

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

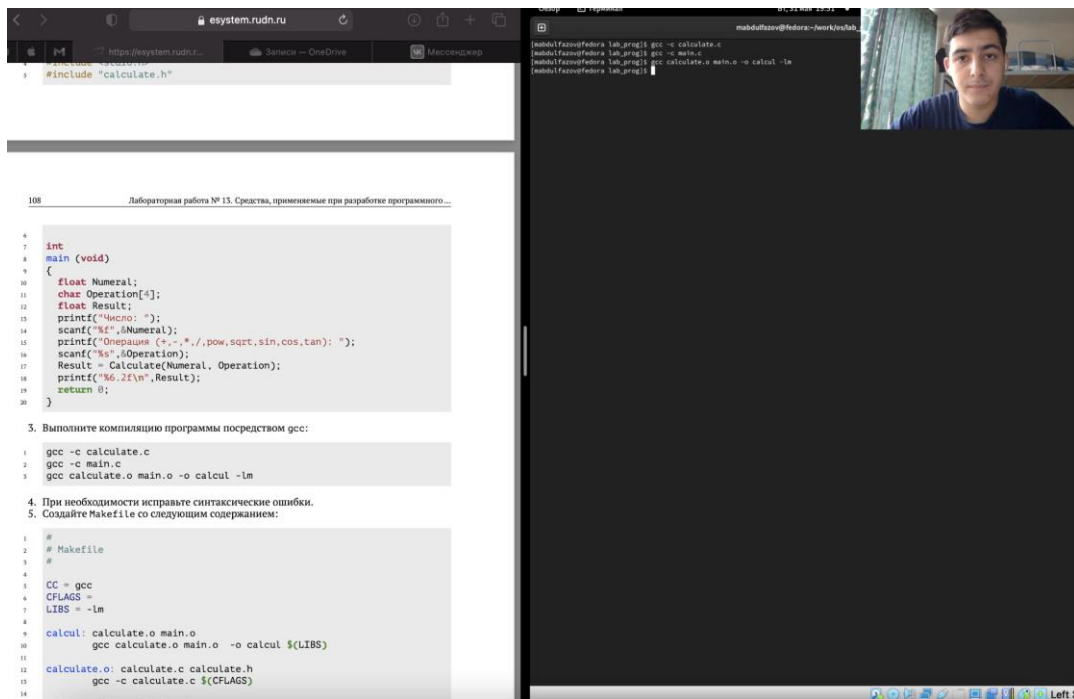
Мансур А. о. Абдулфазов Группа НФИбд-01-21

# 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.

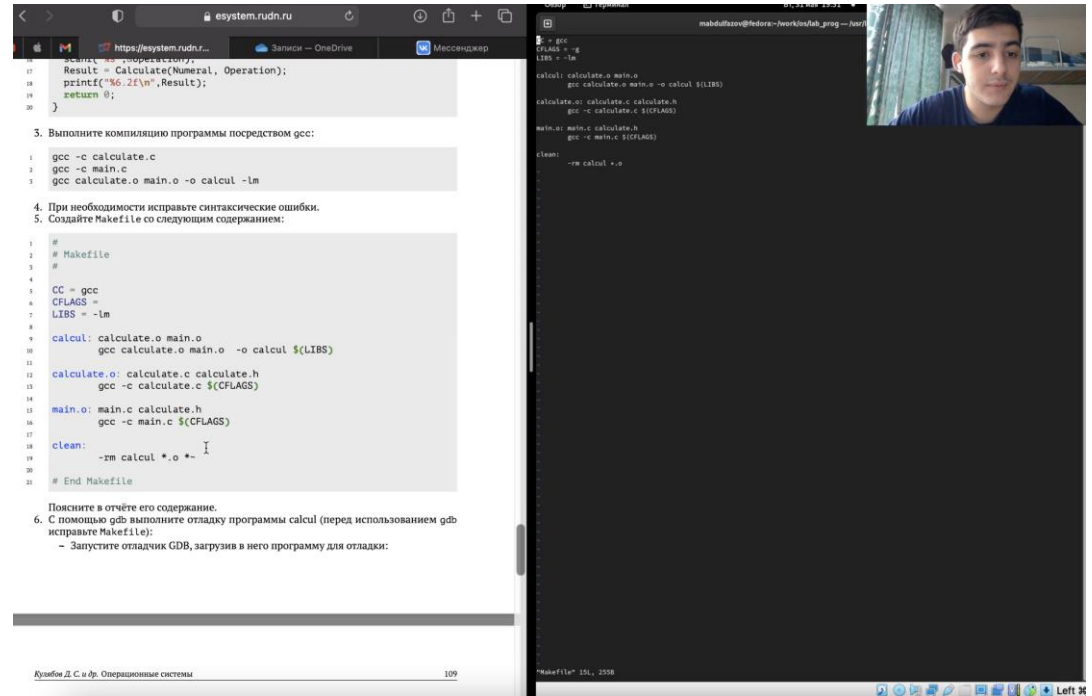
## Выполнение лабораторной работы

1. Скомпилировал все созданные файлы(данные по заданию в лабораторной работе) (рис. 1)



Компиляция файлов

## 2. Исправил синтаксические ошибки в файле Makefile (рис. 2)



The screenshot displays a video lecture interface. On the left, a web browser shows a code editor with a C function `Calculate` and a `Makefile`. The `Makefile` defines targets for `calcul`, `calculate.o`, `main.o`, and a `clean` target. On the right, a terminal window shows the execution of `gcc` commands to compile the program. A small video inset in the top right corner shows the lecturer.

```
17     Result = Calculate(Numeral, Operation);
18     printf("%6.2f\n", Result);
19     return 0;
20 }
```

3. Выполните компиляцию программы посредством gcc:

```
1 gcc -c calculate.c
2 gcc -c main.c
3 gcc calculate.o main.o -o calcul -lm
```

4. При необходимости исправьте синтаксические ошибки.

5. Создайте Makefile со следующим содержанием:

```
1 #
2 # Makefile
3 #
4
5 CC = gcc
6 CFLAGS =
7 LIBS = -lm
8
9 calcul: calculate.o main.o
10     gcc calculate.o main.o -o calcul $(LIBS)
11
12 calculate.o: calculate.c calculate.h
13     gcc -c calculate.c $(CFLAGS)
14
15 main.o: main.c calculate.h
16     gcc -c main.c $(CFLAGS)
17
18 clean:
19     -rm calcul *.o *~
20
21 # End Makefile
```

Поясните в отчёте его содержание.

6. С помощью gdb выполните отладку программы calcul (перед использованием gdb исправьте Makefile):

- Запустите отладчик GDB, загрузив в него программу для отладки:

```
1 # gcc
2 CFLAGS = -g
3 LIBS = -lm
4
5 calcul: calculate.o main.o
6     gcc calculate.o main.o -o calcul $(LIBS)
7
8 calcul.o: calculate.c calculate.h
9     gcc -c calculate.c $(CFLAGS)
10
11 main.o: main.c calculate.h
12     gcc -c main.c $(CFLAGS)
13
14 clean:
15     -rm calcul *.o
```

Курсов Д.С. и др. Операционные системы 109

"Makefile" 151, 2558

## Исправление ошибок

3. Проверил работу калькулятора с помощью отладчика gdb (рис. 3)

eysystem.ruda.ru

Запустить Закрыть Оформить Настроить

Мессенджер

6. С помощью gdb выполните отладку программы calcul (перед использованием gdb испишите Makefile):

- Запустите отладчик GDB, загрузив в него программу для отладки:

```
gdb ./calcul
```

Для запуска программы внутри отладчика введите команду run:

```
run
```

Для двустороннего (по 9 строк) просмотра исходного кода используйте команду list:

```
list
```

Для просмотра строк с 12 по 15 основного файла используйте list с параметрами:

```
list 12,15
```

Для просмотра определённых строк не основного файла используйте list с параметрами:

```
list calculate.c:20,29
```

Установите точку останова в файле calculate.c на строке номер 21:

```
list calculate.c:20,27
break 21
```

Выведите информацию об имеющихся в проекте точке останова:

```
info breakpoints
```

Запустите программу внутри отладчика и убедитесь, что программа остановится в момент прохождения точки останова:

```
run
```

```

mabdufazz@fedora-lab-prng1 grc -c calculate.c
mabdufazz@fedora-lab-prng1 grc -c main.c
mabdufazz@fedora-lab-prng1 grc calculate.c main.c -o calcul -la
mabdufazz@fedora-lab-prng1
mabdufazz@fedora-lab-prng1 vi Makefile
mabdufazz@fedora-lab-prng1 gdb ./calcul
mabdufazz@fedora-lab-prng1
Copyright (C) 2021 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...
(no debugging symbols found in ./calcul)
(gdb) run
Starting program: /home/mabdufazz/work/lab_prog/calcul
Downloading separate debug info for /home/mabdufazz/work/lab_prog/system-supplied-000 at 0x7ffff7c00000...
Downloading separate debug info for /lib64/libc.so.6...
Downloading separate debug info for /lib64/libc-2.34-7.fc35.x86_64 elf/shlib.so...
Using host libthread_db library "/lib64/libthread_db.so.1".
Main thread:
Dereference 0x7ffff7c00000 at 0x7ffff7c00000:
Breakpoint 4 at 0x7ffff7c00000:
[Inferior 1 (process 25237) exited normally.]
(gdb) list
1  /usr/src/debug/glibc-2.34-7.fc35.x86_64/elf/shlib.so: RunTimeError.
2  list 12,15
3
4
5
6
7
8
9
10
11
12  /usr/src/debug/glibc-2.34-7.fc35.x86_64/elf/shlib.so: Dereference 0x7ffff7c00000.
13
14
15  /usr/src/debug/glibc-2.34-7.fc35.x86_64/elf/shlib.so:
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735

```

## Проверка работы калькулятора

#### 4. Проанализировал файлы calculate.c и main.c с помощью утилиты splint (рис. 4)

а команда `bash` покажет весь стек вызываемых функций от начала программы до текущего места.

- Посмотрите, чему равно на этом этапе значение переменной `Numeral`, введя:

```
1 print Numeral
```

На экран должно быть выведено число 5.

- Сравните с результатом вывода на экран после использования команды:

```
1 display Numeral
```

- Уберите точки останова:

```
1 info breakpoints
1 delete 1
```

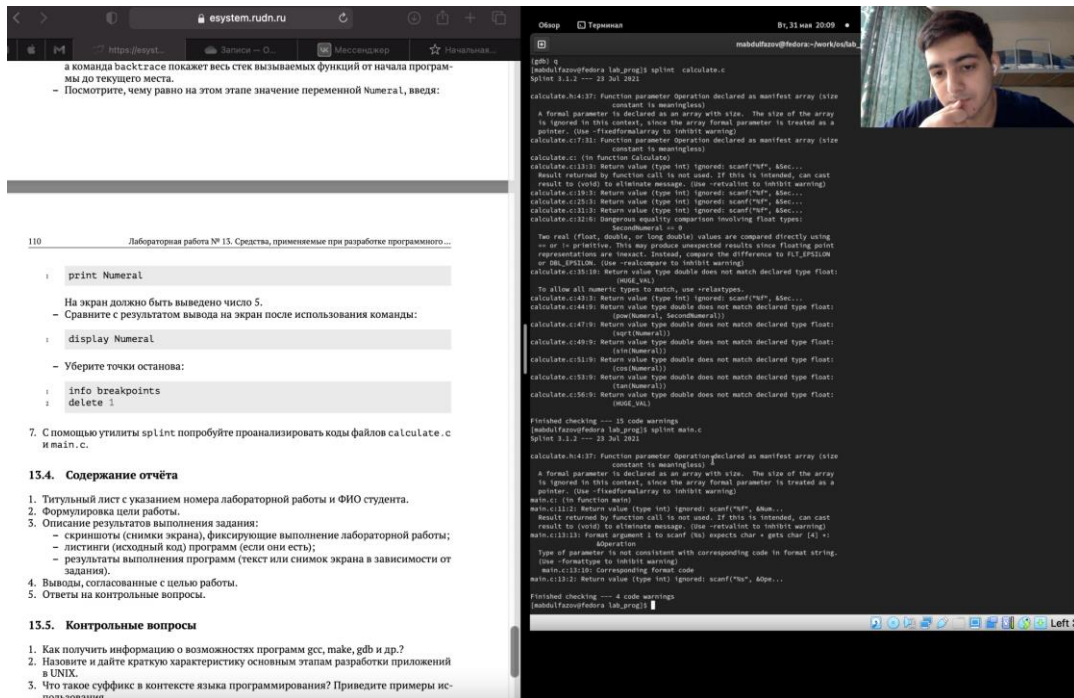
7. С помощью утилиты `splint` попробуйте проанализировать коды файлов `calculate.c` и `main.c`.

### 13.4. Содержание отчёта

- Титульный лист с указанием номера лабораторной работы и ФИО студента.
- Формулировка цели работы.
- Описание результатов выполнения задания:
  - скриншоты (снимки экрана), фиксирующие выполнение лабораторной работы;
  - листинги (исходный код) программ (если они есть);
  - результаты выполнения программ (текст или снимок экрана в зависимости от задания).
- Выводы, согласованные с целью работы.
- Ответы на контрольные вопросы.

### 13.5. Контрольные вопросы

- Как получить информацию о возможностях программ `gcc`, `make`, `gdb` и др.?
- Назовите и дайте краткую характеристику основным этапам разработки приложений в UNIX.
- Что такое суффикс в контексте языка программирования? Приведите примеры использования.



## Анализ файлов

# Вывод

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