Тестирование и отладка

Галопом по Европам

Николай Пономарев

3 ноября 2025 г.

Санкт-Петербургский
государственный университет

Тестирование

Уже было в соседнем курсе, но повторим ещё раз

Тестирование

Уже было в соседнем курсе, но повторим ещё раз

- Любая программа содержит ошибки
- Если программа не содержит ошибок, их содержит алгоритм, который реализует эта программа
- Если ни программа, ни алгоритм ошибок не содержат, такая программа даром никому не нужна

Тестирование не позволяет доказать отсутствие ошибок, оно позволяет лишь найти ошибки, которые в программе присутствуют

Тестирование в Си

- Обширная экосистема для тестирования
  - GoogleTest
  - cmocka
  - Unity Test
  - ...
- НО довольно сложная в настройке и использовании
- Пойдём по простому пути

Пример типичного теста

```
bool balanceOfParentheses(const char* parentheses)
{
    ...
}
bool testCorrectCase()
{
    return balanceOfParentheses("()");
}
bool testIncorrectCases()
{
    return !balanceOfParentheses("((") && !balanceOfParentheses(")(") ;
}
int main(void) {
    if (!testCorrectCase() || !testIncorrectCases()) {
        printf("Tests failed\n");
        return 1;
    }
    return 0;
}
```

Undefined & unspecified behavior[1]

Определение (Unspecified behavior)

behavior, that results from the use of an unspecified value, or other behavior upon which this document provides two or more possibilities and imposes no further requirements on which is chosen in any instance

Определение (Undefined behavior (UB))

behavior, upon use of a nonportable or erroneous program construct or of erroneous data, for which this document imposes no requirements

[1] Определения из стандарта: https://www.open-std.org/JTC1/SC22/WG14/www/docs/n3220.pdf

Примеры unspecified behavior

- Код завершения, возвращаемый в среду выполнения, если тип возвращаемого значения функции `main` не совместим с `int`
- Многие аспекты внутреннего представления типов данных
- Порядок вычисления аргументов функции
- Порядок и непрерывность памяти, выделяемой последовательными вызовами функций `calloc`, `malloc`, `realloc` и `aligned_alloc`

Примеры undefined behavior

- Разыменование нулевого указателя
- Возникновение исключительной ситуации при вычислении выражения (например, если результат математически не определён или не попадает в диапазон представимых значений для своего типа)
- Использование значения указателя на объект, время жизни которого завершилось
- Два объявления одного и того же объекта или функции с несовместимыми типами
- Сложение или вычитание указателя на объект массива (или сразу за ним) и целочисленного типа, которое даёт результат, указывающий не на тот же массив (или сразу за ним)

Отладка

- Устойчивое воспроизведение ошибки
  - Вместо srand(time(NULL)) — srand(<какое-то фиксированное значение>)
  - Ошибка должна воспроизводиться быстро
- Локализация ошибки
  - Аналитически
  - Отладка
- Отладочная гипотеза
  - Похоже на научный подход — гипотеза, эксперимент, уточнение, эксперимент и т.д.
  - Тестовый прогон с отладочной печатью
  - Тестовый прогон под отладчиком

gdb

gdb (the GNU Project debugger) — отладчик проекта GNU, поддерживающий огромное количество архитектур и операционных систем

- В основном консольный
- Есть псевдо-графический режим (см. tui)
- Может интегрироваться с VS Code

Отладка с gdb

- Приложение должно быть собрано с флагом -g
  - Сравните приложение собранное с -g и без него при помощи objdump -S
- Запускать как $ gdb <executable>
- Некоторые полезные команды
  - run — Запустить программу
  - break <название> (b <название>) — Установить точку останова
  - delete <номер> (d <номер>) — Удалить точку останова
  - step (s) — Сделать шаг исполнения (step into)
  - next (n) — Сделать шаг исполнения, не заходя в функции (step over)
  - continue (c) — Продолжить исполнение
  - print <название> — Вывести значение переменной
  - backtrace (bt) — Обратная трассировка (после падения)
  - set history save — Запоминать историю (полезно прописать в .gdbinit)

Домашнее задание

Написать тесты к домашке «Сортированный список».
Тесты должны запускаться по флагу --test, переданному исполняемому файлу.
Также можно настроить запуск тестов через ctest (см. https://coderefinery.github.io/cmake-workshop/testing/)
Не забывайте проверять граничные случаи.
Предпочтительно сдавать в отдельной ветке.