

Министерство высшего образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ  
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)  
Кафедра безопасности информационных систем (БИС)

ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ АССЕМБЛЕР

Отчет по лабораторной работе №2

по дисциплине «Системное программирование»

Студентка гр.737-1

\_\_\_\_\_ Агеева В.С.

\_\_\_\_.\_\_\_\_.2021г

Принял

Руководитель

доцент кафедры БИС

\_\_\_\_\_ Романов А.С.

\_\_\_\_.\_\_\_\_.2021г

Томск 2021

## 1 Введение

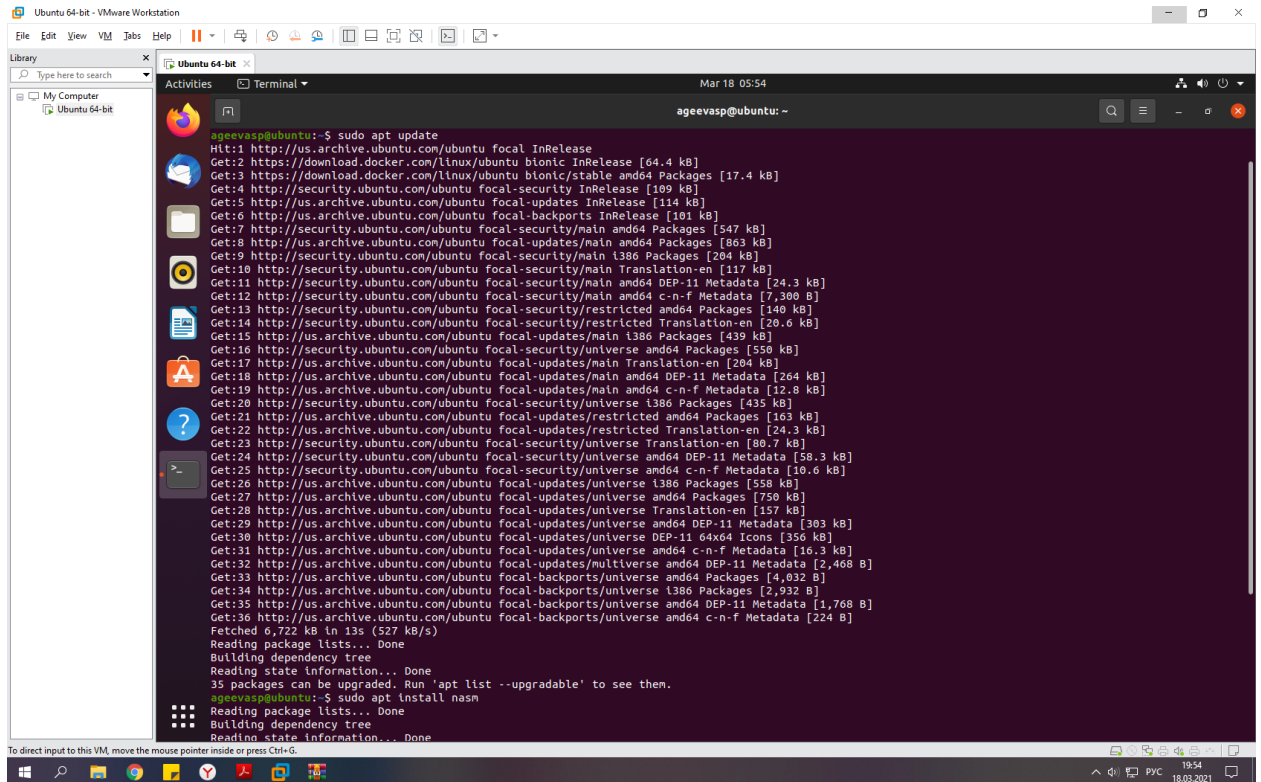
Цель работы: познакомиться со структурой программы на языке Ассемблер, разновидностями и назначением сегментов, способами организации простых и сложных типов данных, познакомиться со средствами создания программ на Ассемблере для ОС Linux.

Задание на лабораторную работу:

1. Задача: дан массив из 10 беззнаковых слов. Инвертировать биты старших байтов всех элементов массива. Найти сумму четных элементов полученного массива.

## 2 Ход работы

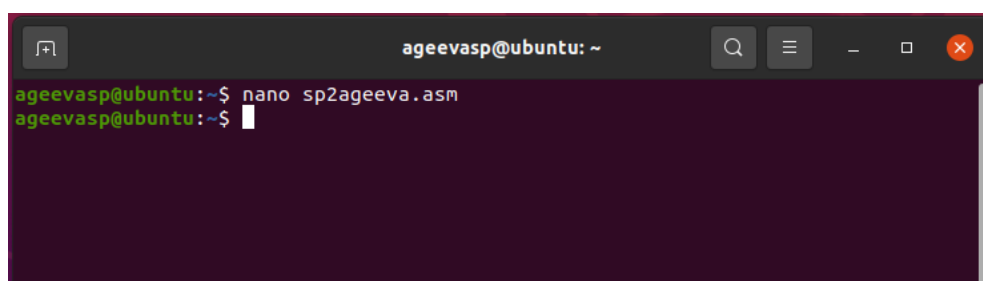
Для начала выполнения лабораторной работы необходимо сначала обновить ubuntu и установить расширенный ассемблер nasm (рисунок 2.1).



```
ageevasp@ubuntu:~$ sudo apt update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 https://download.docker.com/linux/ubuntu bionic InRelease [64.4 kB]
Get:3 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages [17.4 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [109 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [547 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [863 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/main i386 Packages [204 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [117 kB]
Get:11 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [24.3 kB]
Get:12 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [7,300 B]
Get:13 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [140 kB]
Get:14 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [20.6 kB]
Get:15 http://us.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [439 kB]
Get:16 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [550 kB]
Get:17 http://security.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [264 kB]
Get:18 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [264 kB]
Get:19 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [12.8 kB]
Get:20 http://security.ubuntu.com/ubuntu focal-security/universe i386 Packages [435 kB]
Get:21 http://us.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [163 kB]
Get:22 http://us.archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [24.3 kB]
Get:23 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [80.7 kB]
Get:24 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [58.3 kB]
Get:25 http://security.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [10.6 kB]
Get:26 http://us.archive.ubuntu.com/ubuntu focal-updates/universe i386 Packages [558 kB]
Get:27 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [750 kB]
Get:28 http://us.archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [157 kB]
Get:29 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [303 kB]
Get:30 http://us.archive.ubuntu.com/ubuntu focal-updates/universe DEP-11 64x64 Icons [350 kB]
Get:31 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [16.3 kB]
Get:32 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [2,468 B]
Get:33 http://us.archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [4,032 B]
Get:34 http://us.archive.ubuntu.com/ubuntu focal-backports/universe i386 Packages [2,932 B]
Get:35 http://us.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [1,768 B]
Get:36 http://us.archive.ubuntu.com/ubuntu focal-backports/universe amd64 c-n-f Metadata [224 B]
Fetched 6,722 kB in 13s (527 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
35 packages can be upgraded. Run 'apt list --upgradable' to see them.
ageevasp@ubuntu:~$ sudo apt install nasm
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Рисунок 2.1 – Обновление ubuntu и установка nasm

Далее через nano создадим файл, куда запишем код ассемблера (рисунок 2.2).



```
ageevasp@ubuntu:~$ nano sp2ageeva.asm
ageevasp@ubuntu:~$
```

Рисунок 2.2 - Создание файла

На рисунке 2.3 показан код ассемблера.

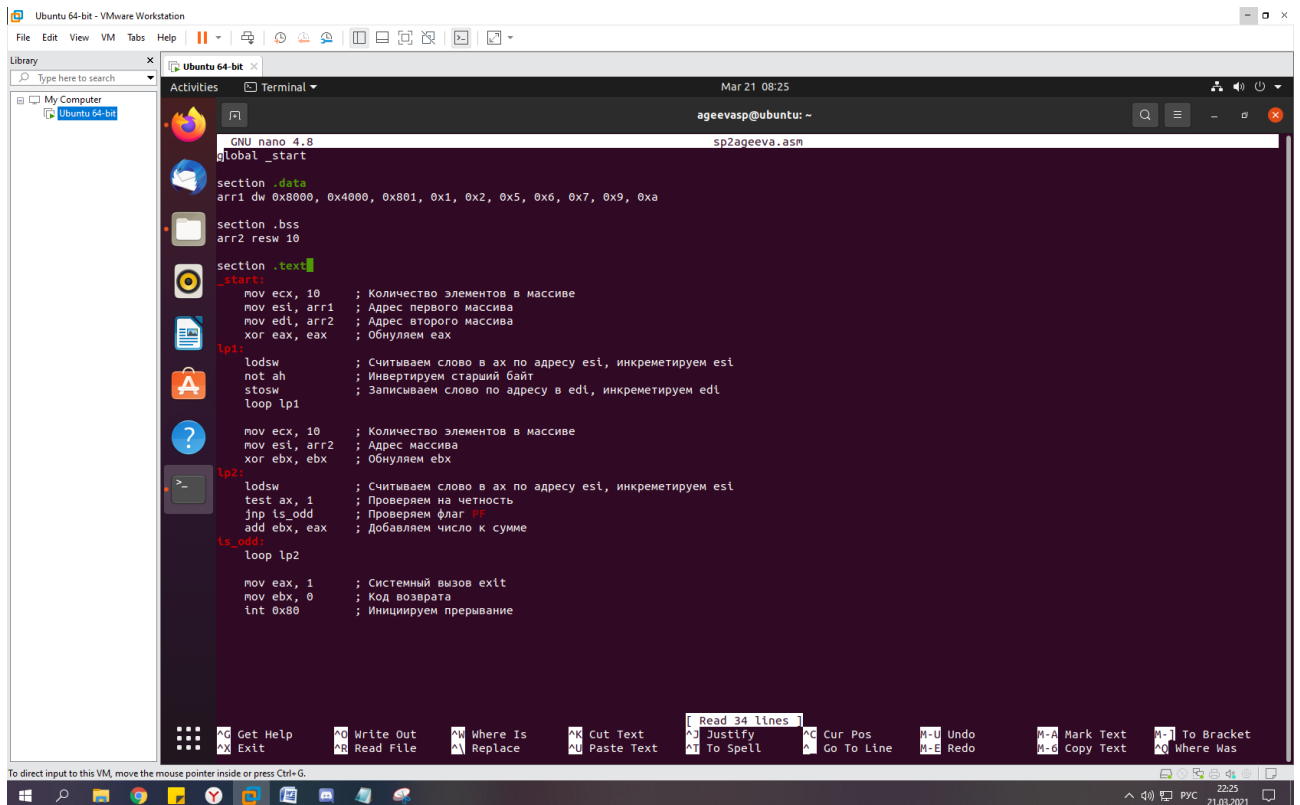


Рисунок 2.3 – Код по варианту на ассемблере

Далее командой `nasm -f elf sp2ageeva.asm` скомпилируем наш код, а командой `ls` проверим, чтобы появился объектный файл `sp2ageeva.o` (рисунок 2.4).

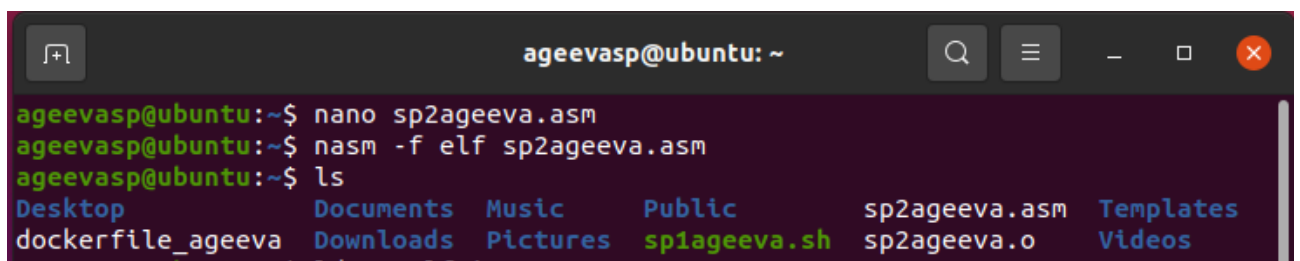
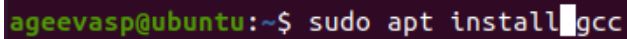


Рисунок 2.4 – Компилирование файла и создание объектного файла

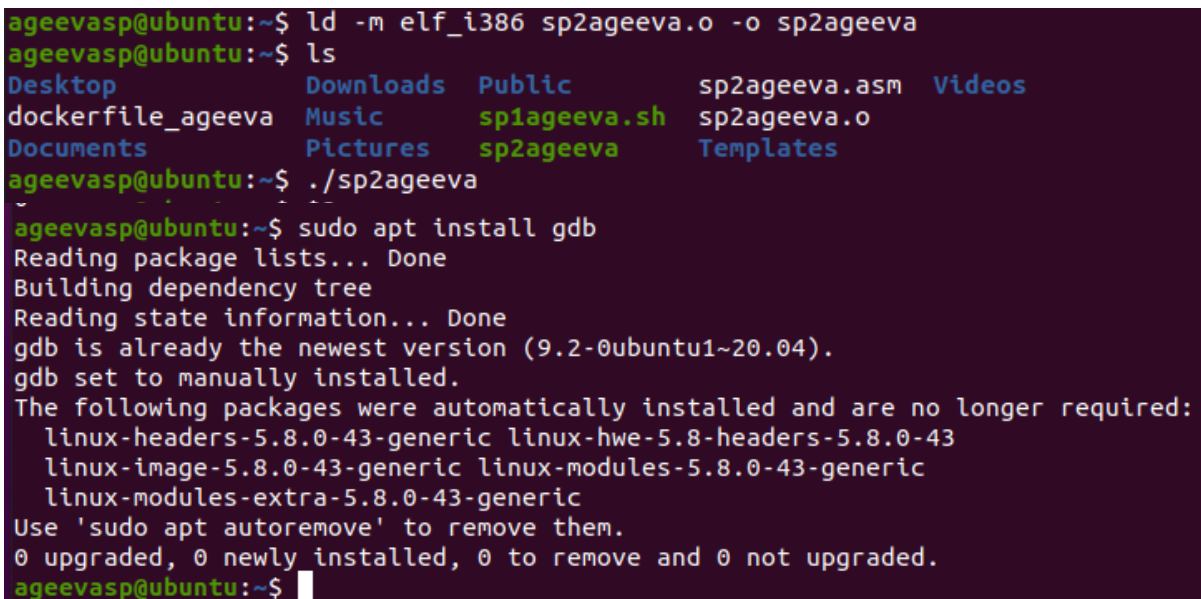
Далее скачаем и установим в системе компилятор `gcc` (рисунок 2.5).



```
ageevasp@ubuntu:~$ sudo apt install gcc
```

Рисунок 2.5 – Скачивание и установка компилятора gcc

Командой `ld -m elf_i386 sp2ageeva.o -o sp2ageeva` происходит связывание. Также для выполнения лабораторной работы нужно скачать и установить отладчик gdb (рисунок 2.6).



```
ageevasp@ubuntu:~$ ld -m elf_i386 sp2ageeva.o -o sp2ageeva
ageevasp@ubuntu:~$ ls
Desktop      Downloads   Public      sp2ageeva.asm  Videos
dockerfile_ageeva  Music      sp1ageeva.sh sp2ageeva.o
Documents    Pictures    sp2ageeva    Templates
ageevasp@ubuntu:~$ ./sp2ageeva
ageevasp@ubuntu:~$ sudo apt install gdb
Reading package lists... Done
Building dependency tree
Reading state information... Done
gdb is already the newest version (9.2-0ubuntu1~20.04).
gdb set to manually installed.
The following packages were automatically installed and are no longer required:
  linux-headers-5.8.0-43-generic linux-hwe-5.8-headers-5.8.0-43
  linux-image-5.8.0-43-generic linux-modules-5.8.0-43-generic
  linux-modules-extra-5.8.0-43-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ageevasp@ubuntu:~$
```

Рисунок 2.6 – Связывание объектного файла с исполняемым файлом и установка отладчика gdb

После того как скачали отладчик, запустим его и проверим работоспособность программы (рисунки 2.7 – 2.11).

```

ageevasp@ubuntu:~$ yasm -f elf sp2ageeva.asm -g dwarf2
ageevasp@ubuntu:~$ ld -m elf_i386 sp2ageeva.o -o sp2ageeva
ageevasp@ubuntu:~$ gdb sp2ageeva
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from sp2ageeva...
(gdb) b _start
Breakpoint 1 at 0x8049000: file sp2ageeva.asm, line 11.
(gdb) r
Starting program: /home/ageevasp/sp2ageeva

Breakpoint 1, _start () at sp2ageeva.asm:11
11      mov ecx, 10      ; Количество элементов в массиве
(gdb) layout regs

```

Рисунок 2.7 – Запуск компилятора

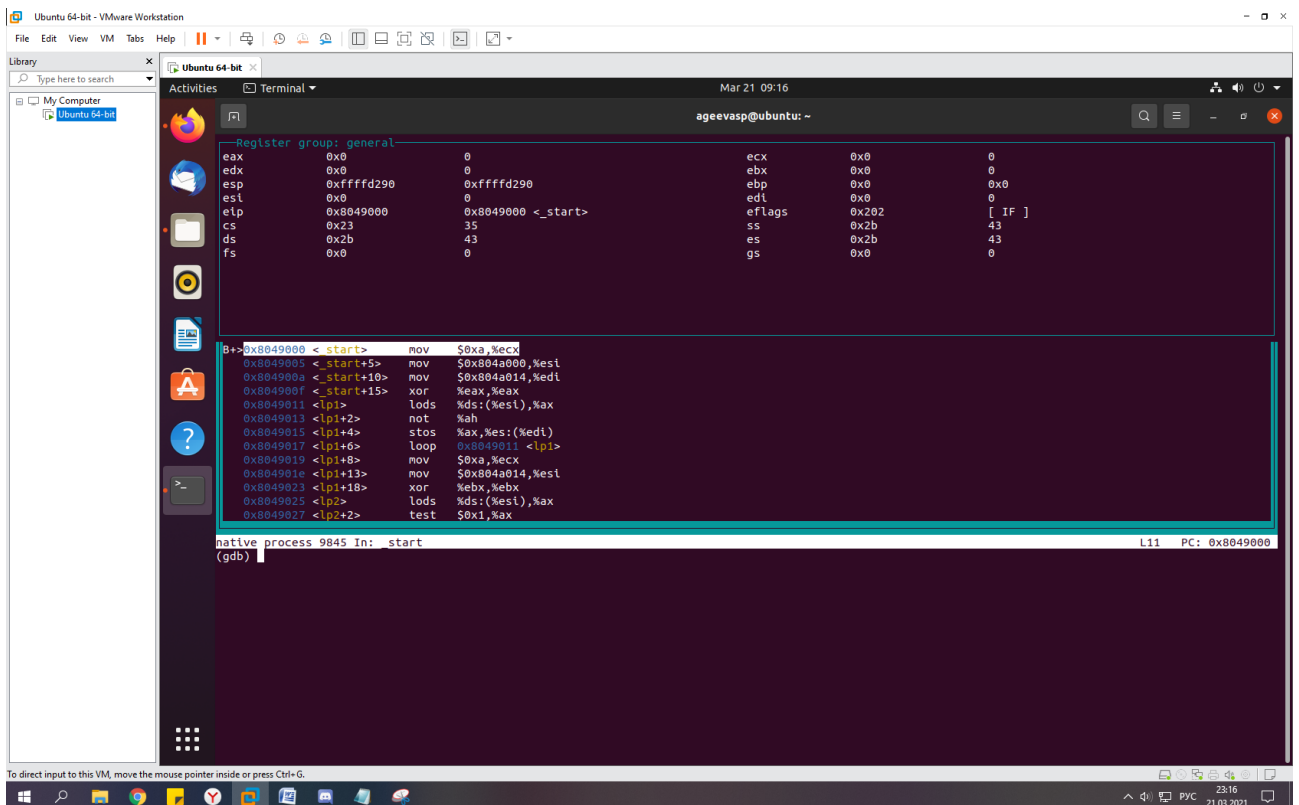


Рисунок 2.8 – Запущенный компилятор

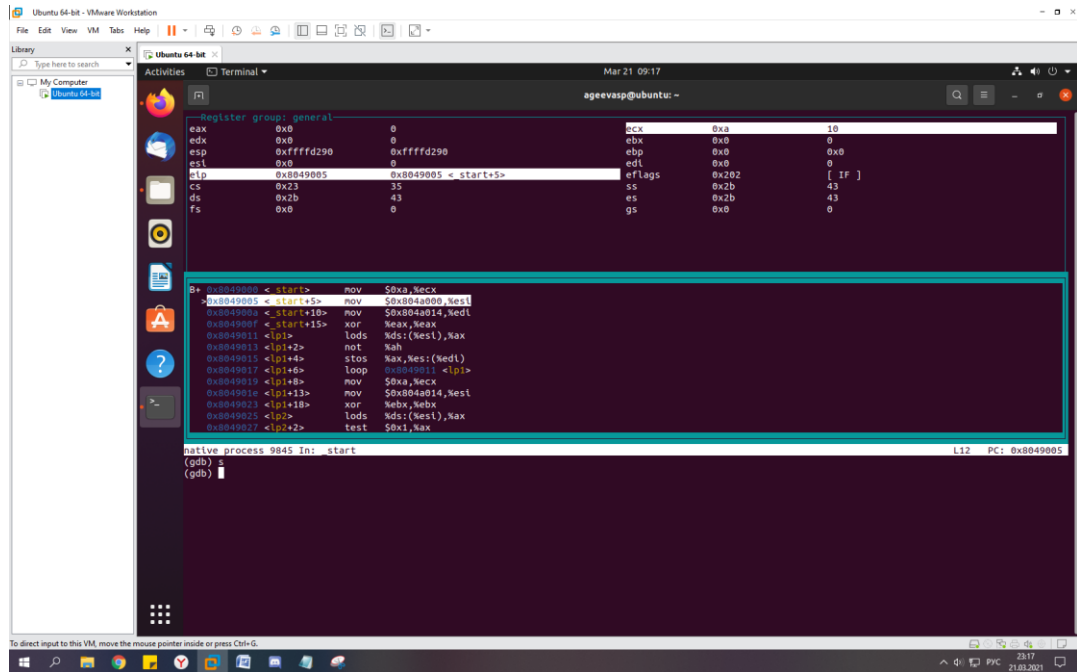


Рисунок 2.9 – Начала работы компилятора

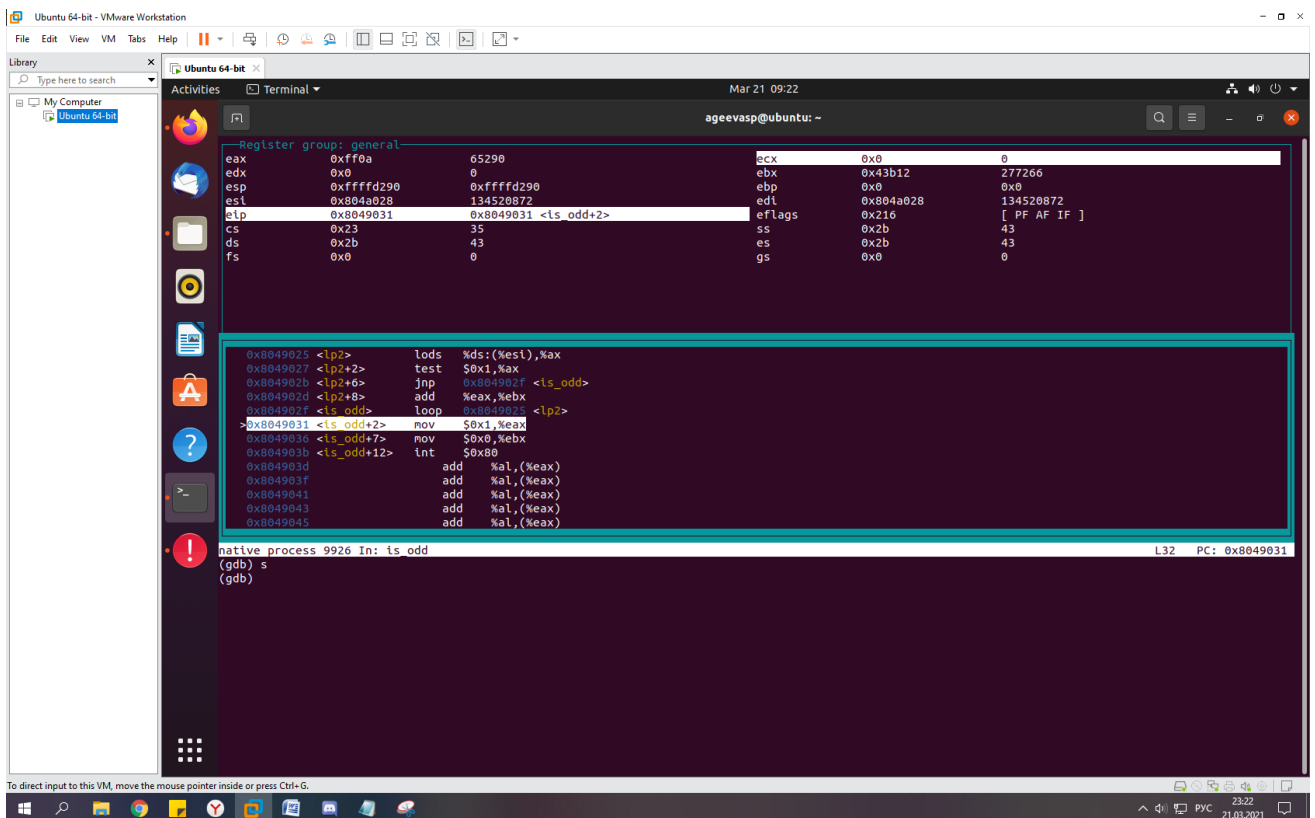


Рисунок 2.10 – Завершение работы компилятора с результатом суммы 43b12

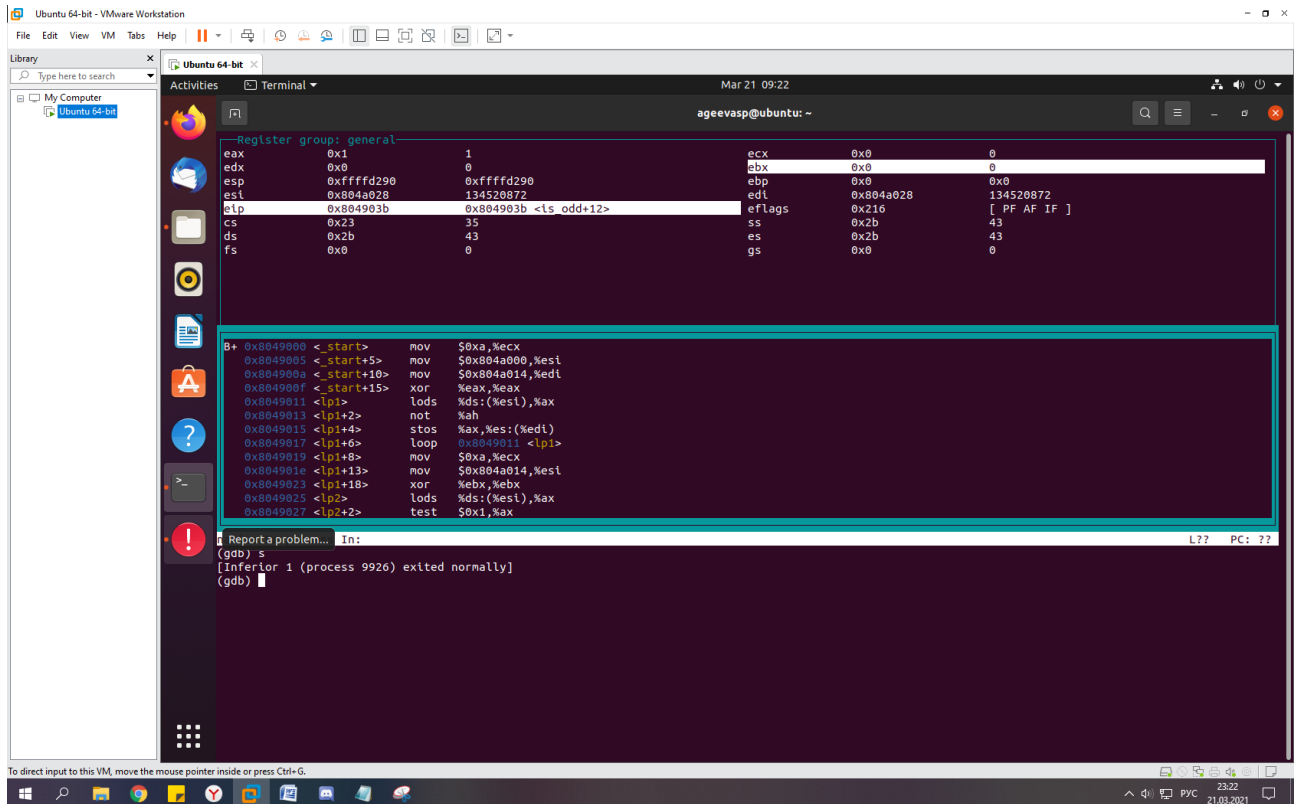


Рисунок 2.11 – Завершение полностью работы компилятора

После того, как мы проверили работоспособность программы через компилятор, нужно дизассемблировать код, для этого нужно ввести команду: `objdump -d -M i386 -M intel-mnemonic sp2ageeva` (рисунок 2.12).



```

ageevasp@ubuntu:~$ objdump -d -M i386 -M intel-mnemonic sp2ageeva
sp2ageeva:      file format elf32-i386

Disassembly of section .text:

08049000 <_start>:
8049000:      b9 0a 00 00 00      mov     ecx,0xa
8049005:      be 00 a0 04 08      mov     esi,0x804a000
804900a:      bf 14 a0 04 08      mov     edi,0x804a014
804900f:      31 c0               xor     eax,eax

08049011 <lp1>:
8049011:      66 ad               lods    ax,WORD PTR ds:[esi]
8049013:      f6 d4               not     ah
8049015:      66 ab               stos    WORD PTR es:[edi],ax
8049017:      e2 f8               loop    8049011 <lp1>
8049019:      b9 0a 00 00 00      mov     ecx,0xa
804901e:      be 14 a0 04 08      mov     esi,0x804a014
8049023:      31 db               xor     ebx,ebx

08049025 <lp2>:
8049025:      66 ad               lods    ax,WORD PTR ds:[esi]
8049027:      66 a9 01 00         test    ax,0x1
804902b:      7b 02               jnp     804902f <is_odd>
804902d:      01 c3               add     ebx,eax

0804902f <is_odd>:
804902f:      e2 f4               loop    8049025 <lp2>
8049031:      b8 01 00 00 00      mov     eax,0x1
8049036:      bb 00 00 00 00      mov     ebx,0x0
804903b:      cd 80               int     0x80
ageevasp@ubuntu:~$

```

Рисунок 2.12 – Дизассемблирование кода

Из рисунка видно следующую информацию:

- адреса;
- машинный код;
- мнемоника;
- операнды.

После того как сделали всю работу на ассемблере, необходимо написать программу на языке С, запустить через компилятор gcc и дизассемблировать. В итоге, сравнить какой код выполняется быстрее (рисунки 2.13 – 2.15).

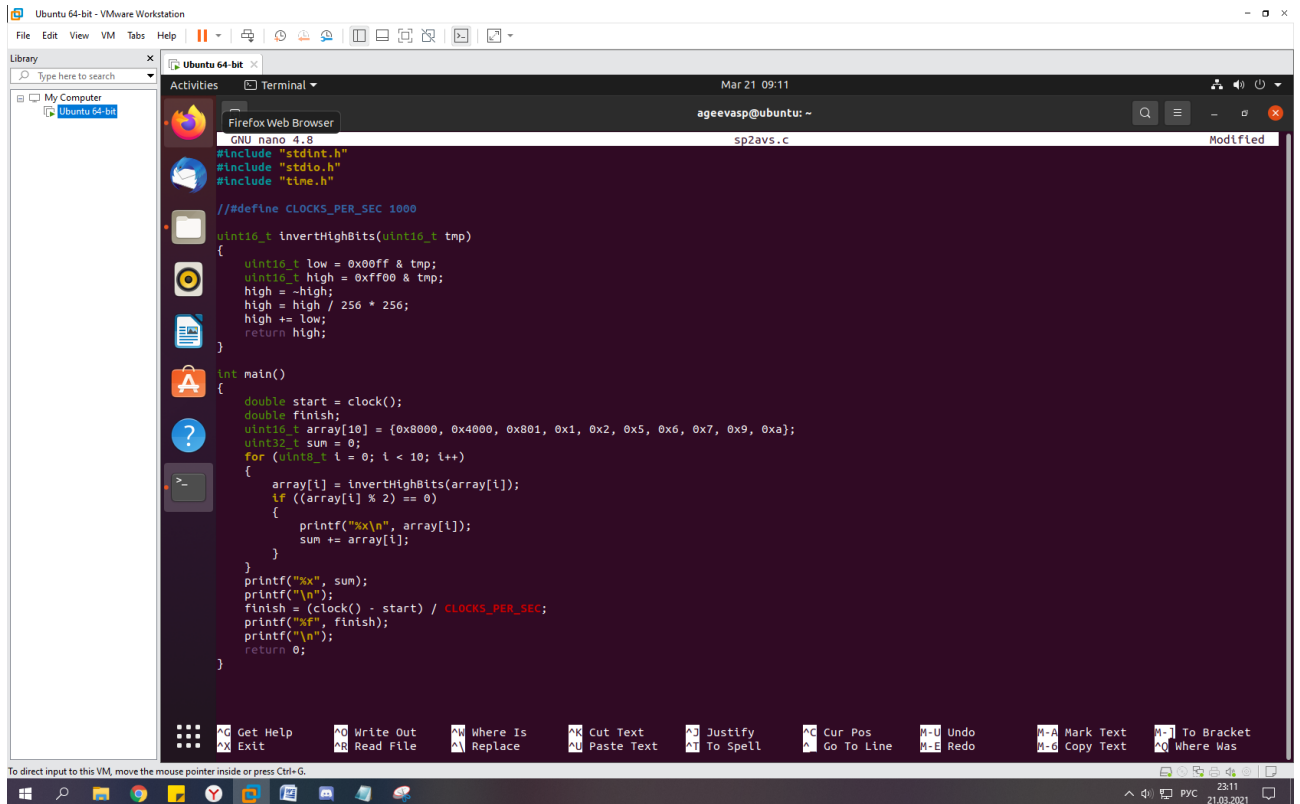


Рисунок 2.13 – Код на языке C

```

ageevasp@ubuntu:~$ gcc sp2avs.c
ageevasp@ubuntu:~$ ./a.out
7f00
bf00
ff02
ff06
ff0a
43b12
0.000058
ageevasp@ubuntu:~$

```

Рисунок 2.14 – Результат программы

```

ageevasp@ubuntu:~$ objdump -d -M i386 -M intel-mnemonic a.out
a.out:          file format elf64-x86-64

Disassembly of section .init:

0000000000000100 <_init>:
 1000:      f3 0f 1e fa          endbr64
 1004:      48                  dec    eax
 1005:      83 ec 08          sub    esp,0x8
 1008:      48                  dec    eax
 1009:      8b 05 d9 2f 00 00    mov    eax,DWORD PTR ds:0x2fd9
 100f:      48                  dec    eax
 1010:      85 c0          test   eax,eax
 1012:      74 02          je     1016 <_init+0x16>
 1014:      ff d0          call   eax
 1016:      48                  dec    eax
 1017:      83 c4 08          add    esp,0x8
 101a:      c3                  ret

Disassembly of section .plt:

00000000000001020 <.plt>:
 1020:      ff 35 82 2f 00 00    push   DWORD PTR ds:0x2f82
 1026:      f2 ff 25 83 2f 00 00    bnd jmp DWORD PTR ds:0x2f83
 102d:      0f 1f 00          nop    DWORD PTR [eax]
 1030:      f3 0f 1e fa          endbr64
 1034:      68 00 00 00 00 00    push   0x0
 1039:      f2 e9 e1 ff ff ff    bnd jmp 1020 <.plt>
 103f:      90                  nop
 1040:      f3 0f 1e fa          endbr64
 1044:      68 01 00 00 00 00    push   0x1
 1049:      f2 e9 d1 ff ff ff    bnd jmp 1020 <.plt>
 104f:      90                  nop
 1050:      f3 0f 1e fa          endbr64
 1054:      68 02 00 00 00 00    push   0x2
 1059:      f2 e9 c1 ff ff ff    bnd jmp 1020 <.plt>
 105f:      90                  nop
 1060:      f3 0f 1e fa          endbr64
 1064:      68 03 00 00 00 00    push   0x3
 1069:      f2 e9 b1 ff ff ff    bnd jmp 1020 <.plt>
 106f:      90                  nop

```

...

```

000000000000013d0 <__libc_csu_fini>:
 13d0:      f3 0f 1e fa          endbr64
 13d4:      c3                  ret

Disassembly of section .fini:

000000000000013d8 <_fini>:
 13d8:      f3 0f 1e fa          endbr64
 13dc:      48                  dec    eax
 13dd:      83 ec 08          sub    esp,0x8
 13e0:      48                  dec    eax
 13e1:      83 c4 08          add    esp,0x8
 13e4:      c3                  ret
ageevasp@ubuntu:~$

```

Рисунок 2.15 – Дизассемблированный код высокого языка программирования

### 3 Заключение

В ходе работы были написаны программы на языке высокого уровня и ассемблера, выполняющие одинаковый функционал.

В ходе анализа дизассемблированного кода было обнаружено, что полученный код высокоуровневой программы многократно (в шесть раз) превосходит по объему аналогичный код программы, написанной на ассемблере. Так же были получены навыки отладки программ с помощью отладчика gdb.

Ссылка на github: <https://github.com/7371avs/SP/tree/main>