исполнение 0.424 мкс/оп

Направления развития

- Повышение скорости динамической трансляции
- Реализация альтернативной модели памяти QEMU для более компактного представления процесса
- Нативное восстановление процесса и встраивание QEMU в виде библиотеки
- Миграция на основе статической трансляции

Результаты

- Рассмотрены существующие технологии миграции процессов.
- Изучен динамический транслятор QEMU.
- Реализован останов процесса Linux:
  - сохранение памяти;
  - сохранение минимального контекста.
- Релизовано восстановление процесса в QEMU.
- Выполнено тестирование производительности.