

# Среди за програмиране на C++

## Windows

*Забележка! Не се предоставя на повечето международни състезания по информатика!*

### Code::Blocks с MinGW

Това е все още най-популярната у нас интегрирана среда за обработка (IDE) за C++ сред състезателите. Предимствата ѝ са, че е лесна за инсталиране, използване и персонализиране. Недостатъци са, че самият продукт не е много поддържан и вече липсват много функционалности на съвременните IDE-та. Понякога програмата забива и трябва да се изключи принудително, което води до загуба на информацията в незапазените файлове.

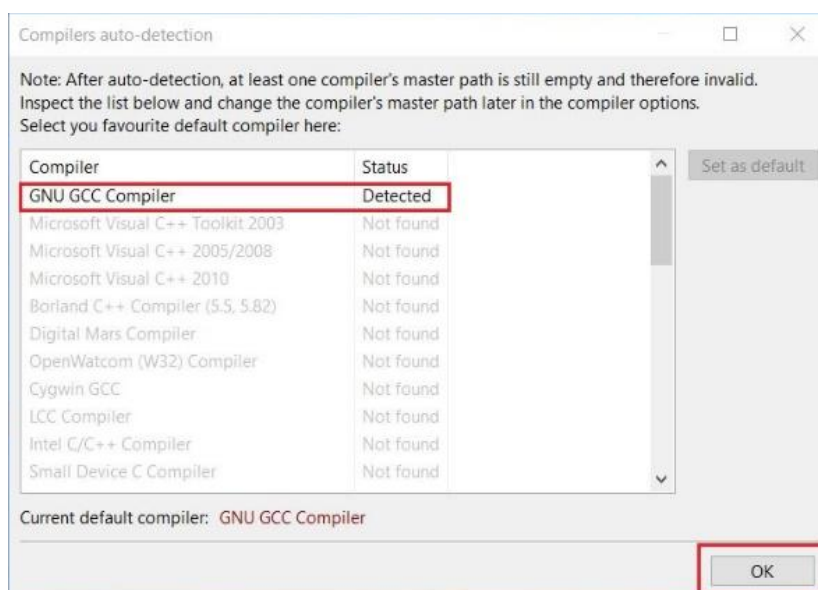
За да се инсталира, е препоръчително да се използва официалният сайт – [www.codeblocks.org](http://www.codeblocks.org). От менюто трябва да се отиде в секция “Downloads” и от тази страница в “Download the binary release”, където може да се изтегли изпълним файл за Windows (.exe) за инсталация на всичко необходимо. Изглед към октомври 2024 г.:



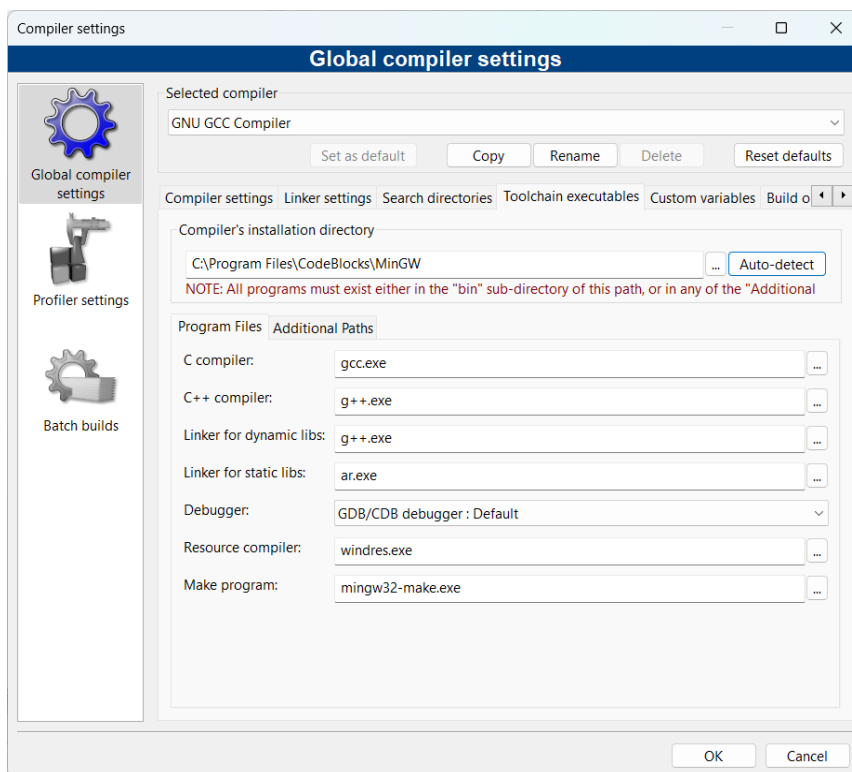
File	Download from
codeblocks-20.03-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03-setup-nonadmin.exe	FossHUB or Sourceforge.net
codeblocks-20.03-nosetup.zip	FossHUB or Sourceforge.net
<b>codeblocks-20.03mingw-setup.exe</b>	FossHUB or Sourceforge.net
codeblocks-20.03mingw-nosetup.zip	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-setup-nonadmin.exe	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-nosetup.zip	FossHUB or Sourceforge.net
codeblocks-20.03mingw-32bit-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03mingw-32bit-nosetup.zip	FossHUB or Sourceforge.net

Трябва да се изтегли инсталационен файл за codeblocks-20.03mingw-setup.exe от линковете в дясната колона. Той ще инсталира не само CodeBlocks, но и MinGW, което само по себе си е програмна среда, която съдържа и компилатора GCC за C/C++. Инсталацията е стандартна, като най-добре се придържайте към опциите по подразбиране.

Ако всичко е наред при първото стартиране на CodeBlocks ще се изпише, че GNU GCC компилаторът е засечен:

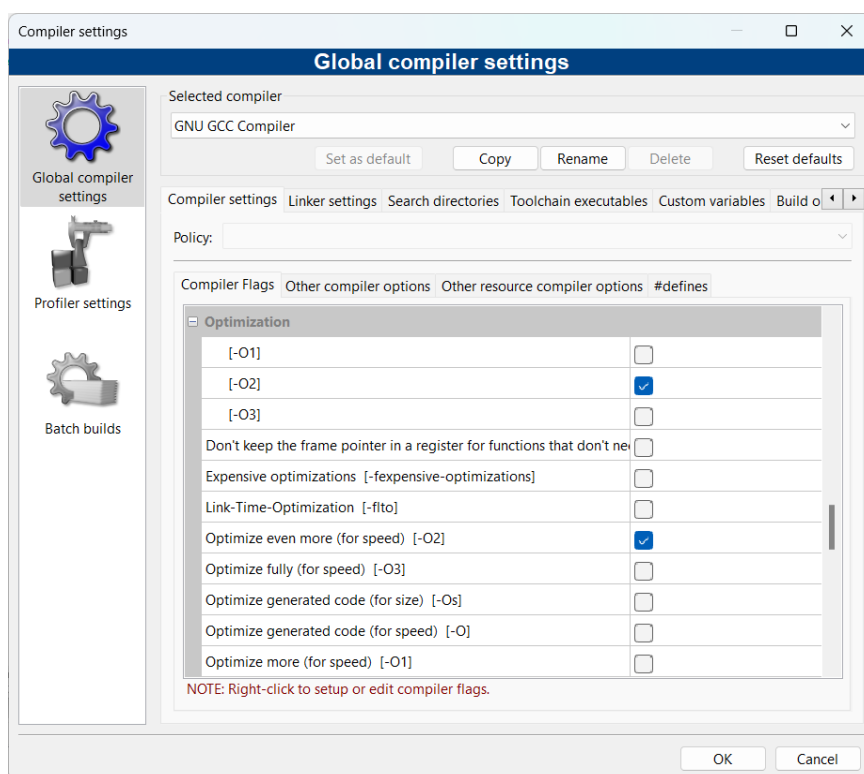


В противен случай, след като се стартира програмата може да опитате да го засечете от Settings → Compiler... → Toolchain executables и да натиснете бутона “Auto-detect”:



Ако това не проработи, трябва от бутона ... да намерите ръчно папката MinGW, която трябва да е подпапка на CodeBlocks папката с инсталация на програмата.

Препоръчително е да включите следните опции на компилатора от Settings → Compiler... → Compiler settings → Compiler Flags, тъй като те се използват от повечето състезателни системи:



Тук може също да настроите компилатора да използва по-нов стандарт, като включите опциите “Have g++ follow the C++17 ISO C++ language standard [-std=c++17]” (или алтернативните опции за C++14 и C++11). Може да проверите дали сте се справили успешно, като пробвате да компилирате програмите [checkCPP14.cpp](#) или [checkCPP17\\_oldMinGW.cpp](#).

Известен е бъг във версията на MinGW, която се инсталира с CodeBlocks. Заради него не може да се компилира програма с хедър файла `#include<bits/stdc++.h>` и включен стандарта C++17 на компилатора. Има различни начини за справяне с този проблем – например, инсталиране на по-нова версия на MinGW. Друг начин е показан на 1-ри ред на програмата [checkCPP17\\_oldMinGW.cpp](#).

## Visual Studio Code

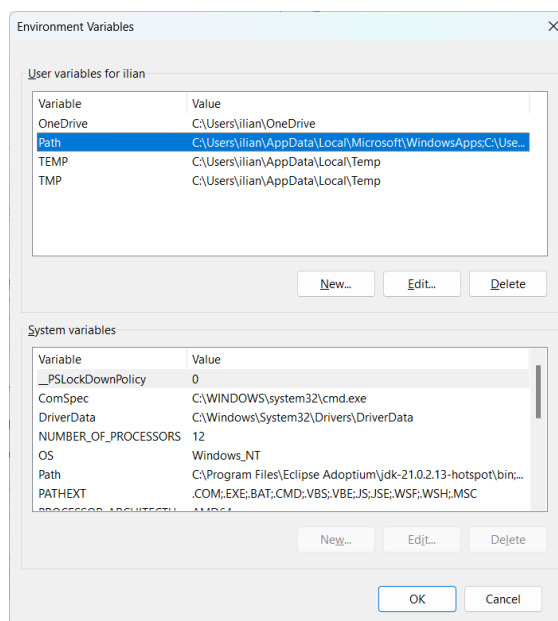
Тази интегрирана среда е доста по-нова и активно развиваща се. Набира популярност и сред състезателите. Малко по-трудно е да се настрои първоначално. За компилиране и изпълняване на написаните програми е най-удобно да се използва вграденият терминал.

За да се инсталира, е препоръчително да се използва официалният сайт - [code.visualstudio.com](#). Още на началната страница има бутон за изтегляне на инсталационен файл. Препоръчително е да се придържате към опциите по подразбиране по време на инсталацията. След като се инсталира, трябва да се добави следният пакет от приставки за C/C++:

<https://marketplace.visualstudio.com/items?itemName=ms-vscode.cpptools-extension-pack>

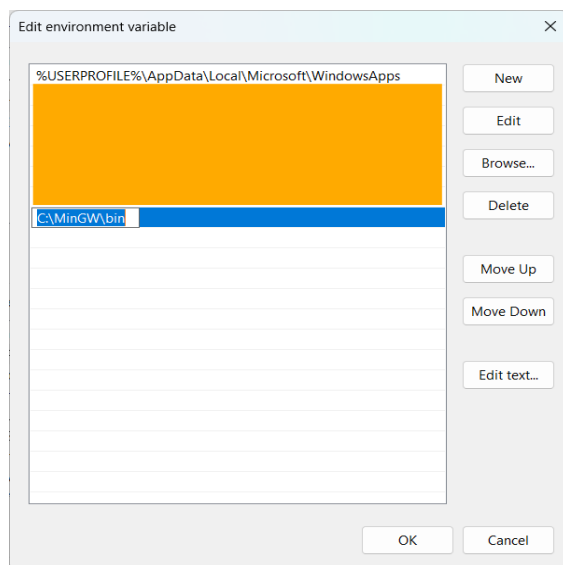
Тук е по-сложно да се настрои компилаторът. Нужно е първо да имате инсталиран компилатор за C++. Стандартно на Windows се инсталира програмната среда MinGW, съдържаща GCC компилатора, например от тук: [sourceforge.net/projects/mingw](#).

След това трябва да се добави пътят към папката с компилатор на C++ към променливите на средата (environment variables). За тази цел в търсачката на Windows напишете “environment variables” и трябва да излязат настройки от Контролния панел, които като отворите изглеждат по следния начин:



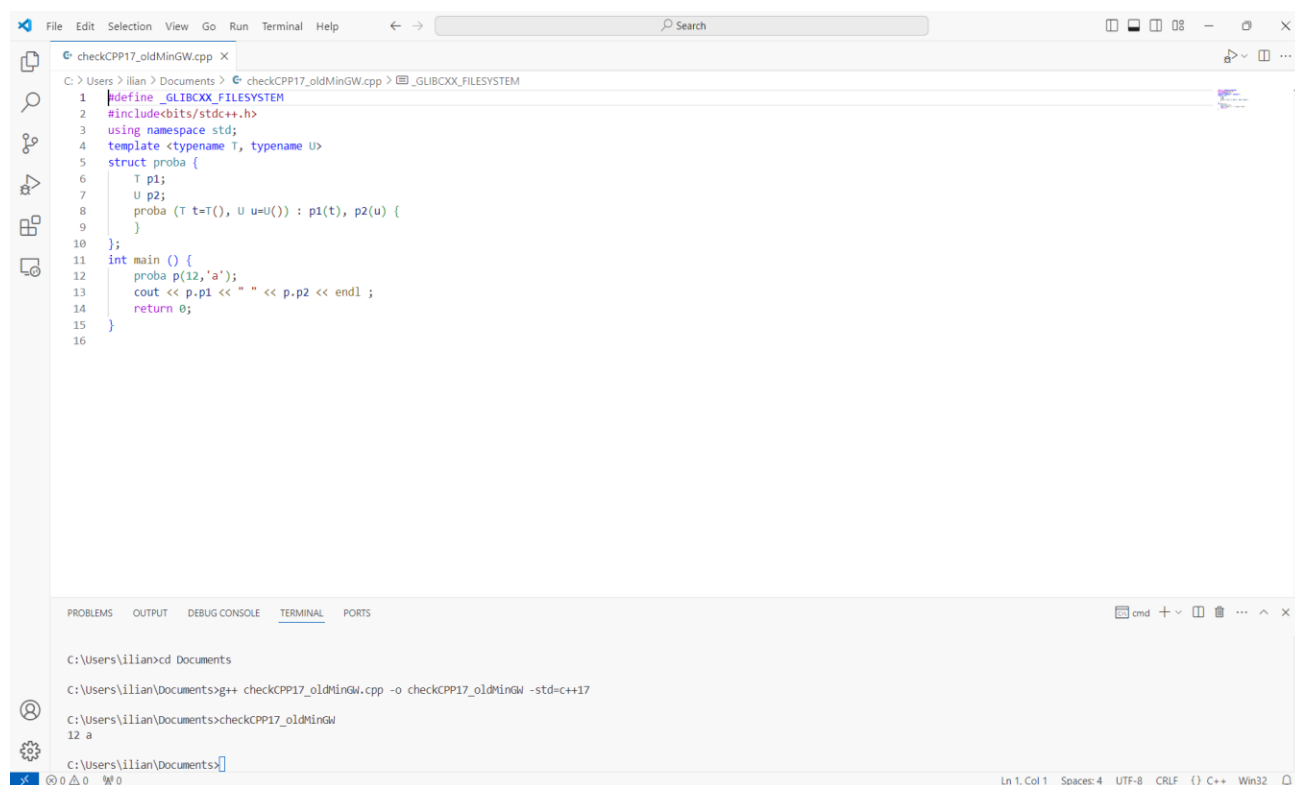
Трябва да маркирате променливата “Path” на “User variables” и след това да натиснете бутонът “Edit...”.

В отворения прозорец е нужно да добавите пътя към папката с компилатора, който обикновено е C:\MinGW\bin, а ако използвате инсталираното MinGW на Codeblocks, то пътят би трябвало да е C:\Program Files\CodeBlocks\MinGW\bin. Накрая натискате "OK" бутоните, докато излезете.



Тази инструкция добавя пътя само в променливите на средата на текущия потребител. Възможно е да ги добавите и за всички потребители на системата като следвате почти същите стъпки. Трябва да отворите настройките чрез текст "system environment variables" в търсачката на Windows. От прозореца, който излиза натиснете бутона "Environment Variables...", намиращ се в долния десен ъгъл. След което се отварят познатите настройки от Контролния панел, но трябва да редактирате променливата "Path" на "System variables".

Вече трябва да можете да компилирате с командата g++ през Visual Studio Code, използвайки вградения терминал. Ето пример за такова компилиране, използващ за терминал Command Prompt, на [checkCPP17\\_oldMinGW.cpp](#) за C++17 стандарта.



```
checkCPP17_oldMinGW.cpp X
C:\Users\ilian\Documents> cd checkCPP17_oldMinGW > _GLIBCXX_FILESYSTEM
1 #define _GLIBCXX_FILESYSTEM
2 #include<bits/stdc++.h>
3 using namespace std;
4 template <typename T, typename U>
5 struct proba {
6     T p1;
7     U p2;
8     proba (T t=T(), U u=U()) : p1(t), p2(u) {
9     }
10 };
11 int main () {
12     proba p(12,'a');
13     cout << p.p1 << " " << p.p2 << endl ;
14     return 0;
15 }
16

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
C:\Users\ilian>cd Documents
C:\Users\ilian\Documents>g++ checkCPP17_oldMinGW.cpp -o checkCPP17_oldMinGW -std=c++17
C:\Users\ilian\Documents>checkCPP17_oldMinGW
12 a
C:\Users\ilian\Documents>
```

Първата команда е за придвижване до пътя, където се намира файла. В случая файлът е в папка Documents, която е подпапка на текущия път (C:\Users\ilian). Втората команда е за компилирането с g++. Първата част е името на файла, след това се задава името на изпълнимия файл (чрез опцията -o) и последното е опция за задаване на стандарта, който да се използва. Третата команда изпълнява получения изпълним файл.

Ако при вас не излиза панела, в който се намира терминала, то може да го покажете през менюто View → Appearance → Panel. Също така, ако сте инсталирали по-нова версия на MinGW, то тогава трябва да може безпроблемно да компилирате и файла [checkCPP17.cpp](#).

# Ubuntu

## Code::Blocks

*Забележка! Не се предоставя на Международната олимпиада по информатика!*

За да го инсталирате е достатъчно да въведете следните команди в терминала:

```
sudo add-apt-repository ppa:damien-moore/codeblocks-stable
sudo apt-get update
sudo apt install codeblocks codeblocks-contrib
```

Тук трябва ръчно да инсталирате компилатор. Може да използвате командите:

```
sudo apt-get update
sudo apt install g++
```

След това трябва да може да стартирате CodeBlocks и да компилирате успешно програмата [checkCPP17.cpp](#). Имайте предвид, че тази версия на Codeblocks за Ubuntu е много нестабилна и често забива!

## Visual Studio Code

Има много начини за инсталиране – може да изтеглите .deb пакет за Linux от официалния сайт, може да го инсталирате през магазина за приложения на Ubuntu или директно през терминала. Командите, които са нужни за инсталация през терминала са следните:

```
sudo apt-get install wget gpg

wget -qO- https://packages.microsoft.com/keys/microsoft.asc | gpg --
dearmor > packages.microsoft.gpg

sudo install -D -o root -g root -m 644 packages.microsoft.gpg
/etc/apt/keyrings/packages.microsoft.gpg

echo "deb [arch=amd64,arm64,armhf signed-
by=/etc/apt/keyrings/packages.microsoft.gpg]
https://packages.microsoft.com/repos/code stable main" |sudo tee
/etc/apt/sources.list.d/vscode.list > /dev/null

rm -f packages.microsoft.gpg

sudo apt install apt-transport-https

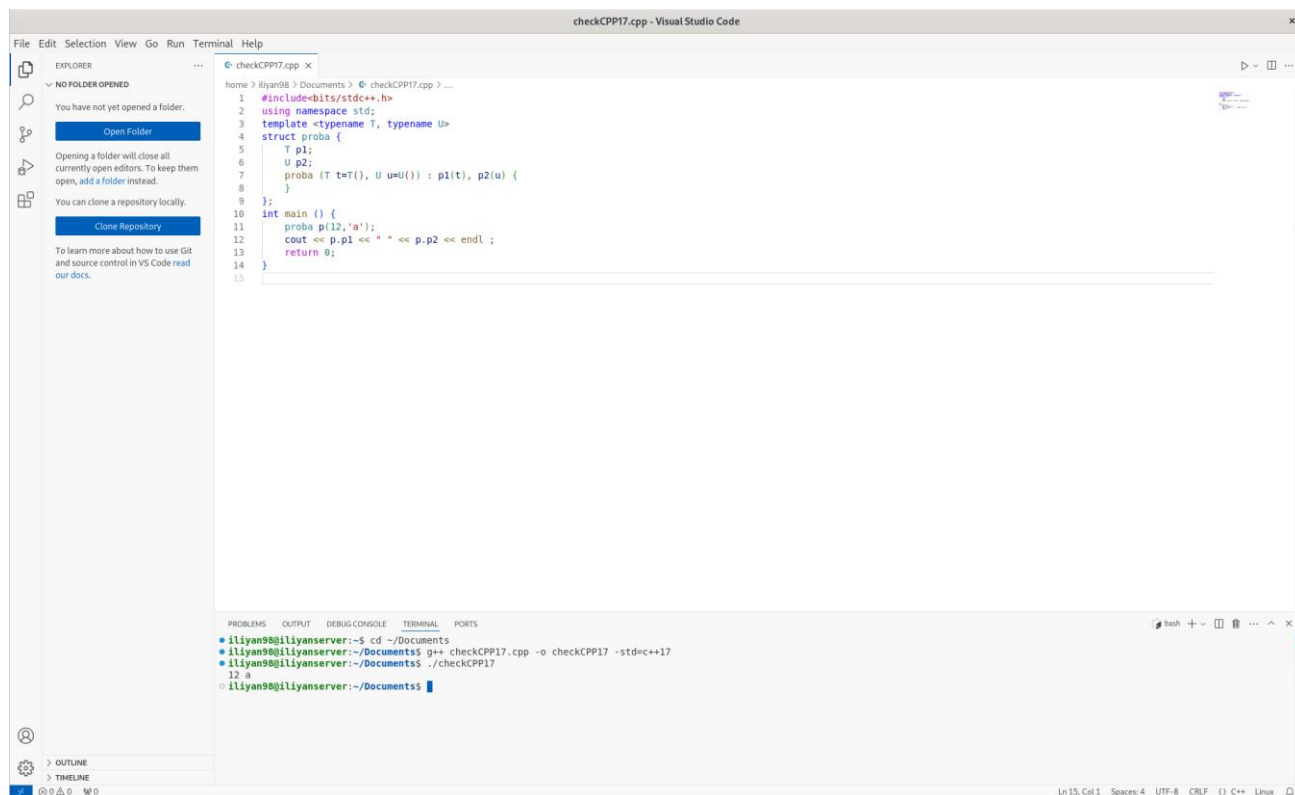
sudo apt update

sudo apt install code # or code-insiders
```

Отново трябва да се добави пакетът от приставки за C/C++, като се следва инструкцията на:

<https://marketplace.visualstudio.com/items?itemName=ms-vscode.cpptools-extension-pack>

Когато е инсталиран и компилатор за C++ (например по инструкциите в предната секция), то вече трябва да могат да се компилират всички примерни програми, подобно на Visual Studio Code на Windows.



The screenshot shows the Visual Studio Code interface with a C++ file named 'checkCPP17.cpp' open. The code in the editor is as follows:

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 template <typename T, typename U>
4 struct proba {
5     T p1;
6     U p2;
7     proba (T t=T(), U u=U()) : p1(t), p2(u) {
8     }
9 };
10 int main () {
11     proba p(12,'a');
12     cout << p.p1 << " * " << p.p2 << endl ;
13     return 0;
14 }
15
```

The terminal at the bottom shows the following commands and output:

```
● iliyang@iliyanserver:~$ cd ~/Documents
● iliyang@iliyanserver:~/Documents$ g++ checkCPP17.cpp -o checkCPP17 -std=c++17
● iliyang@iliyanserver:~/Documents$ ./checkCPP17
12 a
```

The status bar at the bottom right indicates 'Ln 15, Col 1 Spaces: 4 UTF-8 CRLF ( ) C++ Linux'.