

ダウンロードおよびインストール作業はお客様の責任にて行っていただけますようお願いいたします。TAC 出版では、インストールによるトラブル、不具合に対し、いかなる責任も負いません。また、お問い合わせも受け付けておりませんので、あらかじめご了承ください。

参考 gcc のインストール（2通り）

C コンパイラ gcc を使ってプログラミングをするためには、以下の環境（どちらか1通り）が必要です。

- I. Minimalist GNU for Windows (MinGW) をインストールする。
- II. WSL (Windows Subsystem for Linux) をインストールし、さらに、gcc をインストールする。

I. Minimalist GNU for Windows (MinGW) をインストールする。

1. SourceForgeより、mingw-get-setup.exe をダウンロードし、実行する。
2. MinGWをインストールする。
3. Path を設定する。

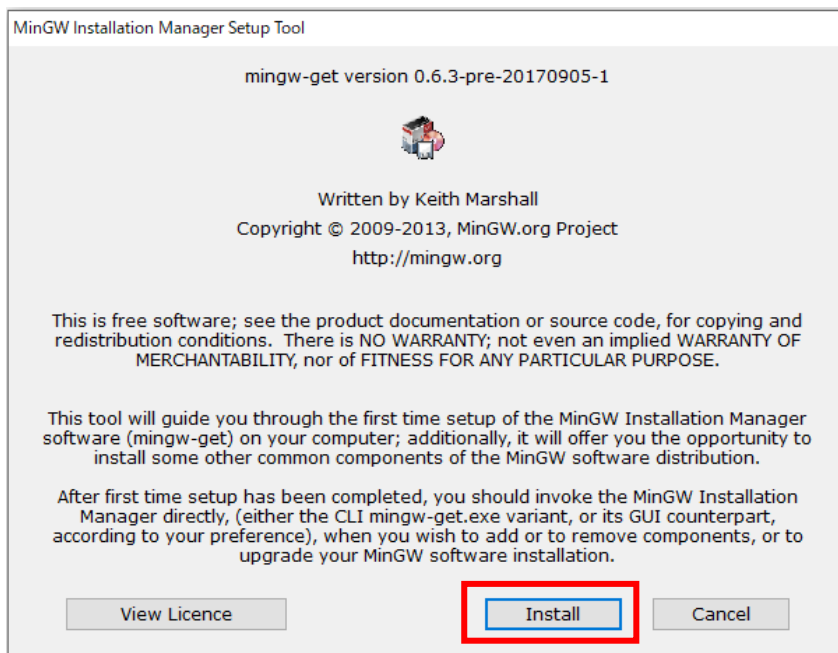
I-1 ダウンロード

<https://sourceforge.net/projects/mingw/> にアクセスし、「ダウンロード」ボタンをクリックする。

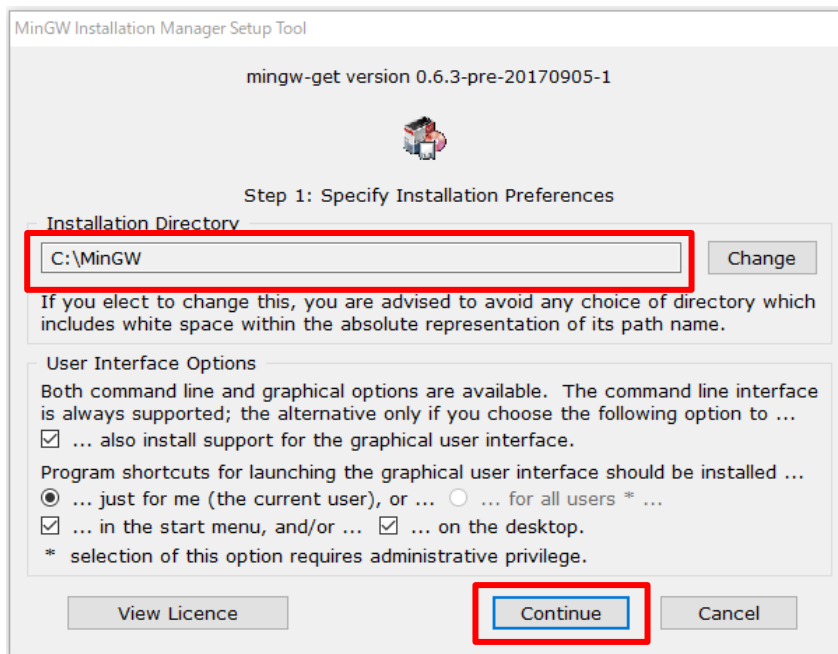
The screenshot displays the SourceForge project page for MinGW. At the top, there is a navigation bar with 'SOURCEFORGE' and links for 'Open Source Software', 'Business Software', 'Resources', 'For Vendors', and 'Help'. Below this, the project title 'MinGW - Minimalist GNU for Windows' is prominently displayed, along with a description: 'A native Windows port of the GNU Compiler Collection (GCC)'. The page also shows '196 Reviews' with a star rating, 'Downloads: 3,427,238 This Week', and 'Last Update: 2021-09-05'. A green 'Download' button is highlighted with a red box. Below the main content, there is a 'Windows' section with tabs for 'Summary', 'Files', 'Reviews', 'Support', 'News', 'Wiki', 'Mailing Lists', 'Tickets', and 'Git'. A notice at the bottom states: 'This project is in the process of moving to osdn.net/projects/mingw/, you can continue to follow us there.'

I-2 MinGW のインストール

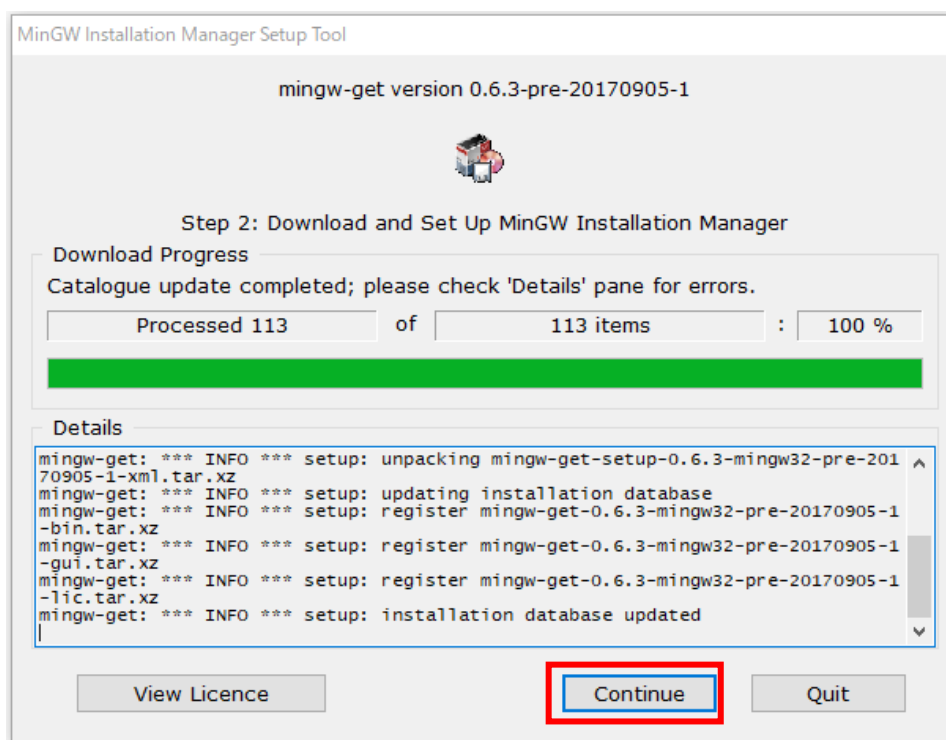
ダウンロードしたセットアップファイル (mingw-get-setup.exe) をダブルクリックして実行すると、次のようなウィンドウが現れますので、Install を選択します。



インストール先のフォルダを尋ねてきますので、デフォルトの “C:\MinGW” でよければ、そのまま Continue を選択します。

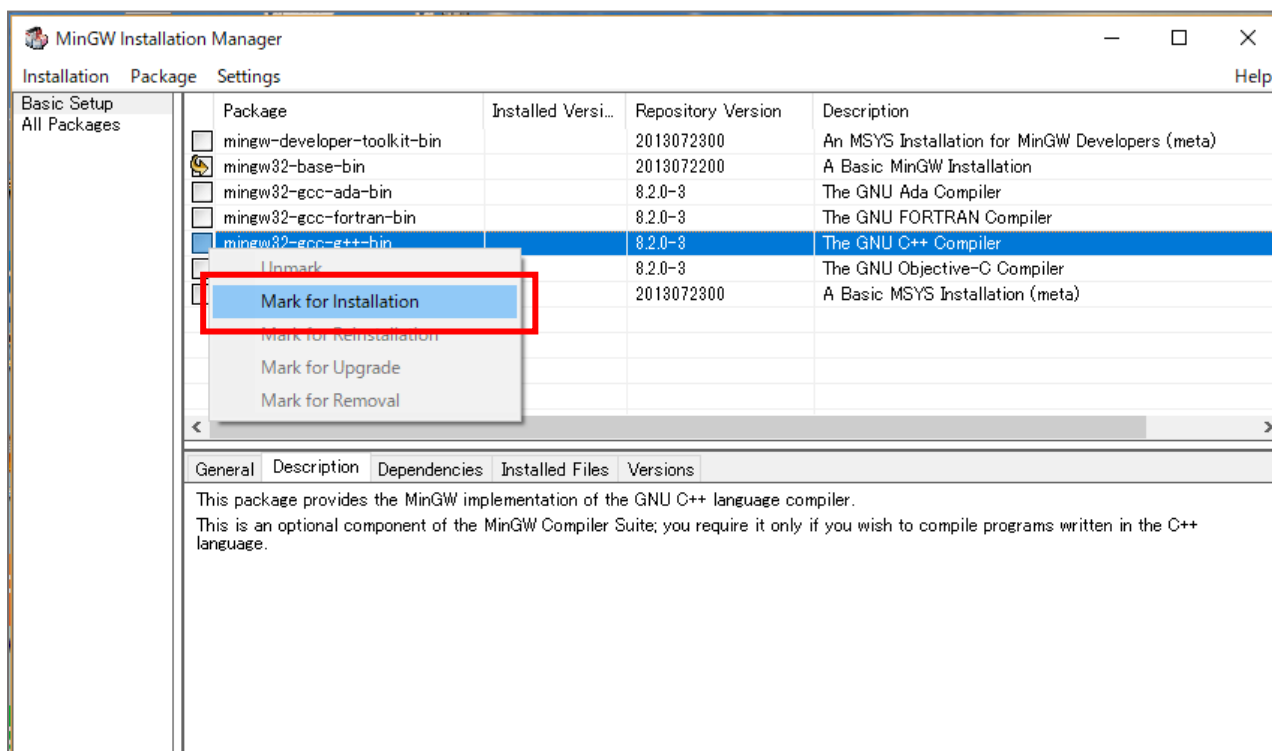


以上のことを行うと、次のようにMinGW を構成するソフトウェアのダウンロードが始まります。ダウンロードが完了したら、Continue を選択します。

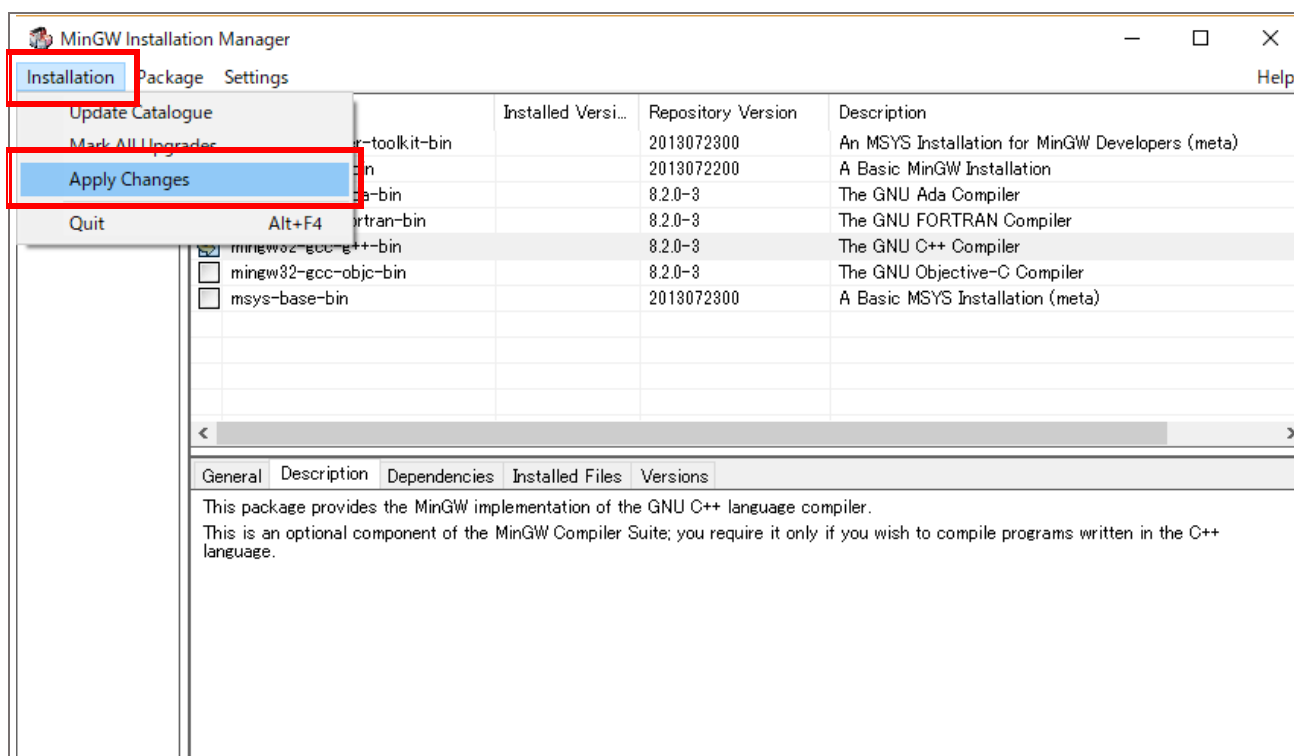


以上のことを行うと、MinGW Installation Manager のウィンドウが現れますので、次の項目をクリックしてポップアップメニューからMark for Installation を選択します。

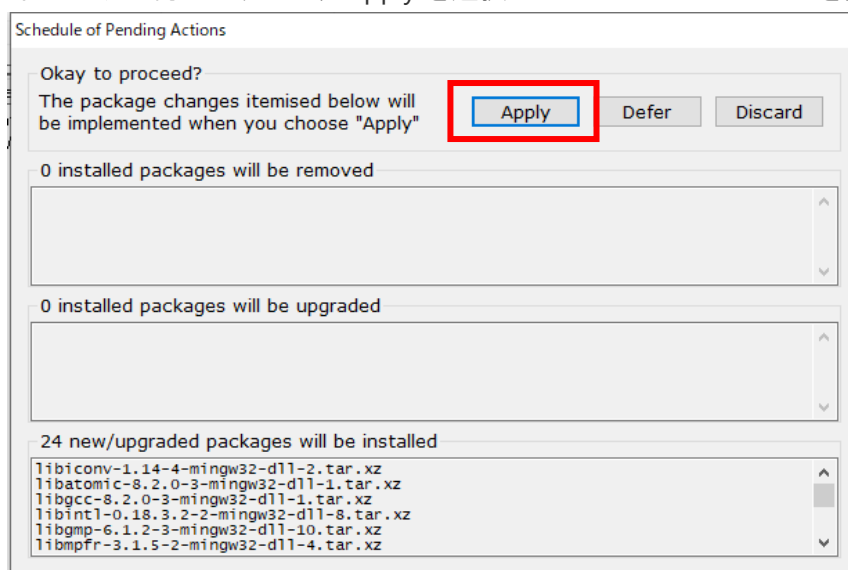
- mingw32-base-bin
- mingw32-gcc-g++-bin



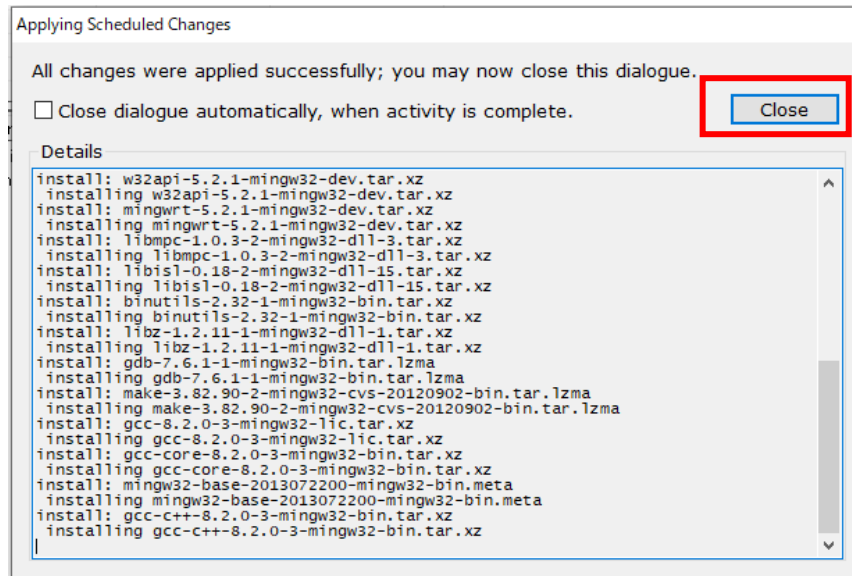
左上のツールバーから Installation → Apply Changes をクリックしてインストールの準備をします。



次のようなウィンドウが現れますので、Apply を選択してインストールを開始します。



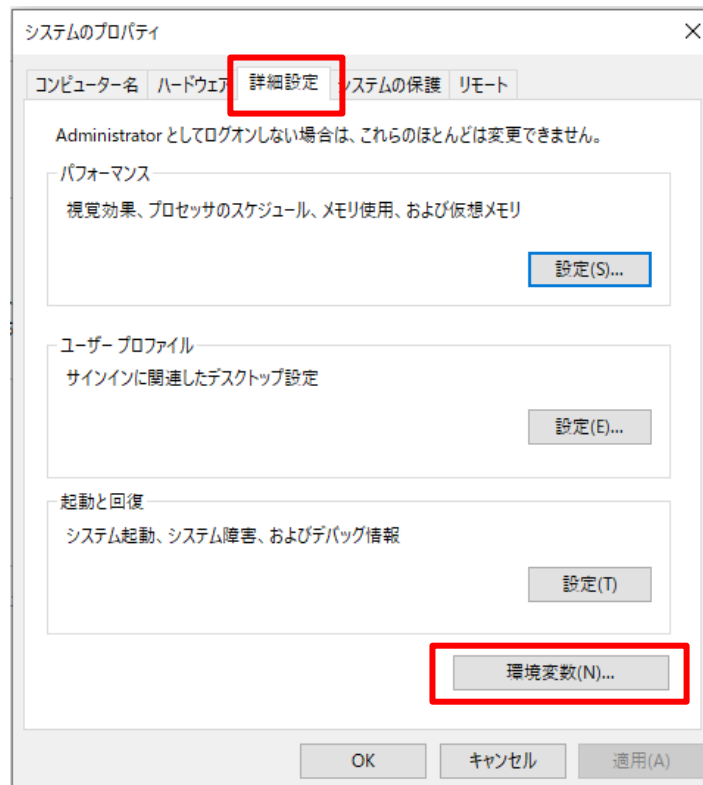
インストールが成功すると、次のようなウィンドウが現れますので、Close を選択すると終了です。



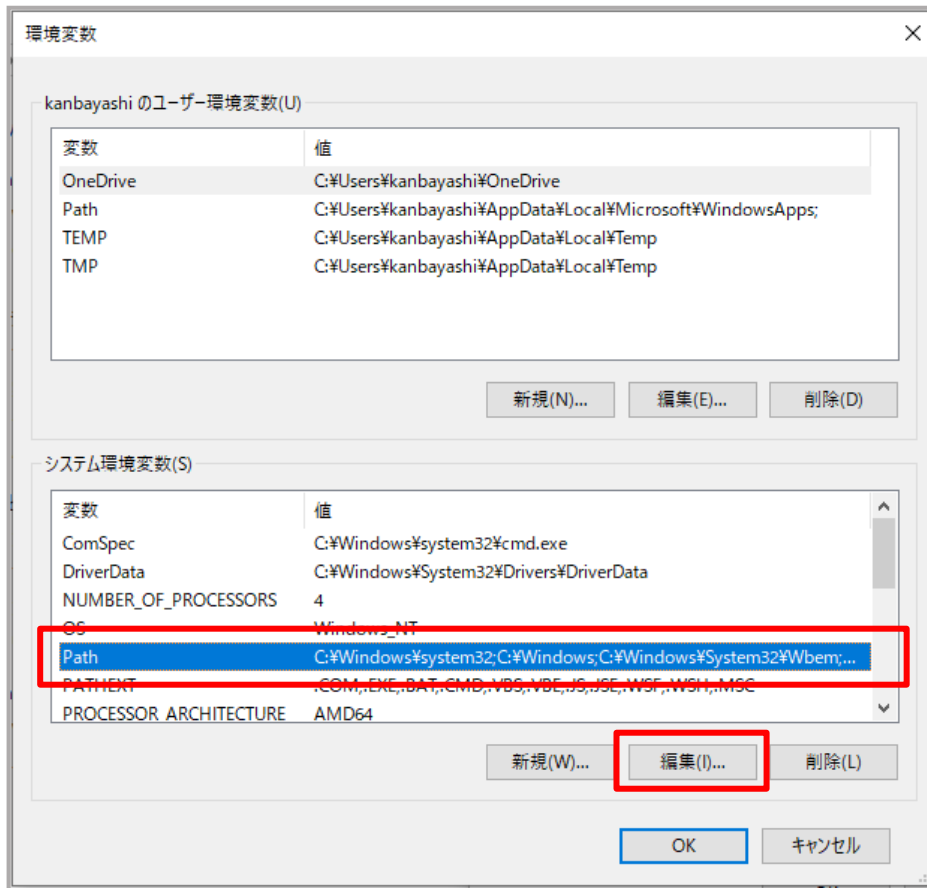
I-3 環境変数の設定

次に環境変数Path を設定します。

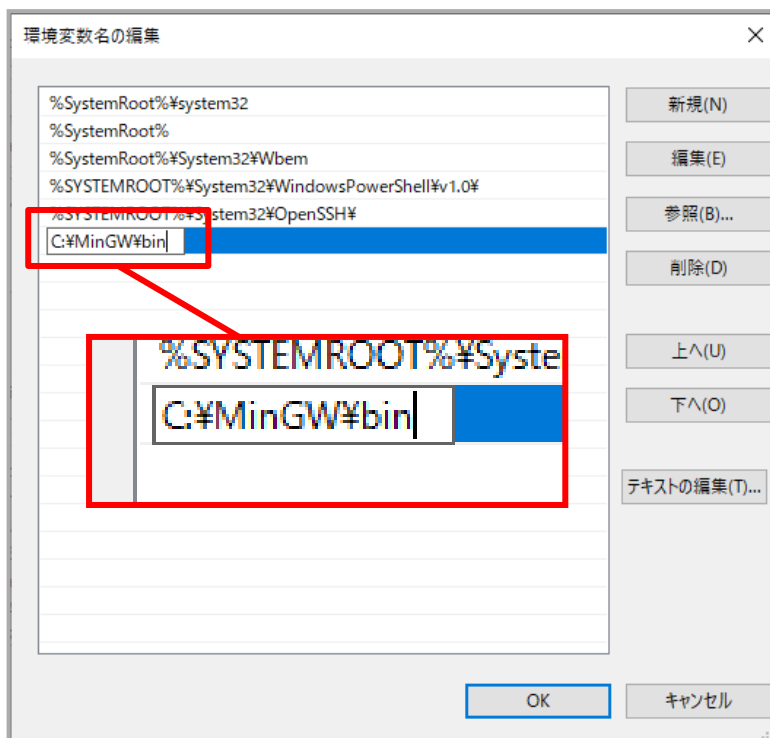
コントロールパネルから、“システムとセキュリティ” → “システム” → “システムの詳細設定” を選択すると、次のようなウィンドウが現れます。



“環境変数(N)” を選択すると、次のようなウィンドウが現れますので、“システム環境変数(S)から” Path” を選択して、“編集” をクリックします。

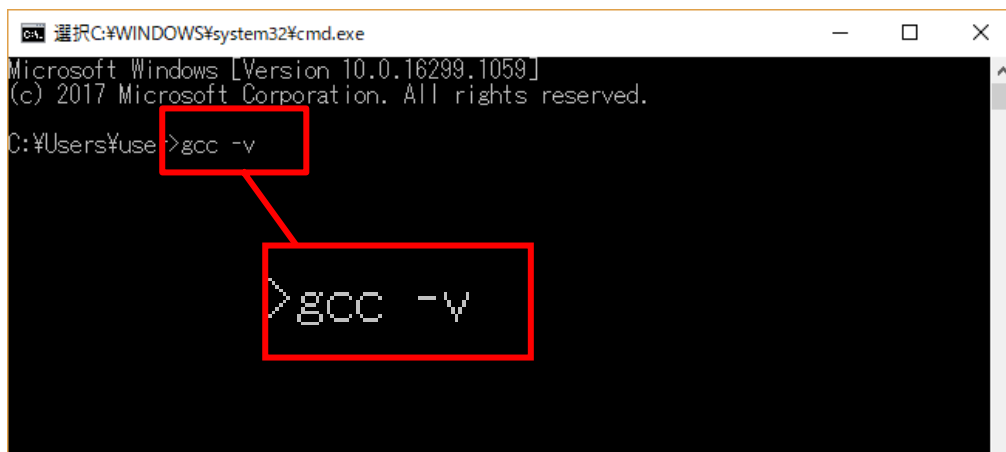
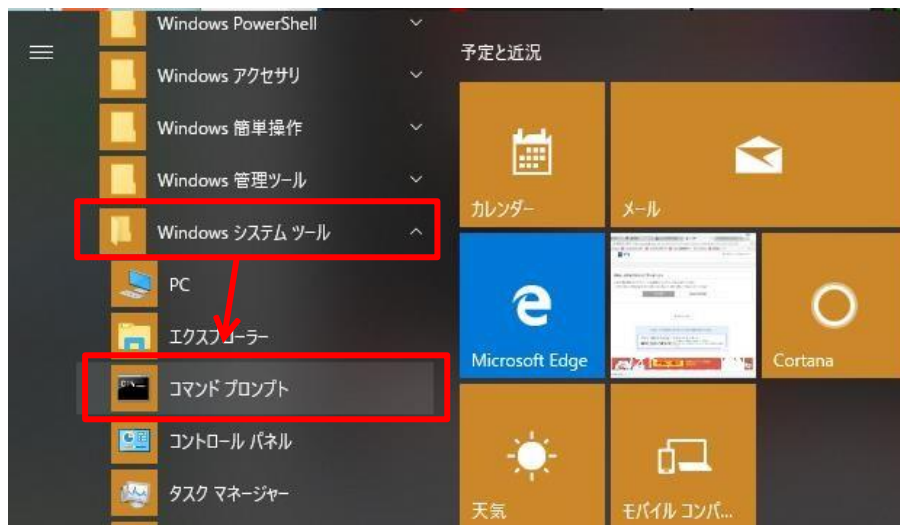


次のように Path のリストが現れますので、「新規(N)」をクリックして、新規入力ができる状態になったら、インストールのときに指定したフォルダを記入します。ここではデフォルトの“C:\MinGW\bin”を記入します。入力が終了したら“OK”をクリックして終了です。

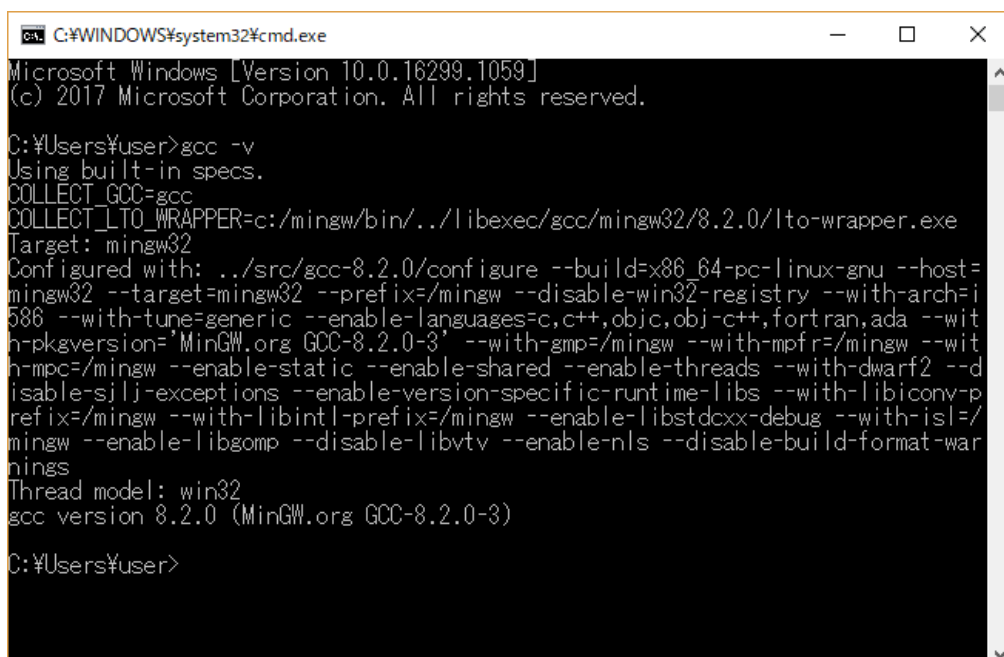


以上で MinGW のインストールが完了しました。

コマンドプロンプトを起動 ([スタート]メニュー→[Windows システムツール]→[コマンドプロンプト]) して, “gcc -v” と入力して確認しましょう。



次のようにバージョン情報が表示されます。これで gcc コンパイラを使用して C のプログラムをコンパイルできるようになりました。

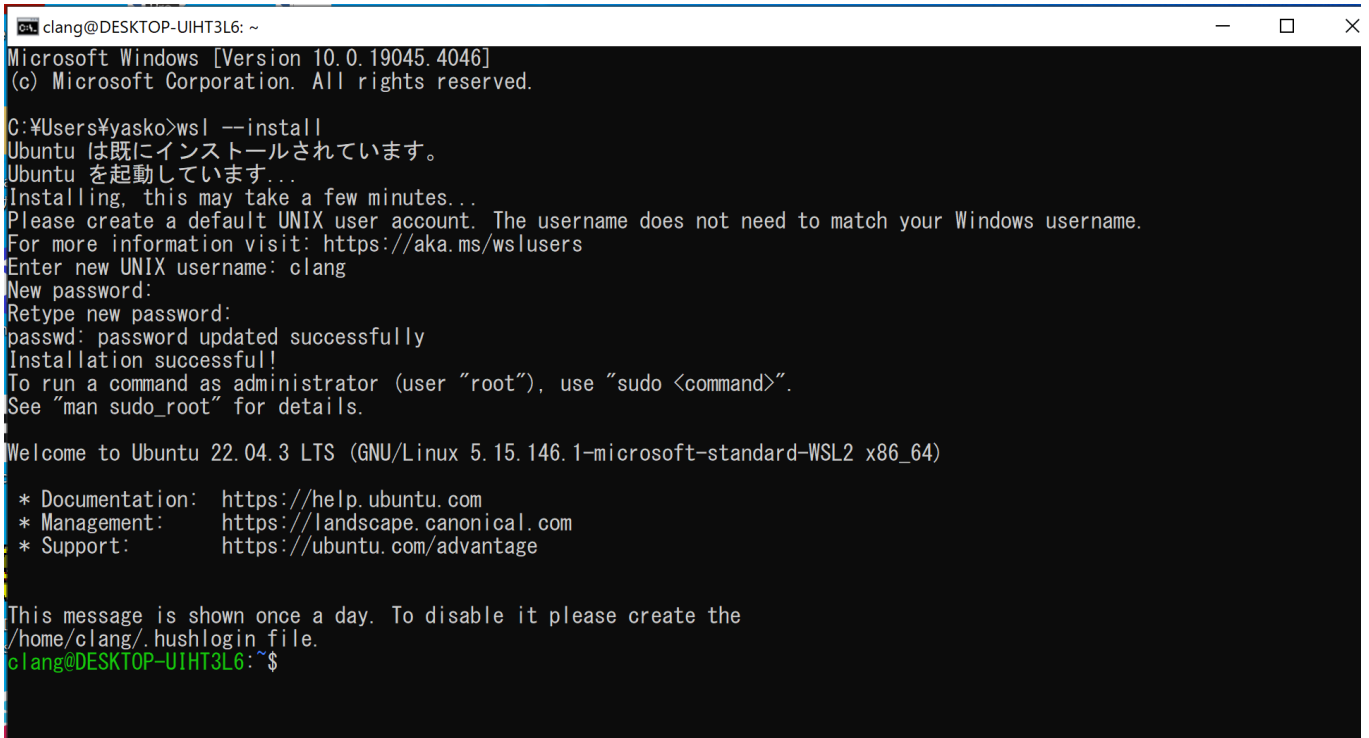


II. WSLをインストールし、さらに、gccをインストールする。

1. コマンドプロンプトを起動 ([スタート]メニュー→[Windowsシステムツール]→[コマンドプロンプト]) してWSLをインストールする。
2. gcc をインストールする。

II-1 コマンドプロンプトを起動してWSLをインストールする。

Prompt% wsl --install[改行]



```
clang@DESKTOP-UIHT3L6: ~
Microsoft Windows [Version 10.0.19045.4046]
(c) Microsoft Corporation. All rights reserved.

C:\Users\¥yasko>wsl --install
Ubuntu は既にインストールされています。
Ubuntu を起動しています...
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: clang
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 22.04.3 LTS (GNU/Linux 5.15.146.1-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/clang/.hushlogin file.
clang@DESKTOP-UIHT3L6:~$
```

「Ubuntuは既にインストールされています。」というメッセージは、表示されないこともあります。
「Enter new UNIX username:」には、適当な名前をアルファベットで入力します。

例) clang

「New password:」には、「管理者のパスワード」を入力します。

「Retype new password:」には、再度「管理者のパスワード」を入力します。

II-2 gcc をインストールする。

WSLをインストールしてもgcc がインストールされていない場合は、gcc が実行できません。

以下のメッセージが表示されます。

Command 'gcc' not found, but can be installed with: sudo apt install gcc

その場合には、gcc をインストールする必要があります。2つのステップが必要です。

- apt (WSLでソフトウェアをインストールしたり、更新する際に使うコマンド)を更新します。
- apt コマンドを用いて gcc をインストールします。

以上の2つのコマンドを実行するには、「管理者のパスワード」を入力する必要があります。

まずは、apt コマンドの更新をします。

Prompt% sudo apt update[改行]

```
clang@DESKTOP-UIHT3L6: ~  
clang@DESKTOP-UIHT3L6:~$ sudo apt update  
[sudo] password for clang:  
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease  
Get:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]  
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]  
Get:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]  
Get:5 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]  
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1472 kB]  
Get:7 http://archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]  
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [253 kB]  
Get:9 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1876 kB]  
Get:10 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [318 kB]  
Get:11 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [853 kB]  
Get:12 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [164 kB]  
Get:13 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.8 kB]  
Get:14 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [37.2 kB]  
Get:15 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7588 B]  
Get:16 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]  
Get:17 http://archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]  
Get:18 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]  
Get:19 http://archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]  
Get:20 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]  
Get:21 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1683 kB]  
Get:22 http://archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [312 kB]  
Get:23 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1933 kB]  
Get:24 http://archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [328 kB]  
Get:25 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1075 kB]  
Get:26 http://archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [246 kB]  
Get:27 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [22.1 kB]  
Get:28 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [42.7 kB]
```

```
clang@DESKTOP-UIHT3L6: ~  
Get:15 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7588 B]  
Get:16 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]  
Get:17 http://archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]  
Get:18 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]  
Get:19 http://archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]  
Get:20 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]  
Get:21 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1683 kB]  
Get:22 http://archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [312 kB]  
Get:23 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1933 kB]  
Get:24 http://archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [328 kB]  
Get:25 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1075 kB]  
Get:26 http://archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [246 kB]  
Get:27 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [22.1 kB]  
Get:28 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [42.7 kB]  
Get:29 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [10.4 kB]  
Get:30 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [472 B]  
Get:31 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [67.1 kB]  
Get:32 http://archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [11.0 kB]  
Get:33 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]  
Get:34 http://archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]  
Get:35 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [27.2 kB]  
Get:36 http://archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.2 kB]  
Get:37 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [644 B]  
Get:38 http://archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]  
Fetched 31.5 MB in 39s (800 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
103 packages can be upgraded. Run 'apt list --upgradable' to see them.  
clang@DESKTOP-UIHT3L6:~$
```

正常に実行された場合は、以上のようなメッセージが表示されます。

続いて、gcc コマンドをインストールします。

Prompt% sudo apt install gcc[改行]

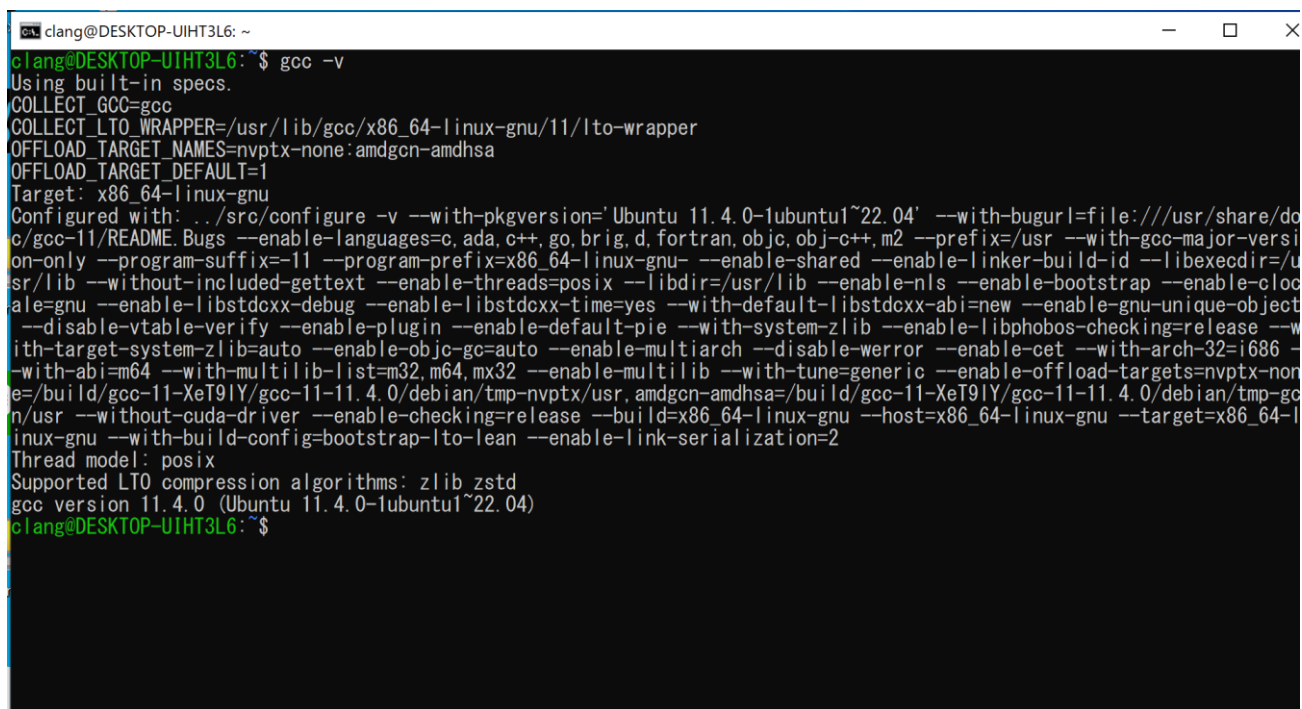
```
clang@DESKTOP-UIHT3L6: ~
clang@DESKTOP-UIHT3L6:~$ sudo apt install gcc
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  cpp cpp-11 fontconfig-config fonts-dejavu-core gcc-11 gcc-11-base libasan6 libatomic1 libc-dev-bin libc-devtools
  libc6 libc6-dev libcc1-0 libcrypt-dev libdeflate0 libfontconfig1 libfreetype6 libgcc-11-dev libgd3 libgomp1 libisl23
  libitm1 libjbig0 libjpeg-turbo8 libjpeg8 liblsan0 libmpc3 libnsl-dev libquadmath0 libtiff5 libtirpc-dev libtsan0
  libubsan1 libwebp7 libxpm4 linux-libc-dev manpages-dev rpcsvc-proto
Suggested packages:
  cpp-doc gcc-11-locales gcc-multilib make autoconf automake libtool flex bison gdb gcc-doc gcc-11-multilib gcc-11-doc
  glibc-doc libgd-tools
Recommended packages:
  libnss-nis libnss-nisplus
The following NEW packages will be installed:
  cpp cpp-11 fontconfig-config fonts-dejavu-core gcc gcc-11 gcc-11-base libasan6 libatomic1 libc-dev-bin libc-devtools
  libc6-dev libcc1-0 libcrypt-dev libdeflate0 libfontconfig1 libfreetype6 libgcc-11-dev libgd3 libgomp1 libisl23
  libitm1 libjbig0 libjpeg-turbo8 libjpeg8 liblsan0 libmpc3 libnsl-dev libquadmath0 libtiff5 libtirpc-dev libtsan0
  libubsan1 libwebp7 libxpm4 linux-libc-dev manpages-dev rpcsvc-proto
The following packages will be upgraded:
  libc6
1 upgraded, 38 newly installed, 0 to remove and 102 not upgraded.
Need to get 52.3 MB of archives.
After this operation, 153 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libc6 amd64 2.35-0ubuntu3.7 [3235 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 gcc-11-base amd64 11.4.0-1ubuntu1~22.04 [20.2 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/main amd64 libisl23 amd64 0.24-2build1 [727 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy/main amd64 libmpc3 amd64 1.2.1-2build1 [46.9 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 cpp-11 amd64 11.4.0-1ubuntu1~22.04 [10.0 MB]
```

「Do you want to continue? [Y/n]」には、y を入力します。

```
clang@DESKTOP-UIHT3L6: ~
Setting up libquadmath0:amd64 (12.3.0-1ubuntu1~22.04) ...
Setting up libmpc3:amd64 (1.2.1-2build1) ...
Setting up libatomic1:amd64 (12.3.0-1ubuntu1~22.04) ...
Setting up fontconfig-config (2.37-2build1) ...
Setting up libjpeg-turbo8:amd64 (2.1.2-0ubuntu1) ...
Setting up libwebp7:amd64 (1.2.2-2ubuntu0.22.04.2) ...
Setting up libubsan1:amd64 (12.3.0-1ubuntu1~22.04) ...
Setting up libnsl-dev:amd64 (1.3.0-2build2) ...
Setting up libcrypt-dev:amd64 (1:4.4.27-1) ...
Setting up libisl23:amd64 (0.24-2build1) ...
Setting up libc-dev-bin (2.35-0ubuntu3.7) ...
Setting up libcc1-0:amd64 (12.3.0-1ubuntu1~22.04) ...
Setting up liblsan0:amd64 (12.3.0-1ubuntu1~22.04) ...
Setting up libitm1:amd64 (12.3.0-1ubuntu1~22.04) ...
Setting up libtsan0:amd64 (11.4.0-1ubuntu1~22.04) ...
Setting up libjpeg8:amd64 (8c-2ubuntu10) ...
Setting up cpp-11 (11.4.0-1ubuntu1~22.04) ...
Setting up fontconfig-config (2.13.1-4.2ubuntu5) ...
Setting up libgcc-11-dev:amd64 (11.4.0-1ubuntu1~22.04) ...
Setting up gcc-11 (11.4.0-1ubuntu1~22.04) ...
Setting up cpp (4:11.2.0-1ubuntu1) ...
Setting up libc6-dev:amd64 (2.35-0ubuntu3.7) ...
Setting up libtiff5:amd64 (4.3.0-6ubuntu0.8) ...
Setting up libfontconfig1:amd64 (2.13.1-4.2ubuntu5) ...
Setting up gcc (4:11.2.0-1ubuntu1) ...
Setting up libgd3:amd64 (2.3.0-2ubuntu2) ...
Setting up libc-devtools (2.35-0ubuntu3.7) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.4) ...
clang@DESKTOP-UIHT3L6:~$
```

最後に、gcc のバージョンを確かめます。

Prompt% gcc -v[改行]

A terminal window titled 'clang@DESKTOP-UIHT3L6: ~' showing the command 'gcc -v' and its output. The output includes details about the GCC version (11.4.0), target architecture (x86_64-linux-gnu), and various compiler options. The terminal text is as follows:

```
clang@DESKTOP-UIHT3L6:~$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/11/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none:amdgc-nvptx-none:amdgc-nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu 11.4.0-1ubuntu1~22.04' --with-bugurl=file:///usr/share/doc/gcc-11/README.Bugs --enable-languages=c,ada,c++,go,brig,d,fortran,objc,obj-c++,m2 --prefix=/usr --with-gcc-major-version-only --program-suffix=-11 --program-prefix=x86_64-linux-gnu- --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --without-included-gettext --enable-threads=posix --libdir=/usr/lib --enable-nls --enable-bootstrap --enable-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --with-default-libstdcxx-abi=new --enable-gnu-unique-object --disable-vtable-verify --enable-plugin --enable-default-pie --with-system-zlib --enable-libphobos-checking=release --with-target-system-zlib=auto --enable-objc-gc=auto --enable-multiarch --disable-werror --enable-cet --with-arch-32=i686 --with-abi=m64 --with-multilib-list=m32,m64,mx32 --enable-multilib --with-tune=generic --enable-offload-targets=nvptx-none=/build/gcc-11-XeT91Y/gcc-11-11.4.0/debian/tmp-nvptx/usr,amdgc-nvptx-none=/build/gcc-11-XeT91Y/gcc-11-11.4.0/debian/tmp-nvptx/usr --without-cuda-driver --enable-checking=release --build=x86_64-linux-gnu --host=x86_64-linux-gnu --target=x86_64-linux-gnu --with-build-config=bootstrap-lto-lean --enable-link-serialization=2
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 11.4.0 (Ubuntu 11.4.0-1ubuntu1~22.04)
clang@DESKTOP-UIHT3L6:~$
```

以上のようにバージョン情報が表示されます。これで gcc コンパイラを使用して C のプログラムをコンパイルできるようになりました。

注意 : WSLを起動すると、Linux (Ubuntu) というオペレーティングシステムが起動するので、ディレクトリ (フォルダ) の指定の仕方が異なります。

<ドライブ>:¥<ディレクトリ1>¥<ディレクトリ2> ⇒
/mnt/<ドライブ>/<ディレクトリ1>/<ディレクトリ2>

例) C:¥Users¥usr ⇒ /mnt/c/Users/user
C:¥Users¥usr¥micro ⇒ /mnt/c/Users/user/micro