

1 Компилятор GCC

1.1 Версия компилятора

Команда:

```
gcc --version
```

Вывод:

```
gcc (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0
```

1.2 Команда для получения исполняемого файла:

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

Список файлов, полученных в результате компиляции:

Имя	Размер, байты	Тип
main.c	109	Исходный код
a-main.i	14978	Препроцессированный C код
a-main.o	1512	Объектный файл
a-main.s	705	Ассемблированный файл
a.out	15960	Исполняемый файл

1.3 Этапы компиляции

1.3.1 Препроцессинг

Команда:

```
cpre main.c -o main.i
```

Вход	main.c
Выход	main.i
Назначение	Обработка препроцессором
Имя утилиты	cpre
Способ передачи параметров	Флаговое

1.3.2 Компиляция

Команда:

```
c99 -S -fverbose-asm -masm=intel main.i
```

Вход	main.i
Выход	main.s
Назначение	Трансляция на язык ассемблера
Имя утилиты	c99
Способ передачи параметров	Флаговое

1.3.3 Ассемблирование

Команда:

```
as main.s -o main.o
```

Вход	main.s
Выход	main.o
Назначение	Ассемблирование в объектный файл
Имя утилиты	as
Способ передачи параметров	Флаговое

1.3.4 Компоновка

Команда:

```
ld \  
-dynamic-linker /lib64/ld-linux-x86-64.so.2 \  
-o output \  
/usr/lib/x86_64-linux-gnu/crt1.o \  
/usr/lib/x86_64-linux-gnu/crti.o \  
-L/usr/lib/x86_64-linux-gnu \  

```

```
-lc \
main.o \
/usr/lib/x86_64-linux-gnu/crti.o -o main.exe
```

Вход	main.o
Выход	main.exe
Назначение	Получение исполняемого файла
Имя утилиты	ld
Способ передачи параметров	Флаговое

Список объектных файлов, с которыми компоуется программа

Имя	Назначение
crt1.o	Содержит точку входа <code>_start</code> , которая инициализирует выполнение программы через вызов <code>__libc_start_main</code> .
crti.o	Определяет начало секций <code>.init</code> (инициализация глобальных объектов) и <code>.fini</code> (финализация).
crti.o	Завершает секции <code>.init</code> и <code>.fini</code> , добавляя код завершения.
libc.so (-lc)	Стандартная библиотека C.
main.o	Скомпилированный объектный код программы.

Список библиотек, с которыми компоуется программа

Имя	Назначение
libc.so	Стандартная библиотека C: функции ввода-вывода (<code>printf</code> , <code>scanf</code>), управление памятью (<code>malloc</code> , <code>free</code>), строковые операции и т.д.
/lib64/ld-linux-x86-64.so.2	Загрузчик динамических библиотек

2 Компилятор clang

2.1 Версия компилятора

Команда:

```
clang --version
```

Вывод:

```
Ubuntu clang version 18.1.3 (1ubuntu1)
```

```
Target: x86_64-pc-linux-gnu
```

```
Thread model: posix
```

```
InstalledDir: /usr/bin
```

2.2. Команда для получения исполняемого файла

Команда:

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

Список файлов, полученных в результате компиляции

Имя файла	Размер, байты	Тип
main.c	109	Исходный код
main.i	15633	Препроцессированный C код
main.s	869	Ассемблированный файл
main.o	1272	Объектный файл
main.bc	2608	Биткод (LLVM IR)
a.out	16008	Исполняемый файл

2.3 Этапы компиляции

2.3.1 Препроцессинг

Команда:

```
clang -E main.c -o main.i
```

Вход:	main.c
Выход:	main.i
Назначение:	Обработка директив препроцессора.
Имя утилиты:	clang (с флагом -E)
Способ передачи параметров:	Флаговое.

2.3.2 Компиляция

Команда:

```
clang -S -fverbose-asm -masm=intel main.i
```

Вход:	main.i
Выход:	main.s
Назначение:	Трансляция в ассемблерный код.
Имя утилиты:	clang (с флагом -S)
Способ передачи параметров:	Флаговое.

2.3.3 Ассемблирование

Команда:

```
as main.s -o main.o
```

Вход:	main.s
Выход:	main.o
Назначение:	Преобразование ассемблерного кода в объектный файл.
Имя утилиты:	as
Способ передачи параметров:	Флаговое.

2.3.4 Компоновка

Команда:

```
clang main.o -o main.exe
```

Вход:	main.o
Выход:	main.exe
Назначение:	Создание исполняемого файла.
Имя утилиты:	clang
Способ передачи параметров:	Флаговое.

Список объектных файлов

Имя	Назначение
crt1.o	Содержит точку входа <code>_start</code> , вызывает <code>__libc_start_main</code> для подготовки <code>main</code> .
crti.o	Начало секций <code>.init</code> (инициализация глобальных объектов) и <code>.fini</code> (финализация).
crtn.o	Завершает секции <code>.init</code> и <code>.fini</code> .
main.o	Скомпилированный объектный код программы.
libc.s	Стандартная библиотека C.

Список библиотек

Имя	Назначение
libc.so	Стандартная библиотека C.
/lib64/ld-linux-x86-64.so.2	Динамический загрузчик.
libgcc / libunwind	Вспомогательные функции LLVM.

3 Дополнительные задания

3.1 Какие динамические библиотеки использует Ваш исполняемый файл?

Команда:

```
ldd a.out
```

Вывод:

```
linux-vdso.so.1 (0x00007fff88571000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f47fc039000)
/lib64/ld-linux-x86-64.so.2 (0x00007f47fc25b000)
```

3.2 Найдите, с помощью какого ключа компилятора gcc можно передавать параметры компилятору с языка ассемблера.

Ответ: флаг `-fverbose-asm`.

Листинг программы:

```
.file "main.c"
.intel_syntax noprefix
# GNU C99 (Ubuntu 13.3.0-6ubuntu2~24.04) version 13.3.0
(x86_64-linux-gnu)
# compiled by GNU C version 13.3.0, GMP version 6.3.0,
MPFR version 4.2.1, MPC version 1.3.1, isl version isl-
0.26-GMP

# GGC heuristics: --param ggc-min-expand=100 --param ggc-
min-heapsize=131072
# options passed: -masm=intel -mtune=generic -march=x86-64
-std=c99 -fasynchronous-unwind-tables -fstack-protector-
strong -fstack-clash-protection -fcf-protection
.text
.section .rodata
.LC0:
```

```

        .string "Year has %d weeks.\n"
        .text
        .globl main
        .type   main, @function
main:
.LFB0:
        .cfi_startproc
        endbr64
        push    rbp #
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        mov rbp, rsp    #,
        .cfi_def_cfa_register 6
# main.c:7:      printf("Year has %d weeks.\n", WEEKS);
        mov esi, 52 #,
        lea rax, .LC0[rip] # tmp84,
        mov rdi, rax    #, tmp84
        mov eax, 0    #,
        call    printf@PLT #
# main.c:8:      return 0;
        mov eax, 0    # _3,
# main.c:9: }
        pop rbp #
        .cfi_def_cfa 7, 8
        ret
        .cfi_endproc
.LFE0:
        .size   main, .-main

```



```

    .ident "GCC: (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0"
    .section .note.GNU-stack,"",@progbits
    .section .note.gnu.property,"a"
    .align 8
    .long 1f - 0f
    .long 4f - 1f
    .long 5
0:
    .string "GNU"
1:
    .align 8
    .long 0xc0000002
    .long 3f - 2f
2:
    .long 0x3
3:
    .align 8
4:

```

3.3 Найдите, с помощью какого ключа компилятора gcc можно передавать параметры компоновщику.

Команда:

```
gcc -Wl,-Map=output.map main.c
```

3.4 Заполните таблицу

	Размер исполняемого файла
g0	15960
g1	16928

g2	17144
g3	43856