Лекция 3

1. **Средства разработки программ в *Linux*:**
   Редактор *vim*, компилятор *gcc*, отладчик *gdb*.
2. **Создание процессов в *Linux*:**
   Системный вызов *fork*.

Редактор *vim* (“ESC :q!” 😄)

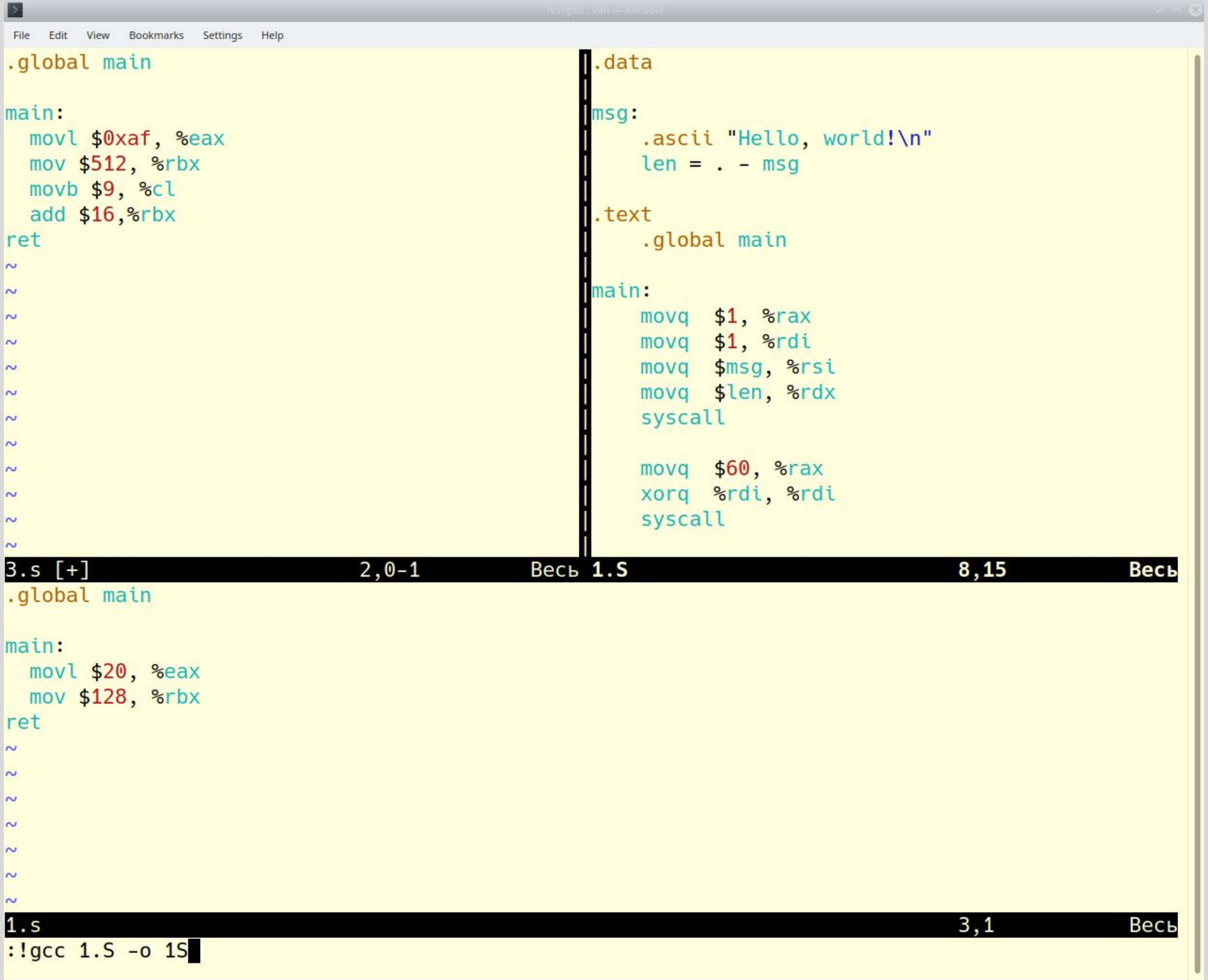
```
malkov@192:~> vimtutor

=    W e l c o m e   t o   t h e   V I M   T u t o r    -    Version 1.7      =
===========================================================================

.........................................................................
                    Lesson 1.1:  MOVING THE CURSOR



  ** To move the cursor, press the h,j,k,l keys as indicated. **
          ^
          k              Hint:  The h key is at the left and moves left.
     < h       l >              The l key is at the right and moves right.
          j                    The j key looks like a down arrow.
          v
.........................................................................
```



```
.global main

main:
  movl $0xaf, %eax
  mov $512, %rbx
  movb $9, %cl
  add $16,%rbx
ret
~
~
~
~
~
~
~
~
~
~
~
~
3.s [+]                          2,0-1          Весь
```

```
.data

msg:
    .ascii "Hello, world!\n"
    len = . - msg

.text
    .global main

main:
    movq  $1, %rax
    movq  $1, %rdi
    movq  $msg, %rsi
    movq  $len, %rdx
    syscall

    movq  $60, %rax
    xorq  %rdi, %rdi
    syscall
1.S                              8,15           Весь
```

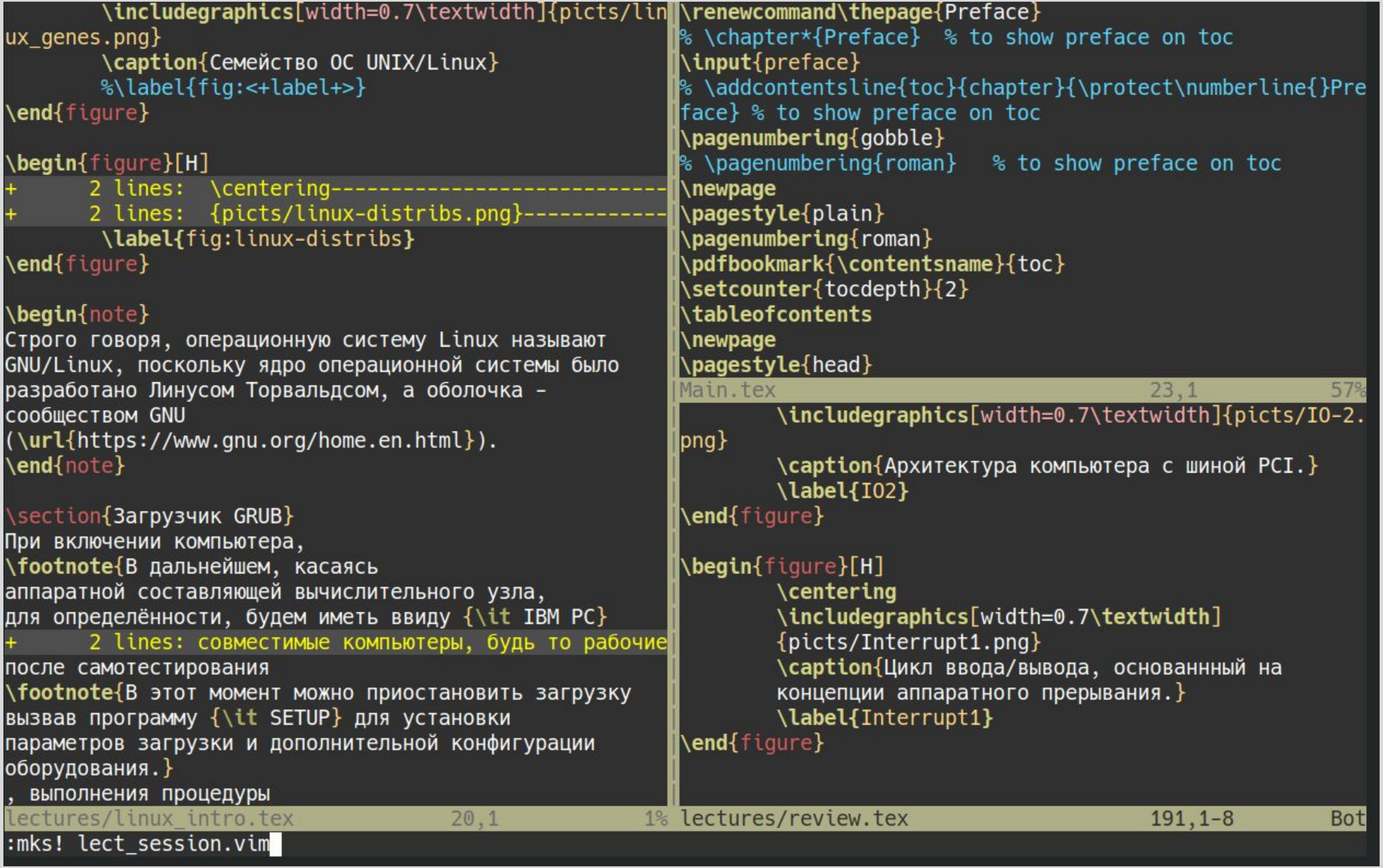
```
.global main

main:
  movl $20, %eax
  mov $128, %rbx
ret
~
~
~
~
~
~
~
1.s                                                           3,1            Весь
:!gcc 1.S -o 1S
```

~> vim 1.s
:split 2.s
:set splitright
:vsplit
CTRL+w w
:!gcc 1.S -o 1S
:!./1S

Hello, world!
Нажмите ENTER или введите команду для продолжения

:wa
:qa



```
        \includegraphics[width=0.7\textwidth]{picts/lin
ux_genes.png}
        \caption{Семейство ОС UNIX/Linux}
        %\label{fig:<+label+>}
\end{figure}

\begin{figure}[H]
+       2 lines:  \centering---------------------------
+       2 lines:  {picts/linux-distribs.png}-----------
        \label{fig:linux-distribs}
\end{figure}

\begin{note}
Строго говоря, операционную систему Linux называют
GNU/Linux, поскольку ядро операционной системы было
разработано Линусом Торвальдсом, а оболочка -
сообществом GNU
(\url{https://www.gnu.org/home.en.html}).
\end{note}

\section{Загрузчик GRUB}
При включении компьютера,
\footnote{В дальнейшем, касаясь
аппаратной составляющей вычислительного узла,
для определённости, будем иметь ввиду {\it IBM PC}
+       2 lines: совместимые компьютеры, будь то рабочие
после самотестирования
\footnote{В этот момент можно приостановить загрузку
вызвав программу {\it SETUP} для установки
параметров загрузки и дополнительной конфигурации
оборудования.}
, выполнения процедуры
lectures/linux_intro.tex                    20,1           1%
```

```
\renewcommand\thepage{Preface}
% \chapter*{Preface}  % to show preface on toc
\input{preface}
% \addcontentsline{toc}{chapter}{\protect\numberline{}Pre
face} % to show preface on toc
\pagenumbering{gobble}
% \pagenumbering{roman}   % to show preface on toc
\newpage
\pagestyle{plain}
\pagenumbering{roman}
\pdfbookmark{\contentsname}{toc}
\setcounter{tocdepth}{2}
\tableofcontents
\newpage
\pagestyle{head}
Main.tex                                      23,1          57%
```

```
        \includegraphics[width=0.7\textwidth]{picts/IO-2.
png}
        \caption{Архитектура компьютера с шиной PCI.}
        \label{IO2}
\end{figure}

\begin{figure}[H]
        \centering
        \includegraphics[width=0.7\textwidth]
        {picts/Interrupt1.png}
        \caption{Цикл ввода/вывода, основаннный на
        концепции аппаратного прерывания.}
        \label{Interrupt1}
\end{figure}

lectures/review.tex                          191,1-8        Bot
```

```
:mks! lect_session.vim
```

Коллекция компиляторов ***gcc***

GCC(1) GNU GCC(1)

NAME

gcc - GNU project C and C++ compiler

SYNOPSIS

gcc [**-c**|**-S**|**-E**] [**-std=**standard]
[**-g**] [**-pg**] [**-O**level]
[**-W**warn...] [**-Wpedantic**]
[**-I**dir...] [**-L**dir...]
[**-D**macro[=defn]...] [**-U**macro]
[**-f**option...] [**-m**machine-option...]
[**-o** outfile] [@file] infile...

Only the most useful options are listed here; see below for the remainder. **g++** accepts mostly
the same options as **gcc**.

Отладчик ***gdb***

```
.global main

main:
  movl $0xaf, %eax
  mov $512, %rbx
  movb $9, %cl
  add $16,%rbx
ret
```

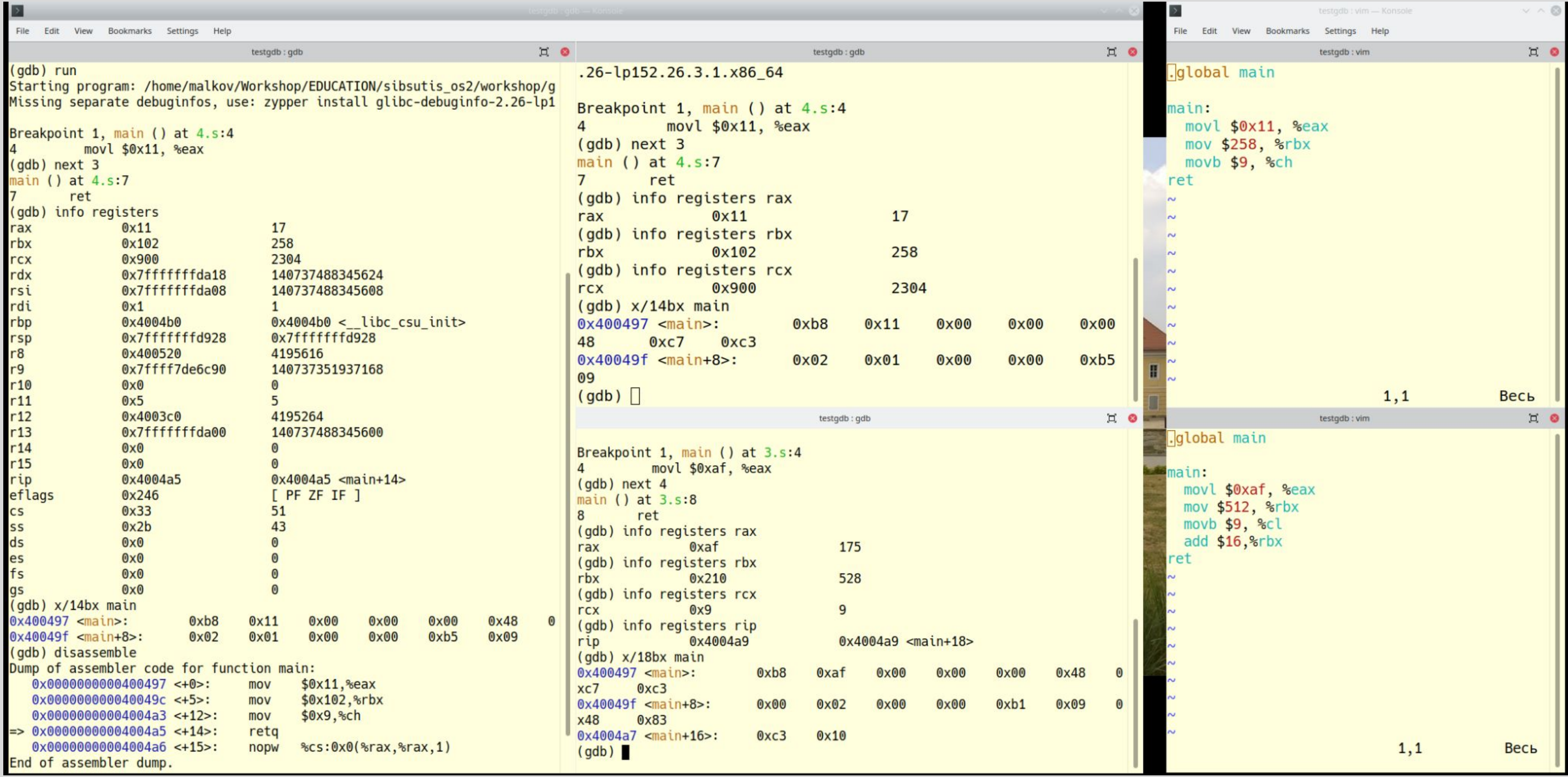
```
> gcc 3.s  -g -o 3
```

```
> gdb 3
```

```
> (gdb) break 1
```

```
> (gdb) run
```

https://www.opennet.ru/base/dev/from_c_to_asm.txt.html



```
(gdb) run
Starting program: /home/malkov/Workshop/EDUCATION/sibsutis_os2/workshop/g
Missing separate debuginfos, use: zypper install glibc-debuginfo-2.26-lp1

Breakpoint 1, main () at 4.s:4
4         movl $0x11, %eax
(gdb) next 3
main () at 4.s:7
7       ret
(gdb) info registers
rax            0x11                17
rbx            0x102               258
rcx            0x900               2304
rdx            0x7fffffffda18      140737488345624
rsi            0x7fffffffda08      140737488345608
rdi            0x1                 1
rbp            0x4004b0            0x4004b0 <__libc_csu_init>
rsp            0x7fffffffd928      0x7fffffffd928
r8             0x400520            4195616
r9             0x7ffff7de6c90      140737351937168
r10            0x0                 0
r11            0x5                 5
r12            0x4003c0            4195264
r13            0x7fffffffda00      140737488345600
r14            0x0                 0
r15            0x0                 0
rip            0x4004a5            0x4004a5 <main+14>
eflags         0x246               [ PF ZF IF ]
cs             0x33                51
ss             0x2b                43
ds             0x0                 0
es             0x0                 0
fs             0x0                 0
gs             0x0                 0
(gdb) x/14bx main
0x400497 <main>:        0xb8    0x11    0x00    0x00    0x00    0x48    0
0x40049f <main+8>:      0x02    0x01    0x00    0x00    0xb5    0x09
(gdb) disassemble
Dump of assembler code for function main:
   0x0000000000400497 <+0>:     mov    $0x11,%eax
   0x000000000040049c <+5>:     mov    $0x102,%rbx
   0x00000000004004a3 <+12>:    mov    $0x9,%ch
=> 0x00000000004004a5 <+14>:    retq
   0x00000000004004a6 <+15>:    nopw   %cs:0x0(%rax,%rax,1)
End of assembler dump.
```

```
.26-lp152.26.3.1.x86_64

Breakpoint 1, main () at 4.s:4
4         movl $0x11, %eax
(gdb) next 3
main () at 4.s:7
7       ret
(gdb) info registers rax
rax            0x11                17
(gdb) info registers rbx
rbx            0x102               258
(gdb) info registers rcx
rcx            0x900               2304
(gdb) x/14bx main
0x400497 <main>:        0xb8    0x11    0x00    0x00    0x00
48      0xc7    0xc3
0x40049f <main+8>:      0x02    0x01    0x00    0x00    0xb5
09
(gdb) □
```

```

Breakpoint 1, main () at 3.s:4
4         movl $0xaf, %eax
(gdb) next 4
main () at 3.s:8
8       ret
(gdb) info registers rax
rax            0xaf                175
(gdb) info registers rbx
rbx            0x210               528
(gdb) info registers rcx
rcx            0x9                 9
(gdb) info registers rip
rip            0x4004a9            0x4004a9 <main+18>
(gdb) x/18bx main
0x400497 <main>:        0xb8    0xaf    0x00    0x00    0x00    0x48    0
xc7     0xc3
0x40049f <main+8>:      0x00    0x02    0x00    0x00    0xb1    0x09    0
x48     0x83
0x4004a7 <main+16>:     0xc3    0x10
(gdb) █
```

```
.global main

main:
  movl $0x11, %eax
  mov $258, %rbx
  movb $9, %ch
ret
~
~
~
~
~
~
~
~
~
~
~
                                 1,1           Весь
```

```
.global main

main:
  movl $0xaf, %eax
  mov $512, %rbx
  movb $9, %cl
  add $16,%rbx
ret
~
~
~
~
~
~
~
~
~
~
~
                                 1,1           Весь
```

```
> gdb 3
(gdb) break main
Breakpoint 1 at 0x400497: file 3.s, line 4.
(gdb) run
Breakpoint 1, main () at 3.s:4
4         movl $0xaf, %eax
(gdb) next 4
main () at 3.s:8
8       ret
(gdb) info registers rax
rax            0xaf                175
```

```
(gdb) info registers rbx
rbx            0x210                 528
(gdb) info registers rcx
rcx            0x9                   9
(gdb) info registers rip
rip            0x4004a9              0x4004a9 <main+18>
(gdb) x/18bx main
0x400497 <main>:        0xb8    0xaf    0x00    0x00    0x00    0x48    0xc7    0xc3
0x40049f <main+8>:      0x00    0x02    0x00    0x00    0xb1    0x09    0x48    0x83
0x4004a7 <main+16>:     0xc3    0x10
```

```
(gdb) disassemble
Dump of assembler code for function main:
   0x0000000000400497 <+0>:     mov    $0xaf,%eax
   0x000000000040049c <+5>:     mov    $0x200,%rbx
   0x00000000004004a3 <+12>:    mov    $0x9,%cl
   0x00000000004004a5 <+14>:    add    $0x10,%rbx
=> 0x00000000004004a9 <+18>:    ret
   0x00000000004004aa <+19>:    nopw   0x0(%rax,%rax,1)
End of assembler dump.
(gdb)
```

```
(gdb) help
List of classes of commands:

aliases -- User-defined aliases of other commands.
breakpoints -- Making program stop at certain points.
data -- Examining data.
files -- Specifying and examining files.
internals -- Maintenance commands.
obscure -- Obscure features.
running -- Running the program.
stack -- Examining the stack.
status -- Status inquiries.
support -- Support facilities.
text-user-interface -- TUI is the GDB text based interface.
tracepoints -- Tracing of program execution without stopping the program.
user-defined -- User-defined commands.
```

```
Type "help" followed by a class name for a list of commands in that class.
Type "help all" for the list of all commands.
Type "help" followed by command name for full documentation.
Type "apropos word" to search for commands related to "word".
Type "apropos -v word" for full documentation of commands related to "word".
Command name abbreviations are allowed if unambiguous.
```

```
(gdb) list main
6       void hTest(int N, int* a, int* b){
7        for(int i=0; i<N;i++)
8           a[i]+=b[i];
9       }
10
11       int main(int argc, char** argv){
12        if(argc<2){
13          fprintf(stderr, "USAGE: lab2 <N>\n");
14          return -1;
15        }

(gdb) b 6
Breakpoint 1 at 0x4007a6: file lab2c.c, line 7.
```

```
(gdb) run 1024
Breakpoint 1, hTest (N=1024, a=0x4032a0,
   b=0x4042b0) at lab2c.c:7
7         for(int i=0; i<N;i++)

(gdb) step
8             a[i]+=b[i];
(gdb) info local
i = 0
(gdb) n 16
8             a[i]+=b[i];
(gdb) info local
i = 8
(gdb) n 16
8             a[i]+=b[i];
(gdb) info local
i = 8
```

```
(gdb) break 8 if i==64
(gdb) c
Continuing.
Breakpoint 2, hTest (N=1024, a=0x4032a0,
   b=0x4042b0) at lab2c.c:8
8           a[i]+=b[i];
(gdb) print i
$1 = 64
(gdb) x/68d a
0x4032a0:      1      5      9      13
0x4032b0:      17     21     25     29
......................................................................
0x403390:      241    245    249    253
0x4033a0:      128    130    132    134
(gdb) c
Continuing.
(gdb) q
```

Профилировщик ***gprof***

```
~/Lecture2> gcc -pg  lab2c.c -o lab2c
/Lecture2> ./lab2c 0
Elapsed time: 2545.74 ms
Lecture2> ls -ltr
-rwxr-xr-x 1 malkov users 15232 сен 17 18:19 lab2c
-rw-r--r-- 1 malkov users  1650 сен 17 18:19 gmon.out
/Lecture2> gprof lab2c  gmon.out > lab2c.prof
/Lecture2> ls -ltr
-rw-r--r-- 1 malkov users  1650 сен 17 18:19 gmon.out
-rw-r--r-- 1 malkov users  5849 сен 17 18:19 lab2c.prof
```

```
/Lecture2> vim lab2c.prof
...........................................................
index   %time    self   children   called     name
[1]    100.0     3.67     2.55                main [1]
                 2.55     0.00      1/1       hTest [2]
-------------------------------------------------------------
                 2.55     0.00      1/1         main [1]
[2]     41.0     2.55     0.00      1       hTest [2]
----------------------------------------------
```

Создание процессов в *Linux*

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
void oldman();
void recreation();

int main(){
  pid_t  child_pid, parent_pid;
  int i=0;
```

```
 fprintf(stdout, "Before RECREATION %i\n",
                                        parent_pid=(int)getpid());

child_pid=fork();

 while(i++<5)
   if(child_pid!=0)
     oldman();
   else
      recreation();
 return 0;
}
```

```
#include  <sys/types.h>
#include  <unistd.h>

void oldman(){
fprintf(stdout, "I'm not yet dead! My ID is %i\n", (int) getpid());
}

void recreation(){
fprintf(stdout, "Who I am? My ID is %i\n", (int) getpid());
}
```

```
~> ./2
Before RECREATION 6169
I'm not yet dead! My ID is 6169
I'm not yet dead! My ID is 6169
I'm not yet dead! My ID is 6169
Who I am? My ID is 6170
I'm not yet dead! My ID is 6169
I'm not yet dead! My ID is 6169
Who I am? My ID is 6170
Who I am? My ID is 6170
Who I am? My ID is 6170
Who I am? My ID is 6170
```

```
~> ./2
Before RECREATION 6154
I'm not yet dead! My ID is 6154
I'm not yet dead! My ID is 6154
Who I am? My ID is 6155
Who I am? My ID is 6155
Who I am? My ID is 6155
Who I am? My ID is 6155
Who I am? My ID is 6155
I'm not yet dead! My ID is 6154
I'm not yet dead! My ID is 6154
I'm not yet dead! My ID is 6154
```

```
 #include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
int main(){
pid_t child_pid, parent_pid;
double s=0.0;

child_pid=fork();
if(child_pid!=0){
  s+=3.14;
  fprintf(stdout, "CHILD: %i s=%g &s=%u\n", (int) getpid(),s,&s);
}
else{
 s+=2.72;
 fprintf(stdout, "PARENT: %i s=%g &s=%u\n", (int) getpid(),s, &s);
}
return 0;
}
```

```
PARENT: 5404 s=2.72 &s=2309295864
CHILD:    5403 s=3.14 &s=2309295864
```

При создании процесса с помощью системного вызова fork() копируется адресное пространство, - переменная s имеет один и тот же адрес. Однако отображение на физическую память для родительского и дочернего процесса различно, - значения переменной s различны.

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
int main(){
pid_t child_pid;
pid_t parent_pid;
double s=0.0;;
FILE* fp;

child_pid=fork();
fp=fopen("test.dat","a+");
if(child_pid!=0){
  s+=3.14;
  fprintf(fp, "CHILD: %i s=%g &s=%u fp=%u\n", (int) getpid(),
s, &s, fp);
}
else{
  s+=2.72;
  fprintf(fp, "PARENT: %i s=%g &s=%u fp=%u\n",(int) getpid(),
  s, &s,fp);
}
fclose(fp);
return 0;
}
```

test.dat

```
PARENT: 5450 s=2.72 &s=760346688 fp=6299664
CHILD: 5449 s=3.14 &s=760346688 fp=6299664
```

Дескрипторы файлов при копировании сохраняются.

СПАСИБО ЗА ВНИМАНИЕ!