

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ по лабораторной работе № 13

дисциплина:Операционные системы

- Студент: Гань Чжаолун
- Группа:НФИБД-01-21
- № ст. билета: 1032198038

МОСКВА 2022 г.

Цель работы:

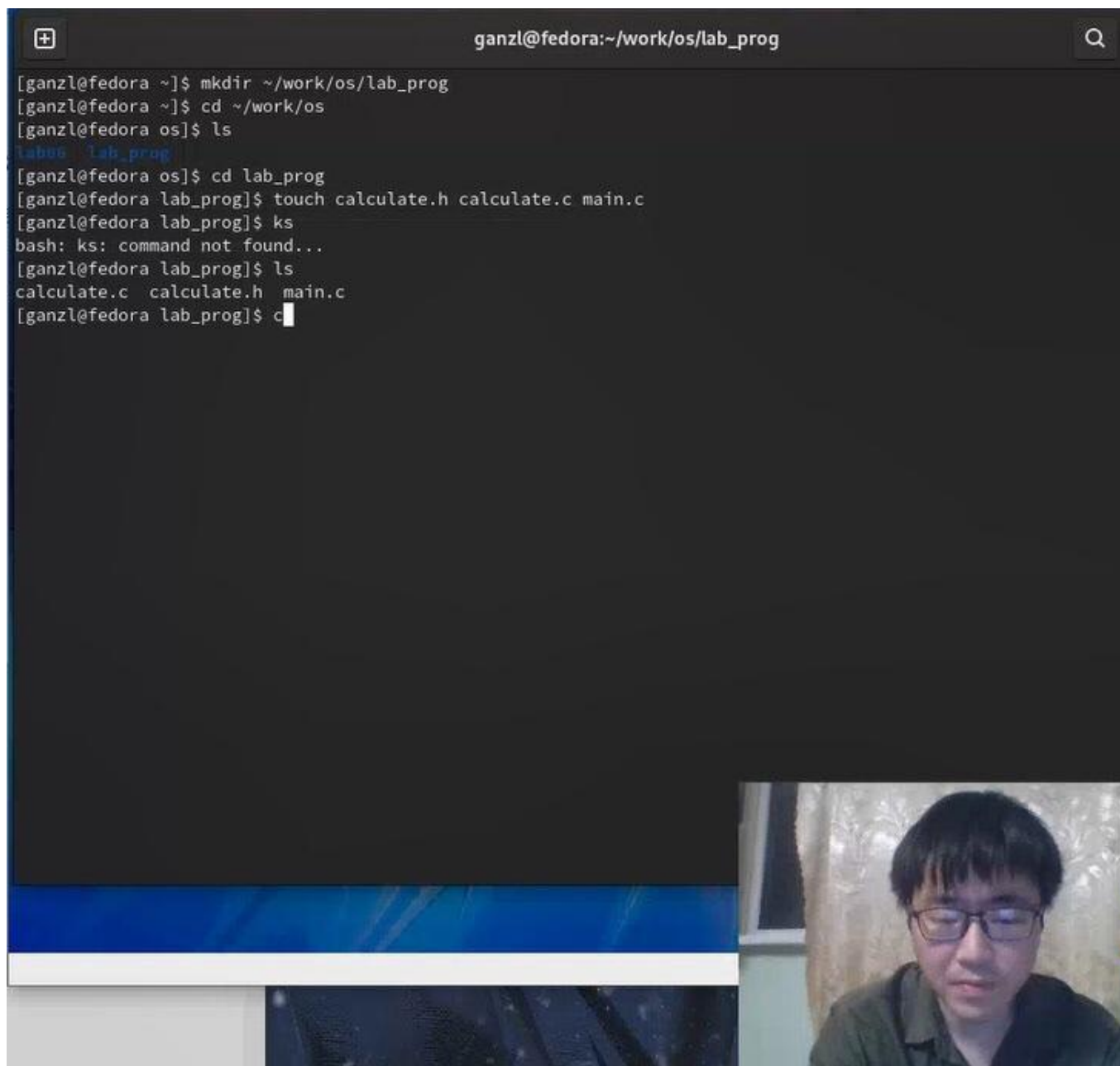
Приобрести простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования С калькулятора с простейшими функциями.

Выполнение работы:

1. В домашнем каталоге создайте подкаталог `~/work/os/lab_prog`.

2. Создайте в нём файлы: `calculate.h`, `calculate.c`, `main.c`.

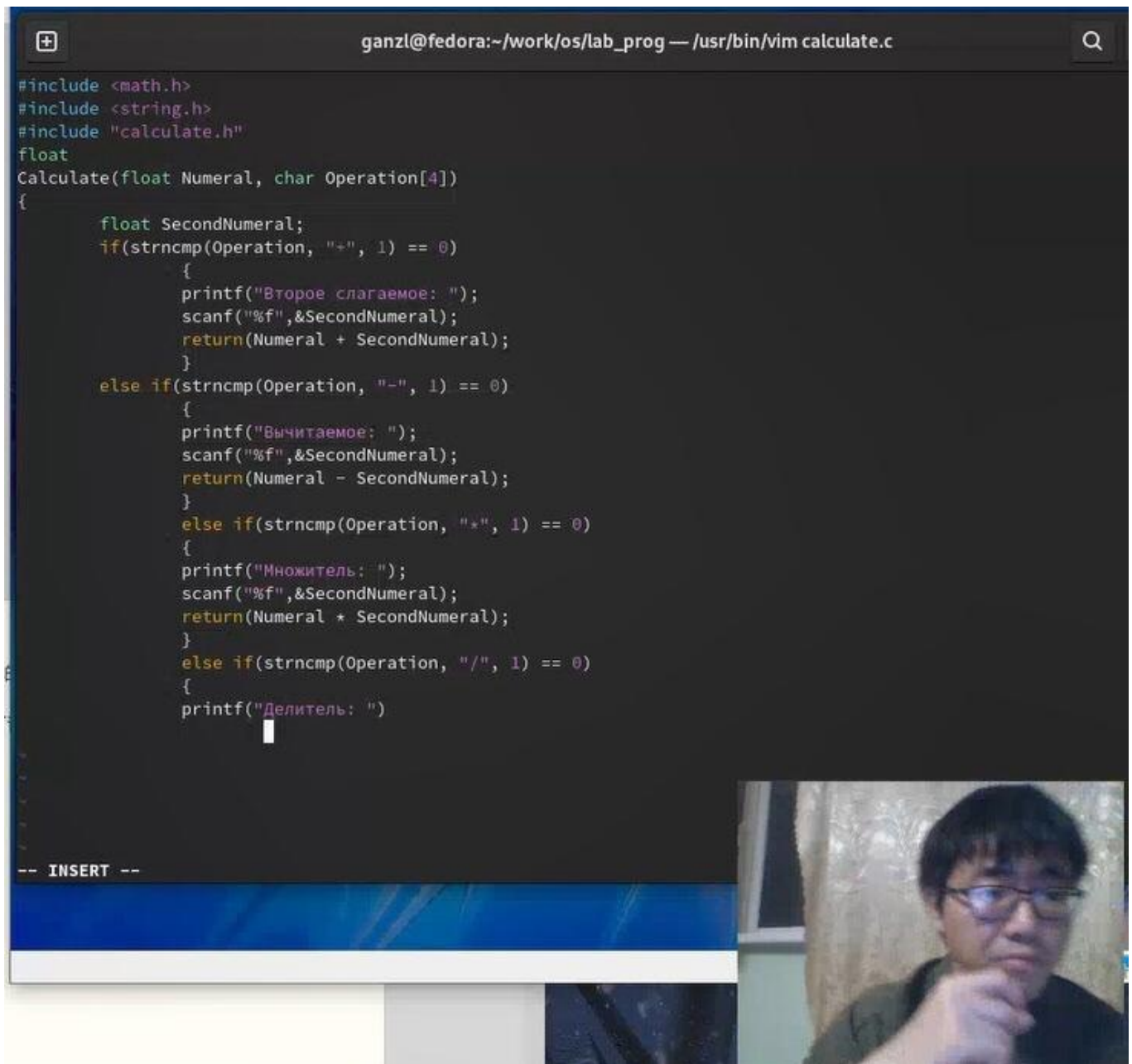
Я создаю каталог `lab_prog` и создаю `calculate` в `calculate.h`, `calculate.c`, `main.c` Тремя файлами.

A screenshot of a terminal window and a video feed. The terminal window, titled 'ganzl@fedora:~/work/os/lab_prog', shows a series of commands: 'mkdir ~/work/os/lab_prog', 'cd ~/work/os', 'ls', 'cd lab_prog', 'touch calculate.h calculate.c main.c', 'ks' (which results in an error), and 'ls' (which shows 'calculate.c calculate.h main.c'). The video feed in the bottom right corner shows a man with glasses and dark hair, wearing a dark shirt, looking down.

```
ganzl@fedora:~/work/os/lab_prog

[ganzl@fedora ~]$ mkdir ~/work/os/lab_prog
[ganzl@fedora ~]$ cd ~/work/os
[ganzl@fedora os]$ ls
labos  lab_prog
[ganzl@fedora os]$ cd lab_prog
[ganzl@fedora lab_prog]$ touch calculate.h calculate.c main.c
[ganzl@fedora lab_prog]$ ks
bash: ks: command not found...
[ganzl@fedora lab_prog]$ ls
calculate.c calculate.h main.c
[ganzl@fedora lab_prog]$ c
```

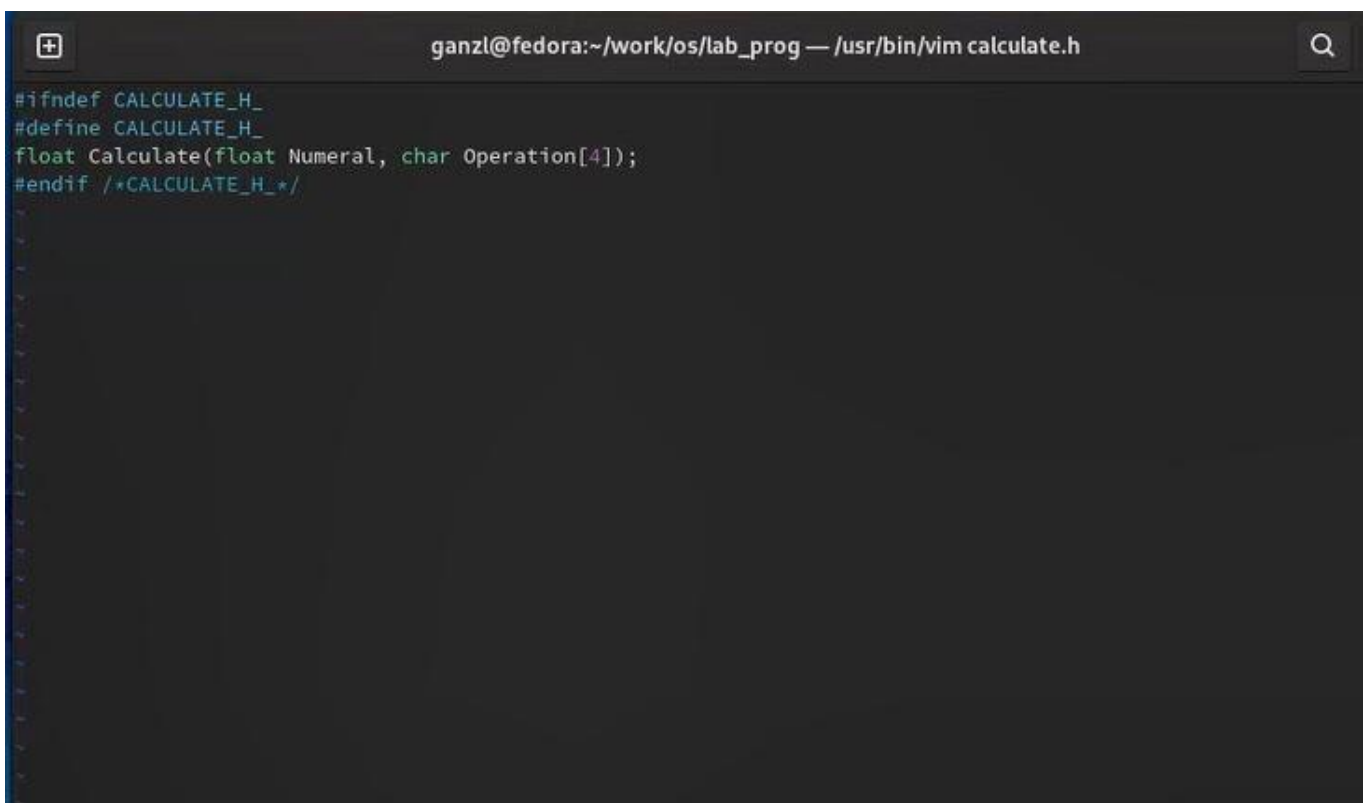
Я начал редактировать эти три файла



```
ganzl@fedora:~/work/os/lab_prog — /usr/bin/vim calculate.c

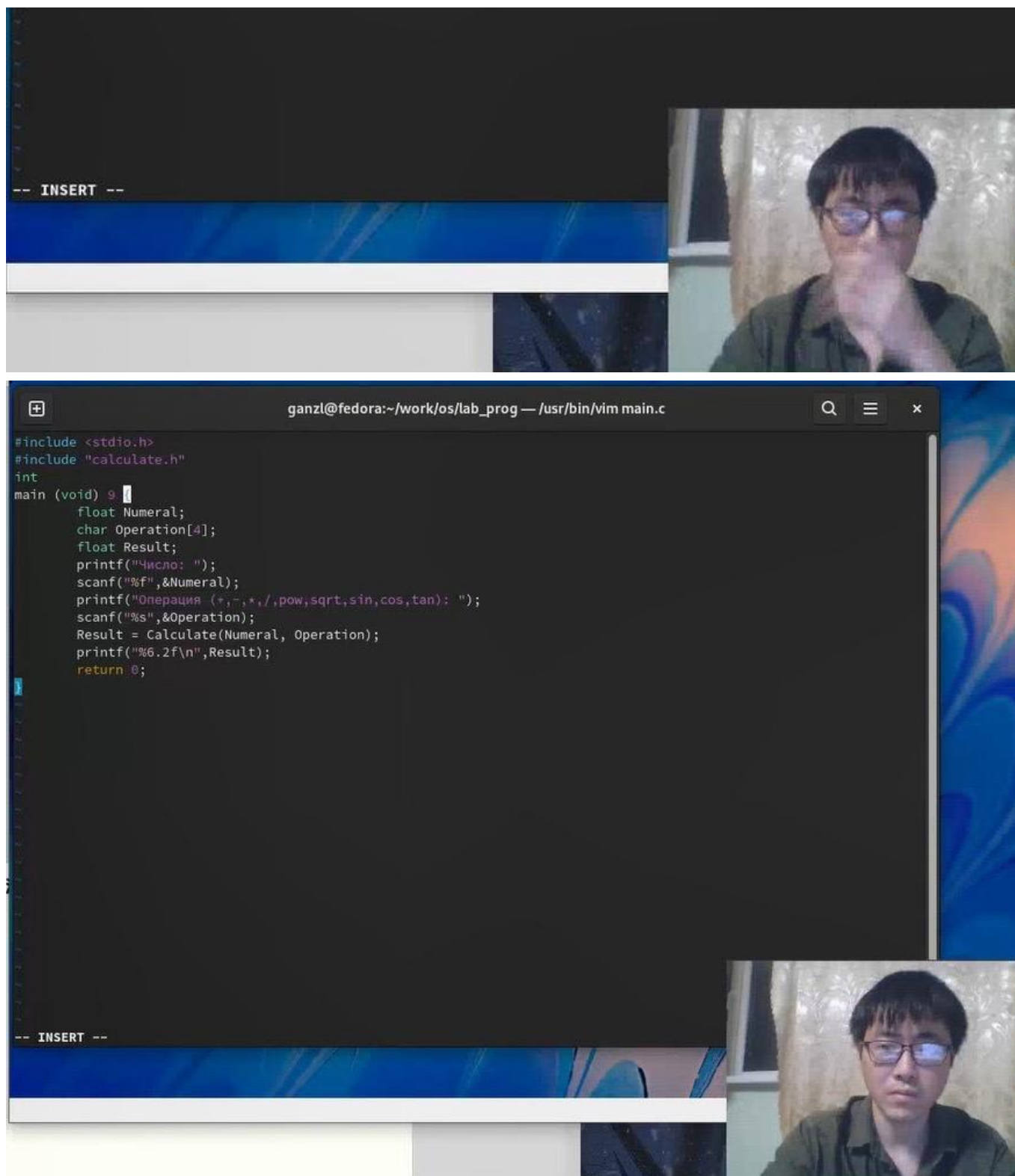
#include <math.h>
#include <string.h>
#include "calculate.h"
float
Calculate(float Numeral, char Operation[4])
{
    float SecondNumeral;
    if(strncmp(Operation, "+", 1) == 0)
    {
        printf("Второе слагаемое: ");
        scanf("%f",&SecondNumeral);
        return(Numeral + SecondNumeral);
    }
    else if(strncmp(Operation, "-", 1) == 0)
    {
        printf("Вычитаемое: ");
        scanf("%f",&SecondNumeral);
        return(Numeral - SecondNumeral);
    }
    else if(strncmp(Operation, "*", 1) == 0)
    {
        printf("Множитель: ");
        scanf("%f",&SecondNumeral);
        return(Numeral * SecondNumeral);
    }
    else if(strncmp(Operation, "/", 1) == 0)
    {
        printf("Делитель: ")
    }

-- INSERT --
```



```
ganzl@fedora:~/work/os/lab_prog — /usr/bin/vim calculate.h

#ifndef CALCULATE_H_
#define CALCULATE_H_
float Calculate(float Numeral, char Operation[4]);
#endif /*CALCULATE_H_*/
```



Я создаю файл makefile и редактирую его


```
ganzl@fedora:~/work/os/lab_prog

[ganzl@fedora lab_prog]$ vi calculate.c
[ganzl@fedora lab_prog]$ vi calculate.h
[ganzl@fedora lab_prog]$ gcc -c calculate.c
[ganzl@fedora lab_prog]$ gcc -c main.c
[ganzl@fedora lab_prog]$ gcc calculate.o main.o -o calcul -lm
[ganzl@fedora lab_prog]$ touch makefile
[ganzl@fedora lab_prog]$ vi ma
main.c  makefile
[ganzl@fedora lab_prog]$ vi ma
main.c  makefile
[ganzl@fedora lab_prog]$ vi makefile
```



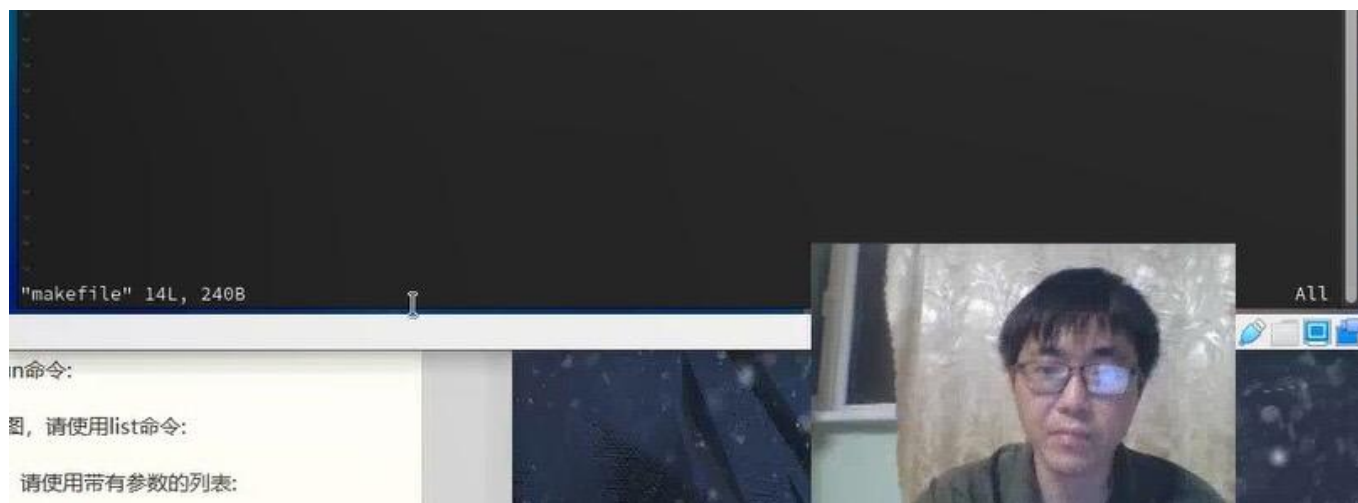
```
ganzl@fedora:~/work/os/lab_prog — /usr/bin/vim makefile

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

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

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

clean:
        -rm calcul *.o
```




Затем я использую команду make для создания файла выполнения и после этого выполняю его. Выполните команды GDB, чтобы проверить его выполнение


```

ganzl@fedora:~/work/os/lab_prog — /home/ganzl/work/os/lab_prog/calcul
[гanzl@fedora lab_prog]$ gcc calculate.o main.o -o calcul -lm
[гanzl@fedora lab_prog]$ gdb ./calcul
GNU gdb (GDB) Fedora 11.2-2.fc35
Copyright (C) 2022 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:
<https://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...
(gdb) run
Starting program: /home/ganzl/work/os/lab_prog/calcul
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Число: 123
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): +
Второе слагаемое: 123
246.00
[Inferior 1 (process 37622) exited normally]
(gdb)


```



```

(gdb) list
1      #include <stdio.h>
2      #include "calculate.h"
3
4      int main (void)
5      {
6          float Numeral;
7          char Operation[4];
8          float Result;
9          printf("Число: ");
10         scanf("%f",&Numeral);
(gdb) list 12,15
12         scanf("%s",&Operation);
13         Result = Calculate(Numeral, Operation);
14         printf("%0.2f\n",Result);
15         return 0;
(gdb) list calculate.c:20,29
20         return(Numeral * SecondNumeral);
21     }
22     else if(strncmp Operation, "/", 1) == 0)
23     {
24         printf("Делитель: ");
25         scanf("%f",&SecondNumeral);
26         if(SecondNumeral == 0)
27         {
28             printf("Ошибка: деление на ноль!");
29             return(HUGE_VAL);
(gdb) list calculate.c:20,27
20         return(Numeral * SecondNumeral);
21     }
22     else if(strncmp Operation, "/", 1) == 0)
23     {
24         printf("Делитель: ");
25         scanf("%f",&SecondNumeral);
26         if(SecondNumeral == 0)
27         {
(gdb) break 21
Breakpoint 1 at 0x401205: file calculate.c, line 22

```



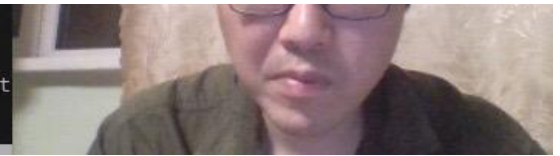

```

(gdb) info breakpoints
Num   Type           Disp Enb Address          What
1     breakpoint     keep y   0x000000000000401295 in Calculate at
(gdb)

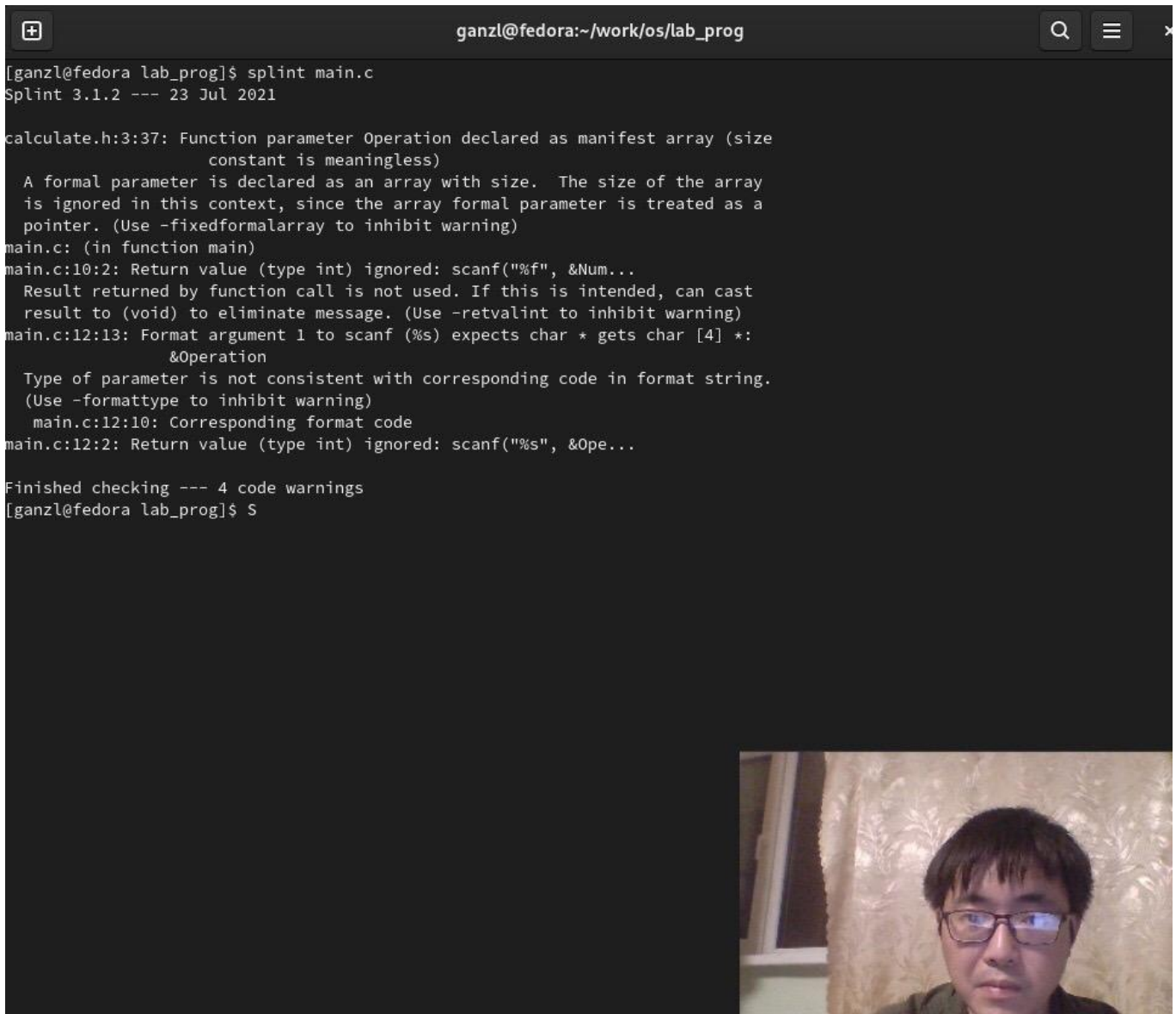
(gdb) run
Starting program: /home/ganzl/work/os/lab_prog/calcul
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Число: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -
Вычитаемое: 5
0.00
[Inferior 1 (process 4104) exited normally]
(gdb) list calculate.c:20,27
20         return Numeral * SecondNumeral;
21     }
22     else if(strncmp(Operation, "/", 1) == 0)
23     {
24         printf("Делитель: ");
25         scanf("%f",&SecondNumeral);
26         if(SecondNumeral == 0)
27         {
(gdb) run
Starting program: /home/ganzl/work/os/lab_prog/calcul
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Число: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): /

Breakpoint 1, Calculate (Numeral=5, Operation=0x7fffffffdf44 "/") at calculate.c:22
22     else if(strncmp(Operation, "/", 1) == 0)
(gdb) backtrace
#0 Calculate (Numeral=5, Operation=0x7fffffffdf44 "/") at calcula
#1 0x0000000000004014eb in main() at main.c:13
(gdb) print Numeral
$1 = 5
(gdb) display Numeral
1: Numeral = 5
(gdb) delete 1
(gdb) info breakpoints
No breakpoints or watchpoints.
(gdb)

```



Я использовал команду split для анализа файлов calculate.c и main.c.



```
ganzl@fedora:~/work/os/lab_prog
[ganzl@fedora lab_prog]$ splint main.c
Splint 3.1.2 --- 23 Jul 2021

calculate.h:3:37: 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:10:2: 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:12:13: 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:12:10: Corresponding format code
main.c:12:2: Return value (type int) ignored: scanf("%s", &Ope...

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



Контрольные вопросы:

1. Как получить информацию о возможностях программ gcc, make, gdb и др.?
 - --help
2. Назовите и дайте краткую характеристику основным этапам разработки приложений в UNIX.
планирование, включающее сбор и анализ требований к функционалу и другим характеристикам разрабатываемого приложения;
3. проектирование, включающее в себя разработку базовых алгоритмов и спецификаций, определение языка программирования;
4. непосредственная разработка приложения:
5. кодирование — по сути создание исходного текста программы (возможно в нескольких вариантах);
6. анализ разработанного кода;
7. сборка, компиляция и разработка исполняемого модуля;
8. тестирование и отладка, сохранение произведённых изменений;
9. документирование.
10. Что такое суффикс в контексте языка программирования? Приведите примеры использования.
 - *.c *.o *.h

11. Каково основное назначение компилятора языка C в UNIX?
 - Скомпилируйте файл
12. Для чего предназначена утилита make?
 - получите файл выполнения
13. Приведите пример структуры Makefile. Дайте характеристику основным элементам этого файла.
 - `calcul: calculate.o main.o` (Обязательно используйте файл после двоеточия, чтобы сгенерировать файл перед двоеточием)
 - `gcc calculate.o main.o -o calcul $(LIBS)` (Специальный код выполнения для создания файлов)
14. Назовите основное свойство, присущее всем программам отладки. Что необходимо сделать, чтобы его можно было использовать?
 - 'main' - это одно из значений его атрибута, что означает, что текущая программа выполняется непосредственно в этой программе, а не вызывается для запуска
15. Назовите и дайте основную характеристику основным командам отладчика gdb.
 - `backtrace`: вывод на экран пути к текущей точке останова (по сути вывод названий всех функций)
 - `break`: установить точку останова (в качестве параметра может быть указан номер строки или название функции)
 - `clear`: удалить все точки останова в функции
 - `continue`: продолжить выполнение программы
 - `delete`: удалить точку останова
 - `display`: добавить выражение в список выражений, значения которых отображаются при достижении точки останова программы
 - `finish`: выполнить программу до момента выхода из функции
 - `info breakpoints`: вывести на экран список используемых точек останова
 - `info watchpoints`: вывести на экран список используемых контрольных выражений
 - `list`: вывести на экран исходный код (в качестве параметра может быть указано название файла и через двоеточие номера начальной и конечной строк)
 - `next`: выполнить программу пошагово, но без выполнения вызываемых в программе функций
 - `print`: вывести значение указываемого в качестве параметра выражения
 - `run`: запуск программы на выполнение
 - `set`: установить новое значение переменной
 - `step`: пошаговое выполнение программы
 - `watch`: установить контрольное выражение, при изменении значения которого программа будет остановлена
16. Опишите по шагам схему отладки программы, которую Вы использовали при выполнении лабораторной работы.
 - Сначала я пишу код, затем использую `split`, чтобы определить, есть ли другая ошибка, затем `make` генерирует файл выполнения, а затем использую `gdb` для конкретного обнаружения после выполнения.
17. Прокомментируйте реакцию компилятора на синтаксические ошибки в программе при его первом запуске.
 - Выведите сообщение об ошибке и остановите компиляцию
18. Назовите основные средства, повышающие понимание исходного кода программы.
 - Gdb
19. Каковы основные задачи, решаемые программой `split`?

- Обнаружение ошибок в коде и вывод подробной информации

Вывод:

- Я научился простейшим навыкам разработки, анализа, тестирования и отладки приложений в таких операционных системах, как UNIX/Linux.