

Отладка

Отладочный вывод

Основные полезные моменты

1. Все отладочные выводы следует делать в поток stderr
2. При отладке полезно использовать макросы, сообщающие, например, номер строки и имя функции
3. Полезно уметь “отключать” отладочные сообщения

Простой пример отладочных выводов “на коленке” (более подробно этот вопрос рассматривается в статье [Logging with GCC](#)):

```
#include <stdio.h>

#define DEBUG

int main(){
    #ifdef DEBUG
        fprintf(stderr, "DEBUG: %s:%s:%d: %s\n", __FILE__, __func__, __LINE__,
                    "Debug message");
    #endif
    return ;
}
```

Почитать подробнее

- [Полезные стандартные макросы](#)
- [Статья про простое логгирование \(Logging with GCC\)](#)
- [Использование syslog](#)

gdb + cli

GDB имеет достаточно простой, но мощный командный интерфейс и хорошую справку по нему. Однако, если достаточно всего лишь узнать в каком месте программа упала (Segmentation fault), может хватить и следующего набора действий:

1. Собрать программу с добавлением отладочных данных:

```
gcc -g myprog.c
```

2. Открыть её в отладчике:

```
gdb ./a.out
```

3. Запустить программу командой run:

```
(gdb) run
```

4. Ввести исходные данные, если ваша программа получает какие-то данные на вход. Если требуется перенаправить на вход вашей программе содержимое файла, запустите ее с помощью

```
(gdb) run < input_file.txt
```

5. gdb распечатает сообщение "Program received signal SIGSEGV, Segmentation fault." с указанием строки и инструкции на которой это произошло. Вы также можете после краха Вашей программы написать

```
(gdb) where
```

и получить подробный стектрейс

6. Вывести на экран значения переменных на этот момент можно с помощью команды [print](#)

- [Краткий tutorial с примерами отладки ломающихся программ](#)
- [Полезное про массивы и работу с памятью](#)
- [Полная официальная документация](#)

gdb + ide

Любая среда разработки или даже мощный текстовый редактор обычно предоставляют вам графический интерфейс для использования gdb при отладке своих программ. Обычно он достаточно наглядный и имеет хорошее описание для каждой IDE.

Вы можете самостоятельно найти описание использования отладчика в вашей любимой IDE. Для CLion можно посмотреть эти источники:

- [Debugging in CLion](#)
- [Debugging in CLion on youtube](#)

From:
<http://se.moevm.info/> - **se.moevm.info**

Permanent link:
<http://se.moevm.info/doku.php/courses:programming:debug?rev=1574138757>

Last update: **2022/12/10 09:08**

