

Установка ПО для курса

Кодомашина

Ревизия 474 (960f6f7)

35 страниц, последнее изменение 22.09.2023

Содержание

1 Disclaimer	2
1.1 Рекомендуемые аппаратные системные требования	2
2 macOS	3
2.1 Intro	3
2.2 Установка Homebrew	3
2.3 Установка Git	11
2.4 Установка Visual Studio Code	11
3 Windows	13
3.1 Intro	13
3.2 Установка и настройка WSL	13
3.3 Установка Git	23
3.4 Установка Visual Studio Code	24

1 Disclaimer

В рамках курса мы используем следующие операционные системы:

- Microsoft Windows 10 (22H2)
- Microsoft Windows 11
- Apple macOS 13 (Ventura)

Описанная документация составлена с использованием последних версий указанных операционных систем, действия и получаемые результаты в предыдущих версиях могут отличаться.

1.1 Рекомендуемые аппаратные системные требования

- CPU с поддержкой аппаратной виртуализации (AMD VT / Intel VT)
- ≥ 8GB RAM

2 macOS

2.1 Intro

В macOS для установки программного обеспечения мы будем использовать пакетный менеджер Homebrew - он позволит максимально упростить последующую установку ПО.

2.2 Установка Homebrew

На сайте проекта - [brew.sh](#) прямо сразу описано как установить Homebrew в одну единственную команду:

```
1 /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Для этого запускаем терминал: Нажимаем на кнопку активации Spotlight в правом верхнем углу экрана:

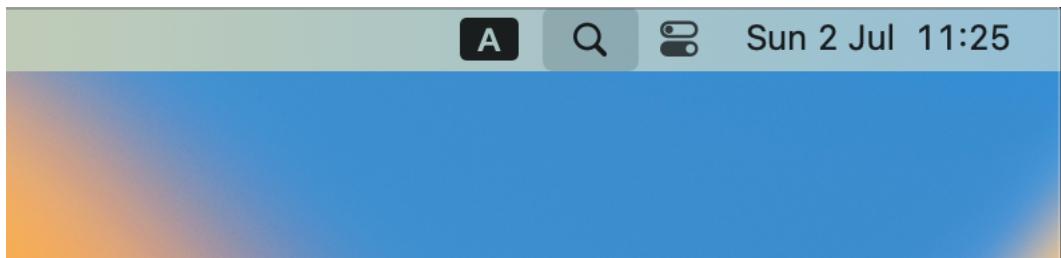


Рис. 1: Кнопка запуска Spotlight

Откроется форма ввода поискового запроса Spotlight:

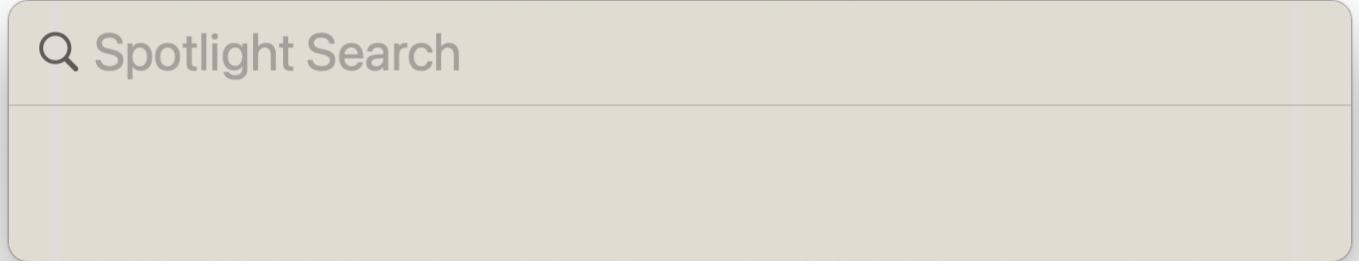


Рис. 2: Spotlight. Форма ввода поискового запроса

Печатаем там слово “Terminal”, он найдет приложение с таким названием:

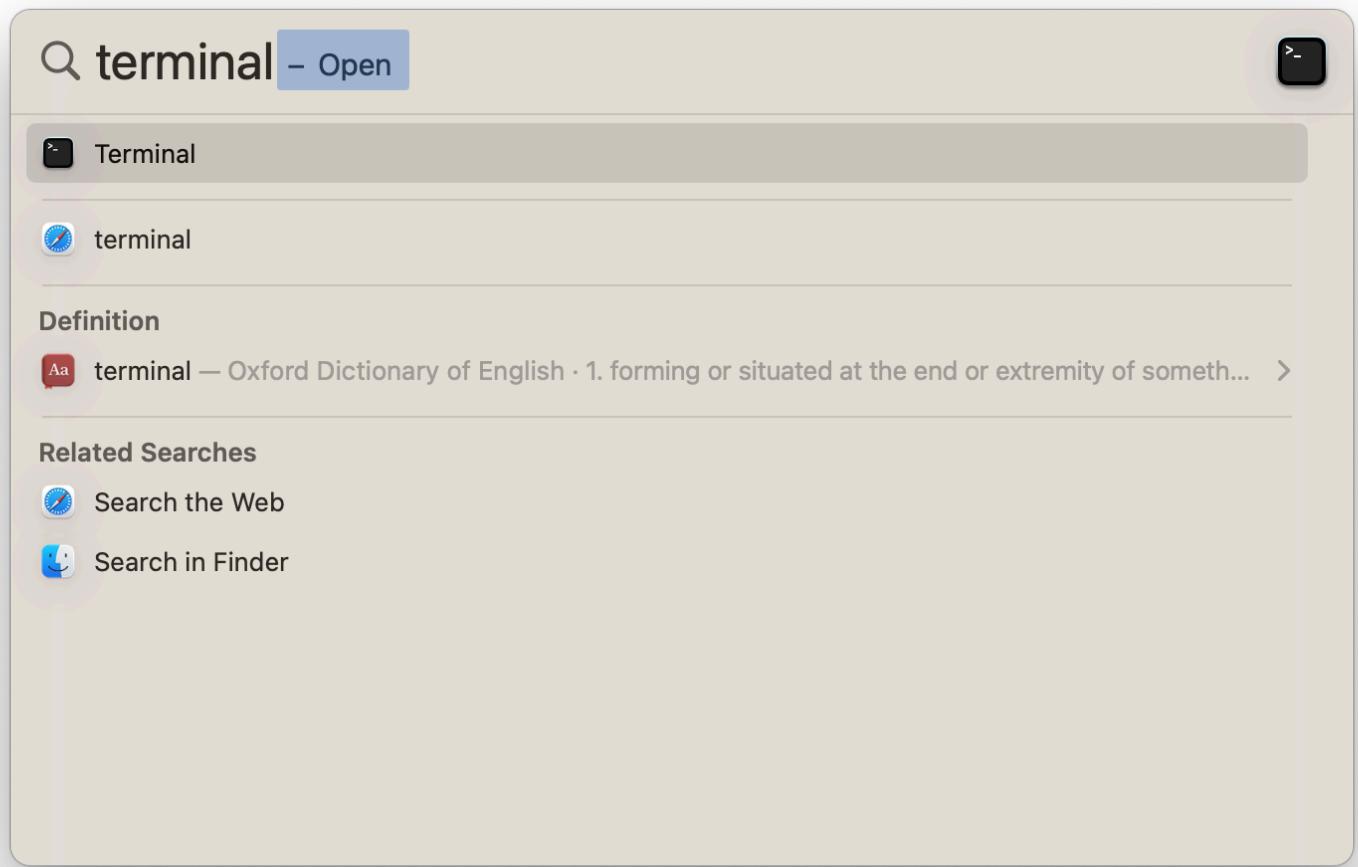


Рис. 3: Spotlight. Терминал

Нажимаем Enter, терминал запустится в виде белого окна с приглашением ввода и мигающим курсором:

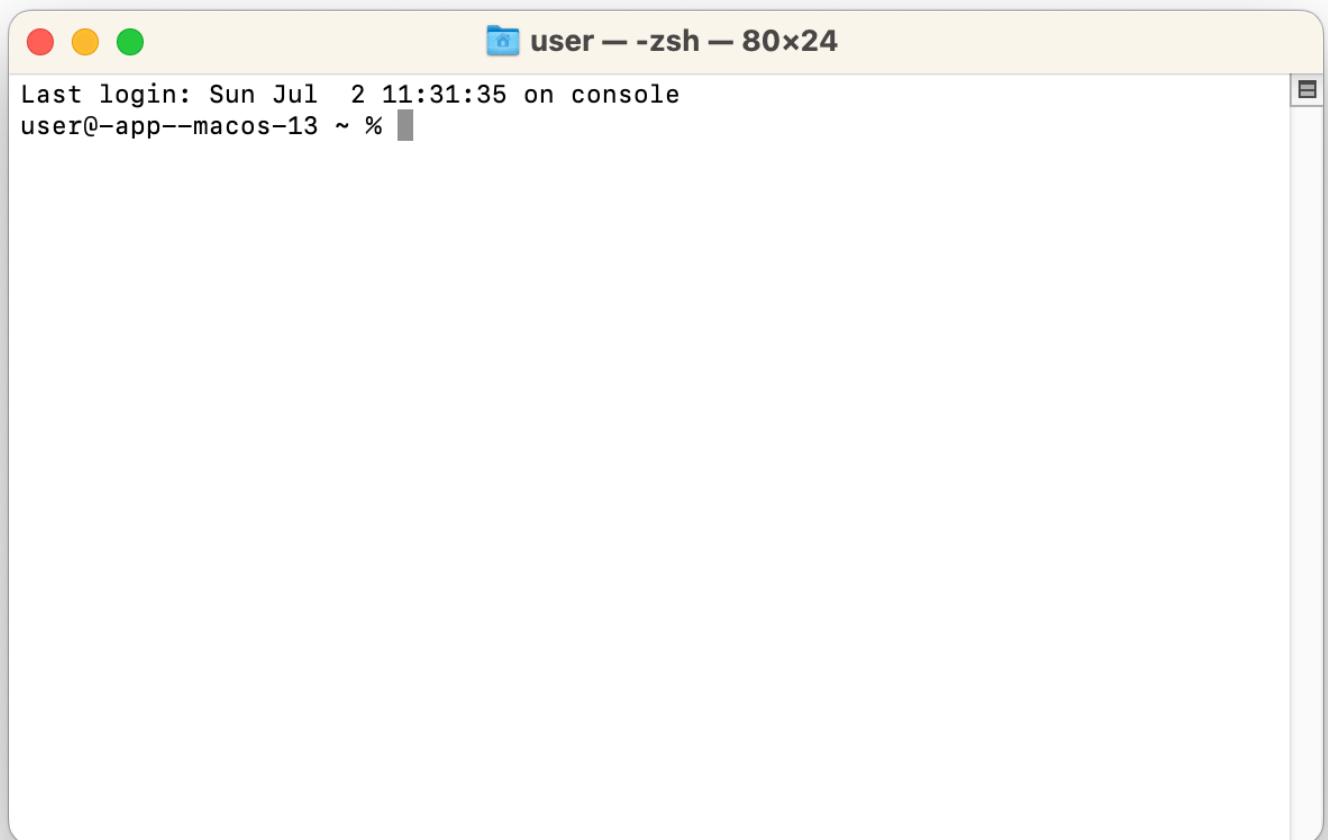


Рис. 4: Терминал macOS

Вводим команду для установки Homebrew:

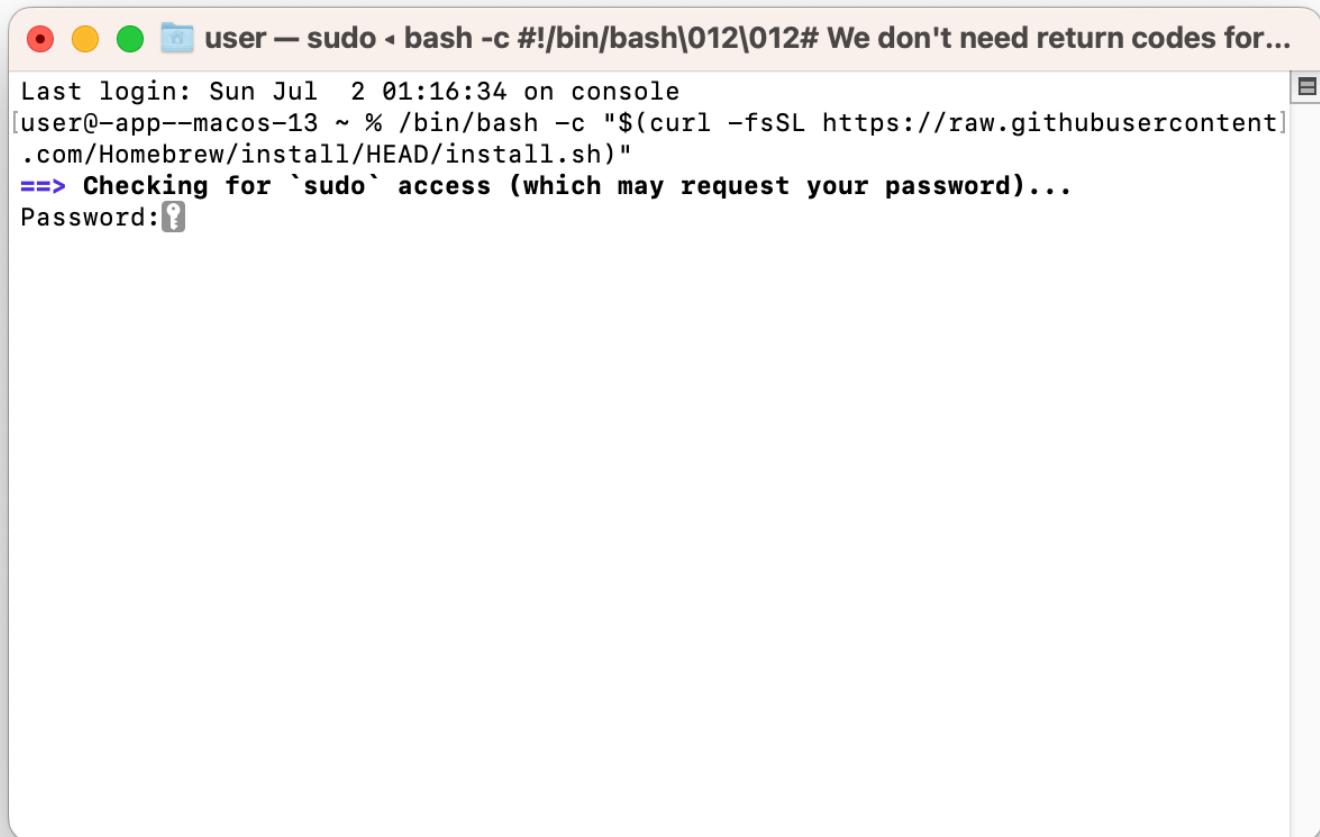
```
1 /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```



Last login: Sun Jul 2 01:16:34 on console
user@app-macos-13 ~ % /bin/bash -c "\$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

Рис. 5: Терминал macOS. Установка Homebrew

После нажатия клавиши Enter установщик запросит пароль текущего пользователя для установки Homebrew:



The screenshot shows a terminal window on a Mac OS X desktop. The title bar reads "user — sudo -c #!/bin/bash\012\012# We don't need return codes for...". The main pane contains the following text:

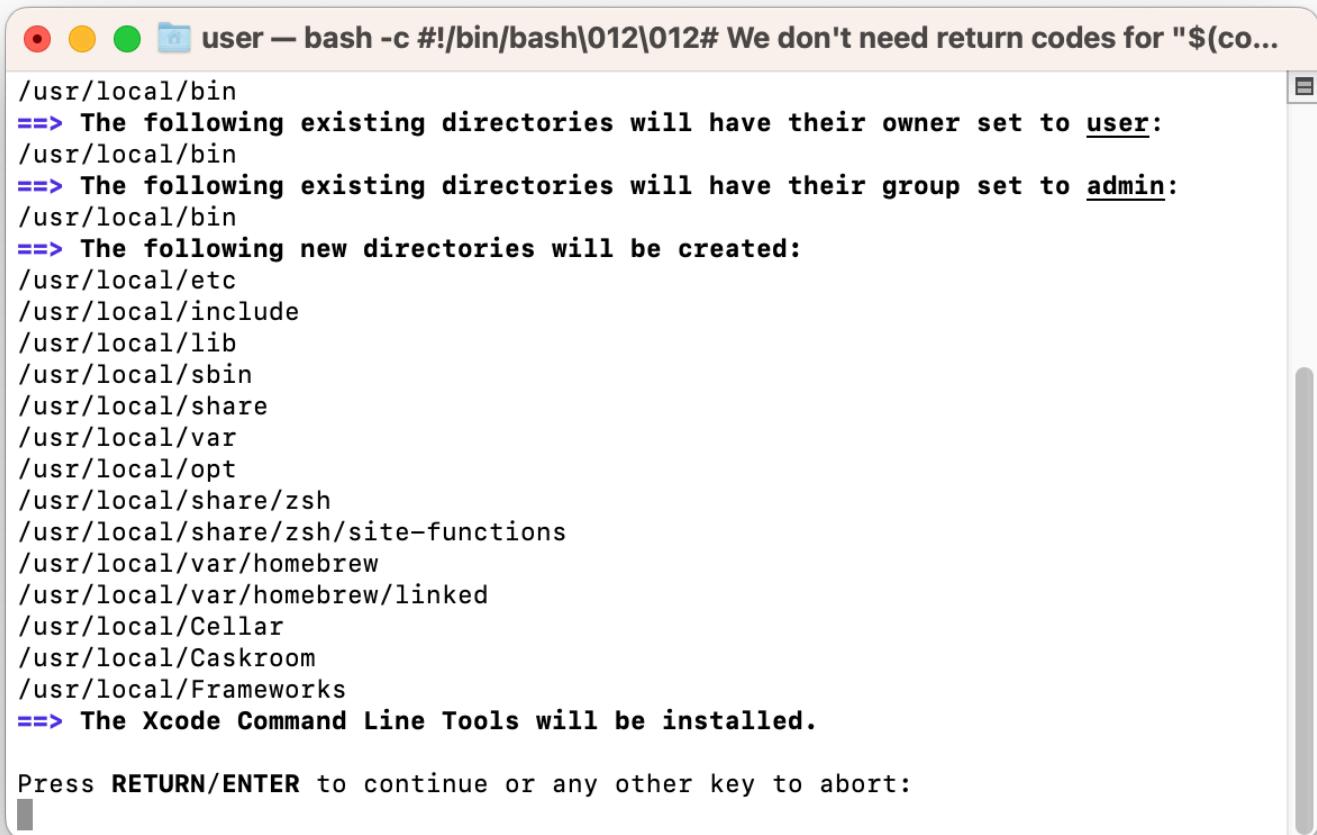
```
Last login: Sun Jul  2 01:16:34 on console  
[user@app--macos-13 ~ % /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"  
==> Checking for `sudo` access (which may request your password)...  
Password: ?
```

Рис. 6: Установка Homebrew. Запрос пароля

Вводим пароль текущего пользователя.

ВНИМАНИЕ: Пароль не будет отображаться при вводе, ни звездочками ни какими-либо иными символами

После запроса пароля, установщик покажет что он собирается изменить в системе и запросит в подтверждение этих действий нажать клавишу Enter:



The screenshot shows a terminal window on a Mac OS X desktop. The title bar says "user — bash -c #!/bin/bash|012|012# We don't need return codes for "\$(co...". The main pane contains the following text:

```
/usr/local/bin  
==> The following existing directories will have their owner set to user:  
/usr/local/bin  
==> The following existing directories will have their group set to admin:  
/usr/local/bin  
==> The following new directories will be created:  
/usr/local/etc  
/usr/local/include  
/usr/local/lib  
/usr/local/sbin  
/usr/local/share  
/usr/local/var  
/usr/local/opt  
/usr/local/share/zsh  
/usr/local/share/zsh/site-functions  
/usr/local/var/homebrew  
/usr/local/var/homebrew/linked  
/usr/local/Cellar  
/usr/local/Caskroom  
/usr/local/Frameworks  
==> The Xcode Command Line Tools will be installed.  
  
Press RETURN/ENTER to continue or any other key to abort:
```

Рис. 7: Установка Homebrew

В процессе установки будет установлен сам менеджер пакетов Homebrew и все его зависимости.

По завершению установки, установщик сообщит, что нужно добавить путь, куда установлен Homebrew в [PATH](#) для того, чтобы все приложения установленные через Homebrew и сам его интерфейс - `brew`, были доступны без указания полного пути:

```
* [new tag]          4.0.8      -> 4.0.8
* [new tag]          4.0.9      -> 4.0.9
HEAD is now at 9a482dea2 Merge pull request #15612 from Homebrew/sponsors-maintainers-man-completions
==> Installation successful!

==> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
https://docs.brew.sh/Analytics
No analytics data has been sent yet (nor will any be during this install run).

==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
https://github.com/Homebrew/brew#donations

==> Next steps:
- Run these two commands in your terminal to add Homebrew to your PATH:
  (echo; echo 'eval "$( /usr/local/bin/brew shellenv )"' ) >> /Users/user/.zprofile
  eval "$( /usr/local/bin/brew shellenv )"
- Run brew help to get started
- Further documentation:
  https://docs.brew.sh

user@app--macos-13 ~ %
```

Рис. 8: Завершение установки Homebrew

Для завершения этого процесса необходимо выполнить следующие команды:

для устройств Apple на основе процессоров Intel:

```
1 (echo; echo 'eval "$( /usr/local/bin/brew shellenv )"' ) >> ~/.zprofile
2 eval "$( /usr/local/bin/brew shellenv )"
```

для устройств Apple на основе собственных процессоров Apple (M1, M2, etc.):

```
1 (echo; echo 'eval "$( /opt/homebrew/bin/brew shellenv )"' ) >> ~/.zprofile
2 eval "$( /opt/homebrew/bin/brew shellenv )"
```

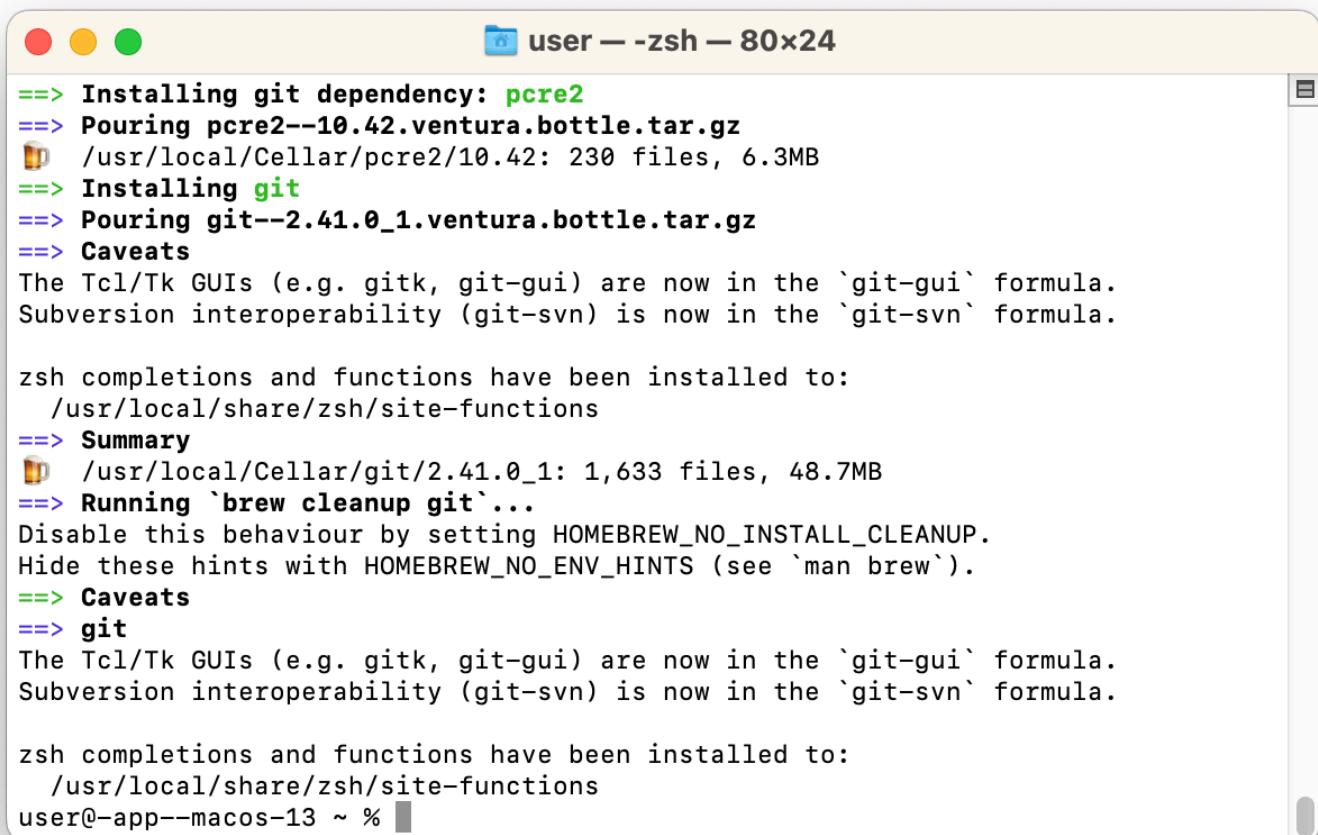
После чего можно пользоваться Homebrew и устанавливать остальные приложения.

2.3 Установка Git

После установки Homebrew установить Git можно командой:

```
1 brew install git
```

По завершению установки, brew сообщит об успешной установке программы:



The screenshot shows a terminal window titled "user — -zsh — 80x24". The window contains the following text output from the "brew install git" command:

```
==> Installing git dependency: pcre2
==> Pouring pcre2--10.42.ventura.bottle.tar.gz
🍺 /usr/local/Cellar/pcre2/10.42: 230 files, 6.3MB
==> Installing git
==> Pouring git--2.41.0_1.ventura.bottle.tar.gz
==> Caveats
The Tcl/Tk GUIs (e.g. gitk, git-gui) are now in the `git-gui` formula.
Subversion interoperability (git-svn) is now in the `git-svn` formula.

zsh completions and functions have been installed to:
/usr/local/share/zsh/site-functions
==> Summary
🍺 /usr/local/Cellar/git/2.41.0_1: 1,633 files, 48.7MB
==> Running `brew cleanup git`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
==> Caveats
==> git
The Tcl/Tk GUIs (e.g. gitk, git-gui) are now in the `git-gui` formula.
Subversion interoperability (git-svn) is now in the `git-svn` formula.

zsh completions and functions have been installed to:
/usr/local/share/zsh/site-functions
user@app--macos-13 ~ %
```

Рис. 9: Установка Git

2.4 Установка Visual Studio Code

После установки Homebrew установить Git можно командой:

```
1 brew install visual-studio-code
```

По завершению установки, brew сообщит об успешной установке программы:

```
zsh completions and functions have been installed to:  
/usr/local/share/zsh/site-functions  
==> Summary  
🍺 /usr/local/Cellar/git/2.41.0_1: 1,633 files, 48.7MB  
==> Running `brew cleanup git`...  
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.  
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).  
==> Caveats  
==> git  
The Tcl/Tk GUIs (e.g. gitk, git-gui) are now in the `git-gui` formula.  
Subversion interoperability (git-svn) is now in the `git-svn` formula.  
  
zsh completions and functions have been installed to:  
/usr/local/share/zsh/site-functions  
[user@app--macos-13 ~ % brew install visual-studio-code  
==> Downloading https://update.code.visualstudio.com/1.79.2/darwin/stable  
==> Downloading from https://az764295.vo.msecnd.net/stable/695af097c7bd098fbf017  
#####
# 100.0%  
==> Installing Cask visual-studio-code  
==> Moving App 'Visual Studio Code.app' to '/Applications/Visual Studio Code.app'  
==> Linking Binary 'code' to '/usr/local/bin/code'  
🍺 visual-studio-code was successfully installed!  
user@app--macos-13 ~ %
```

Рис. 10: Установка Visual Studio Code

3 Windows

3.1 Intro

Для получения однородного опыта работы с Git и Visual Studio Code под Windows мы производим установку Git с использованием WSL - Windows Subsystem for Linux, которая является встроенной системой виртуализации с предварительно подготовленным дистрибутивом Linux - Ubuntu 22.04 (jammy).

Инструкция подготовлена для Windows 10 с последним набором обновлений - 22H2, а так же Windows 11. Действия по установке и настройке в них идентичны.

3.2 Установка и настройка WSL

Установка WSL в последней версии Windows 10 и Windows 11 производится в одно действие: посредством команды `wsl --install`, которую необходимо запустить в командной строке Windows. Для этого нужно нажать Win+R и в появившемся окне напечатать cmd, затем нажать Enter:

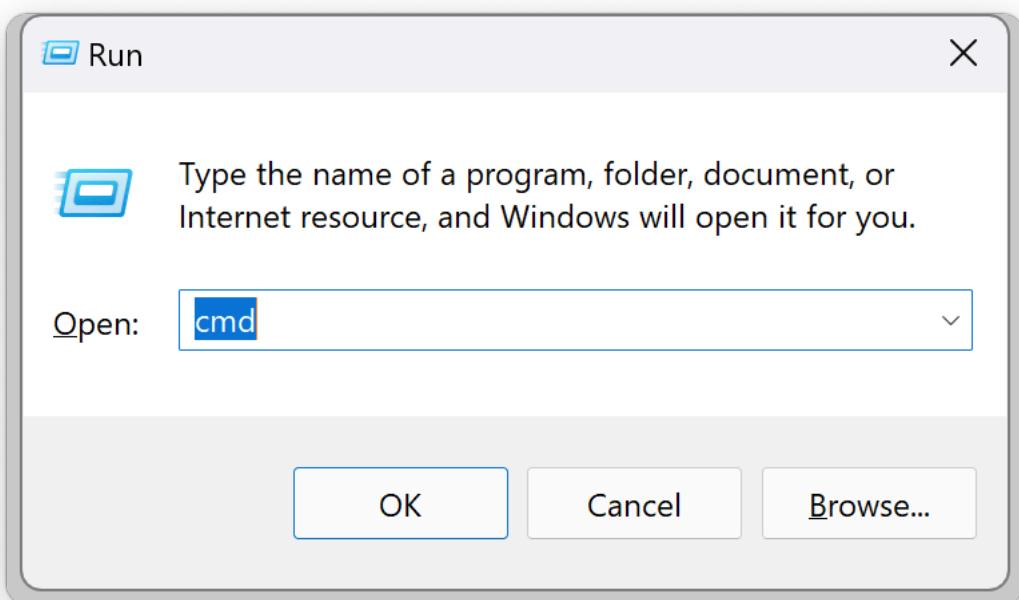


Рис. 11: Окно запуска команд Windows

После чего откроется окно командной строки Windows:

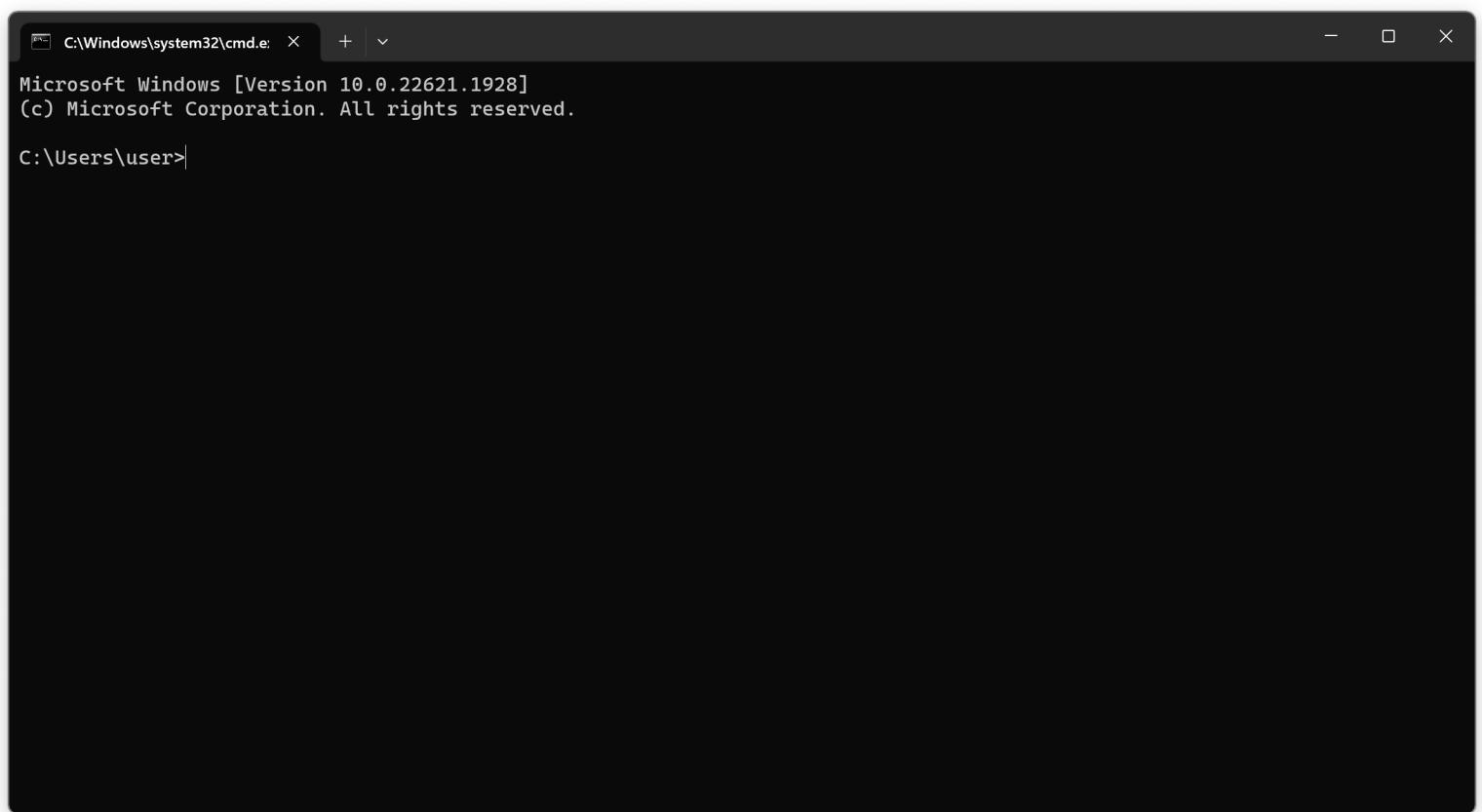
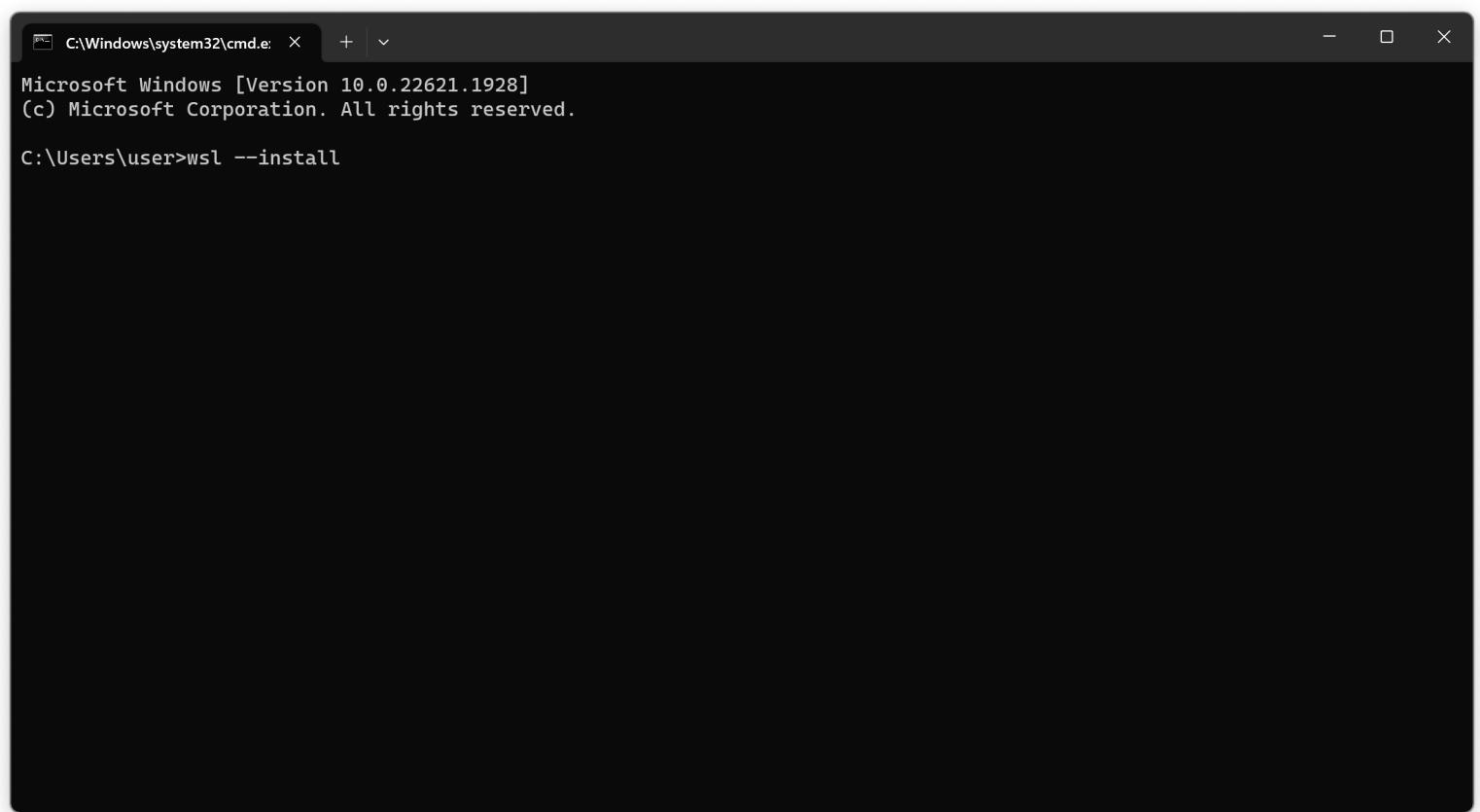


Рис. 12: Окно командной строки Windows

Оно может незначительно отличаться в последних версиях Windows 10 и Windows 11.

Вводим команду `wsl --install` и нажимаем Enter



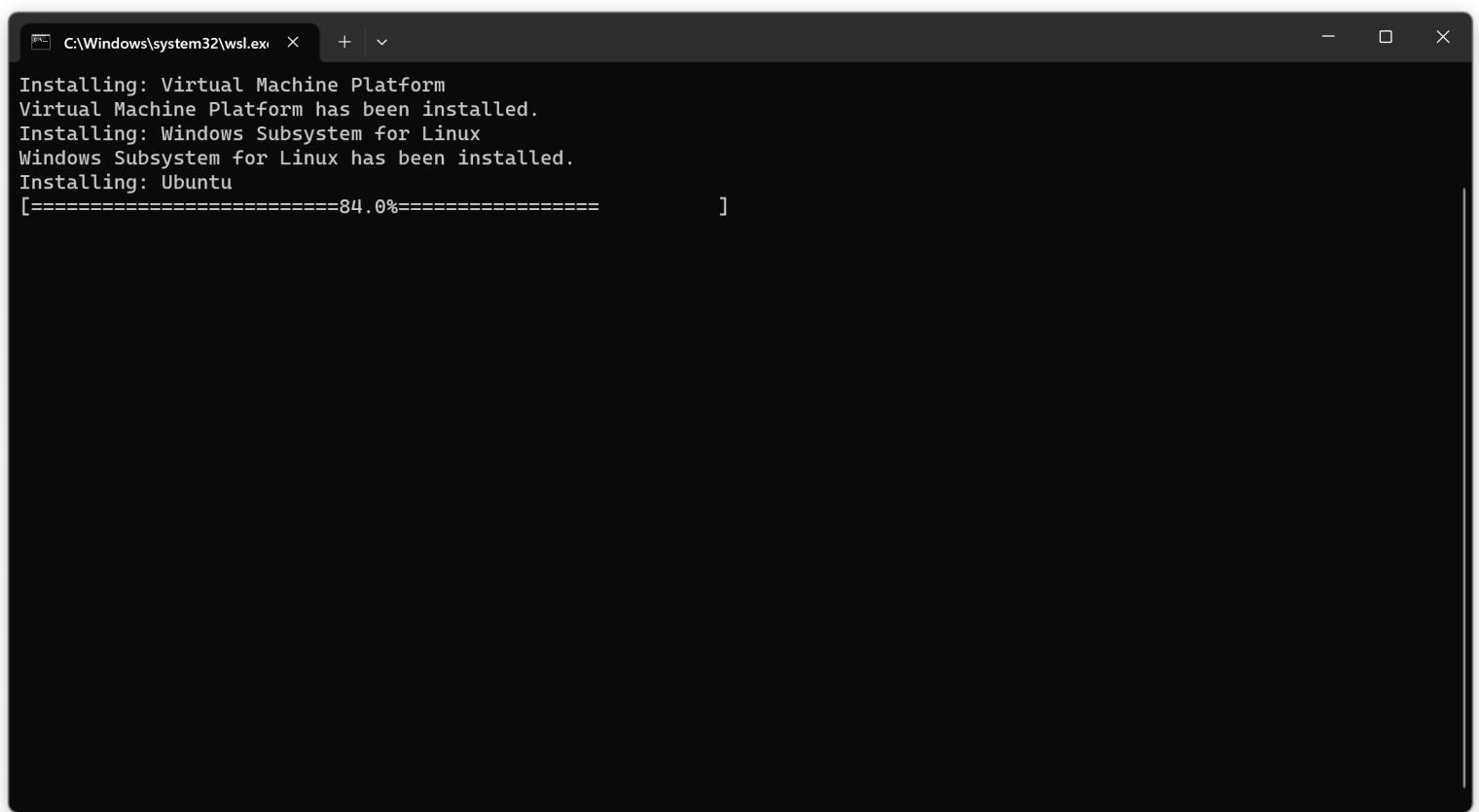
C:\Windows\system32\cmd.e: + v

Microsoft Windows [Version 10.0.22621.1928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>wsl --install

Рис. 13: Окно командной строки Windows. Ввод команды `wsl --install`

После чего начнется установка системных компонентов Windows:



A screenshot of a Windows Command Prompt window titled "C:\Windows\system32\wsl.exe". The window shows the following text output:

```
Installing: Virtual Machine Platform
Virtual Machine Platform has been installed.
Installing: Windows Subsystem for Linux
Windows Subsystem for Linux has been installed.
Installing: Ubuntu
[=====84.0%===== ]
```

Рис. 14: Окно командной строки Windows. Процесс установки системных компонентов Windows

Во время установки компонентов, могут быть запрошены разрешения на модификацию файлов и доступ к различным службам, это абсолютно нормально, следует разрешить доступ, нажав “Yes”:

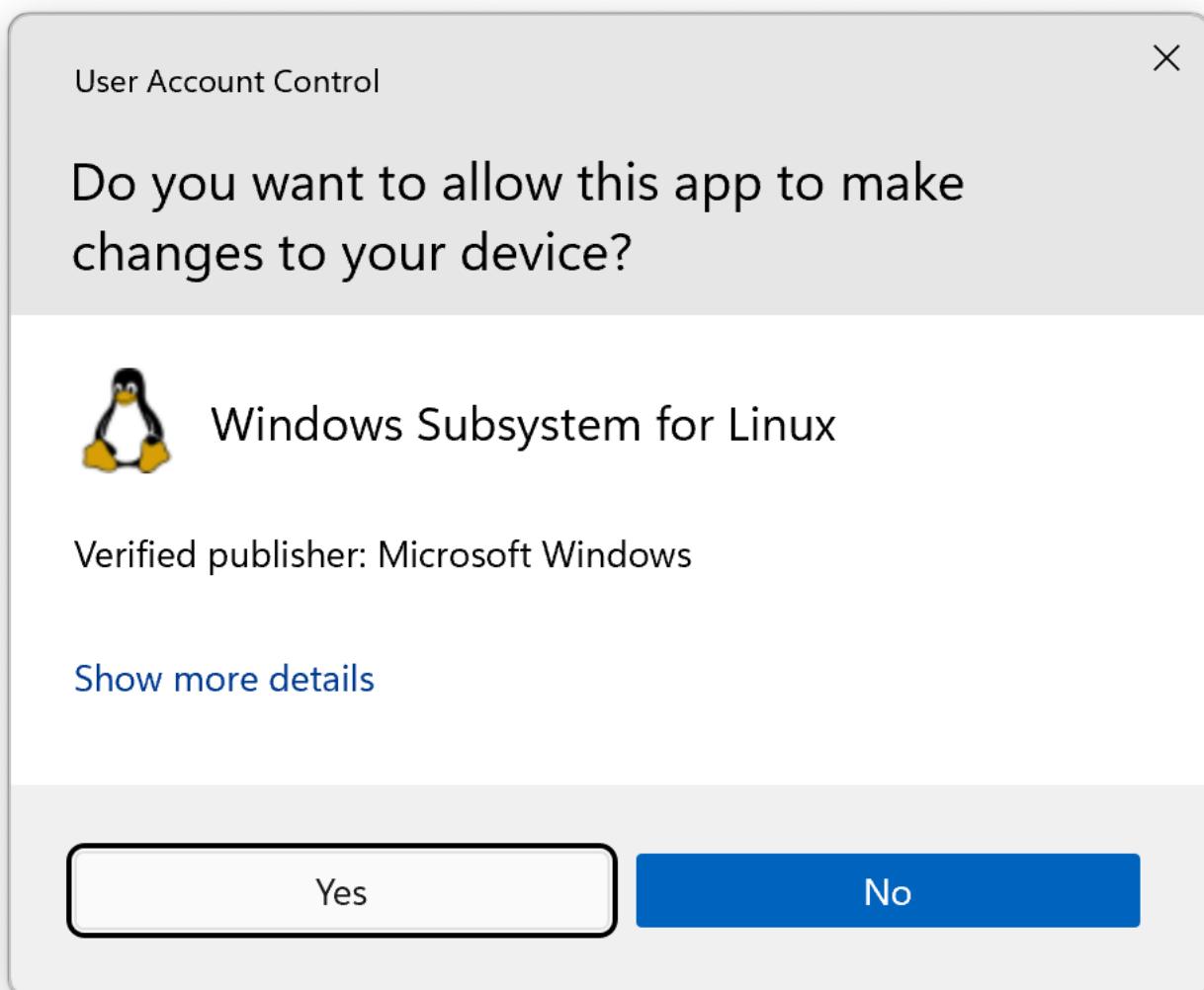
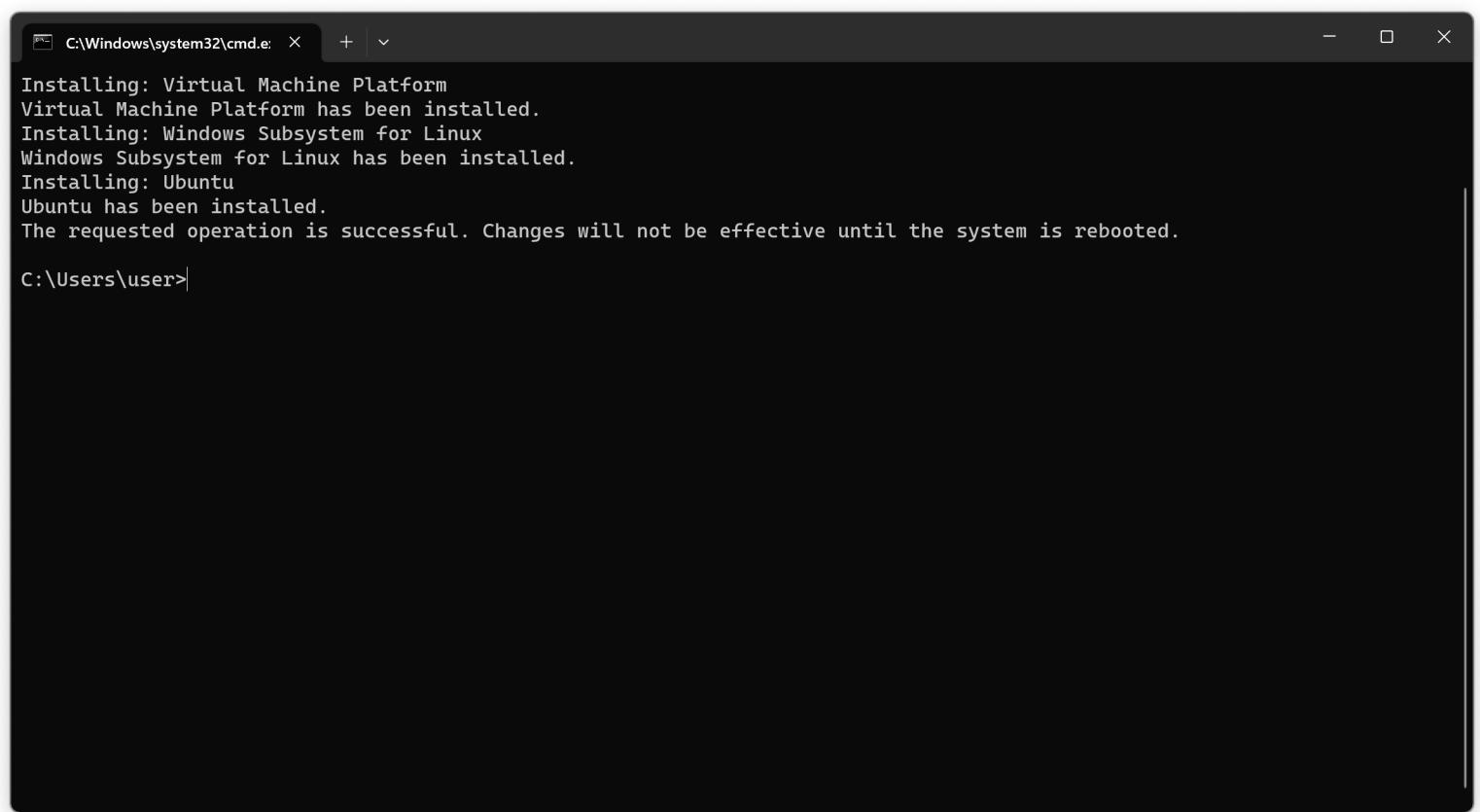


Рис. 15: Запрос подтверждения доступа к системным файлам и службам Windows

После того как установка системных компонентов Windows завершится wsl сообщит, что все готово и изменения вступят в силу после перезагрузки Windows:



A screenshot of a Windows Command Prompt window titled 'C:\Windows\system32\cmd.e'. The window contains the following text output:

```
Installing: Virtual Machine Platform
Virtual Machine Platform has been installed.
Installing: Windows Subsystem for Linux
Windows Subsystem for Linux has been installed.
Installing: Ubuntu
Ubuntu has been installed.
The requested operation is successful. Changes will not be effective until the system is rebooted.

C:\Users\user>
```

Рис. 16: Окно командной строки Windows. Завершение установки

После чего нужно перезагрузить Windows штатным образом: Start -> Power -> Restart. Когда Windows перезагрузится, откроется окно командной строки Windows, в котором продолжится настройка:

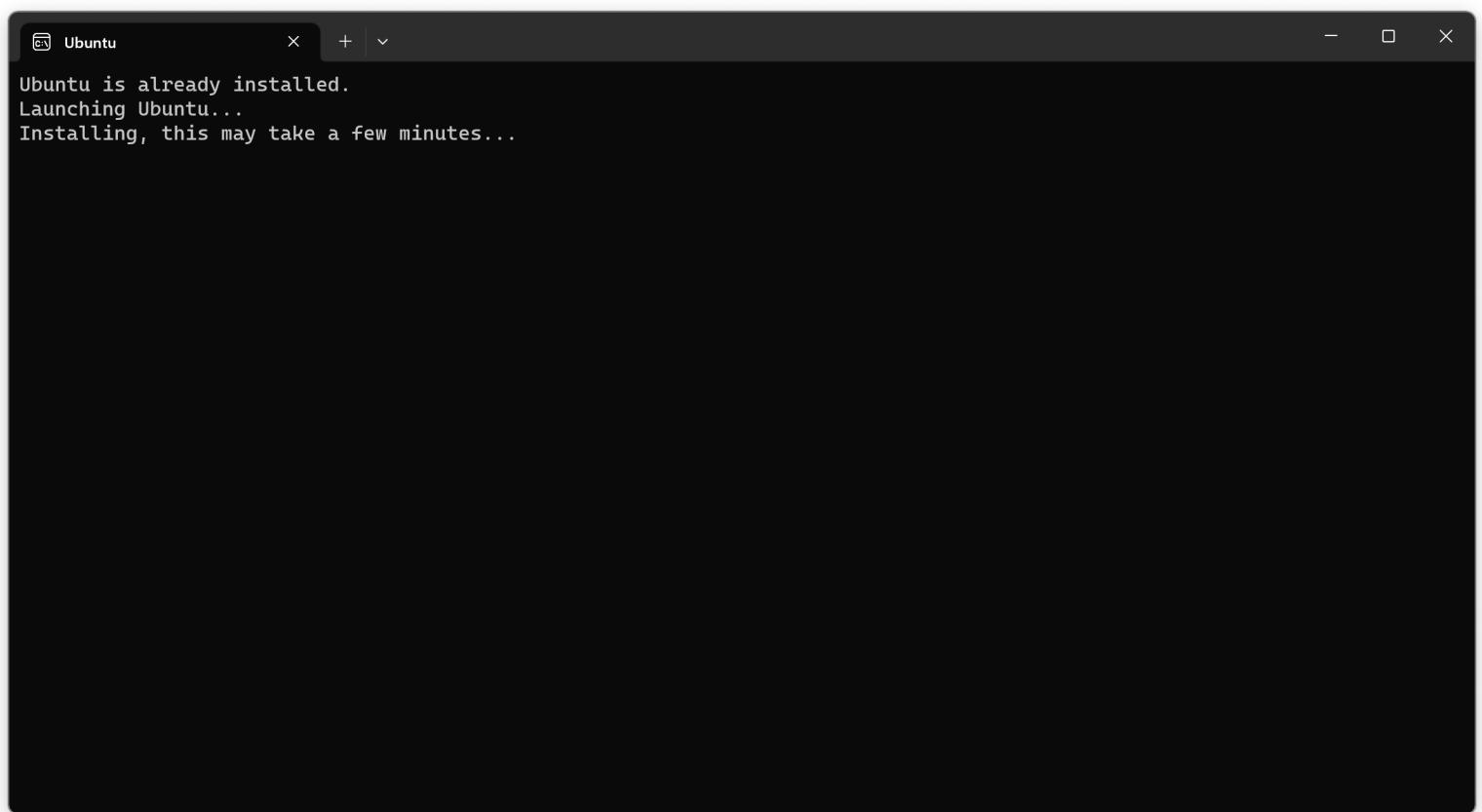


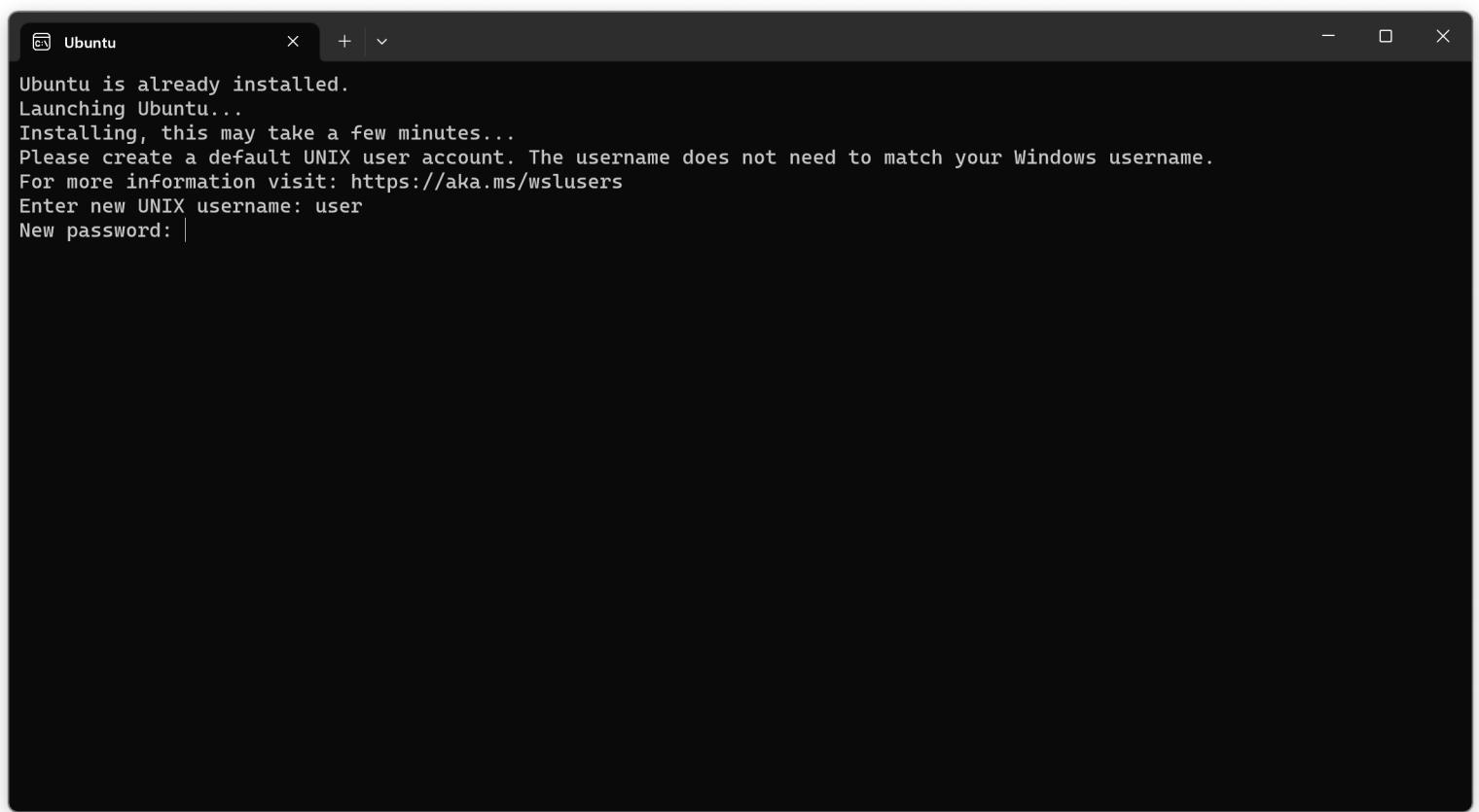
Рис. 17: Окно командной строки Windows. Настройка после перезагрузки

Когда процесс настройки завершится, будет запрошено имя нового пользователя Unix:

```
Ubuntu is already installed.  
Launching Ubuntu...  
Installing, this may take a few minutes...  
Please create a default UNIX user account. The username does not need to match your Windows username.  
For more information visit: https://aka.ms/wslusers  
Enter new UNIX username:
```

Рис. 18: Окно командной строки Windows. Создание нового пользователя Unix

Вводим имя нового пользователя, его можно придумать новое, использовать такое же как имя учетной записи в Windows, либо просто использовать имя пользователя “user”. Затем будет запрошен пароль для этого пользователя:



Ubuntu is already installed.
Launching Ubuntu...
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: <https://aka.ms/wslusers>
Enter new UNIX username: user
New password: |

Рис. 19: Окно командной строки Windows. Пароль для нового пользователя Unix

ВНИМАНИЕ: Пароль не будет отображаться при вводе, ни звездочками ни какими-либо иными символами

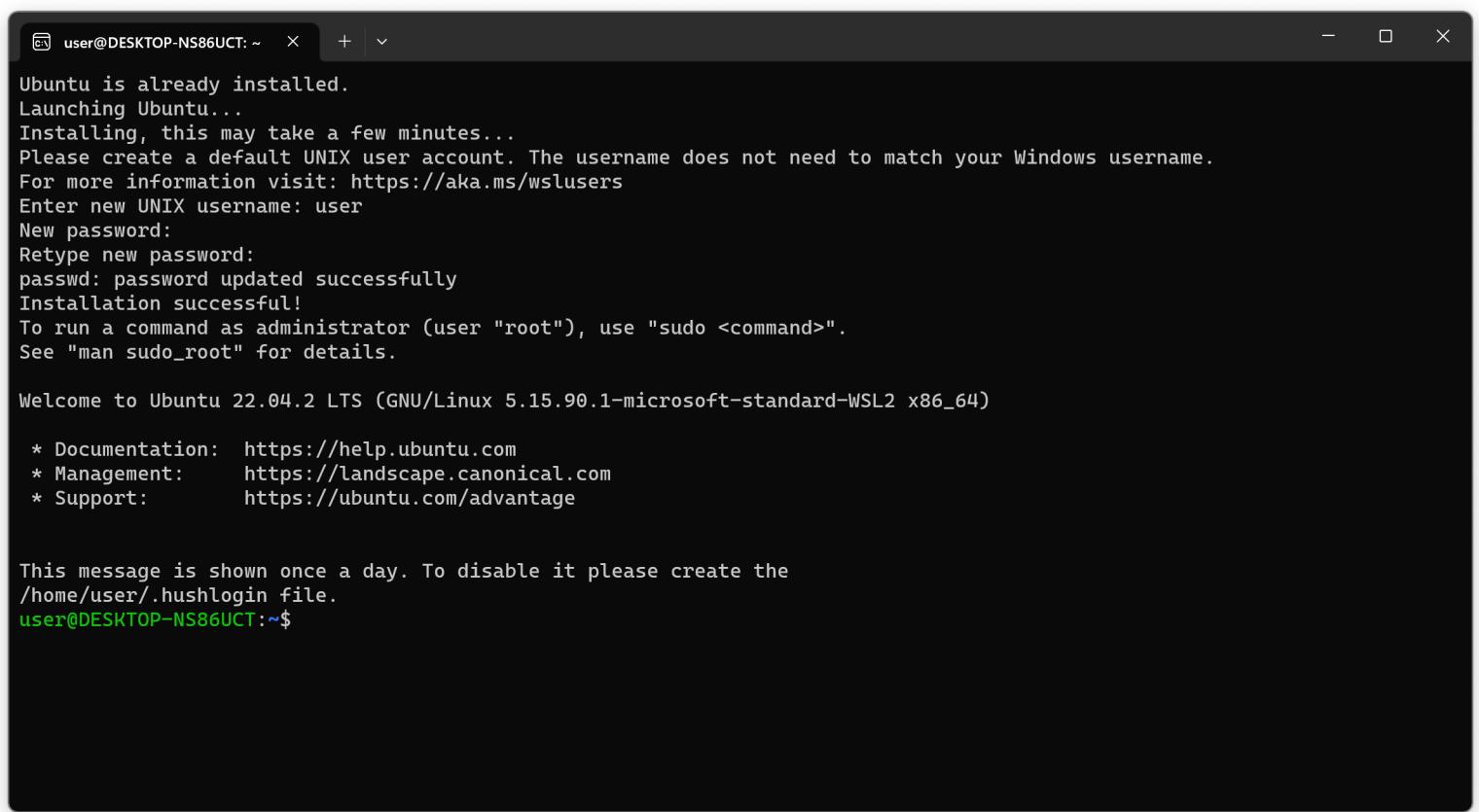
Вводим пароль и нажимаем Enter, после чего пароль будет запрошен еще раз для подтверждения:

The screenshot shows a Windows terminal window titled "Ubuntu". The window contains the following text:

```
Ubuntu is already installed.  
Launching Ubuntu...  
Installing, this may take a few minutes...  
Please create a default UNIX user account. The username does not need to match your Windows username.  
For more information visit: https://aka.ms/wslusers  
Enter new UNIX username: user  
New password:  
Retype new password: |
```

Рис. 20: Окно командной строки Windows. Пароль для нового пользователя Unix, подтверждение

После ввода пароля, WSL Сообщит что успешно установлен и настроен:



user@DESKTOP-NS86UCT: ~

```
Ubuntu is already installed.
Launching Ubuntu...
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: user
New password:
Retype new password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)

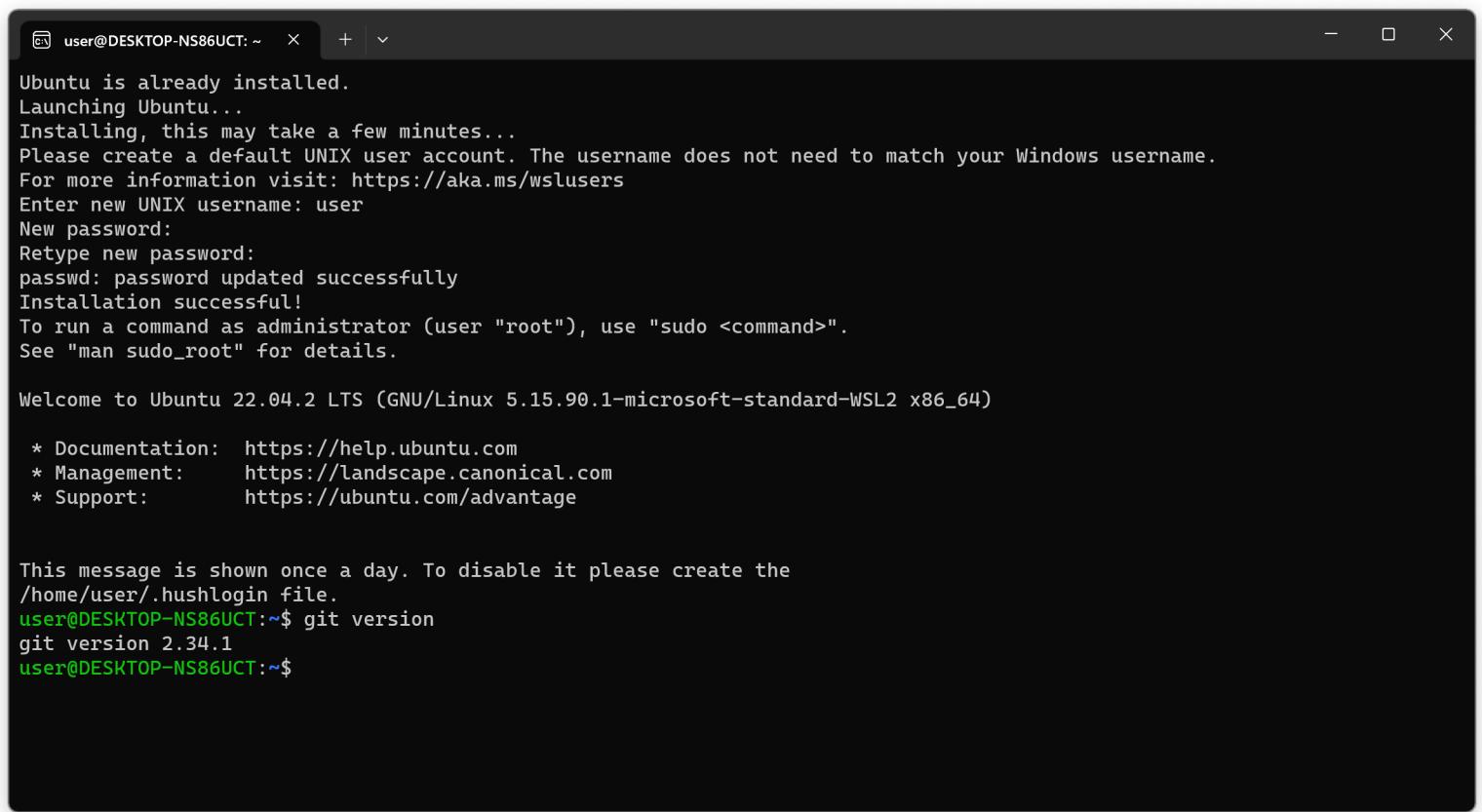
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This message is shown once a day. To disable it please create the
/home/user/.hushlogin file.
user@DESKTOP-NS86UCT:~$
```

Рис. 21: Окно командной строки Windows. Подтверждение установки WSL

3.3 Установка Git

После установки WSL отдельно устанавливать Git не требуется, он уже установлен в образе Ubuntu 22.04. Убедиться в этом можно, введя команду `git version`:



```
user@DESKTOP-NS86UCT: ~ + \xUbuntu is already installed.  
Launching Ubuntu...  
Installing, this may take a few minutes...  
Please create a default UNIX user account. The username does not need to match your Windows username.  
For more information visit: https://aka.ms/wslusers  
Enter new UNIX username: user  
New password:  
Retype new password:  
passwd: password updated successfully  
Installation successful!  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)  
  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/advantage  
  
This message is shown once a day. To disable it please create the  
/home/user/.hushlogin file.  
user@DESKTOP-NS86UCT:~$ git version  
git version 2.34.1  
user@DESKTOP-NS86UCT:~$
```

Рис. 22: Окно командной строки Windows. Подтверждение установки WSL

3.4 Установка Visual Studio Code

На официальном сайте приложения - code.visualstudio.com есть прямая ссылка для скачивания Visual Studio Code:

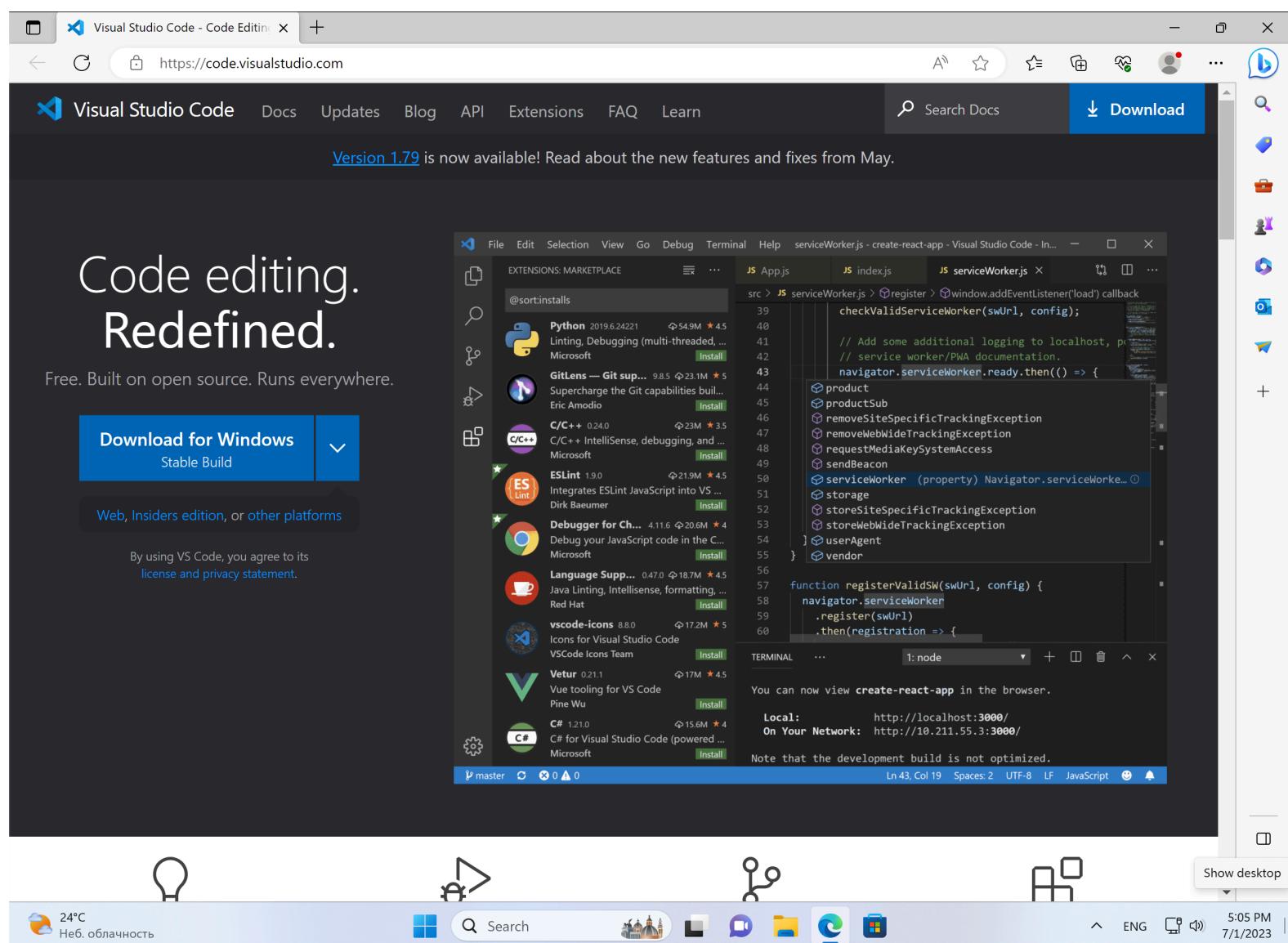


Рис. 23: Сайт code.visualstudio.com

Нажимаем “Download for Windows” - начнется скачивание установщика. Запускаем его кликом левой кнопки мыши из списка загрузок:

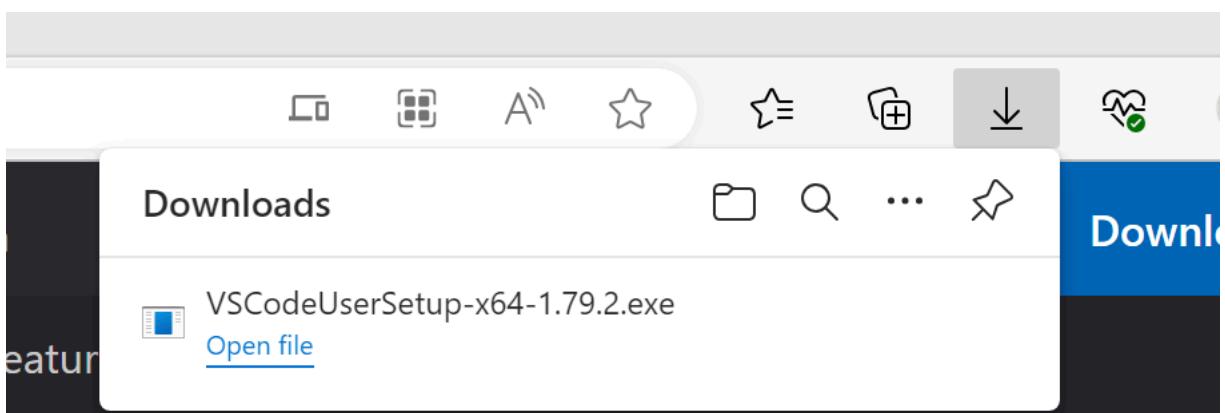


Рис. 24: Список загрузок

Когда приложение запустится - откроется его окно с лицензионным соглашением:

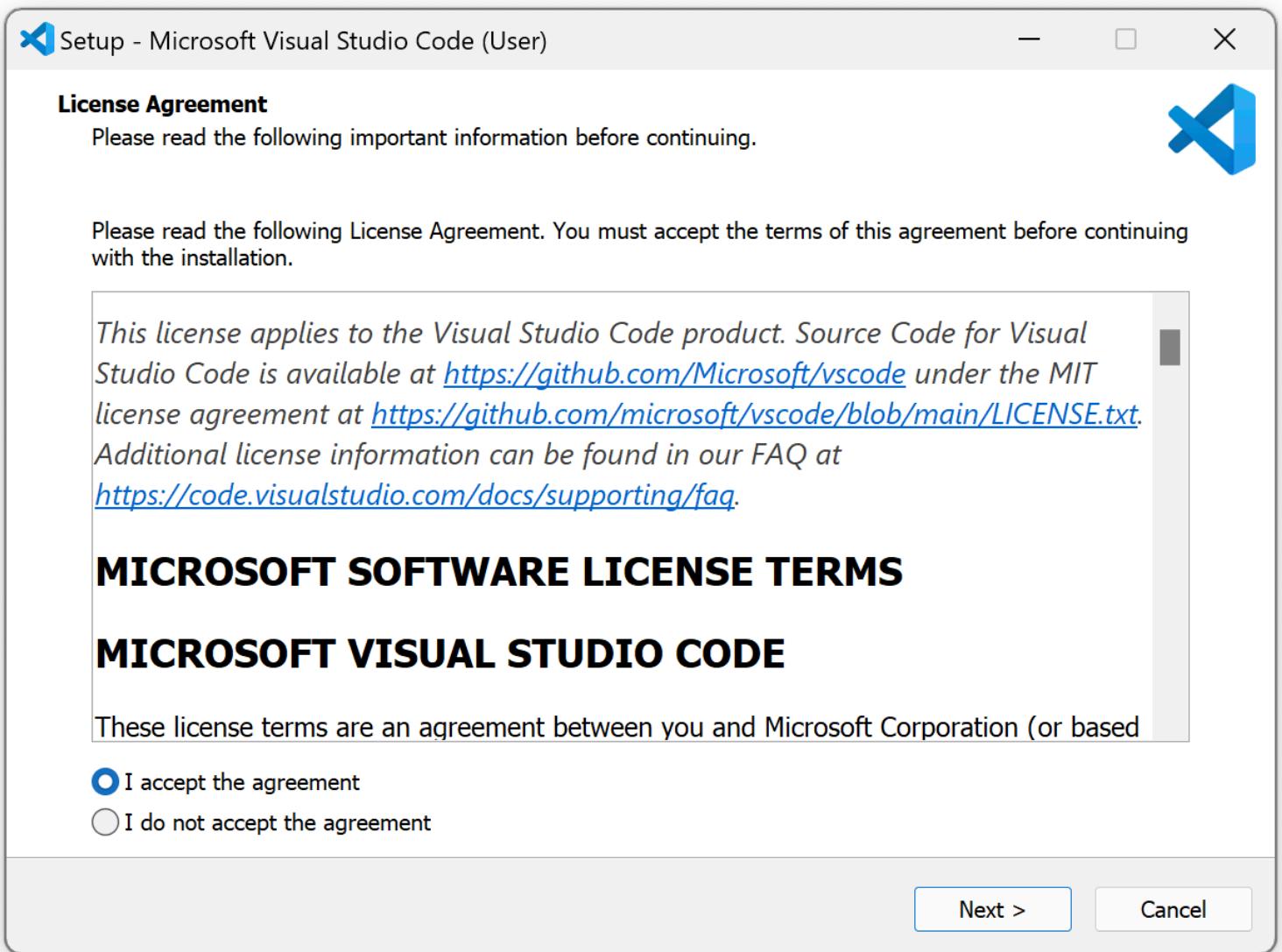


Рис. 25: Установщик Visual Studio Code. Окно лицензии

Выбираем “I accept the agreement” и нажимаем “Next >”, настройки на всех остальных экранах установщика можно не трогать и оставить по умолчанию. В конце будет последнее окно установщика с кнопкой “Install”:

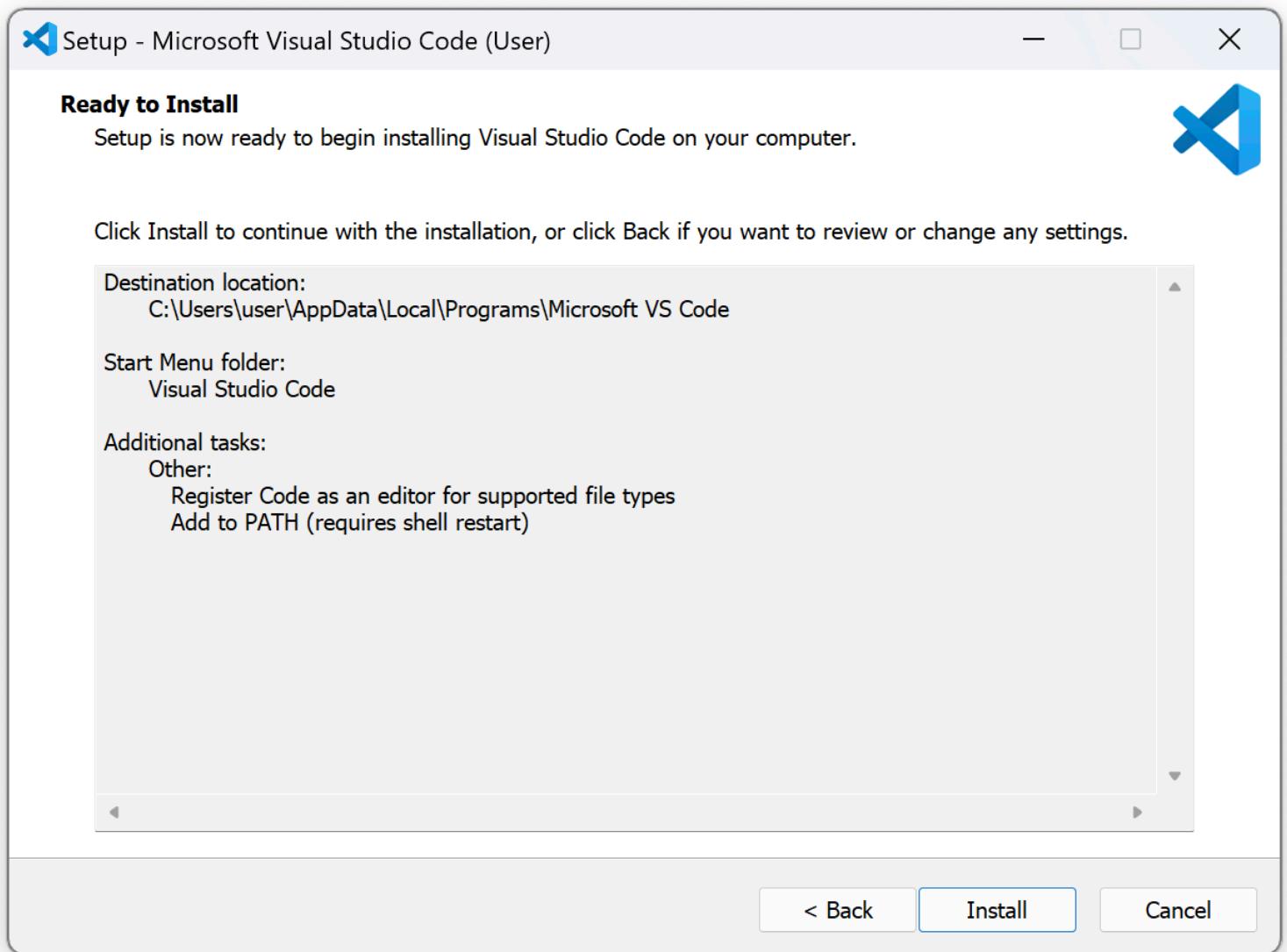


Рис. 26: Установщик Visual Studio Code. Окно подтверждения установки

Нажимаем “Install” и начнется установка приложения Visual Studio Code:

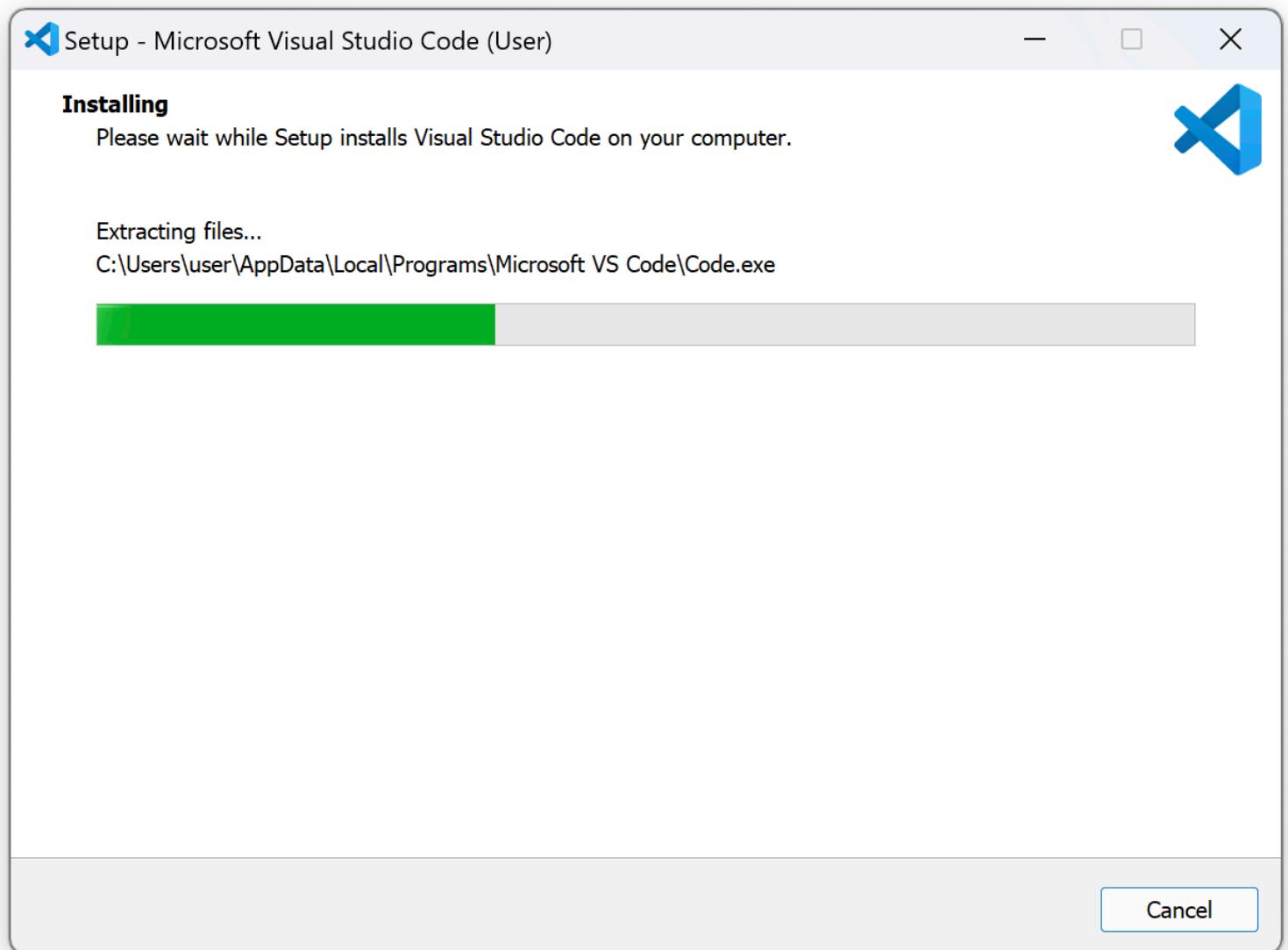


Рис. 27: Установщик Visual Studio Code. Процесс установки

По завершению установки, установщик об этом сообщит и предложит сразу запустить Visual Studio Code:

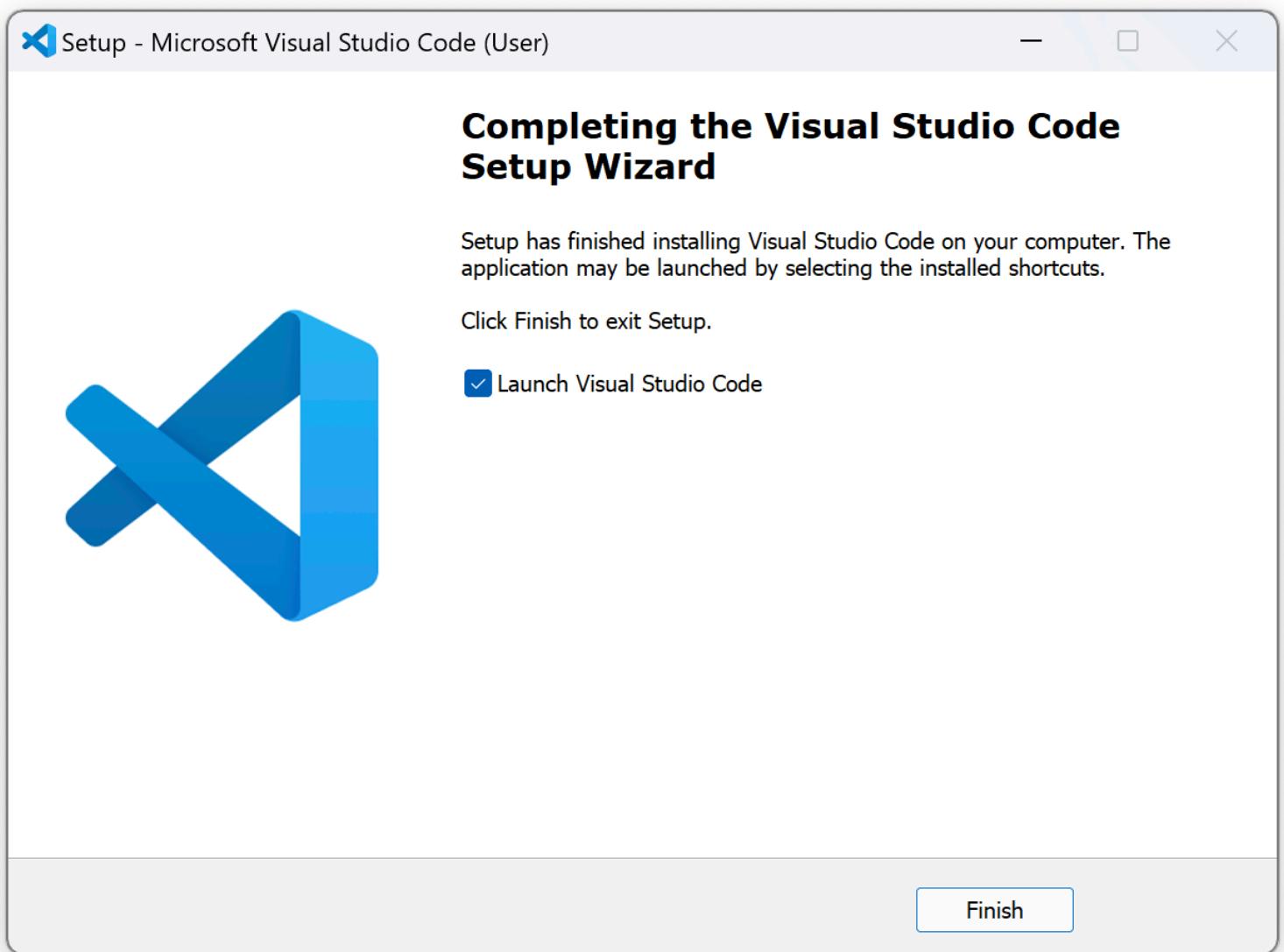


Рис. 28: Установщик Visual Studio Code. Окно подтверждения установки

Нам это подходит - просто нажимаем “Finish”. После чего запустится Visual Studio Code с окном приветствия:

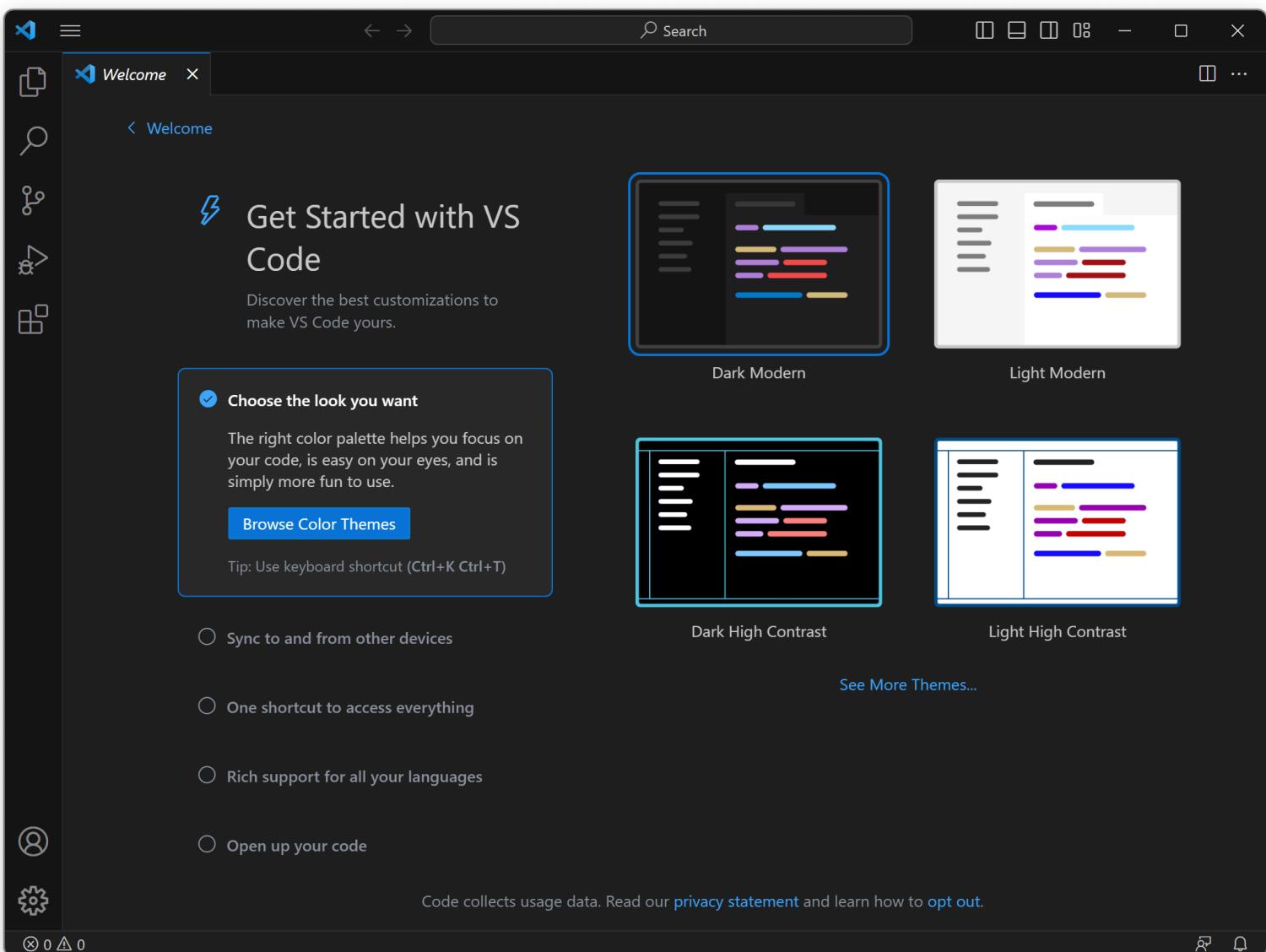


Рис. 29: Visual Studio Code. Окно приветствия

На этом установка Visual Studio Code завершена. Теперь нужно интегрировать Visual Studio Code с WSL, в котором у нас установлен Git:

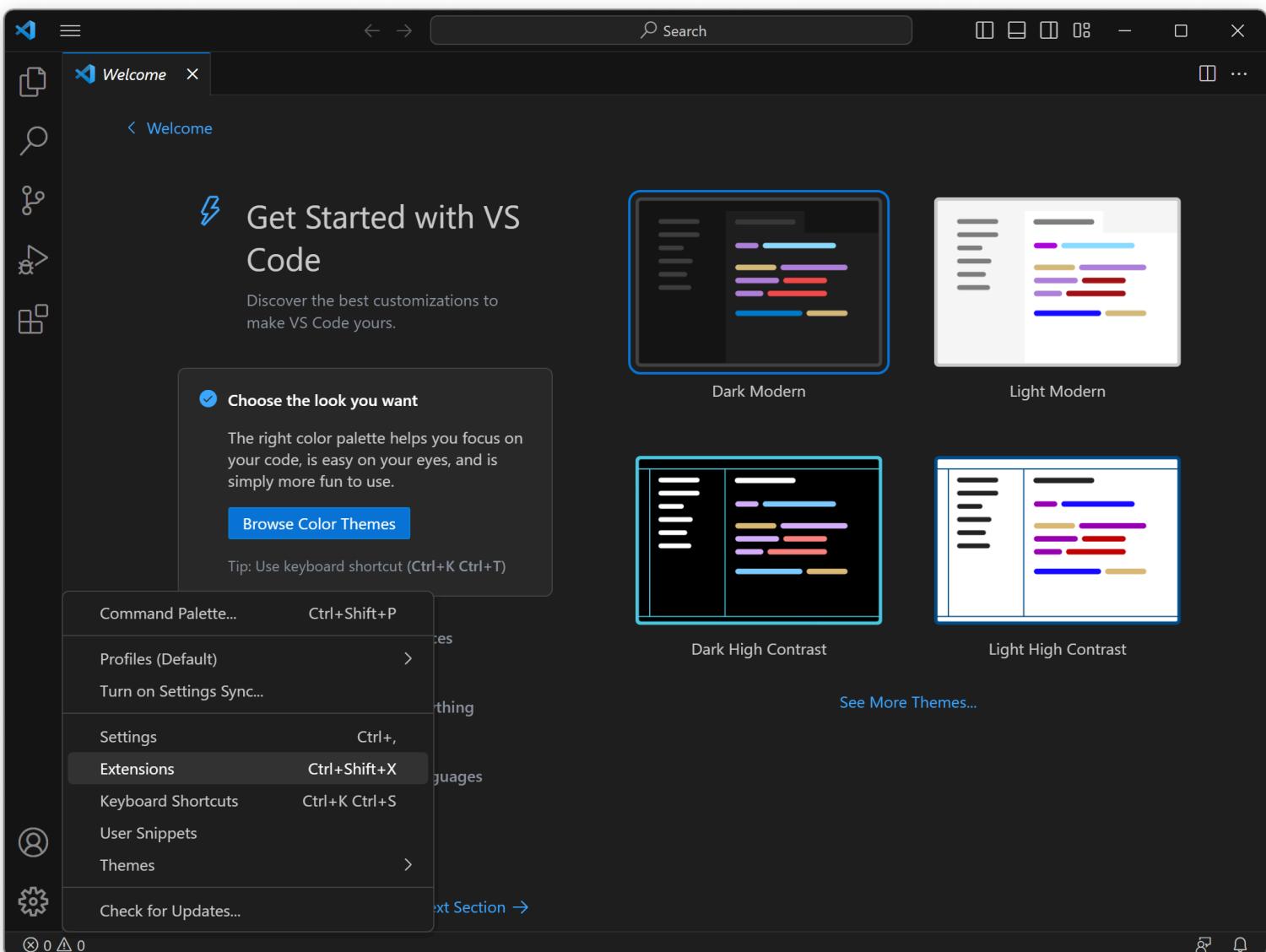


Рис. 30: Visual Studio Code. Окно расширений

Открываем меню расширений Visual Studio Code и в строке поиска (сверху) вводим `ms-vscode-remote.remote-wsl`:

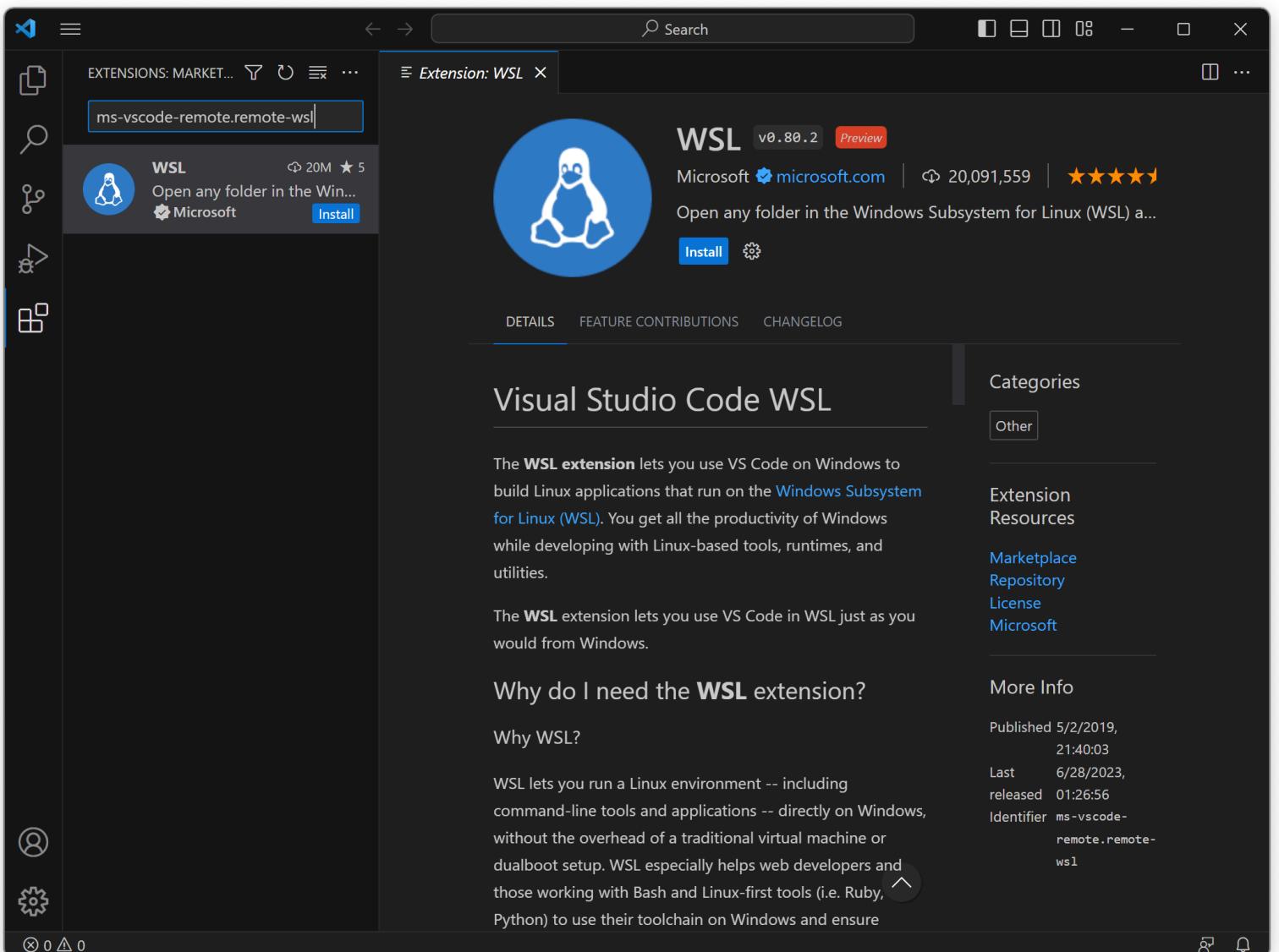


Рис. 31: Visual Studio Code. Окно расширений

Нажимаем “Install” и дожидаемся окончания установки. После чего в левом нижнем углу окна Visual Studio Code появится индикатор соединения с WSL:

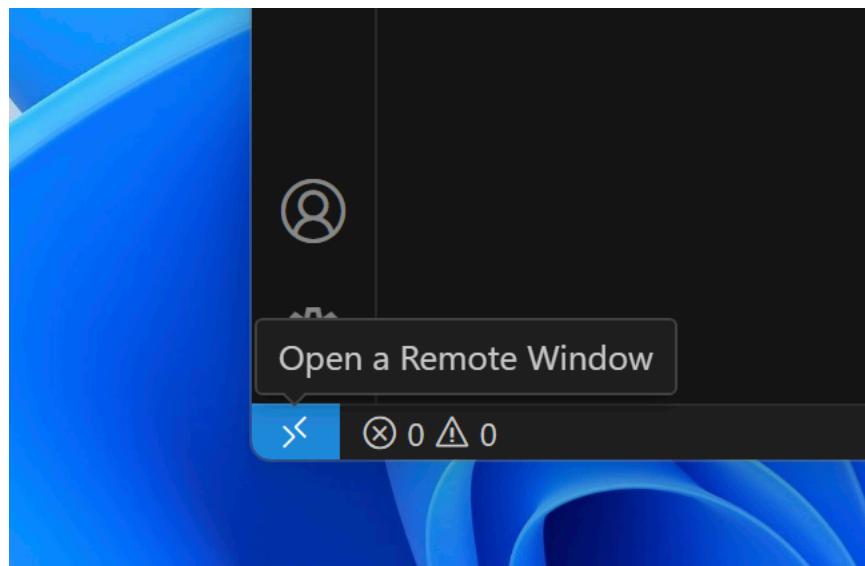


Рис. 32: Visual Studio Code. WSL Remote

На этот индикатор нужно нажать и, в появившейся сверху панели команд, выбрать “Connect to WSL”:

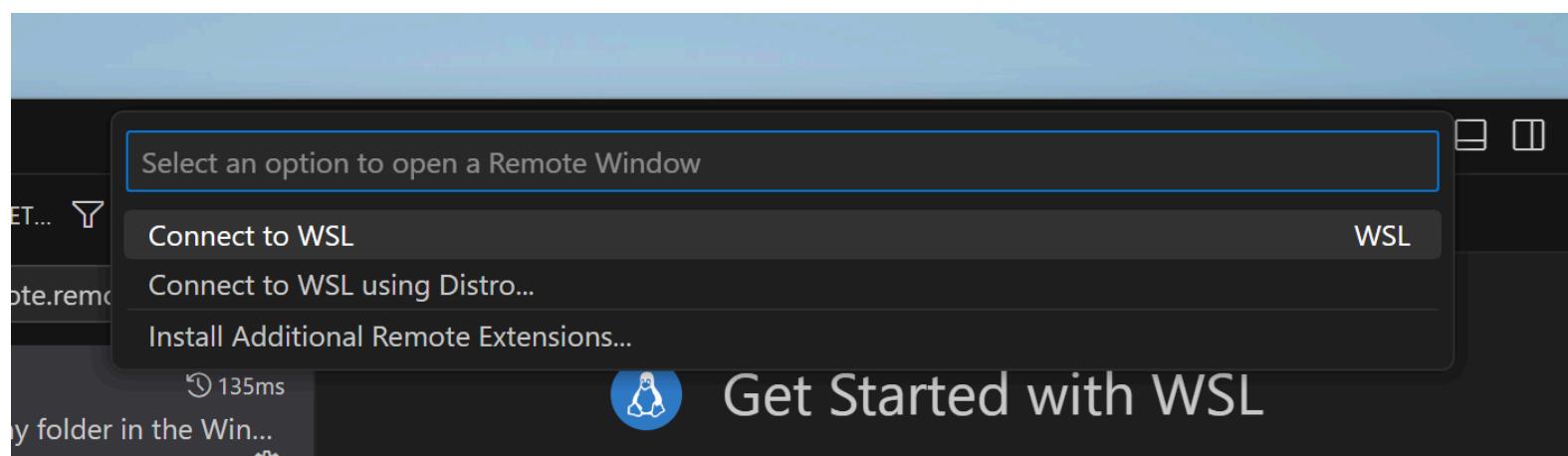


Рис. 33: Visual Studio Code. Панель команд

Начнется установка компонентов Visual Studio Code для WSL:

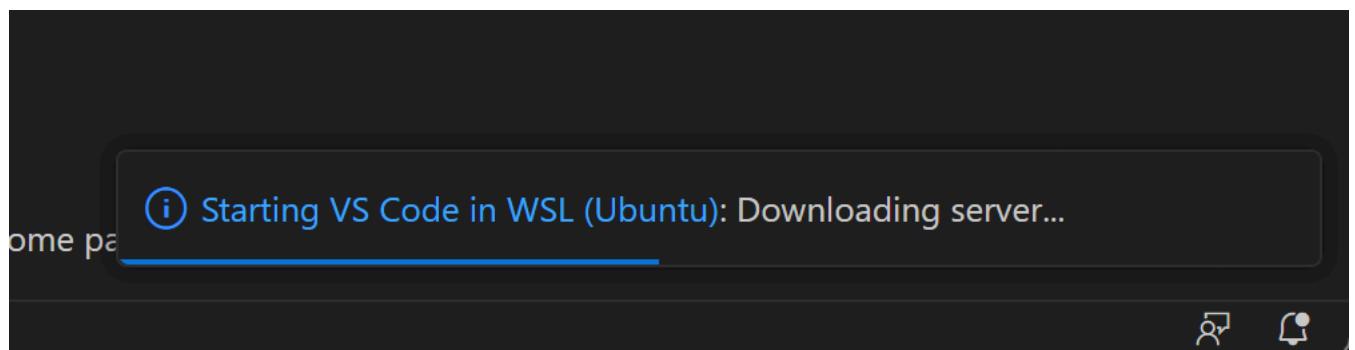


Рис. 34: Visual Studio Code. Установка компонентов Visual Studio Code в WSL

После завершения процесса установки, индикатор в нижнем левом углу покажет, что готов к работе с WSL:

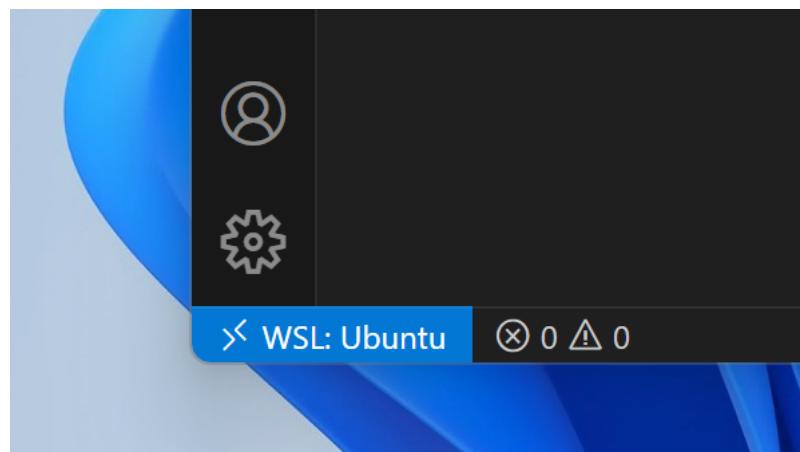


Рис. 35: Visual Studio Code. WSL Remote готов к работе

На этом установка и настройка завершена. Можно пользоваться Visual Studio Code.