

Лабораторная работа номер 4

Палымбетов Амирхан

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	11

Список иллюстраций

3.1	рис1	7
3.2	рис2	8
3.3	рис3	8
3.4	рис4	8
3.5	рис5	8
3.6	рис6	9
3.7	рис7	9
3.8	рис8	10

Список таблиц

1 Цель работы

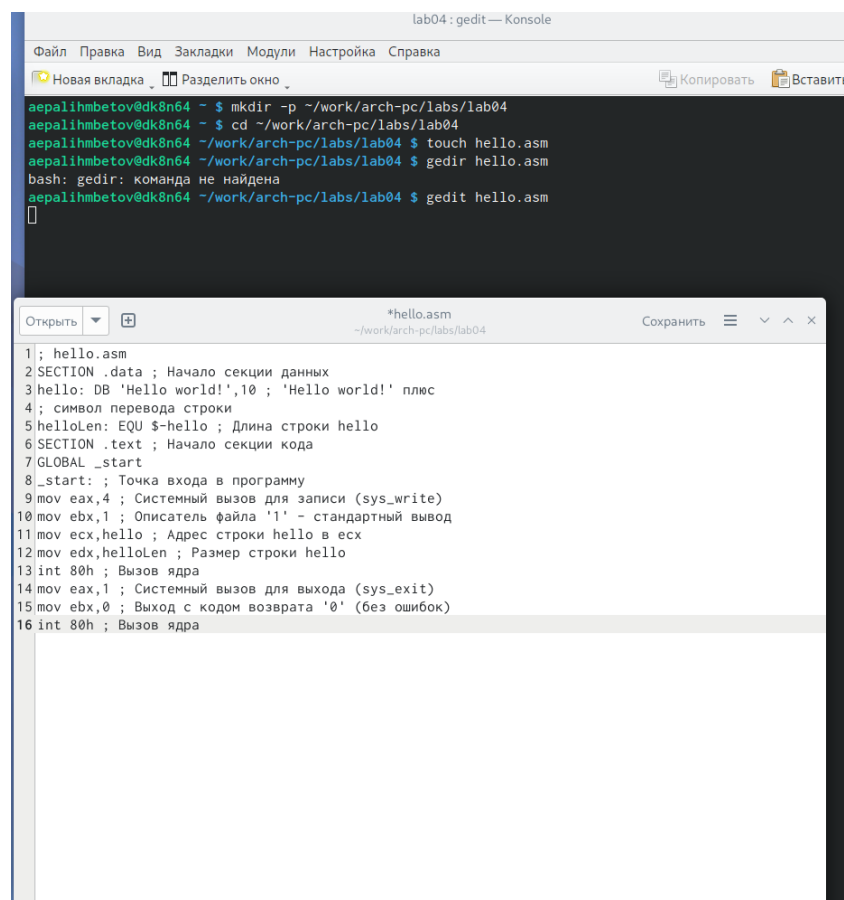
Цель данной лабораторной работы - освоить процедуры компиляции и сборки программ, написанных на ассемблере NASM

2 Задание

1.Создание программы Hello world! 2.Работа с транслятором NASM 3.Работы с расширенным синтаксисом командной строки NASM 4.Работа с компоновщиком LD 5.Запуск исполняемого файла 6.Выполнение самостоятельной работы

3 Выполнение лабораторной работы

1.Создание программы Hello world!: Создаем наш рабочий каталог, а также файл с написанным кодом на ассемблере.



The image shows a terminal window and a text editor window. The terminal window, titled 'lab04: gedit — Konsole', displays the following commands and output:

```
aepalihmbetov@dk8n64 ~$ mkdir -p ~/work/arch-pc/labs/lab04
aepalihmbetov@dk8n64 ~$ cd ~/work/arch-pc/labs/lab04
aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab04$ touch hello.asm
aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab04$ gedit hello.asm
bash: gedit: команда не найдена
aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab04$ gedit hello.asm
```

The text editor window, titled '*hello.asm' and showing the path '~/work/arch-pc/labs/lab04', contains the following assembly code:

```
1; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Hello world!',10 ; 'Hello world!' плюс
4 ; символ перевода строки
5 helloLen: EQU $-hello ; Длина строки hello
6 SECTION .text ; Начало секции кода
7 GLOBAL _start
8 _start: ; Точка входа в программу
9 mov eax,4 ; Системный вызов для записи (sys_write)
10 mov ebx,1 ; Описатель файла '1' - стандартный вывод
11 mov ecx,hello ; Адрес строки hello в ecx
12 mov edx,helloLen ; Размер строки hello
13 int 80h ; Вызов ядра
14 mov eax,1 ; Системный вызов для выхода (sys_exit)
15 mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
16 int 80h ; Вызов ядра
```

Рис. 3.1: рис1

2.Работа с транслятором NASM: При помощи кода превращаем текст файла в объектный код файл будет иметь название hello.o

```

aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab04 $ nasm -f elf hello.asm
aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab04 $ ls
hello.asm  hello.o
aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab04 $

```

Рис. 3.2: рис2

3.Работы с расширенным синтаксисом командной строки NASM

С помощью команды компилирую файл .asm в .o

```

aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab04 $ ls
hello.asm  hello.o  list.lst  obj.o

```

Рис. 3.3: рис3

4.Работа с компоновщиком LD: При помощи компоновщика получаем исполняемый файл а затем задаем ему имя.

```

aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab04 $ ld -m elf_i386 hello.o -o hello
aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab04 $ ls
hello  hello.asm  hello.o  list.lst  obj.o
aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab04 $ ld -m elf_i386 obj.o -o main

```

Рис. 3.4: рис4

!5.Запуск исполняемого файл Тут ничего сложного просто запускаем файл при помощи команды

```

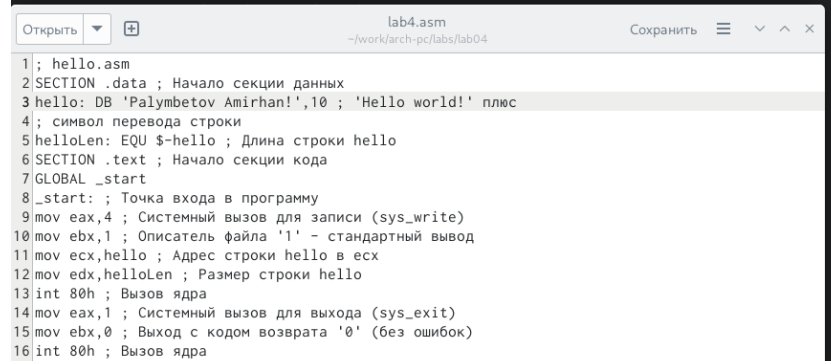
aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab04 $ ./hello
Hello world!

```

Рис. 3.5: рис5

6.Выполнение самостоятельной работы: Копируем файл и слегка изменяем содержание кода далее выполняем все тоже самое что и с предыдущем кодом затем отправляем все изменения на github


```
hello world!
aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab04 $ cp hello.asm lab4.asm
aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab04 $ ^C
aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab04 $ gedit hello.asm
```



```
1; hello.asm
2SECTION .data ; Начало секции данных
3hello: DB 'Palymbetov Amirhan!',10 ; 'Hello world!' плюс
4; символ перевода строки
5helloLen: EQU $-hello ; Длина строки hello
6SECTION .text ; Начало секции кода
7GLOBAL _start
8_start: ; Точка входа в программу
9mov eax,4 ; Системный вызов для записи (sys_write)
10mov ebx,1 ; Описатель файла '1' - стандартный вывод
11mov ecx,hello ; Адрес строки hello в ecx
12mov edx,helloLen ; Размер строки hello
13int 80h ; Вызов ядра
14mov eax,1 ; Системный вызов для выхода (sys_exit)
15mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
16int 80h ; Вызов ядра
```

Рис. 3.6: рис6

```
aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab04 $ nasm -f elf lab4.asm
aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab04 $ nasm -o obj.o -f elf -g -l list.lst lab4.asm
aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab04 $ ld -m elf_i386 lab4.o -o lab4
aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab04 $ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list.lst main obj.o
aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab04 $ ./lab4
Palymbetov Amirhan!
aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab04 $
```

Рис. 3.7: рис7

```

aepalihmbetov@dk8n64 ~/work/arch-pc/labs/lab04 $ cd ~/work/study/2023-2024/Архитектура компьютера/arch-pc
aepalihmbetov@dk8n64 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git add .
aepalihmbetov@dk8n64 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git commit -am 'add new files'
[master 0a546dc] add new files
1 changed, 4311 insertions(+)
Wiki
wiki.gentoo.org: 100644 labs/lab02/lab02_PalymbetovAmirhan.zip
rename labs/lab02/report/{report.docx => lab02_PalymbetovAmirhan.docx} (100%)
rename labs/lab02/report/{report.md => lab02_PalymbetovAmirhan.md} (100%)
rename labs/lab02/report/{report.pdf => lab02_PalymbetovAmirhan.pdf} (99%)
create mode 100644 labs/lab03/lab3_PalymbetovAmirhan.zip
create mode 100644 labs/lab03/report (копия)/Makefile
create mode 100644 labs/lab03/report (копия)/bib/cite.bib
create mode 100644 labs/lab03/report (копия)/image/work1.png
create mode 100644 labs/lab03/report (копия)/image/work2.png
create mode 100644 labs/lab03/report (копия)/image/work3.png
create mode 100644 labs/lab03/report (копия)/image/work4.png
create mode 100644 labs/lab03/report (копия)/image/work5.png
create mode 100644 labs/lab03/report (копия)/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 labs/lab03/report (копия)/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab03/report (копия)/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab03/report (копия)/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab03/report (копия)/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab03/report (копия)/pandoc/filters/pandocxnos/__init__.py
create mode 100644 labs/lab03/report (копия)/pandoc/filters/pandocxnos/core.py
create mode 100644 labs/lab03/report (копия)/pandoc/filters/pandocxnos/main.py
create mode 100644 labs/lab03/report (копия)/pandoc/filters/pandocxnos/pandocattributes.py
rename labs/lab03/{report => report (копия)}/report.docx (100%)
rename labs/lab03/{report => report (копия)}/report.md (100%)
rename labs/lab03/{report => report (копия)}/report.pdf (100%)
create mode 100644 labs/lab03/report/lab3_PalymbetovAmirhan.docx
create mode 100644 labs/lab03/report/lab3_PalymbetovAmirhan.md
create mode 100644 labs/lab03/report/lab3_PalymbetovAmirhan.pdf
aepalihmbetov@dk8n64 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git push
Перечисление объектов: 16, готово.
Подсчет объектов: 100% (16/16), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (10/10), готово.
Запись объектов: 100% (10/10), 10.26 Миб | 3.66 Миб/с, готово.
Всего 10 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 3 local objects.
To github.com:Sauronam/study_2023-2024_arch-pc.git
   9d4541b..0a546dc master -> master
aepalihmbetov@dk8n64 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $

```

Рис. 3.8: рис8

4 Выводы

При выполнении данной лабораторной работы я освоил процедуры компиляции и сборки программ, написанных на ассемблере NASM.