



Трансляторы и среды разработки

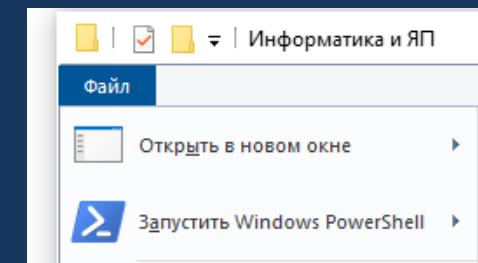
IDE (*Integrated Development Environment*) - интегрированная среда разработки:

- редактирование (синтаксическая подсветка, подсказки, рефакторинг)
- компиляция (интерпретация)
- средства сборки (Linker, компоновщик, редактор связей)
- отладка
- рефакторинг
- коллективное владение кодом, системы контроля версий
- встроенные фреймворки
- профилирование (трассировка памяти времени исполнения кода по переменным и модулям)

Компилятор, Linker, отладчик и другие вспомогательные средства IDE выполнены в виде отдельных программ, исполняемых в **режиме командной строки**

Режим командной строки – базовый (скрытый) режим работы программы в ОС в формате диалога с алфавитно-цифрового терминала (строчный экран + клавиатура). Графический интерфейс (GUI) – надстройка над режимом командной строки, в некоторых сборках ОС (сервер) может отсутствовать.

PowerShell - расширяемое средство автоматизации от Microsoft с открытым исходным кодом, состоящее из оболочки с **интерфейсом командной строки** и сопутствующего языка сценариев





Рефакторинг

Рефакторинг – изменение структуры кода, не изменяющее поведение программы
«генеральная уборка» :

- стилистика кода (эстетика, читаемость)
- повышение модульности, универсальности, управляемости кода

Изменение имен классов, функций, переменных, клонирование модулей, перемещение в проекте, изменение интерфейсов, выделение интерфейсов



Реинжиниринг – генеральная переделка проекта, изменяющая его характеристики , «капитальный ремонт, переезд»



Варианты использования IDE

1. Windows – Microsoft Visual Studio – собственный C/C++-компилятор
2. Windows – CodeBlocks – MinGW – GCC
3. Windows – WSL – Linux(Ubuntu) – CodeBlocks – GCC
4. Linux(Ubuntu) – CodeBlocks – GCC



Среда разработки CodeBlocks

- **CodeBlocks** – кросплатформенная открытая среда разработки под внешние компиляторы (в т.ч. из других IDE) (<http://codeblocks.org>)
- Можно использовать компиляторы MS Visual Studio (не работает отладчик), GCC,G++ в MinGW
- **GNU Compiler Collection** (обычно используется сокращение **GCC**) — набор компиляторов для различных языков программирования, разработанный в рамках проекта GNU. GCC является свободным программным обеспечением. Он используется как стандартный компилятор для свободных UNIX-подобных операционных систем.
- **MinGW** (Minimalist GNU for Windows),— бесплатная среда разработки приложений под Microsoft Windows с открытым исходным кодом. Состоит из *портированного* (перенесенного) GCC, GNU Binutils (ассемблер, компоновщик, менеджер архивации), GNU Debugger, набора свободно распространяемых библиотек импорта и заголовочных файлов для Windows API и прочего.

По поводу предыстории и взаимоотношений **GNU** и **Linux** и их общего предка **UNIX**

<https://losst.ru/kak-pravilno-linux-ili-gnu-linux>



CodeBlocks/Windows (настройка)

<http://codeblocks.org>

The screenshot shows a Windows file explorer window with the following directory structure:

- Program Files > CodeBlocks > MinGW**
 - bin**: Contains executables like `addr2line.exe`, `ar.exe`, `g++.exe`, `gcc.exe`, etc.
 - include**: Contains header files like `stddef.h`, `stdexcept.h`, `stdint.h`, `stdio.h`, etc.
 - lib**
 - etc**
 - licenses**
 - opt**
 - share**
 - x86_64-w64-mingw32**
 - build-info.txt**
- MinGW > x86_64-w64-mingw32 > include**: Shows a detailed view of the `stdio.h` file.

Red boxes highlight the `bin` folder, the `include` folder, and the `stdio.h` file.



CodeBlocks/Windows (настройка)

Settings->Compiler

Settings->Debugger

The image shows two overlapping windows from the CodeBlocks IDE.

Global compiler settings (Top Window):

- Selected compiler:** GNU GCC Compiler (highlighted by a red box).
- Compiler's installation directory:** C:\Program Files\CodeBlocks\MinGW (highlighted by a red box). An "Auto-detect" button is also visible.
- Program Files:** A tabbed section containing compiler-related executable paths.

Debugger settings (Bottom Window):

- GDB/CDB debugger : Default:** A tab under the Debugger settings window.
- Executable path:** C:\Program Files\CodeBlocks\MinGW\bin\gdb.exe (highlighted by a red box).
- Arguments:** (empty)
- Debugger Type:** GDB (radio button selected).
- Debugger initialization commands:** A text area containing startup scripts. A red box highlights the following checked options:
 - Disable startup scripts (-nx) (GDB only)
 - Watch function arguments
 - Watch local variables
 - Catch C++ exceptions
 - Evaluate expression under cursor
- Note:** "These commands will be sent to the debugger on each session start".

Settings->Editor

The image shows the "Configure editor" dialog.

General settings (Tab Bar):

- Editor settings (highlighted by a red box)
- Other editor settings
- C/C++ Editor settings
- Encoding settings (highlighted by a red box)

Encoding:

- Use encoding when opening files:** WINDOWS-1251 (highlighted by a red box).
- Use this encoding:** (dropdown menu)



Создание проекта

New from template

Projects
Build targets
Files
Custom
User templates

Category: <All categories>

X

Go

Cancel

ARM Project AVR Project Arduino Project

Console application D application Direct/X project

FLTK

Code::Blocks plugin DLL Dynamic Link Library

DLL DLL

1

Console application

2

Please select the compiler to use and which configurations you want enabled in your project.

Compiler:

GNU GCC Compiler

Create "Debug" configuration: Debug

"Debug" options

Output dir.: bin\Debug\

Objects output dir.: obj\Debug\

Create "Release" configuration: Release

"Release" options

Output dir.: bin\Release\

Objects output dir.: obj\Release\

TIP: Try right-clicking on the project to open its context menu.

1. Select a category
2. Select a template
3. Press Generate

Console application



Please select the folder where you want the new project to be created as well as its title.

3

Project title:

fufufu

Folder to create project in:

F:\temp\

Project filename:

fufufu.cbp

Resulting filename:

F:\temp\fufufu\fufufu.cbp

temp > fufufu



fufufu.cbp

main.cpp



Компиляция, сборка и исполнение

The screenshot shows the Code::Blocks IDE interface. The top menu bar has 'Fortran' selected. A red box highlights the build button (a green triangle icon) in the toolbar. Another red box highlights the 'Debug' configuration dropdown in the toolbar. The left sidebar shows a project named 'fufufu' with a file 'main.cpp'. The main code editor window displays the following C++ code:

```
1 #include <stdio.h>
2
3 int main() { // Проверка работоспособности
4     printf("a-a-a-a-a-a-a-a-a-a\n");
5     return 0;
6 }
7
```

The output window at the bottom shows the execution results:

```
F:\temp\fufufu\bin\Debug\fufufu.exe
a-a-a-a-a-a-a-a-a-a

Process returned 0 (0x0) execution time : 0.148 s
Press any key to continue.
```

Компиляция и сборка с динамическими библиотеками (g++)

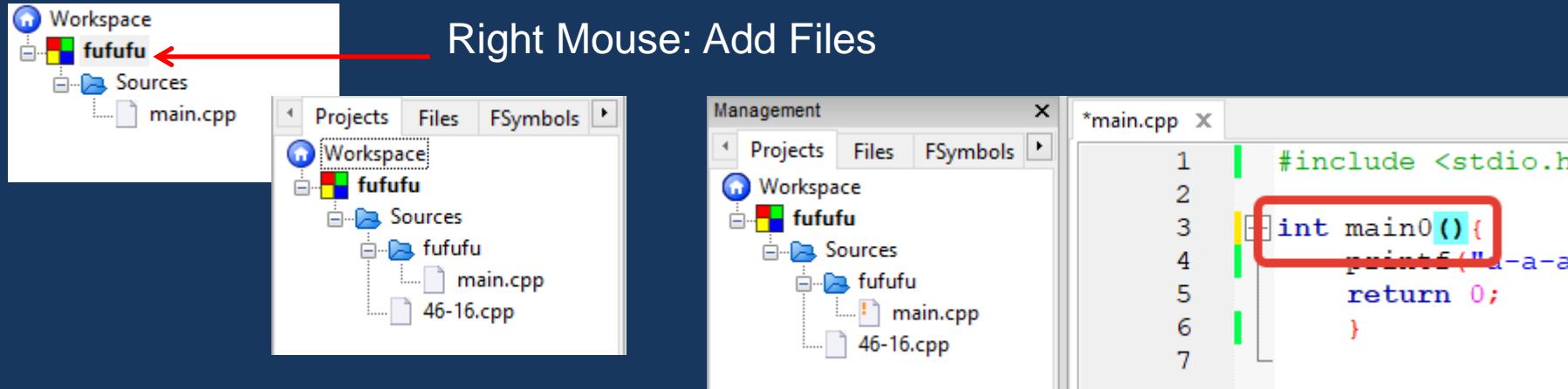
```
----- Run: Debug in fufufu (compiler: GNU GCC Compiler) -----
g++ -Wall -fexceptions -g -Wall -g -c F:\temp\fufufu\main.cpp -o obj\Debug\main.o
g++ -o bin\Debug\fufufu.exe obj\Debug\main.o
Output file is bin\Debug\fufufu.exe with size 53.43 KB
Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 0 warning(s) (0 minute(s), 0 second(s))
```

Исполнение exe в сборке Debug (отладка)

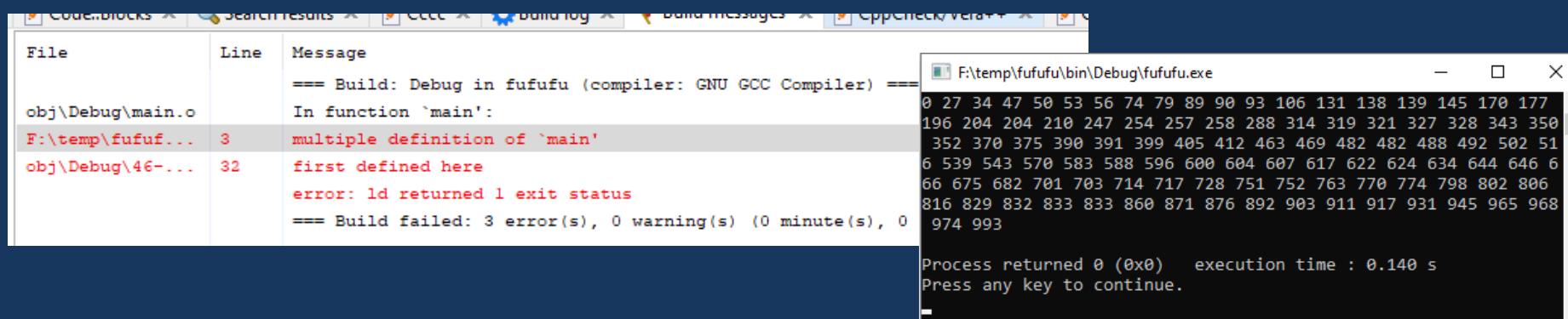
```
----- Run: Debug in fufufu (compiler: GNU GCC Compiler) -----
Checking for existence: F:\temp\fufufu\bin\Debug\fufufu.exe
Set variable: PATH=.;C:\Program Files\CodeBlocks\MinGW\bin;C:\Program Files\CodeBlocks\MinGW;C:\Program Files (x86)\Common Files\Windows\System32\WindowsPowerShell\v1.0;C:\Windows\System32\OpenSSH;C:\Program Files\MongoDB\Server\4.0\bin;C:\Program Files\PU\Local\Microsoft\WindowsApps;C:\Users\Admin\AppData\Local\Programs\Microsoft VS Code\bin
Executing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "F:\temp\fufufu\bin\Debug\fufufu.exe" (in F:\temp\fufufu\.)
```



Добавление сторонних исходников



main должен быть только один (C/C++ - ошибка сборки)



Компиляция и сборка двух файлов в проекте

```
----- Build: Debug in fufufu (compiler: GNU GCC Compiler) -----
g++ .exe -Wall -fexceptions -g -Wall -g -c F:\temp\46-16.cpp -o obj\Debug\46-16.o
g++ .exe -Wall -fexceptions -g -Wall -g -c F:\temp\fufufu\main.cpp -o obj\Debug\main.o
g++ .exe -o bin\Debug\fufufu.exe obj\Debug\46-16.o obj\Debug\main.o
Output file is bin\Debug\fufufu.exe with size 62.85 KB
Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 0 warning(s) (0 minute(s), 0 second(s))
```

Многие действия в
IDE не
выполняются при
исполнении кода
(закрыть окно)



Отладка (debug)

- пошаговое исполнение
- непрерывное до точки останова
- слежение за переменными

```
0.03
Build Debug Fortran wxSmith Tools Tools+ Plugins Doxygen Settings Help
42-03.cpp x
1 //-
2 //-----Простые числа
3 void calc(int val, int A[], int n) {
4     int i,m,k;
5         for (i=0, m=2; i < n-1 && m < val; m++) {
6             for (k=0; k < i; k++)
7                 if (m % A[k]==0) break;
8             if (i==k)
9                 A[i++] = m;
10        }
11    A[i] = 0;
12
13 #include <stdio.h>
14 void main(){
15     int B[100],v=200; calc(v,B,100);
16     for (int i=0; B[i]!=0; i++) printf("%d ",B[i]);
17     puts("");
18 }
```

Debug->Debugging Windows

- Watching - слежение за переменными
- Call stack – стек вызовов функций

«Медицинские факты»:

- программа не обязана делать то, что она должна делать
- слежение программой и сравнение текущего поведения (состояния) с ожидаемым
- лучший отладчик – функция printf (трассировка необходимых значений, история работы программы)

```
42-03.cpp x
1 //-
2 //-----Простые числа
3 void calc(int val, int A[], int n) {
4     int i,m,k;
5         for (i=0, m=2; i < n-1 && m < val; m++) {
6             for (k=0; k < i; k++)
7                 if (m % A[k]==0) break;
8             if (i==k)
9                 A[i++] = m;
10        }
11    A[i] = 0;
12
13 #include <stdio.h>
14 void main(){
15     int B[100],v=200; calc(v,B,100);
16     for (int i=0; B[i]!=0; i++) printf("%d ",B[i]);
17     puts("");
18 }
```

Call stack

Nr	Address	Function	File
0	0x401641	calc (val=200, A=0x61fc80, n=100)	F:\temp\42-03.cpp
1	0x401641	main()	F:\temp\42-03.cpp

Watches

Function	val	A	n
	200	0x61fc80	100

Locals

i	m	k
0	2	0



Отладка (debug)

```
1 //-
2 //-----Простые числа
3 #include <stdio.h>
4 void calc(int val, int A[], int n){
5     int i,m,k;
6     for (i=0, m=2; i < n-1 && m < val; m++) {
7         for (k=0; k < i; k++)
8             if (m % A[k]==0) break;
9         if (i==k) {
10             for(int j=0;j<i;j++)
11                 printf("%d ",A[j]);
12             printf("\n");
13         }
14     A[i] = 0;
15 }
16
17
18 int main(){
19     int B[100],v=50; calc(v,B,100);
20     for (int i=0; B[i]!=0; i++) printf("%d ",B[i]);
21     puts("");
22 }
```

```
Выбрать F:\temp\fufufufu\bin\Debug\fufufufu.exe
2
2 3
2 3 5
2 3 5 7
2 3 5 7 11
2 3 5 7 11 13
2 3 5 7 11 13 17
2 3 5 7 11 13 17 19
2 3 5 7 11 13 17 19 23
2 3 5 7 11 13 17 19 23 29
2 3 5 7 11 13 17 19 23 29 31
2 3 5 7 11 13 17 19 23 29 31 37
2 3 5 7 11 13 17 19 23 29 31 37 41
2 3 5 7 11 13 17 19 23 29 31 37 41 43
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47
Process returned 0 (0x0)   execution time : 0.085 s
Press any key to continue.
```

лучший отладчик – функция printf (трассировка необходимых значений, история работы программы)



Состояние процессора и дизассемблирование

Register	Hex	Interpreted
eax	0x7	7
ecx	0x8	8
edx	0x22ff0c	2293516
ebx	0x7ffd000	2147328000
esp	0x22fed8	0x22fed8
ebp	0x22fee8	0x22fee8
esi	0x0	0
edi	0x0	0
eip	0x4014fc	0x4014fc <sort
eflags	0x206	[PF IF]
cs	0x1b	27
ss	0x23	35
ds	0x23	35
es	0x23	35
fs	0x3b	59
gs	0x0	0

Disassembly

Function: sort (D:\Temp\fufu2\main.cpp:10)
Frame start: 0x22fef0

```
;3 : void sort(int A[], int n){  
0x401460      push   %ebp  
0x401461      mov    %esp,%ebp  
0x401463      sub    $0x10,%esp  
;4 :     int i,found;                                // Количество обменов  
;5 :     do {                                         // Повторять просмотр..  
0x40150f      jmp    0x401466 <sort(int*, int)+6>  
;6 :     found =0;  
0x401466      movl   $0x0,-0x8(%ebp)  
;7 :     for (i=0; i<n-1; i++)  
0x40146d      movl   $0x0,-0x4(%ebp)  
0x401474      mov    0xc(%ebp),%eax  
0x401477      sub    $0x1,%eax  
0x40147a      cmp    -0x4(%ebp),%eax  
0x40147d      jle    0x401509 <sort(int*, int)+169>  
0x401500      addl   $0x1,-0x4(%ebp)  
0x401504      jmp    0x401474 <sort(int*, int)+20>  
;8 :             if (A[i] > A[i+1]) {          // Сравнить соседей  
0x401483      mov    -0x4(%ebp),%eax
```

Дизассемблирование – восстановление исходного текста программы по двоичному коду:

- Си – по командам процессора,
- Java – по байт-коду

Проблемы дизассемблирования – утерянные при трансляции имена, неоднозначная структура кода



Установка CodeBlocks в Ubuntu

1. Установка c/c++

```
sudo apt-get update
```

```
sudo apt install gcc
```

```
sudo apt install g++
```

```
sudo apt-get install build-essential gdb
```

2. Установка CodeBlocks

```
sudo add-apt-repository ppa:damien-moore/codeblocks-stable
```

```
sudo apt update
```

```
sudo apt install codeblocks codeblocks-contrib
```

3. Установка freeGlut

```
sudo apt-get install freeglut3 freeglut3-dev
```

```
sudo apt-get install binutils-gold
```



Установка CodeBlocks в Ubuntu (вар.2)

1. Установка с/c++, freeglut

```
sudo apt install g++ freeglut3 freeglut3-dev
```

```
sudo apt install libxmu-dev libxi-dev
```

2. Установка CodeBlocks

```
sudo apt install codeblocks
```

3. Установка OpenGL

```
sudo apt install build-essential libgl1-mesa-dev
```

```
sudo apt install libglew-dev libsdl2-dev libsdl2-image-dev libglm-dev libfreetype6-dev #  
some libraries
```

4. Утилита проверки состояния OpenGL

```
sudo apt install mesa-utils
```

```
glxinfo | grep OpenGL
```



Режим командной строки

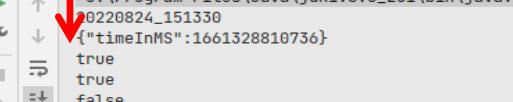
Режим командной строки – базовый (скрытый) режим работы программы в ОС в формате диалога с использованием алфавитно-цифрового терминала (строчный экран + клавиатура). Графический интерфейс (GUI) – надстройка над режимом командной строки, в некоторых сборках ОС (сервер) может отсутствовать.

«Медицинские факты»:

- окно в режиме командной строки – *терминал, консоль*
 - стандартный ввод/вывод - символьные потоки) –
перенаправляются в *консоль*



```
System.out.println(new Gson().toJson(xx));
OwnDateTime tt = new OwnDateTime();
System.out.println(tt.isToday());
tt.hour(12);
System.out.println(tt.isToday());
tt.incMonth();
System.out.println(tt.isToday());
System.out.println(new OwnDateTime().timeOfDayInMin());
```



The screenshot shows the Java code execution interface. The code runs successfully, displaying the output in the terminal window:

```
Run: OwnDateTime x
"C:\Program Files\Java\jdk1.8.0_201\bin\java.exe"
20220824_151330
{"timeInMS":1661328810736}
true
true
false
913
2020-01-01_00.00

Process finished with exit code 0
```

```
d:\Temp\cpp1\dist\Debug\Cygwin-Windows>cpp1 aaa bbb ccc  
cpp1  
aaa  
bbb  
ccc  
abcd
```

```
int main(int narg, char *argv[]) {
    for (int i=0;i<narg;i++)
        puts(argv[i]);
```



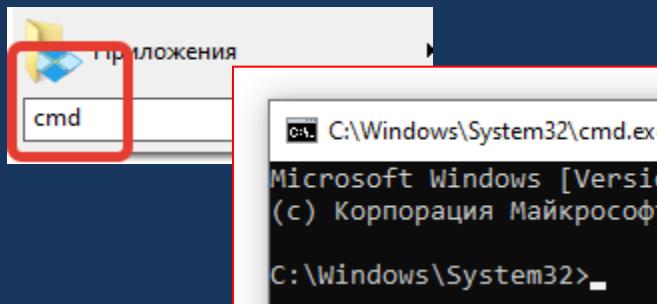
Процессор команд

Командный процессор – системная программа, осуществляющая диалог с пользователем в режиме командной строки:

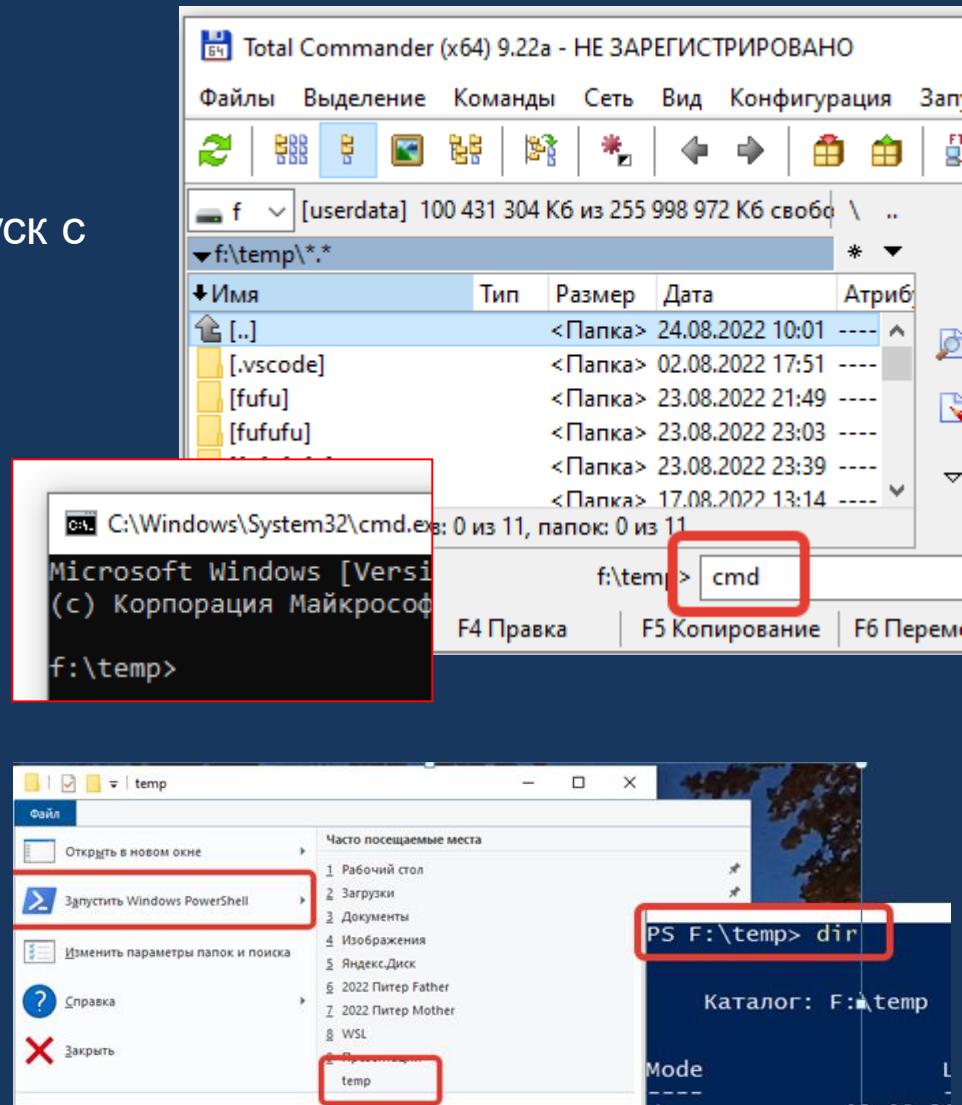
- Windows – cmd, PowerShell
- Unix – Shell
- GNU/Linux - Bash

Total Commander – запуск с выбранным каталогом

Выпадает в списке – cmd.exe
Запуск с каталогом Windows



Проводник – File - PowerShell





Процессор команд Windows Cmd,PowerShell

«Прожиточный минимум»:

Управляющие символы:

- «стрелка вверх, стрелка вниз» перемещение по последовательности введенных команд
- Home и End - в начало и конец редактируемой строки
- CTRL + «стрелка влево/вправо» перемещение на слово
- Esc – удаление редактируемой команды
- F1 - ввод последней выполненной команды по одному символу
- F5 - вывод предыдущей команды
- F7 - выводит список нескольких последних команд
- CTRL+C – копирование, CTRL+V или «правая кнопка мыши» - вставка

Команды:

- cd – смена текущего каталога
- dir – вывод каталога

```
PS F:\> cd temp
PS F:\temp> cd f:/_
PS F:\> cd temp/fufu_-
PS F:\temp\fufu> cd ../_
```



Процессор команд Linux/WSL Bash

Команды:

- cd – смена текущего каталога
- xxx –help – ключ вывода подсказки для xxx
- sudo – префикс исполнения от имени суперпользователя
- apt установщик пакетов

```
/home/romanow/桌面/programm/11c.  
romanow@DESKTOP-LM8R924:/mnt/f/temp$ cd  
romanow@DESKTOP-LM8R924:~$ cd /  
romanow@DESKTOP-LM8R924:/$ cd /mnt/f/temp/fufu  
romanow@DESKTOP-LM8R924:/mnt/f/temp/fufu$ cd ../  
romanow@DESKTOP-LM8R924:/mnt/f/temp$
```

```
romanow@DESKTOP-LM8R924:/mnt/f/temp$ ls -g  
total 180  
-rwxrwxrwx 1 romanow 19752 Aug 26 21:03 42-03  
-rwxrwxrwx 1 romanow 446 Aug 23 23:23 42-03.cpp  
-rwxrwxrwx 1 romanow 19608 Aug 24 10:01 44-06  
-rwxrwxrwx 1 romanow 715 Aug 23 20:35 44-06.cpp  
-rwxrwxrwx 1 romanow 54882 Aug 17 14:27 44-06.exe  
-rwxrwxrwx 1 romanow 6744 Aug 23 19:59 44-06.o  
-rwxrwxrwx 1 romanow 23712 Aug 2 16:35 46-16  
-rwxrwxrwx 1 romanow 1220 Aug 23 22:21 46-16.cpp  
-rwxrwxrwx 1 romanow 17040 Jul 15 21:33 46-16.exe  
-rwxrwxrwx 1 romanow 3360 Jul 15 22:45 46-16.o
```



Настройка путей по умолчанию

Каталоги, в которых ОС ищет исполняемые файлы (программы) по имени. Перечислены в системной переменной PATH. При отсутствии пути в PATH в командной строке необходимо вводить полный путь до файла. Иногда IDE требуют настройки PATH для вызова компилятора

1. Windows PowerShell window showing the command `PS F:\temp> sysdm.cpl` being run.

2. "Свойства системы" (System Properties) dialog box. The "Переменные среды..." (Environment Variables...) button is highlighted with a red box.

3. "Изменить переменную среды" (Change Environment Variable) dialog box. The "Path" variable under "Системные переменные" (System variables) is selected and highlighted with a red box. The "Изменить..." (Edit...) button is also highlighted with a red box.

4. "Изменить переменную среды" (Change Environment Variable) dialog box. The "Path" variable is listed in the list box. A new entry, `C:\Program Files\CodeBlocks\MinGW\bin\`, has been added and is highlighted with a red box. The "Создать" (Create) button is also highlighted with a red box.



Воспроизведение команд GCC CodeBlocks

Windows PowerShell

```
PS F:\temp\fufufufu> g++.exe -Wall -fexceptions -g -Wall -g -c ..\42-03.cpp -o obj\Debug\42-03.o
PS F:\temp\fufufufu> g++.exe -o bin\Debug\fufufufu.exe obj\Debug\42-03.o
PS F:\temp\fufufufu> bin\Debug\fufufufu.exe
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47
PS F:\temp\fufufufu> -
```

