

## Задание №2

### в рамках вычислительного практикума

#### Этапы получения исполняемого файла

Исходный код я взял из 5 задания первой лабораторной работы

Листинг кода №1.

```
#include <stdio.h>

#define OK 0
#define ERR_IO 1
#define ERR_RANGE 2

/*
    Функция, которая принимает делимое (n) и делитель (d). Число n
    делится на d до тех пор,
    пока остаток от деления равен нулю. Выводит в консоль
    делители числа n. Число n изменяется по указателю
*/
void division(int *n, int d)
{
    int new_n = *n;
    while (new_n % d == 0)
    {
        new_n /= d;
        printf("%d ", d);
    }
    *n = new_n;
}

/*
    Функция solve() проверяет деление числа n на делитель div.
    Сначала проверяется деление на 2, затем
    перебираются нечетные числа, которые меньше n. Если деление
    происходит без остатка, то управление передается
    функции division(), которая выводит делители в консоль.
*/
void solve(int n)
{
    int div = 3;
```

```

    if (n % 2 == 0)
        division(&n, 2);

    while (n > 1)
    {
        if (n % div == 0)
            division(&n, div);
        div += 2;
    }
}

int main(void)
{
    int n;
    printf("Введите натуральное число N: ");

    if (scanf("%d", &n) != 1)
    {
        printf("Некорректный ввод");
        return ERR_IO;
    }
    else if (n < 1)
    {
        printf("Некорректный диапазон");
        return ERR_RANGE;
    }
    else
        solve(n);
    return OK;
}

```

## 1. компилятор gcc

### 1.1 Версия компилятора:

```
gcc version 11.4.0 (Ubuntu 11.4.0-1ubuntu1~22.04)
```

### 1.2 Команда для получения исполняемого файла

```
gcc -std=c99 -Wall -Werror -v -save-temps hello.c
```

### 1.3 Список файлов, полученных в результате компиляции:

Имя	Размер, байты	Тип
hello.c	1714	Исходный код C

a-hello.i	14602	Файл с информацией о компиляции
a-hello.s	3234	Файл ассемблера
a-hello.o	2496	Объектный файл
a.out	16128	Исполняемый файл

#### 1.4 Обработка препроцессором:

```
/usr/lib/gcc/x86_64-linux-gnu/11/cc1 -E -quiet -v -imultiarch x86_64-linux-gnu hello.c -mtune=generic -march=x86-64 -std=c99 -Wall -Werror -fpch-preprocess -fasynchronous-unwind-tables -fstack-protector-strong -Wformat-security -fstack-clash-protection -fcf-protection -o a-hello.i
```

Вход	hello.c
Выход	ignoring nonexistent directory "/usr/local/include/x86_64-linux-gnu" ignoring nonexistent directory "/usr/lib/gcc/x86_64-linux-gnu/11/include-fixed" ignoring nonexistent directory "/usr/lib/gcc/x86_64-linux-gnu/11/../../../../x86_64-linux-gnu/include" #include "... " search starts here: #include <...> search starts here: /usr/lib/gcc/x86_64-linux-gnu/11/include /usr/local/include /usr/include/x86_64-linux-gnu /usr/include End of search list. End of search list.
Назначение	2. Удаление комментариев 3. Вставка файлов 4. Текстовые замены <b>5. Условная компиляция</b>
Имя утилиты	cc1
Способ передачи данных	По командной строке

## 1.5 Трансляция на язык ассемблера

```
/usr/lib/gcc/x86_64-linux-gnu/11/cc1 -fpreprocessed a-hello.i -quiet -  
dumpdir a- -dumpbase hello.c -dumpbase-ext .c -mtune=generic -  
march=x86-64 -Wall -Werror -std=c99 -version -fasynchronous-  
unwind-tables -fstack-protector-strong -Wformat-security -fstack-  
clash-protection -fcf-protection -o a-hello.s
```

Вход	a-hello.i
Выход	GNU C99 (Ubuntu 11.4.0-1ubuntu1~22.04) version 11.4.0 (x86_64-linux-gnu) compiled by GNU C version 11.4.0, GMP version 6.2.1, MPFR version 4.1.0, MPC version 1.2.1, isl version isl-0.24-GMP GGC heuristics: --param ggc-min-expand=100 --param ggc-min-heapsize=131072 GNU C99 (Ubuntu 11.4.0-1ubuntu1~22.04) version 11.4.0 (x86_64-linux-gnu) compiled by GNU C version 11.4.0, GMP version 6.2.1, MPFR version 4.1.0, MPC version 1.2.1, isl version isl-0.24-GMP GGC heuristics: --param ggc-min-expand=100 --param ggc-min-heapsize=131072 Compiler executable checksum: 50eaa2331df977b8016186198deb2d18 COLLECT_GCC_OPTIONS='-std=c99' '-Wall' '-Werror' '-v' '-save-temps' '-mtune=generic' '-march=x86-64' '-dumpdir' 'a-' as -v --64 -o a-hello.o a-hello.s
Назначение	<ol style="list-style-type: none"><li>1. Упростить реализацию и отладки транслятора</li><li>2. Повысить переносимость с одной платформы на другую</li></ol>
Имя утилиты	cc1
Способ передачи данных	По командной строке

## 1.6 Ассемблирование в объектный файл:

Команда для перевода программы с языка ассемблера в машинный код:

```
as hello.c -o hello.o
```

Вход	main.s
Выход	GNU ассемблер, версия 2.38 (x86_64-linux-gnu); используется BFD версии (GNU Binutils for Ubuntu) 2.38
Назначение	Перевод программы с языка ассемблера в машинный код
Имя утилиты	as
Способ передачи данных	По командной строке

## 1.7 Компоновка

```
/usr/lib/gcc/x86_64-linux-gnu/11/collect2 -plugin /usr/lib/gcc/x86_64-linux-gnu/11/liblto_plugin.so -plugin-opt=/usr/lib/gcc/x86_64-linux-gnu/11/lto-wrapper -plugin-opt=-fresolution=a.res -plugin-opt=-pass-through=lgcc -plugin-opt=-pass-through=lgcc_s -plugin-opt=-pass-through=lc -plugin-opt=-pass-through=lgcc -plugin-opt=-pass-through=lgcc_s --build-id --eh-frame-hdr -m elf_x86_64 --hash-style=gnu --as-needed -dynamic-linker /lib64/ld-linux-x86-64.so.2 -pie -z now -z relro /usr/lib/gcc/x86_64-linux-gnu/11/../../../../x86_64-linux-gnu/Scrt1.o /usr/lib/gcc/x86_64-linux-gnu/11/../../../../x86_64-linux-gnu/crti.o /usr/lib/gcc/x86_64-linux-gnu/11/crtbeginS.o -L/usr/lib/gcc/x86_64-linux-gnu/11 -L/usr/lib/gcc/x86_64-linux-gnu/11/../../../../x86_64-linux-gnu -L/usr/lib/gcc/x86_64-linux-gnu/11/../../../../lib -L/lib/x86_64-linux-gnu -L/lib/./lib -L/usr/lib/x86_64-linux-gnu -L/usr/lib/./lib -L/usr/lib/gcc/x86_64-linux-gnu/11/../../../../a-hello.o -lgcc --push-state --as-needed -lgcc_s --pop-state -lc -lgcc --push-state --as-needed -lgcc_s --pop-state /usr/lib/gcc/x86_64-linux-gnu/11/crtendS.o /usr/lib/gcc/x86_64-linux-gnu/11/../../../../x86_64-linux-gnu/crtn.o COLLECT_GCC_OPTIONS='-std=c99' '-Wall' '-Werror' '-v' '-save-temps' '-mtune=generic' '-march=x86-64' '-dumpdir' 'a.'
```

Вход	hello.o
Выход	a.out
Назначение	<ol style="list-style-type: none"> <li>1. Объединение нескольких объектных файлов в единый исполняемый файл</li> <li>2. Связывание переменных и функций</li> <li>3. Добавление специального кода, который подготавливает окружение</li> </ol>

	для вызова функции main, а после ее завершения выполняет обратные действия
Имя утилиты	collect2
Способ передачи данных	По командной строке

Список объектных файлов, с которыми компонуется программа:

Имя	Назначение
Scrt1.o	Используется вместо crt1.o при создании PIE. PIE - независимый от позиции исполняемый файл (-fPIE - pie)
Crti.o	Определяет прологи функций для разделов .init и .fini. Таким образом их можно вызывать напрямую. Также они предназначены для поддержки старой системы конструкторов и деструкторов, в которой все разделы .init/.fini объединены во время компоновки.
crtbeginS.o	Используется вместо crtbegin.o при создании общих объектов/PIE.
crtendS.o	GCC использует это для поиска начала деструкторов.
Crtn.o	Определяет эпилоги функций для разделов .init/.fini

Список библиотек, с которыми компонуется программа:

Имя	Назначение
libc	стандартная библиотека языка Си, содержащая основные функции и макросы, необходимые для работы программы.
libgcc_s	статическая версия библиотеки GCC, предоставляющая поддержку для работы программ, скомпилированных с GCC.

### 3. Компилятор Clang

#### 2.1 Версия компилятора:

```
Ubuntu clang version 14.0.0-1ubuntu1.1
```

#### 2.2 Команда для получения исполняемого файла:

```
clang -std=c99 -Wall -Werror -v -save-temps main.c -lm
```

#### 2.3 Список файлов, полученных в результате компиляции:

Имя	Размер, байты	Тип
hello.c	1714	Исходный код C
hello.i	14779	Файл с информацией о компиляции
hello.bc	3344	Двоичный файл
hello.s	4183	Файл ассемблера
hello.o	2288	Объектный файл
a.out	16128	Исполняемый файл

#### 2.4 Обработка препроцессором:

```
/usr/lib/llvm-14/bin/clang" -cc1 -triple x86_64-pc-linux-gnu -E -save-temps=cwd -disable-free -clear-ast-before-backend -disable-llvm-verifier -discard-value-names -main-file-name hello.c -mrelocation-model pic -pic-level 2 -pic-is-pie -mframe-pointer=all -fmath-errno -ffp-contract=on -fno-rounding-math -mconstructor-aliases -funwind-tables=2 -target-cpu x86-64 -tune-cpu generic -debugger-tuning=gdb -v -fcoverage-compilation-dir=/home/whxqh/lab_01_05_04 -resource-dir /usr/lib/llvm-14/lib/clang/14.0.0 -internal-isystem /usr/lib/llvm-14/lib/clang/14.0.0/include -internal-isystem /usr/local/include -internal-isystem /usr/bin/../lib/gcc/x86_64-linux-gnu/11/../../../../x86_64-linux-gnu/include -internal-externc-isystem /usr/include/x86_64-linux-gnu -internal-externc-isystem /include -internal-externc-isystem /usr/include -Wall -Werror -std=c99 -fdebug-compilation-dir=/home/whxqh/lab_01_05_04 -ferror-limit 19 -fgnuc-version=4.2.1 -fcolor-diagnostics -faddrsig -D_GCC_HAVE_DWARF2_CFI_ASM=1 -o hello.i -x c hello.c
```

Вход	hello.c
Выход	clang -cc1 version 14.0.0 based upon LLVM 14.0.0 default target x86_64-pc-linux-gnu ignoring

	nonexistent directory "/usr/bin/./lib/gcc/x86_64-linux-gnu/11/././././x86_64-linux-gnu/include" ignoring nonexistent directory "/include" #include "..." search starts here: #include <...> search starts here: /usr/lib/llvm-14/lib/clang/14.0.0/include /usr/local/include /usr/include/x86_64-linux-gnu /usr/include End of search list.
Назначение	<ol style="list-style-type: none"> <li>1. Удаление комментариев</li> <li>2. Вставка файлов</li> <li>3. Текстовые замены</li> <li>4. Условная компиляция</li> </ol>
Имя утилиты	clang
Способ передачи данных	По командной строке

## 2.5 Получение битового кода:

```
/usr/lib/llvm-14/bin/clang" -cc1 -triple x86_64-pc-linux-gnu -emit-llvm-bc -emit-llvm-uselists -save-temps=cwd -disable-free -clear-ast-before-backend -disable-llvm-verifier -discard-value-names -main-file-name hello.c -mrelocation-model pic -pic-level 2 -pic-is-pie -mframe-pointer=all -fmath-errno -ffp-contract=on -fno-rounding-math -mconstructor-aliases -funwind-tables=2 -target-cpu x86-64 -tune-cpu generic -mllvm -treat-scalable-fixed-error-as-warning -debugger-tuning=gdb -v -fcoverage-compilation-dir=/home/whxeqh/lab_01_05_04 -resource-dir /usr/lib/llvm-14/lib/clang/14.0.0 -Wall -Werror -std=c99 -fdebug-compilation-dir=/home/whxeqh/lab_01_05_04 -ferror-limit 19 -fgnuc-version=4.2.1 -fcolor-diagnostics -disable-llvm-passes -faddrsig -D __GCC_HAVE_DWARF2_CFI_ASM=1 -o hello.bc -x cpp-output hello.i
```

Вход	hello.i
Выход	clang -cc1 version 14.0.0 based upon LLVM 14.0.0 default target x86_64-pc-linux-gnu #include "..." search starts here: End of search list.
Назначение	Clang компилирует исходный код из файла hello.i в LLVM IR битовый код (hello.bc). Для



	последующей оптимизации и трансляции
Имя утилиты	clang
Способ передачи данных	По командной строке

## 2.6 Трансляция на язык ассемблера:

```
/usr/lib/llvm-14/bin/clang" -cc1 -triple x86_64-pc-linux-gnu -S -save-temps=cwd -disable-free -clear-ast-before-backend -disable-llvm-verifier -discard-value-names -main-file-name hello.c -mrelocation-model pic -pic-level 2 -pic-is-pie -mframe-pointer=all -fmath-errno -ffp-contract=on -fno-rounding-math -mconstructor-aliases -funwind-tables=2 -target-cpu x86-64 -tune-cpu generic -mllvm -treat-scalable-fixed-error-as-warning -debugger-tuning=gdb -v -fcoverage-compilation-dir=/home/whxeqh/lab_01_05_04 -resource-dir /usr/lib/llvm-14/lib/clang/14.0.0 -Wall -Werror -td=c99 -fdebug-compilation-dir=/home/whxeqh/lab_01_05_04 -ferror-limit 19 -fgnuc-version=4.2.1 -fcolor-diagnostics -faddrsig -D_GCC_HAVE_DWARF2_CFI_ASM=1 -o hello.s -x ir hello.bc
```

Вход	hello.bc
Выход	clang -cc1 version 14.0.0 based upon LLVM 14.0.0 default target x86_64-pc-linux-gnu
Назначение	Clang выполняет трансляцию LLVM IR битового кода в код на языке ассемблера для последующей сборки в машинный код.
Имя утилиты	clang
Способ передачи данных	По командной строке

## 2.7 Ассемблирование в объектный файл:

```
/usr/lib/llvm-14/bin/clang" -cc1as -triple x86_64-pc-linux-gnu -filetype obj -main-file-name hello.c -target-cpu x86-64 -fdebug-compilation-dir=/home/whxeqh/lab_01_05_04 -dwarf-version=5 -mrelocation-model pic -mrelax-all --mrelax-relocations -o hello.o hello.s
```

Вход	hello.s
Выход	hello.o
Назначение	Clang выполняет ассемблирование файла на языке ассемблера в объектный файл для дальнейшей компоновки в исполняемый код.
Имя утилиты	clang
Способ передачи данных	По командной строке

## 2.8 Компоновка:

```
"/usr/bin/ld" -pie -z relro --hash-style=gnu --build-id --eh-frame-hdr -m elf_x86_64 -dynamic-linker /lib64/ld-linux-x86-64.so.2 -o a.out /lib/x86_64-linux-gnu/Scrt1.o /lib/x86_64-linux-gnu/crti.o /usr/bin/../lib/gcc/x86_64-linux-gnu/11/crtbeginS.o -L/usr/bin/../lib/gcc/x86_64-linux-gnu/11 -L/usr/bin/../lib/gcc/x86_64-linux-gnu/11/../../../../lib64 -L/lib/x86_64-linux-gnu -L/lib/../lib64 -L/usr/lib/x86_64-linux-gnu -L/usr/lib/../lib64 -L/usr/lib/llvm-14/bin/../lib -L/lib -L/usr/lib hello.o -lm -lgcc --as-needed -lgcc_s --no-as-needed -lc -lgcc --as-needed -lgcc_s --no-as-needed /usr/bin/../lib/gcc/x86_64-linux-gnu/11/crtendS.o /lib/x86_64-linux-gnu/crtn.o
```

Вход	hello.o
Выход	a.out
Назначение	Выполняет компоновку объектного файла hello.o в исполняемый файл a.out. Компоновщик (ld) добавляет к объектному файлу необходимые библиотеки и файлы для создания исполняемого кода.
Имя утилиты	ld
Способ передачи данных	По командной строке

Список объектных файлов, с которыми компонуется программа:

Имя	Назначение
Scrt1.o	Используется вместо crt1.o при создании PIE. PIE - независимый от позиции исполняемый файл (-fPIE - pie)

Crti.o	Определяет прологи функций для разделов .init и .fini. Таким образом их можно вызывать напрямую. Также они предназначены для поддержки старой системы конструкторов и деструкторов, в которой все разделы .init/.fini объединены во время компоновки.
crtbeginS.o	Используется вместо crtbegin.o при создании общих объектов/PIE.
crtendS.o	GCC использует это для поиска начала деструкторов.
Crtn.o	Определяет эпилоги функций для разделов .init/.fini

Список библиотек, с которыми компоуется программа:

Имя	Назначение
libc	стандартная библиотека языка Си, содержащая основные функции и макросы, необходимые для работы программы.
libgcc_s	статическая версия библиотеки GCC, предоставляющая поддержку для работы программ, скомпилированных с GCC.
libgcc	библиотека с реализацией некоторых операций, необходимых для работы с языком С.
lm	библиотека математических функций.

### 3. Дополнительные задания:

#### 3.1

```
$ ldd a.out
linux-vdso.so.1 (0x00007fff41b9f000)
libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x000071b043b5b000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x000071b043800000)
/lib64/ld-linux-x86-64.so.2 (0x000071b043c58000)
```

1. `linux-vdso.so.1`: Виртуальная динамическая библиотека ядра Linux, обеспечивающая доступ к системным вызовам.
2. `libm.so.6`: Математическая библиотека для выполнения операций, таких как тригонометрия и логарифмы.
3. `libc.so.6`: Стандартная библиотека языка Си, предоставляющая основные функции, включая работу с потоками и вводом/выводом.
4. `/lib64/ld-linux-x86-64.so.2`: Динамический загрузчик для программ на архитектуре `x86_64` под Linux, отвечает за загрузку и разрешение символов.

### 3.2

С помощью ключа `-S`. Он указывает сгенерировать только ассемблерный код

```
gcc -std=c99 -Wall -Werror -S main.i
```

### 3.3

```
gcc -o app.exe main.o -Wl, Map=map_info.map -lm
```

`-Wl, Map=map_info.map` - опция, которая передает параметр `Map=map_info.map` компоновщику, чтобы создать `map` файл, содержащий информацию о символах и секциях программы

### 3.4

```
gcc -std=C99 -Wall -Werror -g0 hello.c -o hello_g0
gcc -std=c99 -Wall -Werror -g1 hello.c -o hello_g1
gcc -std=c99 -Wall -Werror -g2 hello.c -o hello_g2
gcc -std=c99 -Wall -Werror -g3 hello.c -o hello_g3

ls -l

-rwxrwxr-x 1 whxeqh whxeqh 16152 map 16 19:09 a.out
-rwxrwxrwx 1 whxeqh whxeqh 181 map 9 15:06
build_debug_asan.sh
-rwxrwxrwx 1 whxeqh whxeqh 196 map 9 15:12
build_debug_msan.sh
-rwxrwxrwx 1 whxeqh whxeqh 136 map 9 16:10 build_debug.sh
-rwxrwxrwx 1 whxeqh whxeqh 183 map 9 15:07
build_debug_ubsan.sh
```

```
-rwxrwxrwx 1 whxeh whxeh 128 мар 9 16:09 build_release.sh
-rwxrwxrwx 1 whxeh whxeh 28 мар 9 15:17 clean.sh
drwxrwxr-x 4 whxeh whxeh 4096 мар 9 15:21 func_tests
-rw-rw-r-- 1 whxeh whxeh 3344 мар 16 19:09 hello.bc
-rw-rw-r-- 1 whxeh whxeh 1714 мар 9 12:03 hello.c
-rwxrwxr-x 1 whxeh whxeh 16128 мар 16 20:15 hello_g0
-rwxrwxr-x 1 whxeh whxeh 17296 мар 16 20:16 hello_g1
-rwxrwxr-x 1 whxeh whxeh 17728 мар 16 20:16 hello_g2
-rwxrwxr-x 1 whxeh whxeh 44008 мар 16 20:16 hello_g3
-rw-rw-r-- 1 whxeh whxeh 14779 мар 16 19:09 hello.i
-rw-rw-r-- 1 whxeh whxeh 2288 мар 16 19:09 hello.o
-rw-rw-r-- 1 whxeh whxeh 4183 мар 16 19:09 hello.s
```

Имя	Размер, байт
hello_go	16128
hello_g1	17296
hello_g2	17728
hello_g3	44008