Разработка слоя совместимости с Linux приложениями в ОС Embox

Докладчик:
Остроухов Антон
Группа 371

Научный руководитель:
Козлов Антон Павлович,
ассистент каф. системного программирования

Введение

- Существует большое количество свободного ПО для разных нужд
- GNU/Linux является самой большой свободной ОС
- В случае необходимости использования существующего ПО в специализированных ОС затрагивается сложный процесс переноса
  - Embox: не все задачи требуют реального времени

Постановка задачи

Создать слой совместимости с Linux приложениями в ОС Embox.

1. Сделать обзор существующих подходов
2. Создать прототип слоя совместимости
3. Создать инфраструктуру для переноса приложений из GNU/Linux

Существующие подходы

- Совместимость на уровне API
  - Embox
- Эмуляция
  - FreeBSD, Windows Subsystem for Linux
- Ядро Linux как подсистема
  - L4Linux (порт Linux для микроядра seL4)
  - Windows Subsystem for Linux 2
  - Linux Kernel Library

Linux Kernel Library

- Linux Kernel Library (LKL) - ядро Linux для использования в произвольном приложении
  - Приложения для чтения/записи файловых систем
  - Использование сетевого стека Linux
- Поддерживает работу в разных ОС

Подходит для использования в качестве основы для слоя совместимости.

Поддержка x86 в LKL

Основная платформа для разработки Embox - x86.

Изначально в LKL x86 не поддерживалась.

- Поиск и изучение причин, эксперименты
- Обсуждение технических трудностей на форуме, code review
- В результате была обеспечена поддержка x86

Интеграция LKL в систему сборки Embox

Все компоненты Embox представлены в виде модулей.

- Создан модуль LKL (файл *Mybuild*)
- Создан скрипт для корректной сборки и компоновки LKL в Embox (файл *Makefile*)
- Дополнена реализация libc в Embox

Модификация LKL

- Модифицирован собственный скрипт сборки LKL для правильной сборки средствами Embox
- Модифицирована и дополнена часть исходного кода с реализацией набора примитивов для POSIX-совместимых ОС
  - Исправлена функция возврата идентификационного номера потока
  - Проблема несовместимости в названиях функций у LKL и Embox

Инфраструктура для переноса приложений

Инфраструктура для произвольного приложения должна включать:

- Скрипты для сборки этого приложения (*Mybuild*, *Makefile*)
- Базовое приложение, включающее инициализацию LKL

Инфраструктура для переноса приложений

Была создана основа, включающая необходимые компоненты.

Ниже представлен фрагмент кода для инициализации LKL.

```
#include <lkl_host.h>
#include <lkl.h>

int main() {
    ...
    snprintf(cmdline, sizeof(cmdline), "mem=64M
loglevel=8");
    ret = lkl_start_kernel(&lkl_host_ops, cmdline);
    ...
```

Отладка

После запуска приложения исполнение останавливалось в определенном месте.

- Проблема в поиске оптимального алгоритма для технологии RAID6 (модуль ядра Linux)
  - Является зависимостью модуля файловой системы Btrfs
  - Требует дополнительной конфигурации LKL

Работа приложения

```
root@embox:/#lkltest
[    0.000000] Linux version 4.19.0+ (me@vbox) (gcc version 5.3.1 20160413 (Ubuntu 5.3.1-14ubuntu2)) #4 Sat May 11 02:56:43 EDT 2019
[    0.000000] bootmem address range: 0x2293000 - 0x6292000
[    0.000000] Built 1 zonelists, mobility grouping on.  Total pages: 16255
[    0.000000] Kernel command line: mem=64M logleve
[    0.000000] Dentry cache hash table entries: 8192 (order: 3, 32768 bytes)
[    0.000000] Inode-cache hash table entries: 4096 (order: 2, 16384 bytes)
[    0.000000] Memory available: 64896k/65532k RAM
[    0.000000] SLUB: HWalign=32, Order=0-3, MinObjects=0, CPUs=1, Nodes=1
[    0.000000] NR_IRQS: 1024
[    0.000000] lkl: irqs initialized
[    0.000000] lkl: no time or timer support provided by host
[    0.000000] pid_max: default: 4096 minimum: 301
[    0.000000] Mount-cache hash table entries: 1024 (order: 0, 4096 bytes)
[    0.000000] Mountpoint-cache hash table entries: 1024 (order: 0, 4096 bytes)
[    0.000000] console [lkl_console0] enabled
[    0.000000] clocksource: jiffies: mask: 0xffffffff max_cycles: 0xffffffff, max_idle_ns: 19112604462750000 ns
[    0.000000] random: get_random_u32 called from bucket_table_alloc.isra.6+0x94/0x1f0 with crng_init=0
[    0.000000] NET: Registered protocol family 16
[    0.000000] NET: Registered protocol family 2
[    0.000000] tcp_listen_portaddr_hash hash table entries: 512 (order: 0, 4096 bytes)
[    0.000000] TCP established hash table entries: 1024 (order: 0, 4096 bytes)
[    0.000000] TCP bind hash table entries: 1024 (order: 0, 4096 bytes)
```

Начало инициализации ядра Linux

Работа приложения

```
[    0.000000] jitterentropy: Initialization failed with host not compliant with requirements: 1
[    0.000000] io scheduler noop registered
[    0.000000] io scheduler deadline registered
[    0.000000] io scheduler cfq registered (default)
[    0.000000] io scheduler mq-deadline registered
[    0.000000] io scheduler kyber registered
[    0.000000] NET: Registered protocol family 10
[    0.000000] NET: Unregistered protocol family 10
[    0.000000] sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver
[    0.000000] Warning: unable to open an initial console.
[    0.000000] This architecture does not have kernel memory protection.
[    0.000000] Run /init as init process
kernel started: Success
lkl_start_kernel returned 0
root@embox:/#
```

Конец инициализации ядра Linux

Результаты

1. Сделан обзор существующих подходов
2. Создана начальная версия слоя совместимости на основе LKL
3. Создана инфраструктура для переноса приложений из GNU/Linux