

# Лабораторная работа №13

Средства, применяемые при разработке программного обеспечения в ОС UNIX/Linux

---

Кузнецова С. В.

30 апреля 2023

Российский университет дружбы народов, Москва, Россия

## Информация

---

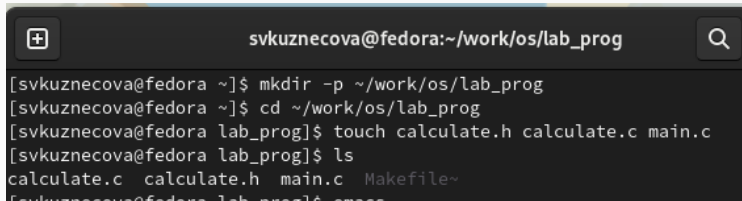
- Кузнецова София Вадимовна
- студент прикладной информатики
- Российский университет дружбы народов

## Выполнение лабораторной работы

---

## Создание подкаталога, файлов и переход в emacs

В домашнем каталоге создадим подкаталог `~/work/os/lab_prog`. После чего создадим в нём файлы: `calculate.h`, `calculate.c`, `main.c` и выполним проверку. Перейдём в `emacs`.

A terminal window with a dark background. The title bar shows a plus icon, the username and host 'svkuznecova@fedora', the current directory '~/work/os/lab\_prog', and a search icon. The terminal text shows the following commands and output:

```
[svkuznecova@fedora ~]$ mkdir -p ~/work/os/lab_prog
[svkuznecova@fedora ~]$ cd ~/work/os/lab_prog
[svkuznecova@fedora lab_prog]$ touch calculate.h calculate.c main.c
[svkuznecova@fedora lab_prog]$ ls
calculate.c calculate.h main.c Makefile~
[svkuznecova@fedora lab_prog]$ emacs
```

Рис. 1: Создание нового подкаталога и файлов в нём

# Перенос первого скрипта

В emacs откроем созданный файл calculate.c и приступим к переносу в него скрипта из файла.



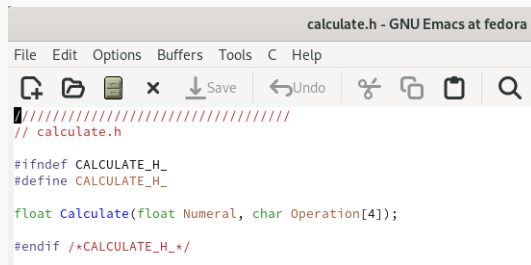
```
calculate.c - GNU Emacs at Fedora
File Edit Options Buffers Tools C Help
Save Undo Copy Paste Find
// calculate.c
#include <stdio.h>
#include <math.h>
#include <string.h>
#include <unistd.h>

float
Calculate(float Numeral, char operation[4])
{
    float SecondNumeral;
    if(strlen(operation) == 3)
    {
        printf("stoppe charseee: ");
        scanf("%f", &SecondNumeral);
        return(Numeral + SecondNumeral);
    }
    else if(strlen(operation) == 1)
    {
        printf("beveraseee: ");
        scanf("%f", &SecondNumeral);
        return(Numeral - SecondNumeral);
    }
    else if(strlen(operation) == 2)
    {
        printf("baseromee: ");
        scanf("%f", &SecondNumeral);
        return(Numeral * SecondNumeral);
    }
    else if(strlen(operation) == 1)
    {
        printf("baseromee: ");
        scanf("%f", &SecondNumeral);
        if(SecondNumeral == 0)
        {
            printf("baseromee: genesse na nohi: ");
            return(DOUBLE_VAL);
        }
        else
            return(Numeral / SecondNumeral);
    }
    else if(strlen(operation) == 3)
    {
        printf("baseromee: ");
    }
}
```

Рис. 2: Перенос скрипта для calculate.c

## Перенос второго скрипта

После того как мы перенесли и сохранили скрипт для первого файла, открываем файл `calculate.h` и также приступим в него скрипта, но уже для второго файла. Выполняем сохранение.



```
calculate.h - GNU Emacs at fedora
File Edit Options Buffers Tools C Help
[Icons: New, Open, Save, Close, Save All, Undo, Redo, Copy, Paste, Find]
////////////////////////////////////
// calculate.h

#ifndef CALCULATE_H_
#define CALCULATE_H_

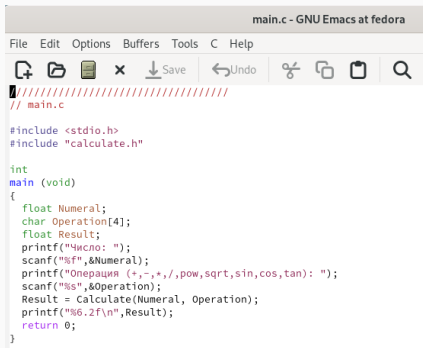
float Calculate(float Numeral, char Operation[4]);

#endif /*CALCULATE_H_*/
```

Рис. 3: Перенос скрипта для `calculate.h`

## Перенос третьего скрипта

Теперь нам нужно перенести последний третий скрипт в файл main.c. После чего также выполняем сохранение и закрываем emacs.



```
main.c - GNU Emacs at fedora
File Edit Options Buffers Tools C Help
[Icons: Open, Save, Undo, Redo, Search]
// main.c
#include <stdio.h>
#include "calculate.h"

int
main (void)
{
    float Numeral;
    char Operation[4];
    float Result;
    printf("Число: ");
    scanf("%f", &Numeral);
    printf("Операция (+, -, *, /, pow, sqrt, sin, cos, tan): ");
    scanf("%s", &Operation);
    Result = Calculate(Numeral, Operation);
    printf("%6.2f\\n", Result);
    return 0;
}
```

Рис. 4: Перенос скрипта для main.c



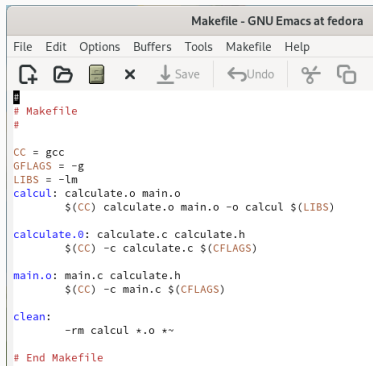
В терминале выполняем компиляцию программ посредством gcc.

```
[svkuznecova@fedora lab_prog]$ gcc -c calculate.c
[svkuznecova@fedora lab_prog]$ gcc -c main.c
[svkuznecova@fedora lab_prog]$ gcc calculate.o main.o calcul -lm
/usr/bin/ld: невозможно найти calcul: Нет такого файла или каталога
collect2: ошибка: выполнение ld завершилось с кодом возврата 1
[svkuznecova@fedora lab_prog]$ gcc calculate.o main.o -o calcul -lm
[svkuznecova@fedora lab_prog]$ ls
calcul calculate.c calculate.h calculate.o main.c main.o Makefile~
[svkuznecova@fedora lab_prog]$
```

Рис. 5: Компиляция программы

## Создание и изменение Makefile

Создадим Makefile и внесём туда небольшие изменения. В переменную CFLAGS добавил опцию -g, необходимую для компиляции объектных файлов и их использования в программе отладчика GDB. Сделал так, что утилита компиляции выбирается с помощью переменной CC.



```
# Makefile
#

CC = gcc
GFLAGS = -g
LIBS = -lm
calcul: calculate.o main.o
    $(CC) calculate.o main.o -o calcul $(LIBS)

calculate.o: calculate.c calculate.h
    $(CC) -c calculate.c $(CFLAGS)

main.o: main.c calculate.h
    $(CC) -c main.c $(CFLAGS)

clean:
    -rm calcul *.o *~

# End Makefile
```

Рис. 6: Создание и изменение Makefile

# Отладка программы calcul с помощью gdb

С помощью gdb выполняем отладку программы calcul. После чего запустим программу командой run.

```
(tsahorin@fedora lab_prog)$ make clean
rm calcul *.o *~
(tsahorin@fedora lab_prog)$ make calculate.o
gcc -c -o calculate.o calculate.c
(tsahorin@fedora lab_prog)$ make main.o
gcc -c -o main.c
(tsahorin@fedora lab_prog)$ make calcul
gcc calculate.o main.o -o calcul -lm
(tsahorin@fedora lab_prog)$ gdb ./calcul
GNU gdb (GDB) Fedora 11.1-1.fc15
Copyright (C) 2012 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-redhat-linux-gnu".
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 ./calcul...
This GDB supports auto-downloading debuginfo from the following URLs:
<http://debuginfod.fedoraproject.org/>
Enable debuginfo for this session? (y or n) y
Debuginfod has been enabled.
To make this setting permanent, add "set debuginfod enabled on" to .gdbinit.
(No debugging symbols found in ./calcul)
(gdb) run
Starting program: /home/tsahorin/work/os/lab_prog/calcul
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Welcome! 5
Multipawn (+,-,*,/,pow,sqrt,sin,cos,tan): +
Multipawn: 6
28.00
[Inferior 1 (process 60371) exited normally]
(gdb)
```

Рис. 7: Отладка программы calcul

Воспользовавшись утилитой splint проанализируем коды файлов `calculate.c` и `main.c`. С помощью утилиты splint выяснилось, что в файлах `calculate.c` и `main.c` присутствует функция чтения `scanf`, возвращающая целое число (тип `int`), но эти числа не используются и нигде не сохраняются. Утилита вывела предупреждение о том, что в файле `calculate.c` происходит сравнение вещественного числа с нулем. Также возвращаемые значения (тип `double`) в функциях `pow`, `sqrt`, `sin`, `cos` и `tan` записываются в переменную типа `float`, что свидетельствует о потере данных.

# Анализ файла calculate.c утилитой splint

```
[ismahorin@fedora lab_prog]$ splint calculate.c
Splint 3.1.2 --- 23 Jul 2021

calculate.h:37: Function parameter Operation declared as manifest array (size
      constant is meaningless)
  A formal parameter is declared as an array with size. The size of the array
  is ignored in this context, since the array formal parameter is treated as a
  pointer. (Use -fixedformalarray to inhibit warning)
calculate.c:10:31: Function parameter Operation declared as manifest array
      (size constant is meaningless)
calculate.c: (in function Calculate)
calculate.c:16:7: Return value (type int) ignored: scanf("%f", &Sec...
  Result returned by function call is not used. If this is intended, can cast
  result to (void) to eliminate message. (Use -retvalint to inhibit warning)
calculate.c:22:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:28:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:34:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:35:10: Dangerous equality comparison involving float types:
      SecondNumeral == 0
  Two real (float, double, or long double) values are compared directly using
  == or != primitive. This may produce unexpected results since floating point
  representations are inexact. Instead, compare the difference to FLT_EPSILON
  or DBL_EPSILON. (Use -realcompare to inhibit warning)
calculate.c:38:10: Return value type double does not match declared type float:
      (HUGE_VAL)
  To allow all numeric types to match, use +relaxtypes.
calculate.c:46:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:47:13: Return value type double does not match declared type float:
      (pow(Numeral, SecondNumeral))
calculate.c:50:11: Return value type double does not match declared type float:
      (sqrt(Numeral))
calculate.c:52:11: Return value type double does not match declared type float:
      (sin(Numeral))
calculate.c:54:11: Return value type double does not match declared type float:
      (cos(Numeral))
calculate.c:56:11: Return value type double does not match declared type float:
      (tan(Numeral))
calculate.c:60:13: Return value type double does not match declared type float:
      (HUGE_VAL)

Finished checking --- 15 code warnings
[ismahorin@fedora lab_prog]$
```

Рис. 8: Анализ файла calculate.c

## Анализ файла main.c утилитой splint

```
[ismahorin@fedora lab_prog]$ splint main.c
Splint 3.1.2 --- 23 Jul 2021

calculate.h:7:37: Function parameter Operation declared as manifest array (size
                    constant is meaningless)
    A formal parameter is declared as an array with size.  The size of the array
    is ignored in this context, since the array formal parameter is treated as a
    pointer. (Use -fixedformalarray to inhibit warning)
main.c: (in function main)
main.c:14:3: Return value (type int) ignored: scanf("%f", &Num...
    Result returned by function call is not used. If this is intended, can cast
    result to (void) to eliminate message. (Use -retvalint to inhibit warning)
main.c:16:14: Format argument 1 to scanf (%s) expects char * gets char [4] *:
                &Operation
    Type of parameter is not consistent with corresponding code in format string.
    (Use -formattype to inhibit warning)
    main.c:16:11: Corresponding format code
main.c:16:3: Return value (type int) ignored: scanf("%s", &Ope...

Finished checking --- 4 code warnings
[ismahorin@fedora lab_prog]$
```

Рис. 9: Анализ файла main.c

## Выводы

---

В ходе выполнения лабораторной работы мы приобрели простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования С калькулятора с простейшими функциями.



Спасибо за внимание!