**Отчет по ИДЗ №3**

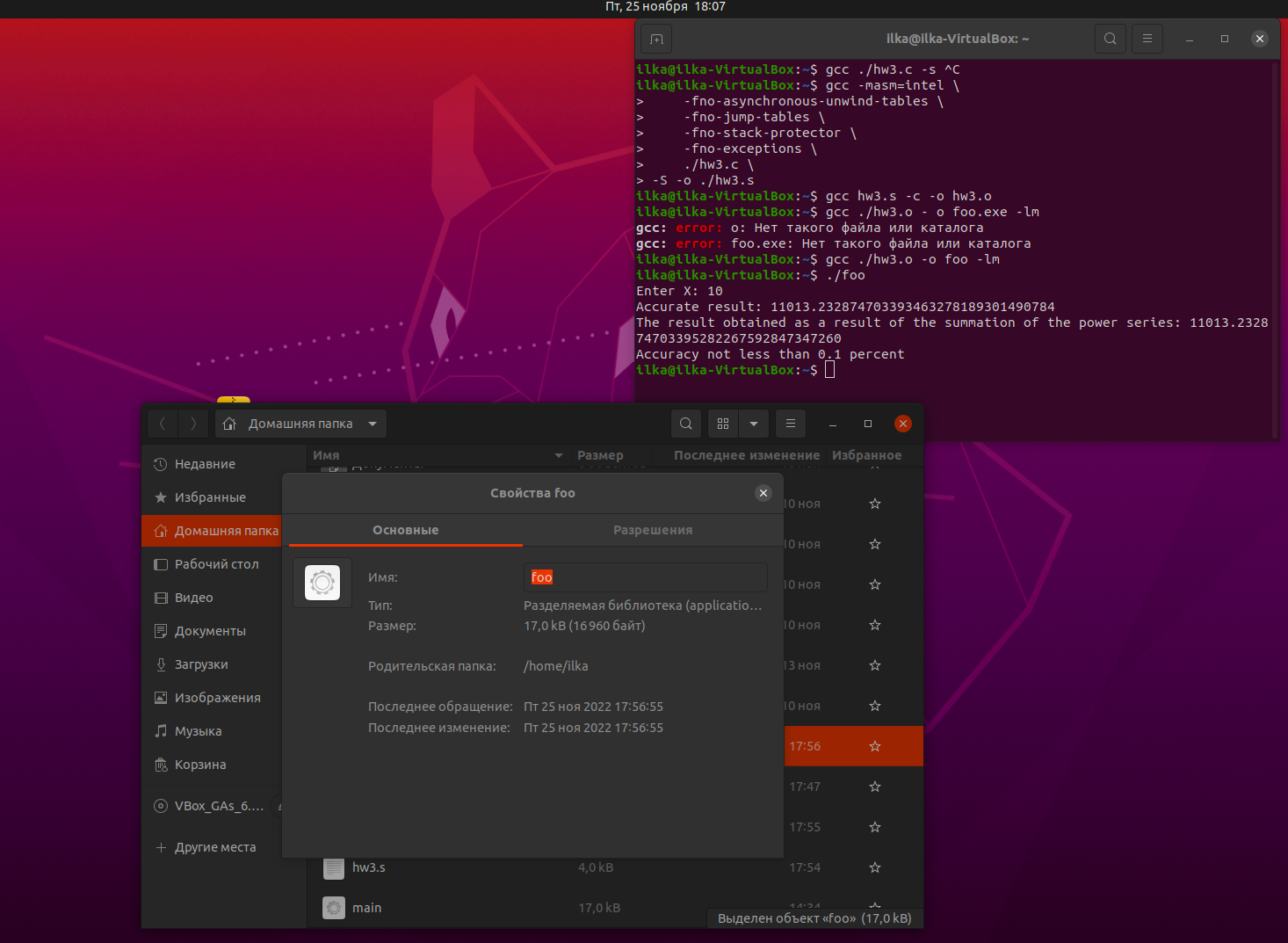
**Морина Ильи БПИ 215**

**Вариант 2**

**Часть 1 – создание программы.**

Первым делом написали код программы на C и разбили его на подпрограммы (функции).

Суть задачи заключается в создании программы, которая вычислять значение гиперболического синуса с помощью степенного ряда.  
Сассемблировали программу, создали объектный файл, создали “разделяемую библиотеку” и запустили программу. Видим, что все работает корректно, переходим к комментирование и редактированию ассемблерного кода.



Данные программы:

Первоначальный вес (без оптимизации)

hw3.c – ﻿1,1 kB (1 111 байт)

hw3o – ﻿ ﻿3,3 kB (3 336 байт)

hw3.s – ﻿4,0 kB (3 956 байт)

foo - ﻿17,0 kB (16 960 байт)

Вес с оптимизацией asm кода.

hw3.c – ﻿﻿ ﻿1,1 kB (1 111 байт)

hw3.o - ﻿3,3 kB (3 304 байт)

hw3.s – ﻿ ﻿ ﻿13,1 kB (13 143 байта)

(с комментариями)

Foo - ﻿17,0 kB (16 960 байт)

**Часть 2 – редактирование и оптимизация ассемблерного кода.**

**Редактирование**

1. Убираем первую строку “file “hw3.c””
2. Удаляем мусор созданный компилятором.

**Оптимизация**

1. Вместо -12[rbp] используем r13d

2. Превращение подобных пар

**mov eax, DWORD PTR [rdx+rax]**

**mov esi, eax**

в **mov esi, DWORD PTR [rdx+rax] (без промежуточных регистров)**

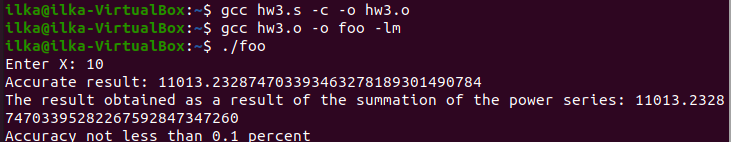
-4[rbp] в программе не нашлось, поэтому оптимизация с помощью r12d не получилась.

**Часть 3 – Пишем комментарии.**

Все комментарии находятся в файле с ассемблерным кодом (.S). Попытался привести подробное описаний (из сайтов) некоторых инструкций, те инструкции, которые встречались очень часто (mov, movsx, movzx, pop и тд) комментировал не везде, так как привел подробное их описание выше.

**Часть 4 – написание и прогон тестов.**

Сначала собираем программу с новым asm файлом.



Видим, что все работает, переходим к написанию тестов.

**Пишем тесты:**

**Тест 1** – Входное значение X 0.3

**Тест 2** – Входное значение X -63

**Тест 3** – Входное значение X -23.3

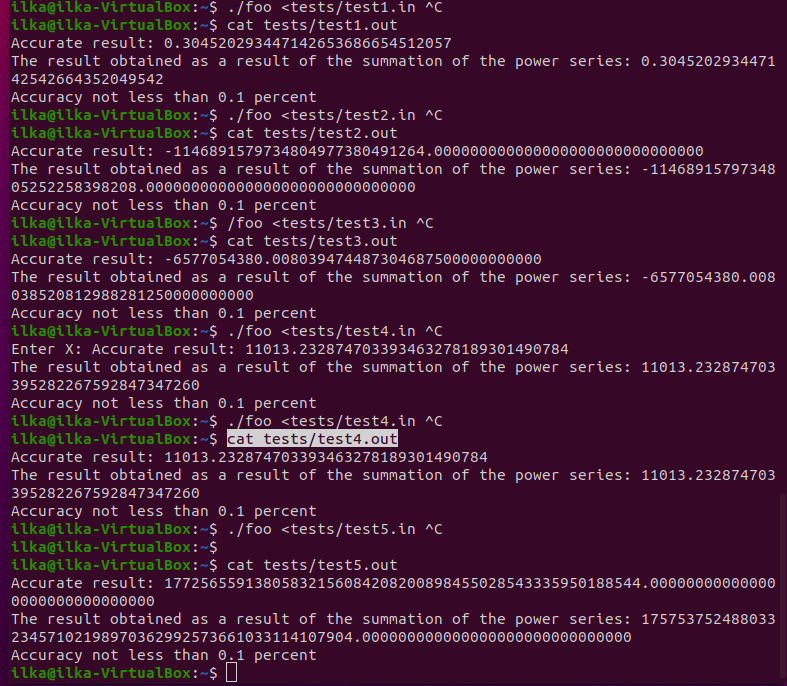
**Тест 4** – Входное значение X 10

**Тест 5** – Входное значение X 121



Сначала тестируем программу на C затем на ассемблерe, так как считаем программу на C корректной + она не изменяется в дальнейшем (в отличие от asm файла).

В файлы с правильными результатами тестов (.out) записываем результаты программы на C.



**Видим, что все тесты программа проходит корректно.**

**Часть 5 – подведение итогов и комментарии.**

**Критерии по которым была написана работа, соответствуют критериям на оценку 6.**

1. Программа содержит код на С и ассемблере.
2. Комментарии (в ассемблерном коде)
3. Рефакторинг (в ассемблерном коде)
4. Оптимизацию (использование регистров)
5. Тестовое покрытие.
6. Отчет.

**Комментарии к программе.**

1. В программе точность вывода до 30 знаков после запятой.
2. Необходимо вводить значения для X в интервале от -63 до 121.
3. Установлено путем тестов, что на значениях, не лежащих в интервале: Либо теряется точность, либо программа ничего не выводит.
4. Так как программа использует библиотеки, то пришлось использовать инструемент lm (файл foo – разделяемая библиотека), из-за чего возникает боязнь, что при проверке что-то пойдет не так, но надеюсь, что я все сделал правильно.

Заранее спасибо за проверку ИДЗ ☺

Прикладываю скрины кода:

