

Задание №2
Этапы получения исполняемого файла

Оглавление

Программа, которую мы используем в дальнейшем.....	2
Компилятор gcc.....	3
Компилятор clang.....	6
Дополнительные задания.....	9
Вывод.....	9

Программа, которую мы используем в дальнейшем

```
// Доколин Георгий ИУ7-22Б
#include <stdio.h>

// Создание макросов
#define a 5
#define b 7

int main()
{
    // Выводим сумму чисел
    printf("%d", a + b);
    return 0;
}
```

Компилятор gcc

Версия компилятора

gcc (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0

Команда для получения исполняемого файла

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

Ключ **-v** используется для отображения программ, вызванных компилятором
Ключ **-save-temps** используется для того, чтобы не удалять промежуточные файлы

Список файлов, полученных в результате компиляции

Имя	Размеры, байты	Тип
main.c	224	Текстовый
a-main.i	13964	Текстовый
a-main.s	687	Текстовый
a-main.o	1496	Объектный
a.out	15960	Исполняемый

Этап обработки препроцессором

Команда для выполнения этапа	/usr/lib/gcc/x86_64-linux-gnu/11/cc1 -E -quiet -v -imultiarch x86_64-linux-gnu main.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-main.i
Вход	main.c
Выход	a-main.i
Назначение	Получение единицы трансляции
Имя утилиты	cc1
Способ передачи параметров	Параметры командной строки

Трансляция на язык ассемблера

Команда для выполнения этапа	/usr/lib/gcc/x86_64-linux-gnu/11/cc1 -fpreprocessed a-main.i -quiet -dumpbase a-main.i -dumpbase-ext .i -mtune=generic -march=x86-64 -Wall -Werror -std=c99 -version -o a-main.s -fasynchronous-unwind-tables -fstack-protector-strong -Wformat-security -fstack-clash-protection -fcf-protection
Вход	a-main.i
Выход	a-main.s

Назначение	Перевод единицы трансляции на язык Ассемблера
Имя утилиты	cc1
Способ передачи параметров	Параметры командной строки

Ассемблирование

Команда для выполнения этапа	as -v --64 -o a-main.o a-main.s
Вход	a-main.s
Выход	a-main.o
Назначение	Перевод программы с языка ассемблера в машинный код.
Имя утилиты	as
Способ передачи параметров	Параметры командной строки

Компоновка

Команда для выполнения этапа	<pre> /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=/tmp/ccBQWKSd.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 -o a.out /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-main.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 </pre>
Вход	a-main.o
Выход	a.out
Назначение	Получение исполняемого файла из объектного.
Имя утилиты	collect2
Способ передачи параметров	Параметры командной строки

Список объектных файлов, с которыми компонуется программа

Имя	Назначение
Scrt1.o	Файл проводит инициализацию перед вызовом функции main.
crti.o	Файл проводит инициализацию окружения, устанавливая различные значения переменных окружения.
crtbeginS.o	Файл хранит в себе код для запуска функции main.
a-main.o	Файл содержит код нашей программы.
crtendS.o	Файл для завершения функции main.
crtn.o	Файл, который служит для завершения работы компоновщика и удаления всего что использовалось при его работе.

Список библиотек, с которыми компонуется программа

Имя	Назначение
-lgcc	Libgcc используется для работы со строками, арифметическими операциями, для работы которых процессору нужно преобразование в иной вид.
-lgcc_s	Libgcc_s используется для того чтобы включить все зависимости в исполняемый файл
-lc	GNU C Library используется для работы с функциями ввода, вывода, выделения памяти, обработки файлов и так далее.

Компилятор clang

Версия компилятора

Ubuntu clang version 14.0.0-1ubuntu1.1

Команда для получения исполняемого файла

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

Список файлов, полученных в результате компиляции

Имя	Размеры, байты	Тип
main.c	224	Текстовый
main.i	14143	Текстовый
main.s	850	Текстовый
main.o	1256	Объектный
main.bc	2360	Бинарный
a.out	16032	Исполняемый

Препроцессирование

Команда для выполнения этапа	<code>"/usr/lib/llvm-14/bin/clang" -cc1 -triple x86_64-pc-linux-gnu -E -disable-free -clear-ast-before-backend -disable-llvm-verifier -discard-value-names -main-file-name main.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/george/Desktop/PTP/LABA_2 -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 -fdebug-compilation-dir=/home/george/Desktop/PTP/LABA_2 -ferror-limit 19 -fgnuc-version=4.2.1 -faddrsig -D__GCC_HAVE_DWARF2_CFI_ASM=1 -o main.i -x c main.c</code>
Вход	main.c
Выход	main.i
Назначение	Получение единицы трансляции
Имя утилиты	cc1
Способ передачи параметров	Параметры командной строки

Перевод на язык ассемблера

Команда для выполнения этапа	"/usr/lib/llvm-14/bin/clang" -cc1 -triple x86_64-pc-linux-gnu -S -disable-free -clear-ast-before-backend -disable-llvm-verifier -discard-value-names -main-file-name main.i -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/george/Desktop/PTP/LABA_2 -resource-dir /usr/lib/llvm-14/lib/clang/14.0.0 -Wall -Werror -std=c99 -fdebug-compilation-dir=/home/george/Desktop/PTP/LABA_2 -ferror-limit 19 -fgnuc-version=4.2.1 -faddrsig -D__GCC_HAVE_DWARF2_CFI_ASM=1 -o main.s -x cpp-output main.i
Вход	main.i
Выход	main.s
Назначение	Перевод программы на язык ассемблера
Имя утилиты	cc1
Способ передачи параметров	Параметры командной строки

Ассемблирование

Команда для выполнения этапа	"/usr/lib/llvm-14/bin/clang" -cc1as -triple x86_64-pc-linux-gnu -filetype obj -main-file-name main.s -target-cpu x86-64 -fdebug-compilation-dir=/home/george/Desktop/PTP/LABA_2 -dwarf-debug-producer "Ubuntu clang version 14.0.0-1ubuntu1.1" -dwarf-version=5 -mrelocation-model pic --mrelax-relocations -o main.o main.s
Вход	main.s
Выход	main.o
Назначение	Перевод кода программы в машинный код
Имя утилиты	cc1as
Способ передачи параметров	Параметры командной строки

Компановка

Команда для выполнения этапа	<code>"/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 main.o -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</code>
Вход	main.o
Выход	a.out
Назначение	Получение исполняемого файла
Имя утилиты	ld
Способ передачи параметров	Параметры командной строки

Список объектных файлов, с которыми компоуется программа

Имя	Назначение
Scrt1.o	Файл проводит инициализацию перед вызовом функции main.
crti.o	Файл проводит инициализацию окружения, устанавливая различные значения переменных окружения.
crtbeginS.o	Файл хранит в себе код для запуска функции main.
a-main.o	Файл содержащий код нашей программы.
crtendS.o	Файл для завершения функции main.
crtn.o	Файл, который служит для завершения работы компоновщика и удаления всего что использовалось при его работе.

Список библиотек, с которыми компоуется программа

Имя	Назначение
-lgcc	Libgcc используется для работы со строками, арифметическими операциями, для работы которых процессору нужно преобразование в иной вид.
-lgcc_s	Libgcc_s используется для того чтобы включить все зависимости в исполняемый файл
-lc	GNU C Library используется для работы с функциями ввода, вывода, выделения памяти, обработки файлов и так далее.

Дополнительные задания

1. Какие динамические библиотеки использует Ваш исполняемый файл?

```
linux-vdso.so.1 (0x00007ffc931bd000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007d7f44400000)
/lib64/ld-linux-x86-64.so.2 (0x00007d7f44761000)
```

2. Найдите, с помощью какого ключа компилятора gcc можно передавать параметры компилятору с языка ассемблера. Используйте этот параметр для получения ассемблерного листинга Вашей программы.

Ассемблерный листинг можно получить при помощи ключа -
Xassembler.

3. Найдите, с помощью какого ключа компилятора gcc можно передавать параметры компоновщику. Используйте этот параметр для получения так называемого map-файла для Вашей программы.

При помощи ключа **-Wl** можно передавать параметры компоновщику.
Команда для получения map файла:

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

4. Размер файла от уровня отладочной информации

	Размер исполняемого файла
g0	16032
g1	16880
g2	16944
g3	16944

Вывод

При выполнении лабораторной работы, мной были получены навыки работы с компиляторами **clang** и **gcc**, так же я подробно познакомился с процессом создания исполняемого файла.