

Практическая часть

Тестирование программ

Тестирование программы подразумевает наличие набора тестов, на которых её надо проверить. В простейшем случае, программа принимает в качестве параметров имена файлов или работает со стандартными потоками ввода-вывода.

Т.к. работа с тестовыми файлами обычно не вызывает проблем, то имеет смысл ввод однотипных данных с клавиатуры свести тоже к работе с файлами. В этом помогут стандартные возможности командного интерпретатора:

- перенаправление ввода на ввод из файла: `./a.out < input.txt`
- перенаправление вывода в файл: `./a.out > output.txt`

Имена и расширения файлов могут быть любыми, методы можно комбинировать:

- `./a.out < input.txt > output.txt`

Отладка программ

В unix-системах, как правило, доступен отладчик `gdb`, поэтому кратко опишем, как с ним работать.

Предварительно программа должна быть откомпилирована с информацией для отладчика, т. е. должна включать в себя ключ «-g», пример: **`c99 -g main.c`**

Основные команды

- `g,run`: запуск (`run`), параметры для `argv` перечисляются после команды. Начнется выполнение до первой точки останова или завершения программы; Можно переопределять каналы ввода-вывода
- `b, breakpoint`: установка точки останова. Принимает в качестве параметра:
 - номер строки (можно с именем файла, если не в этом же файле)
 - имя функции (например, `main`)
- после остановки можно продолжить работу:
 - `c,continue`: продолжить до следующей точки останова или конца выполнения
 - `n, next`: до следующей строки программы (не заходя в вызов функции, например)
 - `s,step`: выполнить один шаг (в отличие от `next`, зайдет во вызов функции и остановится)
- `i,info`: Получение информации:
 - `i b`: получение информации об установленных точках останова

- `p,print`: вывод значение переменной/выражения:
 - `p a` — вывод значения переменной `a`
 - `p a<1` — вывод результата сравнения `a` и `1`
- `d,display`: аналогично `print`, только результат будет выводиться при каждом останове программы;
- `q,quit`: выход из отладчика.

Работа с отладчиком, примеры.

Пример 1: программа падает по неправильной работе с указателем в неизвестном месте.

Решение:

- компилируем, запускаем в отладчике: `gdb a.out`
- Запускаем (если требуется, то указываем параметры после `r`): **r**
- взаимодействуем с программой, программа падает
- спрашиваем, где упала (внутри отладчика): **where**
- получаем номер строки, где произошло неверное обращение и список вложенных вызовов функций (файл, строка), которые привели нас к проблеме.
- Находим относящееся к нашему коду
 - Скажем, если мы передали `NULL`-указатель в функцию `strlen`, то упадем мы внутри этой функции, а надо смотреть, где мы её вызвали.
- Смотрим значение переменных, анализируем состояние программы, добавляем точки останова, запускаем новое тестирование и т. д.