Московский физико-технический институт (государственный университет)
Физтех-школа радиотехники и компьютерных технологий
Кафедра информатики и вычислительной техники

# Поддержка работы со звуковой подсистемой ALSA в бинарном компиляторе уровня приложений x86->Elbrus

Выполнила: Носкова Е.С., М01-903а
Научный руководитель: к. ф.-м. н.Рожин А.Ф.

Москва 2019

# Звуковая архитектура ALSA

**ALSA** (**A**dvanced **L**inux **S**ound **A**rchitecture) – архитектура звуковых драйверов, обеспечивающая поддержку множества звуковых карт

Преимущества ALSA:

- Широкая поддержка аудиоаппаратуры, от потребительских звуковых карт до профессиональных мультиканальных аудио интерфейсов
- Поддержка Open Sound Subsystem(OSS), обеспечивающего бинарную совместимость с большинством программ использующих OSS

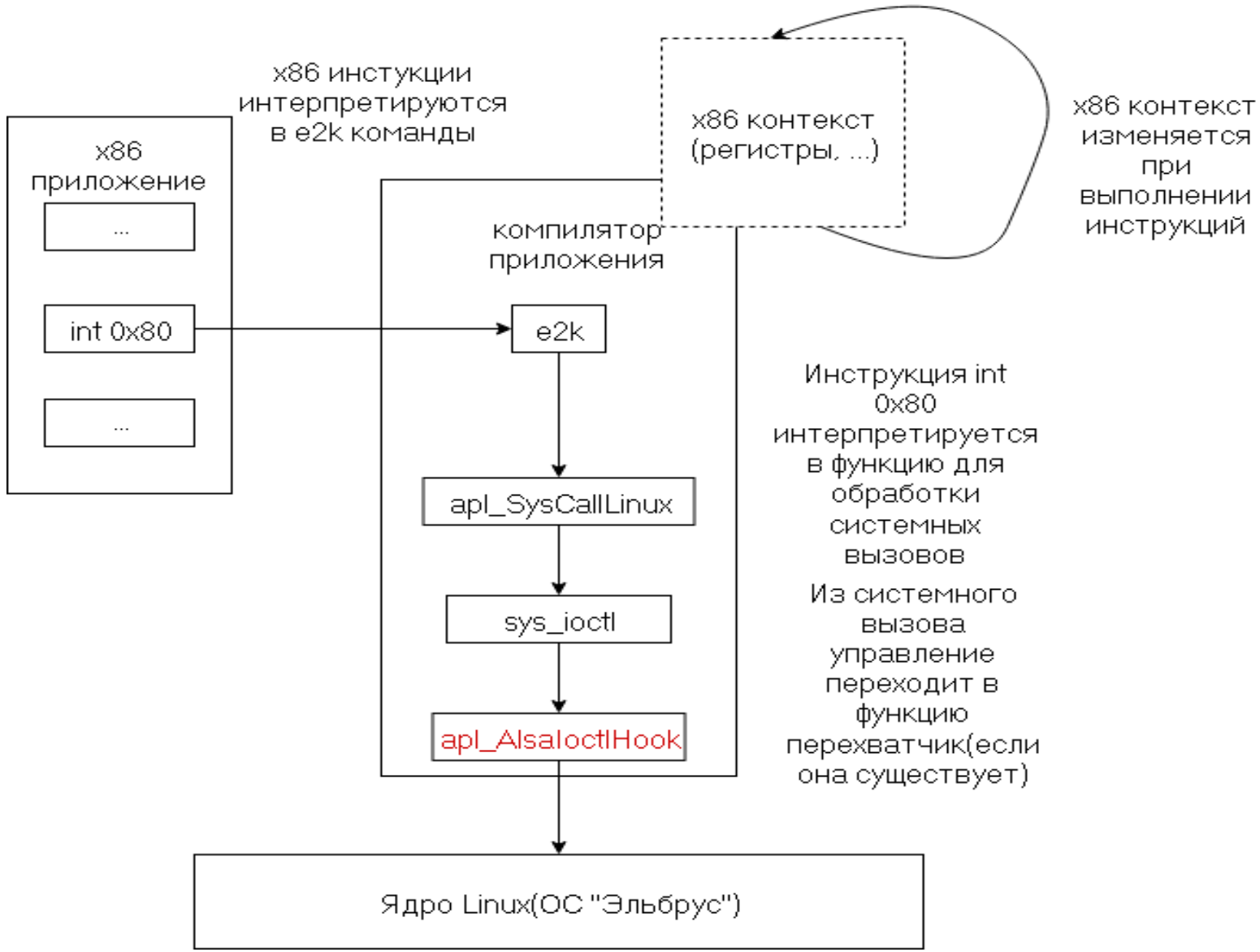
# Цель работы

Реализовать поддержку работы со звуковой подсистемой ALSA в бинарном компиляторе уровня приложений x86->Elbrus

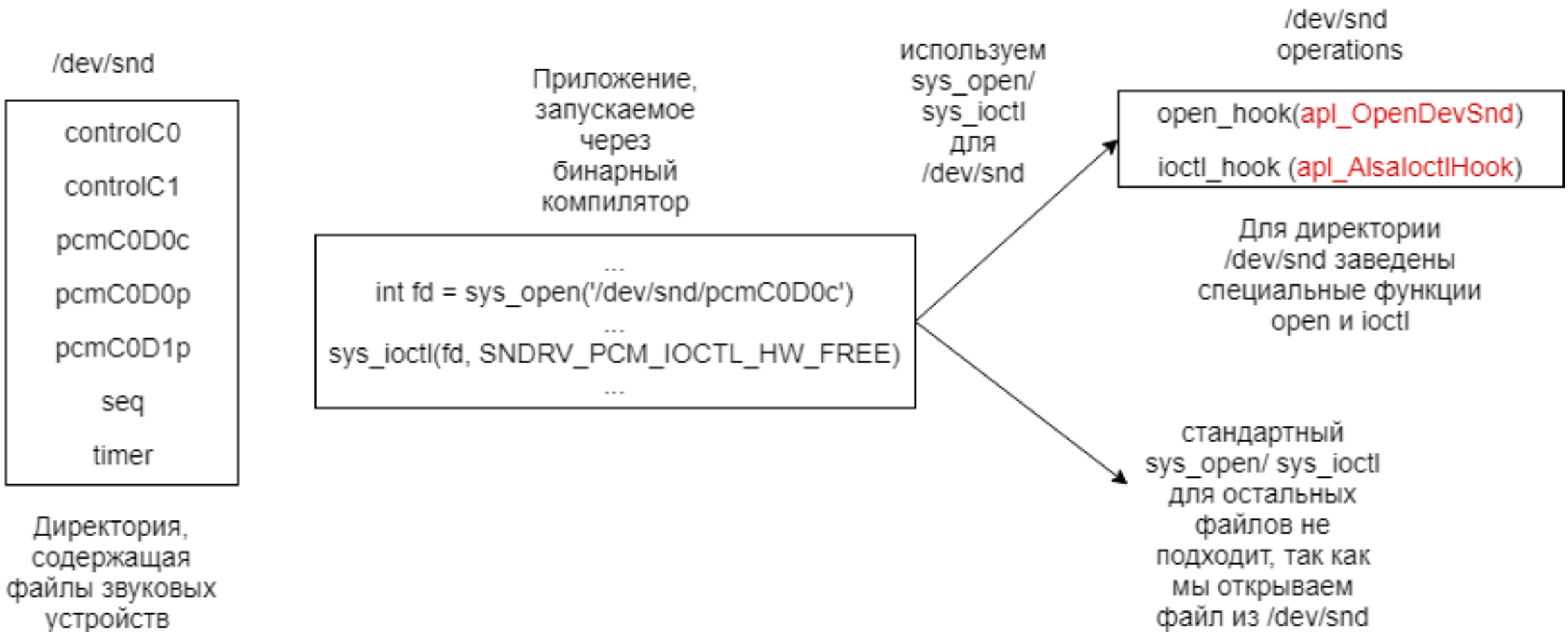
Задачи:

- Формирование принципа обработки запросов от ALSA
- Преобразование адресов, приходящих из x86-приложения
- Поддержка запросов от 32-битных приложений
- Снятие защиты на запись с транслируемого кода
- Тестирование полученного решения

# Обработка системных вызовов компилятором приложений



# Специальная обработка звуковых файлов



# Формирование принципа обработки запросов от ALSA

Проблема:

Поддержать в списке системных вызовов ioctl для работы с ALSA-устройствами

- директория /dev/snd помечается как специальная -для особой обработки компилятором приложений
- создается отдельная ioctl функция для ALSA-устройств
- при открытии файла, находящегося в /dev/snd , функция ioctl, отвечающая за этот файл подменяется на ioctl, который используется только для ALSA-устройств

В дальнейшем при вызове ioctl для данного устройства всегда будет вызываться только специальный для ALSA ioctl

# Преобразование адресов, приходящих из x86-приложения
# Вторичное пространство

Проблема:

Адрес, пришедший из x86 приложения, должен быть преобразован к адресу, с которым может работать ядро

Выполняя сегментную операцию...

LDDSW 0x8200000

...реально обращаемся по адресу

0x8200000 + 1 << 13
(смещение вторичного пространства)

Виртуальная память процесса


Решение:

В реализованных функциях конвертации к исходному адресу добавляется смещение вторичного пространства

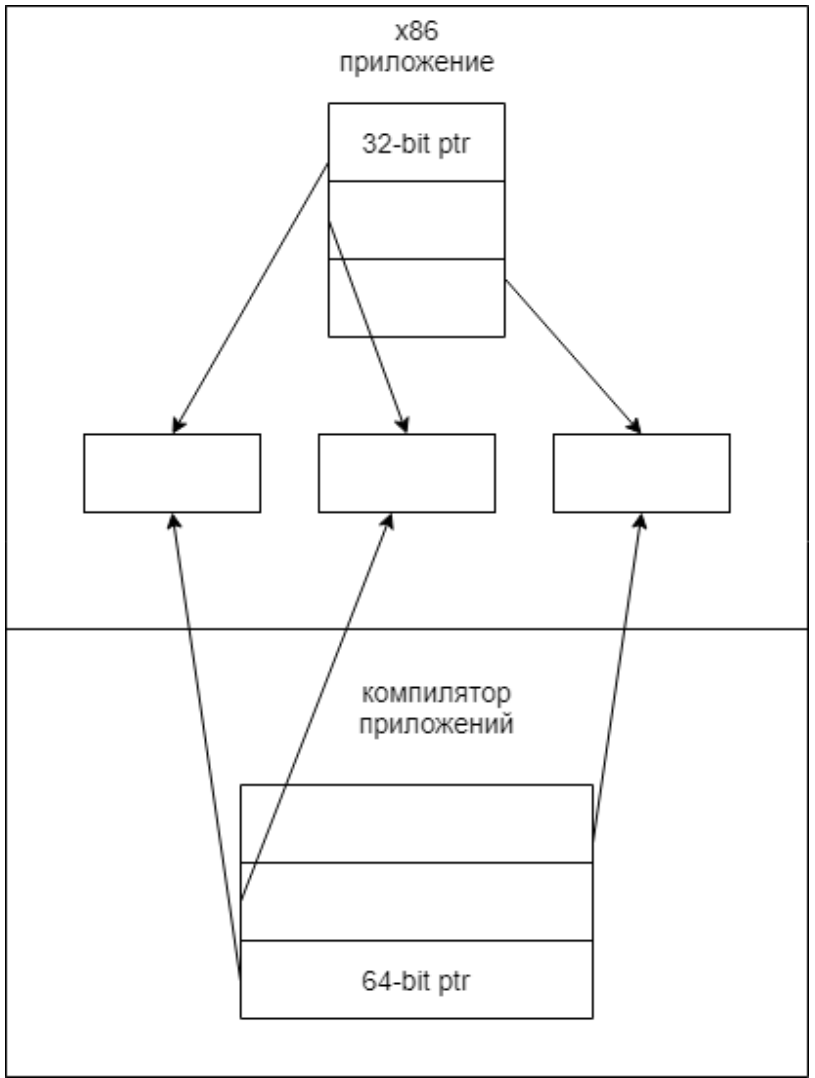
# Преобразование адресов, приходящих из x86-приложения
# Двойные указатели

Двойной указатель  - указатель на массив указателей

Преобразование адреса  -
добавление смещение начала вторичного пространства к адресу

Для аргументов, в состав которых входят двойные указатели:

- Заводятся дополнительные массивы указателей, содержащие преобразованные и непреобразованные адреса
- В функции конвертации производится конвертация из массива непреобразованных в массив преобразованных адресов
- Сформированный массив передается в ядро



# Поддержка запросов от 32-битных приложений Compat-ioctl

Compat ioctl - это ioctl, предназначенный для работы с 32-битным приложением на 64-битном ядре со своими внутренними преобразованиями аргумента

Проблема:

- при преобразовании адреса к адресу вторичного пространства, он становится 64 битным и следовательно существующая реализация compat не может быть использована

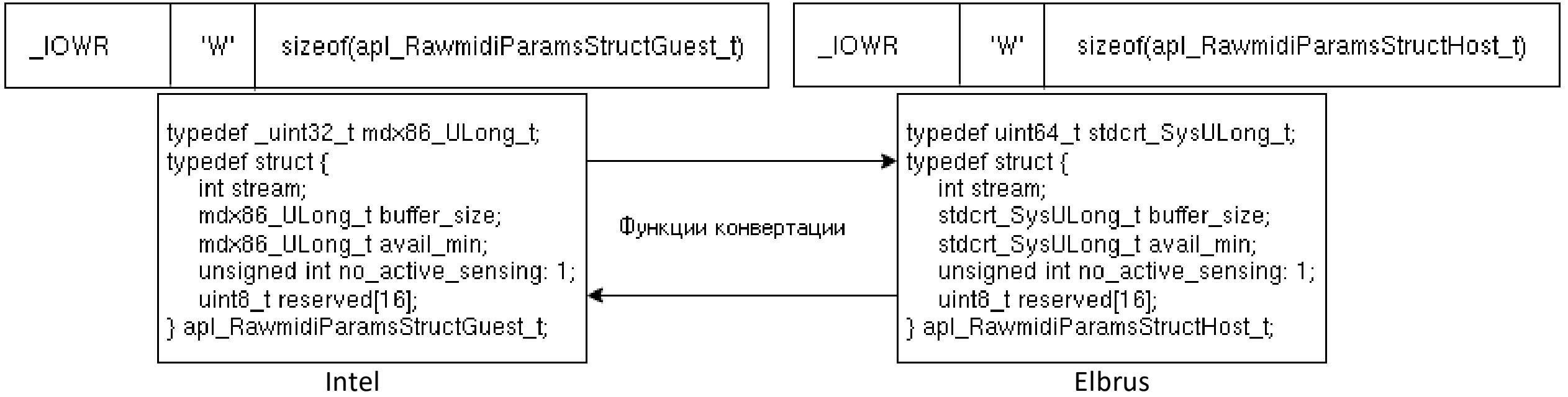
# Поддержка запросов от 32-битных приложений Размер структуры

Проблема:

В 32-битных(Intel) и 64-битных(Elbrus) архитектурах long имеет разные размеры, что влечет за собой разные размеры аргументов, а следовательно и разные команды в X86 и Elbrus

Решение:

Реализованы функции конвертации, в которых происходит преобразование структур



# Снятие защиты на запись с транслируемого кода

Для обеспечения обратной совместимости x86 код, для которого была создана трансляция(Elbrus код) накрывается «защитой», запрещающей его изменять незаметно для компилятора приложений

Отдавая указатель в ядро мы должны гарантировать возможность записи

Проблема:

- Наложение «защиты» может в некоторых случаях вывести из строя компилятор приложений

Решение:

- Снимать «защиту» и запрещать ее установку для некоторых участков памяти, в которых расположены данные, потенциально подверженные изменениям со стороны ядра на время исполнения системного вызова

# Тестирование

Разработанный модуль был протестирован на программном обеспечении, использующем ALSA:

- Alsamixer -программа для настройки параметров ALSA
- MPlayer - медиаплеер

# Результаты

Поддержана работа со звуковой подсистемой ALSA в бинарном компиляторе уровня приложений

X86->Elbrus

- Сформирован принцип обработки запросов от ALSA
- Реализована поддержка запросов от 32-битных приложений
- Добавлено снятие защиты с транслируемого кода
- Осуществлена специальная обработка двойных указателей
- Реализовано 113 ALSA-ioctl команд
- Проведено тестирование полученного решения