



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**  
**К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ**  
**НА ТЕМУ:**

«Брух брах помогите((((()»

Студент группы <ИУ7-83Б>

\_\_\_\_\_  
(Подпись, дата)

<М. Ю. Нитенко>

\_\_\_\_\_  
(И.О. Фамилия)

Руководитель ВКР

\_\_\_\_\_  
(Подпись, дата)

<А. А. Оленев>

\_\_\_\_\_  
(И.О. Фамилия)

Нормоконтролер

\_\_\_\_\_  
(Подпись, дата)

<Это кто>

\_\_\_\_\_  
(И.О. Фамилия)

2022 г.

## **РЕФЕРАТ**

Расчетно-пояснительная записка 18 с., 1 рис., 0 табл., X ист., X прил.

**КЛЮЧЕВЫЕ СЛОВА**

# СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b>	<b>8</b>
<b>1 Аналитическая часть</b>	<b>9</b>
1.1 QEMU . . . . .	9
1.1.1 Оптимизации используемые в QEMU . . . . .	9
1.2 bruh . . . . .	10
<b>2 Конструкторская часть</b>	<b>13</b>
<b>3 Технологическая часть</b>	<b>14</b>
<b>4 Исследовательская часть</b>	<b>15</b>
<b>ЗАКЛЮЧЕНИЕ</b>	<b>16</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>17</b>
<b>ПРИЛОЖЕНИЕ А</b>	<b>18</b>

## **ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ**

## ВВЕДЕНИЕ

Трансляция кода x86 -> ARM.

Процессоры архитектуры ARM занимают большую долю рынка, еще в 2015 году они составляли 35% от рынка процессоров, однако в основном они использовались в портативных устройствах [1]. С появлением процессоров M1 от компании Apple большое число людей начало пользоваться компьютерами на основе архитектуры ARM в домашней обстановке (типа personal computers). Однако, программы собранные под архитектуру x86 не смогут работать на таких компьютерах, им необходим или транслятор, такой как Rosetta 2, или виртуальная машина поддерживающая необходимую архитектуру.

Таким образом что? Боремся с Rosetta 2 что ли? Чето заставило задуматься... Ну пока просто обзор стратегий по оптимизации онлайн и офлайн трансляции кода x86 -> ARM.

# 1 Аналитическая часть

Проблемы эмулятора:

- управление кэшем транслированного кода;
- выделение регистров;
- оптимизация условных блоков;
- direct block chaining????? lmao;
- управление памятью;
- поддержка самоизменяемого кода;
- поддержка исключений;
- поддержка аппаратных прерываний;
- эмуляция режима пользователя.

## 1.1 QEMU

(инфа 2005 года...) В QEMU реализована динамическая трансляция инструкций. QEMU транслирует команды при помощи кода заранее сгенерированного при помощи GCC.

Каждая инструкция процессора разбивается на микрооперации, эти микрооперации реализованы на языке C. Микрооперации выбираются так, чтобы их количество было много меньше количества всех возможных комбинаций инструкций и операндов процессора.

Программа входящая в QEMU — dyngen использует объектные файлы с микроинструкциями и генерирует на их основе динамический генератор кода, именно он используется во время выполнения программы для трансляции. [2]

### 1.1.1 Оптимизации используемые в QEMU

Главной идеей является возможность передачи константных параметров микрооперациям. Для этого GCC генерирует специальный код (?) для каждого константного параметра (я щяс умру..).

Например:

```
movl_T0_r1 # T0 = r1
```

```
addl_T0_im -16 # T0 = T0 - 16
movl_r1_T0 # r1 = T0
```

Таким образом уменьшается количество необходимых микроопераций, так как можно реализовать операции загрузки всех необходимых регистров во временные регистры и проводить операции именно с ними. Эти регистры обычно хранятся в регистрах хоста (например T0, T1 и T2 хранились бы в `rax`, `rbx`, `rcx` на x86).

## 1.2 bruh

Список:

- первое;
- второе;
- пятое;
- десятое.

Формула:

$$c^2 = a^2 + b^2 \tag{1}$$

Ссылаемся на рисунок 1. Информация из источника [?].

### Листинг 1: Пример кода

```
1 package main
2
3 import (
4     "bufio"
5     "fmt"
6     "net/http"
7 )
8
9 func main() {
10     resp, err := http.Get("http://gobyexample.com")
```

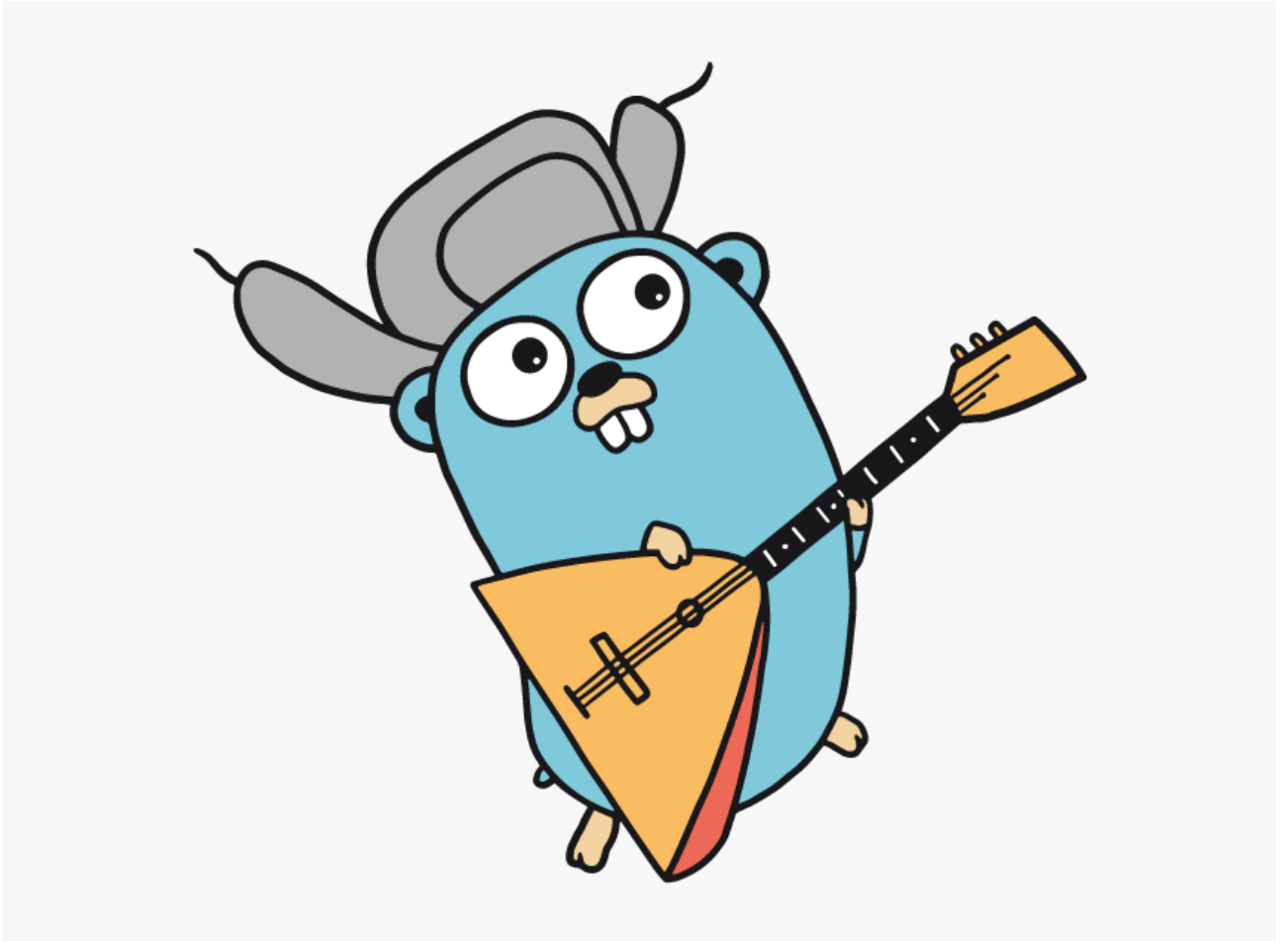


Рисунок 1 – Пример рисунка



```
11     if err != nil {
12         panic(err)
13     }
14     defer resp.Body.Close()
15
16     fmt.Println("Response status:", resp.Status)
17
18     scanner := bufio.NewScanner(resp.Body)
19     for i := 0; scanner.Scan() && i < 5; i++ {
20         fmt.Println(scanner.Text())
21     }
22
23     if err := scanner.Err(); err != nil {
24         panic(err)
25     }
26 }
```

## **2 Конструкторская часть**

### **3 Технологическая часть**

## **4 Исследовательская часть**

## **ЗАКЛЮЧЕНИЕ**

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Annual Report 2015: Strategic Report [Электронный ресурс]. – Режим доступа: [https://media.corporate-ir.net/media\\_files/IROL/19/197211/2016CustomWork/ARM\\_Strategic\\_Report.pdf](https://media.corporate-ir.net/media_files/IROL/19/197211/2016CustomWork/ARM_Strategic_Report.pdf), свободный – (24.11.2021) (преза арма как оформить??)
2. Bellard F. QEMU, a Fast and Portable Dynamic Translator [Текст] / Bellard F. // FREENIX Track: 2005 USENIX Annual Technical Conference. – 2005. – С. 41-42.

## **ПРИЛОЖЕНИЕ А**