

Балтийский государственный технический университет  
«ВОЕНМЕХ» им. Д. Ф. Устинова

Факультет «О» Естественнонаучный

Кафедра О7 «Информационные системы и программная инженерия»

**Практическая работа №2**  
по дисциплине «Компьютерный практикум»  
на тему «ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ ОТЛАДЧИКА GDB»

Выполнил:  
Студент Целихович А.Л.  
Группа О712Б  
Преподаватель:

Санкт-Петербург  
2022 г

## **1 Постановка задачи**

### **1.1 Проверка вычисляемого значения на каждом шагу цикла**

Написать программу 1 в соответствии с вариантом при помощи любого текстового редактора. Скомпилировать программу с добавлением в файл отладочной информации. Далее, используя отладчик GDB, проверить значение вычисляемого в цикле выражения на каждом шаге цикла. При использовании нескольких циклов проверить все значения. В отчете привести используемые команды отладчика и полученный результат.

### **1.2 Проверка стека на каждом входе/выходе функции**

Написать программу 2 в соответствии с вариантом при помощи любого текстового редактора. Для ввода и вывода строки использовать отдельные функции. После скомпилировать программу с добавлением в файл отладочной информации. Затем, используя отладчик GDB, проверить содержимое стека при входе в функции ввода и вывода строки и выходе из них. В отчете привести содержимое стека и используемые команды.

### **1.3 Проверка стека на каждом входе/выходе функции статической библиотеки**

Написать программу 3 в соответствии с вариантом при помощи любого текстового редактора. Для ввода и вывода строки использовать отдельные функции, помещенные в статическую библиотеку. После скомпилировать программу с добавлением в файл отладочной информации. Далее, используя отладчик GDB, проверить содержимое стека при входе в функции ввода и вывода строки и выходе из них. В отчете привести содержимое стека и используемые команды.

### **1.4 Вариативная часть задания**

В практической работе выполняется 6 вариант. Для него предусмотрены следующие программы:

— Программа 1: определить, является ли данное натуральное число  $N$  факториалом какого-нибудь числа, если «да», то какого;

— Программа 2: дана строка символов. Определить, является ли она правильным скобочным выражением;

— Программа 3: дана символьная строка. Проверить, все ли слова после точки начинаются с заглавной буквы. Если нет – исправить.

## 2 Выполнение работы

### 2.1 Проверка вычисляемого значения на каждом шагу цикла

Выполнение первой части практической работы представлено на рисунках 2.1.1, 2.1.2 и 2.1.3.

```
Microsoft Windows [Version 10.0.19044.2006]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\User>cd C:\Program Files\CodeBlocks\MinGW\bin

C:\Program Files\CodeBlocks\MinGW\bin>g++ -g "C:\Users\User\Desktop\Lab3\CompPracticum\Lab2\Part1\Prog1.cpp" -o "C:\Users\User\Desktop\Lab3\CompPracticum\Lab2\Part1\Prog1.exe"

C:\Program Files\CodeBlocks\MinGW\bin>
```

Рисунок 2.1.1 – Компиляция с отладчиком программы

```
Microsoft Windows [Version 10.0.19044.2006]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\User>cd C:\Program Files\CodeBlocks\MinGW\bin

C:\Program Files\CodeBlocks\MinGW\bin>gdb "C:\Users\User\Desktop\TEST\Prog1.exe"
GNU gdb (GDB) 8.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-w64-mingw32".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from C:\Users\User\Desktop\TEST\Prog1.exe...done.
(gdb) run
Starting program: C:\Users\User\Desktop\TEST\Prog1.exe
[New Thread 11272.0x5d90]
[New Thread 11272.0x20b0]
[New Thread 11272.0x32ac]
[New Thread 11272.0xe48]
6
720
[Thread 11272.0x20b0 exited with code 0]
[Thread 11272.0xe48 exited with code 0]
[Thread 11272.0x32ac exited with code 0]
[Inferior 1 (process 11272) exited normally]
(gdb) █
```

Рисунок 2.1.2 – Включение отладчика

```

(gdb) break 9
Breakpoint 1 at 0x40157e: file C:\Users\User\Desktop\Lebs\3 CompPracticum\Lab2\Part1\Prog1.cpp, line 9.
(gdb) run
Starting program: C:\Users\User\Desktop\TEST\Prog1.exe
[New Thread 27612.0x2380]
[New Thread 27612.0x3e98]
[New Thread 27612.0x6824]
[New Thread 27612.0x1abc]
4
Thread 1 hit Breakpoint 1, main () at C:\Users\User\Desktop\Lebs\3 CompPracticum\Lab2\Part1\Prog1.cpp:9
9          while(N!=1) x*=N--;
(gdb) print x
$1 = 1
(gdb) step
Thread 1 hit Breakpoint 1, main () at C:\Users\User\Desktop\Lebs\3 CompPracticum\Lab2\Part1\Prog1.cpp:9
9          while(N!=1) x*=N--;
(gdb) print x
$2 = 4
(gdb) step
Thread 1 hit Breakpoint 1, main () at C:\Users\User\Desktop\Lebs\3 CompPracticum\Lab2\Part1\Prog1.cpp:9
9          while(N!=1) x*=N--;
(gdb) print x
$3 = 12
(gdb) step
Thread 1 hit Breakpoint 1, main () at C:\Users\User\Desktop\Lebs\3 CompPracticum\Lab2\Part1\Prog1.cpp:9
9          while(N!=1) x*=N--;
(gdb) print x
$4 = 24
(gdb) step
10          cout << x << endl;
(gdb) print x
$5 = 24
(gdb) step
24
12          return 0;
(gdb) step
13      }(gdb) step
0x00000000004013c7 in __tmainCRTStartup ()
(gdb) step
Single stepping until exit from function __tmainCRTStartup,
which has no line number information.
[Thread 27612.0x3e98 exited with code 0]
[Thread 27612.0x6824 exited with code 0]
[Thread 27612.0x1abc exited with code 0]
[Inferior 1 (process 27612) exited normally]
(gdb)

```

Рисунок 2.1.3 – Запуск программы с проверкой вычисляемого значения на каждом шагу цикла

## 2.2 Проверка стека на каждом входе/выходе функции

Выполнение второй части практической работы представлено на рисунках 2.2.1 и 2.2.2.

```

Microsoft Windows [Version 10.0.19044.2006]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\User>cd C:\Program Files\CodeBlocks\MinGW\bin

C:\Program Files\CodeBlocks\MinGW\bin>g++ -g "C:\Users\User\Desktop\TEST\Prog2.cpp" -o "C:\Users\User\Desktop\TEST\Prog2.exe"

C:\Program Files\CodeBlocks\MinGW\bin>

```

Рисунок 2.2.1. – Компиляция с отладчиком программы

```

C:\Program Files\CodeBlocks\MinGW\bin>gdb "C:\Users\User\Desktop\TEST\Prog2.exe"
GNU gdb (GDB) 8.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-w64-mingw32".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from C:\Users\User\Desktop\TEST\Prog2.exe...done.
(gdb) break 19
Breakpoint 1 at 0x401634: file C:\Users\User\Desktop\TEST\Prog2.cpp, line 19.
(gdb) break 21
Breakpoint 2 at 0x40165c: file C:\Users\User\Desktop\TEST\Prog2.cpp, line 21.
(gdb) break 32
Breakpoint 3 at 0x40180b: file C:\Users\User\Desktop\TEST\Prog2.cpp, line 32.
(gdb) break 34
Breakpoint 4 at 0x401833: file C:\Users\User\Desktop\TEST\Prog2.cpp, line 34.
(gdb) run
Starting program: C:\Users\User\Desktop\TEST\Prog2.exe
[New Thread 20876.0x3650]
[New Thread 20876.0xdd8]
[New Thread 20876.0x41b8]
[New Thread 20876.0x4ddc]

Thread 1 hit Breakpoint 1, main () at C:\Users\User\Desktop\TEST\Prog2.cpp:20
20         str=Input();
(gdb) where
#0  main () at C:\Users\User\Desktop\TEST\Prog2.cpp:20
(gdb) continue
Continuing.
(9*8)+(69-2))

Thread 1 hit Breakpoint 2, main () at C:\Users\User\Desktop\TEST\Prog2.cpp:22
22         for(int i=0; i<str.length(); i++) {
(gdb) where
#0  main () at C:\Users\User\Desktop\TEST\Prog2.cpp:22
(gdb) continue
Continuing.
This string is NOT a correct parenthesis expression!

Thread 1 hit Breakpoint 3, main () at C:\Users\User\Desktop\TEST\Prog2.cpp:33
33         Output(str);
(gdb) where
#0  main () at C:\Users\User\Desktop\TEST\Prog2.cpp:33
(gdb) continue
Continuing.
You have given for processing: (9*8)+(69-2))

Thread 1 hit Breakpoint 4, main () at C:\Users\User\Desktop\TEST\Prog2.cpp:35
35         return 0;
(gdb) where
#0  main () at C:\Users\User\Desktop\TEST\Prog2.cpp:35
(gdb) continue
Continuing.
[Thread 20876.0x4ddc exited with code 0]
[Thread 20876.0x41b8 exited with code 0]
[Thread 20876.0xdd8 exited with code 0]
[Inferior 1 (process 20876) exited normally]
(gdb) quit

C:\Program Files\CodeBlocks\MinGW\bin>

```

Рисунок 2.2.2 – Запуск программы с проверкой стека на наличие элементов на входе/выходе из функций

## 2.3 Проверка стека на каждом входе/выходе функции статической библиотеки

Выполнение третьей части практической работы представлено на рисунках 2.3.1 и 2.3.2.

```
Microsoft Windows [Version 10.0.17134.1204]
(c) 2018 Microsoft Corporation. Все права защищены.

C:\Users\TEMP.PRIM.010\Desktop\TEST>
C:\Users\TEMP.PRIM.010\Desktop\TEST>g++ Prog3.cpp
C:/Programs/MinGW32/bin/g++.exe: /lib/gcc/mingw32/5.3.0/../../../../x86_64-w64-mingw32/lib/../../../libmingw32.a(crt_c.o)(crt_c.o): undefined reference to 'std::basic_string<char, std::char_traits<char>, std::allocator<char>>::append(char const*)'
C:\Users\TEMP.PRIM.010\Desktop\TEST>g++ -c Prog3.cpp
C:\Users\TEMP.PRIM.010\Desktop\TEST>g++ Prog3.o -L. -luser32
C:\Users\TEMP.PRIM.010\Desktop\TEST>
```

Рисунок 2.3.1 – Создание статической библиотеки с компиляцией программы с отладчиком

```
C:\Users\TEMP.PRIM.010\Desktop\TEST>gdb a.exe
GNU gdb (GDB) 8.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-w64-mingw32".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.exe...done.
(gdb) break 9
Breakpoint 1 at 0x401582: file Prog3.cpp, line 9.
(gdb) break 11
Breakpoint 2 at 0x40158b: file Prog3.cpp, line 11.
(gdb) break 20
Breakpoint 3 at 0x40169f: file Prog3.cpp, line 20.
(gdb) break 22
Breakpoint 4 at 0x4016ab: file Prog3.cpp, line 22.
(gdb) run
Starting program: C:\Users\TEMP.PRIM.010\Desktop\TEST\a.exe
[New Thread 3708.0x1ee0]
[New Thread 3708.0x1720]
[New Thread 3708.0x2e44]
[New Thread 3708.0x6ac]

Thread 1 hit Breakpoint 1, main () at Prog3.cpp:10
10      str=Input();
(gdb) where
#0  main () at Prog3.cpp:10
(gdb) continue
Continuing.
Enter size line: 13
Enter the line: str. 8n rt No

Thread 1 hit Breakpoint 2, main () at Prog3.cpp:12
12      for(int i=0; str[i]!='\0'; i++) {
(gdb) where
#0  main () at Prog3.cpp:12
(gdb) continue
Continuing.

Thread 1 hit Breakpoint 3, main () at Prog3.cpp:21
21      Output(str);
(gdb) where
#0  main () at Prog3.cpp:21
(gdb) continue
Continuing.
String after processing: str. 8n Rt No

Thread 1 hit Breakpoint 4, main () at Prog3.cpp:23
23      delete(str);
(gdb) where
#0  main () at Prog3.cpp:23
(gdb) continue
Continuing.
[Thread 3708.0x1720 exited with code 0]
[Thread 3708.0x6ac exited with code 0]
[Thread 3708.0x2e44 exited with code 0]
[Inferior 1 (process 3708) exited normally]
(gdb) quit

C:\Users\TEMP.PRIM.010\Desktop\TEST>
```

Рисунок 2.3.2 – Запуск программы с проверкой стека на наличие элементов на входе/выходе из статической библиотеки

## 3 Тексты программ

### 3.1 Проверка вычисляемого значения на каждом шагу цикла

```
#include <iostream>

using namespace std;

int main() {
    int N, x=1;
    cin >> N;
    if(N>0) {
        while(N!=1) x*=N--;
        cout << x << endl;
    } else cout << "Is not a factorial!" << endl;
    return 0;
}
```

### 3.2 Проверка стека на каждом входе/выходе функции

```
#include <iostream>
#include <stack>

using namespace std;

string Input() {
    string s;
    cin >> s;
    return s;
}

void Output(string s) {
    cout << "You have given for processing: " << s << endl;
}

int main() {
    string str="";
    stack<char> l;
    //Enter funct Input()
    str=Input();
    //Exit func Input()
    for(int i=0; i<str.length(); i++) {
        if(l.empty()) {
            if(str[i]=='(' || str[i]==')') l.push(str[i]);
        } else {
            if(l.top()=='(' && str[i]==')') l.pop();
            else if(str[i]=='(' || str[i]==')') l.push(str[i]);
        }
    }
    if(l.empty()) cout << "This is a string correctly parenthesized expression!" << endl;
    else cout << "This string is NOT a correct parenthesis expression!" << endl;
    //Enter funct Output()
    Output(str);
    //Exit func Output()
    return 0;
}
```



### 3.3 Проверка стека на каждом входе/выходе функции статической библиотеки

Код программы №3 хранится в нескольких файлах. Содержимое файла hworld.h:

```
#ifndef HWORLD_H_INCLUDED
#define HWORLD_H_INCLUDED

char* Input();
void Output(char* s);

#endif
```

Содержимое файла Put.cpp:

```
#include <iostream>
#include "hworld.h"

char* Input() {
    int N; char *s;
    std::cout << "Enter size line: ";
    std::cin >> N;
    s=new char[N+1];
    while (getchar() != '\n');
    std::cout << "Enter the line: ";
    std::cin.getline(s, N+1);
    s[N]='\0';
    return s;
}

void Output(char* s) {
    printf("String after processing: %s\n", s);
    return;
}
```

Содержимое файла Prog3.cpp:

```
#include <iostream>
#include <stack>
#include "hworld.h"

int main() {
    char *str;
    std::stack<char> l;
    //Enter funct Input()
    str=Input();
    //Exit funct Input()
    for(int i=0; str[i]!='\0'; i++) {
        if(l.empty() && str[i]=='.') l.push(str[i]);
        else if(!l.empty()) {
            if(str[i-1]==' ' && (str[i]>'Z' || str[i]<'A') &&
(str[i]<='z' && str[i]>='a')) {
                str[i]-=32;
            }
        }
    }
    //Enter funct Output()
    Output(str);
    //Exit funct Output()
    delete(str);
    return 0;
}
```