

# Работа на ЭВМ и программирование (группа 112)

Занятие 9

# Контактная информация

- Шундеев Александр Сергеевич
- [alex.shundeev@gmail.com](mailto:alex.shundeev@gmail.com)
- <http://group112.github.io/sem1.html>

# Электронная почта

- Тема письма

- 112 Фамилия Имя Отчество
- 112 Фамилия Имя

- Пример

- 112 Иванов Иван Иванович
- 112 Иванов Иван

Указатель на функцию

# Объявление переменной

```
int (*p1)(void);
```

- p1 - переменная типа указатель на функцию (адрес) без входных параметров, возвращающую значение типа int

```
double (*p2)(double);
```

- p2 - переменная типа указатель на функцию (адрес) с входным параметром типа double, возвращающую значение типа double

```
int (*p3)(double, int);
```

- p3 - переменная типа указатель на функцию (адрес) с двумя входными параметрами типа double и int, возвращающую значение типа int

# Пример

```
int f1(int x, int y)
{
    return x + y;
}
```

```
int f2(int x, int y)
{
    return x * y;
}
```

# Пример

```
int main(void)
{
    int r;
    int (*p)(int, int);

    p = f1;
    r = p(1, 2);
    printf("%d\n", r);

    p = f2;
    r = p(1, 2);
    printf("%d\n", r);

    return 0;
}
```

# Пример

```
int main(void)
{
    int r;
    int (*p)(int, int);

    p = f1;
    r = p(1, 2);
    printf("%d\n", r);

    p = f2;
    r = p(1, 2);
    printf("%d\n", r);

    return 0;
}
```

```
$ ./prog
3
2
```



# Задача на сортировку №3

(передача указателя на функцию сравнения)

# Алгоритм сортировки и отношение порядка

Алгоритм сортировки (в большинстве случаев) не зависит от определения используемого отношения порядка

Естественный порядок  $<$  может быть заменен на другой  $<$

## Пример альтернативного отношения

- Разобьем множество целых чисел на две части:  $X_1$  - простые числа,  $X_2$  - остальные (составные, 1, 0, отрицательные). Считаем, что любой элемент из  $X_1$  «больше» любого элемента из  $X_2$ . Внутри  $X_1$  ( $X_2$ ) элементы сравниваются с помощью естественного порядка.

# Функции сравнения (определение)

## Прототип

■ `int cmp(T x, T y);` (T - `int` или `double`)

## Интерпретация возвращаемого значения

- `< 0`                    если  $x < y$
- `== 0`                    если  $x = y$
- `> 0`                    если  $x > y$

# Функция сравнения (естественный порядок)

```
int cmp(int x, int y)
{
    return x - y;
}
```

# Функция сравнения (использование)

```
int sort(int *a, int n)
{
    ...
    x < y
    ...
}

...

sort(a, n);
```

```
int sort(int (*cmp)(int, int), int *a, int n)
{
    ...
    cmp(x, y) < 0
    ...
}

...

sort(cmpA, a, n);
```

# Структура программы

```
...
void sort(int (*cmp)(int, int), int *a, int n);
int cmpA(int x, int y);
int cmpB(int x, int y);

int main(void)
{
    ...
    sort(cmpA, a, n);
    ...
    sort(cmpB, a, n);
    ...
    return 0;
}
```

```
void sort(int (*cmp)(int, int), int *a, int n);
int cmpA(int x, int y);
int cmpB(int x, int y);

void sort(int (*cmp)(int, int), int *a, int n)
{
    ...
}

int cmpA(int x, int y)
{
    return x - y;
}

int cmpB(int x, int y)
{
    ...
}
```

# Запуск программы

```
$ ./prog
```

# Запуск программы

\$ ./prog

Input array: 5 5 1 4 3 2

Ввод массива (сначала размер, потом элементы)



# Запуск программы

\$ ./prog

Input array: 5 5 1 4 3 2

Input cmp (a/b): a

Выбор функции сравнения

(a - естественный порядок, b - альтернативный)

# Запуск программы

\$ ./prog

Input array: 5 5 1 4 3 2

Input cmp (a/b): a

1 2 3 4 5

# Запуск программы

\$ ./prog

Input array: 5 5 1 4 3 2

Input cmp (a/b): a

1 2 3 4 5

\$ ./prog

Input array: 5 5 1 4 3 2

Input cmp (a/b): b

1 4 2 3 5

# Отладчик gdb

[http://group112.github.io/doc/sem1/2018/2018\\_sem1\\_gdb.pdf](http://group112.github.io/doc/sem1/2018/2018_sem1_gdb.pdf)

```
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 double average(double *a, int n);
5
6 int main(void) {
7     FILE *fi, *fo;
8     double *a, x, r;
9     int n, i;
10
11     fi = fopen("input.txt", "r");
12     if(!fi)
13     {
14         fprintf(stderr, "Can't open input.txt ...\n");
15         return -1;
16     }
17
18     if(fscanf(fi, "%d", &n) != 1)
19     {
20         fprintf(stderr, "Can't read parameter `n` ...\n");
21         return -1;
22     }
23
24     if(n < 1)
25     {
26         fprintf(stderr, "Wrong parameter `n` ...\n");
27         return -1;
28     }
29
30     a = malloc(sizeof(double) * n);
31     if(!a)
32     {
33         fprintf(stderr, "Memory allocation error ...\n");
34         return -1;
```

```
34     return -1;
35 }
36
37 for(i = 0; i < n; i++)
38     if(fscanf(fi, "%lf", a + i) != 1)
39     {
40         fprintf(stderr, "Can't read element a[%d] ...\n", i);
41         return -1;
42     }
43
44 if(fscanf(fi, "%lf", &x) != EOF)
45 {
46     fprintf(stderr, "Wrong input data ...\n");
47     return -1;
48 }
49
50 fclose(fi);
51
52 r = average(a, n);
53
54 fo = fopen("output.txt", "w");
55 if(!fo)
56 {
57     fprintf(stderr, "Can't open output.txt ...\n");
58     return -1;
59 }
60
61 fprintf(fo, "%lf", r);
62
63 fclose(fo);
64
65 return 0;
66 }
67
```

```
1 #include <stdio.h>
2
3 double average(double *a, int n);
4
5 double average(double *a, int n)
6 {
7     double s = 0.;
8     int i;
9
10    for(i = 0; i < n; i++)
11        s += a[i];
12
13    return s / n;
14 }
15
```

# Сценарий 1

(исследование поведения программы)



```
Терминал - student@group112: ~/112
Файл  Правка  Вид  Терминал  Вкладки  Справка
student@group112:~/112$ gcc -g -Wall -Wextra -Wfloat-equal -Werror -pedantic -std=c99 main.c average.c -o prog
student@group112:~/112$ echo "5 1 2 3 4 5" > input.txt
student@group112:~/112$ gdb -q prog
Reading symbols from prog...
(gdb) █
```

Использование ключа **-g** в команде компиляции

```
Терминал - student@group112: ~/112
Файл Правка Вид Терминал Вкладки Справка
student@group112:~/112$ gcc -g -Wall -Wextra -Wfloat-equal -Werror -pedantic -std=c99 main.c average.c -o prog
student@group112:~/112$ echo "5 1 2 3 4 5" > input.txt
student@group112:~/112$ gdb -q prog
Reading symbols from prog...
(gdb) b main.c:24
Breakpoint 1 at 0x12b7: file main.c, line 24.
(gdb) b main.c:44
Breakpoint 2 at 0x139c: file main.c, line 44.
(gdb) █
```

Создание точки остановки: **b** <имя файла>:<номер строки>

```
Терминал - student@group112: ~/112
Файл Правка Вид Терминал Вкладки Справка
student@group112:~/112$ gcc -g -Wall -Wextra -Wfloat-equal -Werror -pedantic -std=c99 main.c average.c -o prog
student@group112:~/112$ echo "5 1 2 3 4 5" > input.txt
student@group112:~/112$ gdb -q prog
Reading symbols from prog...
(gdb) b main.c:24
Breakpoint 1 at 0x12b7: file main.c, line 24.
(gdb) b main.c:44
Breakpoint 2 at 0x139c: file main.c, line 44.
(gdb) i b
Num      Type      Disp Enb Address      What
1        breakpoint keep y   0x00000000000012b7 in main at main.c:24
2        breakpoint keep y   0x000000000000139c in main at main.c:44
(gdb) d 2
(gdb) i b
Num      Type      Disp Enb Address      What
1        breakpoint keep y   0x00000000000012b7 in main at main.c:24
(gdb) b main.c:44
Breakpoint 3 at 0x139c: file main.c, line 44.
(gdb) i b
Num      Type      Disp Enb Address      What
1        breakpoint keep y   0x00000000000012b7 in main at main.c:24
3        breakpoint keep y   0x000000000000139c in main at main.c:44
(gdb) █
```

Просмотр созданных точек остановки: **i b**

Удаление: **d <номер>**

```
Терминал - student@group112: ~/112
Файл Правка Вид Терминал Вкладки Справка
student@group112:~/112$ gcc -g -Wall -Wextra -Wfloat-equal -Werror -pedantic -std=c99 main.c average.c -o prog
student@group112:~/112$ echo "5 1 2 3 4 5" > input.txt
student@group112:~/112$ gdb -q prog
Reading symbols from prog...
(gdb) b main.c:24
Breakpoint 1 at 0x12b7: file main.c, line 24.
(gdb) b main.c:44
Breakpoint 2 at 0x139c: file main.c, line 44.
(gdb) i b
Num      Type      Disp Enb Address      What
1        breakpoint  keep y  0x00000000000012b7 in main at main.c:24
2        breakpoint  keep y  0x000000000000139c in main at main.c:44
(gdb) d 2
(gdb) i b
Num      Type      Disp Enb Address      What
1        breakpoint  keep y  0x00000000000012b7 in main at main.c:24
(gdb) b main.c:44
Breakpoint 3 at 0x139c: file main.c, line 44.
(gdb) i b
Num      Type      Disp Enb Address      What
1        breakpoint  keep y  0x00000000000012b7 in main at main.c:24
3        breakpoint  keep y  0x000000000000139c in main at main.c:44
(gdb) r
Starting program: /home/student/112/prog

Breakpoint 1, main () at main.c:24
24      if(n < 1)
(gdb) █
```

Запуск программы: **r**

```
Терминал - student@group112: ~/112
Файл Правка Вид Терминал Вкладки Справка
student@group112:~/112$ gcc -g -Wall -Wextra -Wfloat-equal -Werror -pedantic -std=c99 main.c average.c -o prog
student@group112:~/112$ echo "5 1 2 3 4 5" > input.txt
student@group112:~/112$ gdb -q prog
Reading symbols from prog...
(gdb) b main.c:24
Breakpoint 1 at 0x12b7: file main.c, line 24.
(gdb) b main.c:44
Breakpoint 2 at 0x139c: file main.c, line 44.
(gdb) i b
Num      Type      Disp Enb Address      What
1        breakpoint  keep y  0x00000000000012b7 in main at main.c:24
2        breakpoint  keep y  0x000000000000139c in main at main.c:44
(gdb) d 2
(gdb) i b
Num      Type      Disp Enb Address      What
1        breakpoint  keep y  0x00000000000012b7 in main at main.c:24
(gdb) b main.c:44
Breakpoint 3 at 0x139c: file main.c, line 44.
(gdb) i b
Num      Type      Disp Enb Address      What
1        breakpoint  keep y  0x00000000000012b7 in main at main.c:24
3        breakpoint  keep y  0x000000000000139c in main at main.c:44
(gdb) r
Starting program: /home/student/112/prog

Breakpoint 1, main () at main.c:24
24      if(n < 1)
(gdb) p n
$1 = 5
(gdb) p n*n + 1
$2 = 26
(gdb) █
```

Просмотр значения выражения: **p** <выражение>

```
student@group112:~/112$ gdb -q prog
Reading symbols from prog...
(gdb) b main.c:24
Breakpoint 1 at 0x12b7: file main.c, line 24.
(gdb) b main.c:44
Breakpoint 2 at 0x139c: file main.c, line 44.
(gdb) i b
Num      Type      Disp Enb Address      What
1        breakpoint keep y   0x00000000000012b7 in main at main.c:24
2        breakpoint keep y   0x000000000000139c in main at main.c:44
(gdb) d 2
(gdb) i b
Num      Type      Disp Enb Address      What
1        breakpoint keep y   0x00000000000012b7 in main at main.c:24
(gdb) b main.c:44
Breakpoint 3 at 0x139c: file main.c, line 44.
(gdb) i b
Num      Type      Disp Enb Address      What
1        breakpoint keep y   0x00000000000012b7 in main at main.c:24
3        breakpoint keep y   0x000000000000139c in main at main.c:44
(gdb) r
Starting program: /home/student/112/prog

Breakpoint 1, main () at main.c:24
24      if(n < 1)
(gdb) p n
$1 = 5
(gdb) p n*n + 1
$2 = 26
(gdb) c
Continuing.

Breakpoint 3, main () at main.c:44
44      if(fscanf(fi, "%lf", &x) != EOF)
(gdb) █
```

Продолжить выполнение программы: **c**

```
(gdb) b main.c:44
Breakpoint 2 at 0x139c: file main.c, line 44.
(gdb) i b
Num      Type      Disp Enb Address      What
1        breakpoint keep y   0x00000000000012b7 in main at main.c:24
2        breakpoint keep y   0x000000000000139c in main at main.c:44
(gdb) d 2
(gdb) i b
Num      Type      Disp Enb Address      What
1        breakpoint keep y   0x00000000000012b7 in main at main.c:24
(gdb) b main.c:44
Breakpoint 3 at 0x139c: file main.c, line 44.
(gdb) i b
Num      Type      Disp Enb Address      What
1        breakpoint keep y   0x00000000000012b7 in main at main.c:24
3        breakpoint keep y   0x000000000000139c in main at main.c:44
(gdb) r
Starting program: /home/student/112/prog

Breakpoint 1, main () at main.c:24
24      if(n < 1)
(gdb) p n
$1 = 5
(gdb) p n*n + 1
$2 = 26
(gdb) c
Continuing.

Breakpoint 3, main () at main.c:44
44      if(fscanf(fi, "%lf", &x) != EOF)
(gdb) p a[0]
$3 = 1
(gdb) p a[0] + a[1]
$4 = 3
(gdb) █
```

```
Терминал - student@group112: ~/112
Файл  Правка  Вид  Терминал  Вкладки  Справка
1      breakpoint      keep y    0x000000000000012b7 in main at main.c:24
2      breakpoint      keep y    0x0000000000000139c in main at main.c:44
(gdb) d 2
(gdb) i b
Num      Type          Disp Enb Address          What
1        breakpoint      keep y    0x000000000000012b7 in main at main.c:24
(gdb) b main.c:44
Breakpoint 3 at 0x139c: file main.c, line 44.
(gdb) i b
Num      Type          Disp Enb Address          What
1        breakpoint      keep y    0x000000000000012b7 in main at main.c:24
3        breakpoint      keep y    0x0000000000000139c in main at main.c:44
(gdb) r
Starting program: /home/student/112/prog

Breakpoint 1, main () at main.c:24
24      if(n < 1)
(gdb) p n
$1 = 5
(gdb) p n*n + 1
$2 = 26
(gdb) c
Continuing.

Breakpoint 3, main () at main.c:44
44      if(fscanf(fi, "%lf", &x) != EOF)
(gdb) p a[0]
$3 = 1
(gdb) p a[0] + a[1]
$4 = 3
(gdb) x/5fg a
0x55555555a490: 1      2
0x55555555a4a0: 3      4
0x55555555a4b0: 5
(gdb) █
```

Просмотр массива: `x/<количество элементов><тип элемента><размер элемента> <адрес>`



```
Терминал - student@group112: ~/112
Файл  Правка  Вид  Терминал  Вкладки  Справка
1      breakpoint      keep y    0x000000000000012b7 in main at main.c:24
2      breakpoint      keep y    0x0000000000000139c in main at main.c:44
(gdb) d 2
(gdb) i b
Num      Type          Disp Enb Address          What
1        breakpoint    keep y    0x000000000000012b7 in main at main.c:24
(gdb) b main.c:44
Breakpoint 3 at 0x139c: file main.c, line 44.
(gdb) i b
Num      Type          Disp Enb Address          What
1        breakpoint    keep y    0x000000000000012b7 in main at main.c:24
3        breakpoint    keep y    0x0000000000000139c in main at main.c:44
(gdb) r
Starting program: /home/student/112/prog

Breakpoint 1, main () at main.c:24
24      if(n < 1)
(gdb) p n
$1 = 5
(gdb) p n*n + 1
$2 = 26
(gdb) c
Continuing.

Breakpoint 3, main () at main.c:44
44      if(fscanf(fi, "%lf", &x) != EOF)
(gdb) p a[0]
$3 = 1
(gdb) p a[0] + a[1]
$4 = 3
(gdb) x/5fg a
0x55555555a490: 1      2
0x55555555a4a0: 3      4
0x55555555a4b0: 5
(gdb) █
```

Тип элемента: **f** - числа с плавающей точкой, **d** - знаковые целые числа

```

Num      Type      Disp Enb Address      What
1        breakpoint keep y   0x00000000000012b7 in main at main.c:24
(gdb) b main.c:44
Breakpoint 3 at 0x139c: file main.c, line 44.
(gdb) i b
Num      Type      Disp Enb Address      What
1        breakpoint keep y   0x00000000000012b7 in main at main.c:24
3        breakpoint keep y   0x000000000000139c in main at main.c:44
(gdb) r
Starting program: /home/student/112/prog

Breakpoint 1, main () at main.c:24
24          if(n < 1)
(gdb) p n
$1 = 5
(gdb) p n*n + 1
$2 = 26
(gdb) c
Continuing.

Breakpoint 3, main () at main.c:44
44          if(fscanf(fi, "%lf", &x) != EOF)
(gdb) p a[0]
$3 = 1
(gdb) p a[0] + a[1]
$4 = 3
(gdb) x/5fg a
0x55555555a490: 1      2
0x55555555a4a0: 3      4
0x55555555a4b0: 5
(gdb) c
Continuing.
[Inferior 1 (process 2762) exited normally]
(gdb) q
student@group112:~/112$

```

Размер элемента: **g** - 8 байт, **w** - 4 байта

# Сценарий 2

(выявление причины аварийного завершения программы)

```

1 #include <stdio.h>
2
3 double average(double *a, int n);
4
5 double average(double *a, int n)
6 {
7     double s = 0.;
8     int i;
9
10    a[n + 100000] = 1;
11
12    for(i = 0; i < n; i++)
13        s += a[i];
14
15    return s / n;
16 }
17

```

"average.c" 17L, 214C записано

17,0-1

Весь

## Моделирование ошибки SEGMENTATION FAULT (строка 10)

```
student@group112:~/112$ gcc -g -Wall -Wextra -Wfloat-equal -Werror -pedantic -std=c99 main.c average.c -o prog
```

```
student@group112:~/112$ ./prog
```

Ошибка сегментирования (стек памяти сброшен на диск)

```
student@group112:~/112$ █
```

```
student@group112:~/112$ gcc -g -Wall -Wextra -Wfloat-equal -Werror -pedantic -std=c99 main.c average.c -o prog
```

```
student@group112:~/112$ ./prog
```

Ошибка сегментирования (стек памяти сброшен на диск)

```
student@group112:~/112$ gdb -q prog
```

Reading symbols from prog...

(gdb) r

Starting program: /home/student/112/prog

Program received signal SIGSEGV, Segmentation fault.

0x00005555555554d4 in average (a=0x55555555a490, n=5) at average.c:10

10           a[n + 1000000] = 1;

(gdb) █

```
student@group112:~/112$ gcc -g -Wall -Wextra -Wfloat-equal -Werror -pedantic -std=c99 main.c average.c -o prog
student@group112:~/112$ ./prog
Ошибка сегментирования (стек памяти сброшен на диск)
student@group112:~/112$ gdb -q prog
Reading symbols from prog...
(gdb) r
Starting program: /home/student/112/prog

Program received signal SIGSEGV, Segmentation fault.
0x0000555555554d4 in average (a=0x55555555a490, n=5) at average.c:10
10      a[n + 100000] = 1;
(gdb) bt
#0  0x0000555555554d4 in average (a=0x55555555a490, n=5) at average.c:10
#1  0x000055555555404 in main () at main.c:52
(gdb) up
#1  0x000055555555404 in main () at main.c:52
52      r = average(a, n);
(gdb) down
#0  0x0000555555554d4 in average (a=0x55555555a490, n=5) at average.c:10
10      a[n + 100000] = 1;
(gdb) █
```

Просмотр стека вызовов функций: `bt`

```
Терминал - student@group112: ~/112
Файл Правка Вид Терминал Вкладки Справка
student@group112:~/112$ gcc -g -Wall -Wextra -Wfloat-equal -Werror -pedantic -std=c99 main.c average.c -o prog
student@group112:~/112$ ./prog
Ошибка сегментирования (стек памяти сброшен на диск)
student@group112:~/112$ gdb -q prog
Reading symbols from prog...
(gdb) r
Starting program: /home/student/112/prog

Program received signal SIGSEGV, Segmentation fault.
0x0000555555554d4 in average (a=0x55555555a490, n=5) at average.c:10
10      a[n + 100000] = 1;
(gdb) bt
#0  0x0000555555554d4 in average (a=0x55555555a490, n=5) at average.c:10
#1  0x000055555555404 in main () at main.c:52
(gdb) up
#1  0x000055555555404 in main () at main.c:52
52      r = average(a, n);
(gdb) down
#0  0x0000555555554d4 in average (a=0x55555555a490, n=5) at average.c:10
10      a[n + 100000] = 1;
(gdb) █
```

Переход «вверх» («вниз») по стеку вызовов функций: `up` ( `down` )



```
student@group112:~/112$ gcc -g -Wall -Wextra -Wfloat-equal -Werror -pedantic -std=c99 main.c average.c -o prog
student@group112:~/112$ ./prog
Ошибка сегментирования (стек памяти сброшен на диск)
student@group112:~/112$ gdb -q prog
Reading symbols from prog...
(gdb) r
Starting program: /home/student/112/prog

Program received signal SIGSEGV, Segmentation fault.
0x0000555555554d4 in average (a=0x55555555a490, n=5) at average.c:10
10      a[n + 100000] = 1;
(gdb) bt
#0  0x0000555555554d4 in average (a=0x55555555a490, n=5) at average.c:10
#1  0x000055555555404 in main () at main.c:52
(gdb) up
#1  0x000055555555404 in main () at main.c:52
52      r = average(a, n);
(gdb) down
#0  0x0000555555554d4 in average (a=0x55555555a490, n=5) at average.c:10
10      a[n + 100000] = 1;
(gdb) c
Continuing.

Program terminated with signal SIGSEGV, Segmentation fault.
The program no longer exists.
(gdb) q
student@group112:~/112$
```