SECCON Beginners CTF勉強会

2025年 事前準備資料

Reversing講義

事前準備のゴール

① Ghidraをインストールし，指定のファイルを読み込み，一番右のパネル内に
`puts("Hello World!");`が表示される (スライドP.3~)

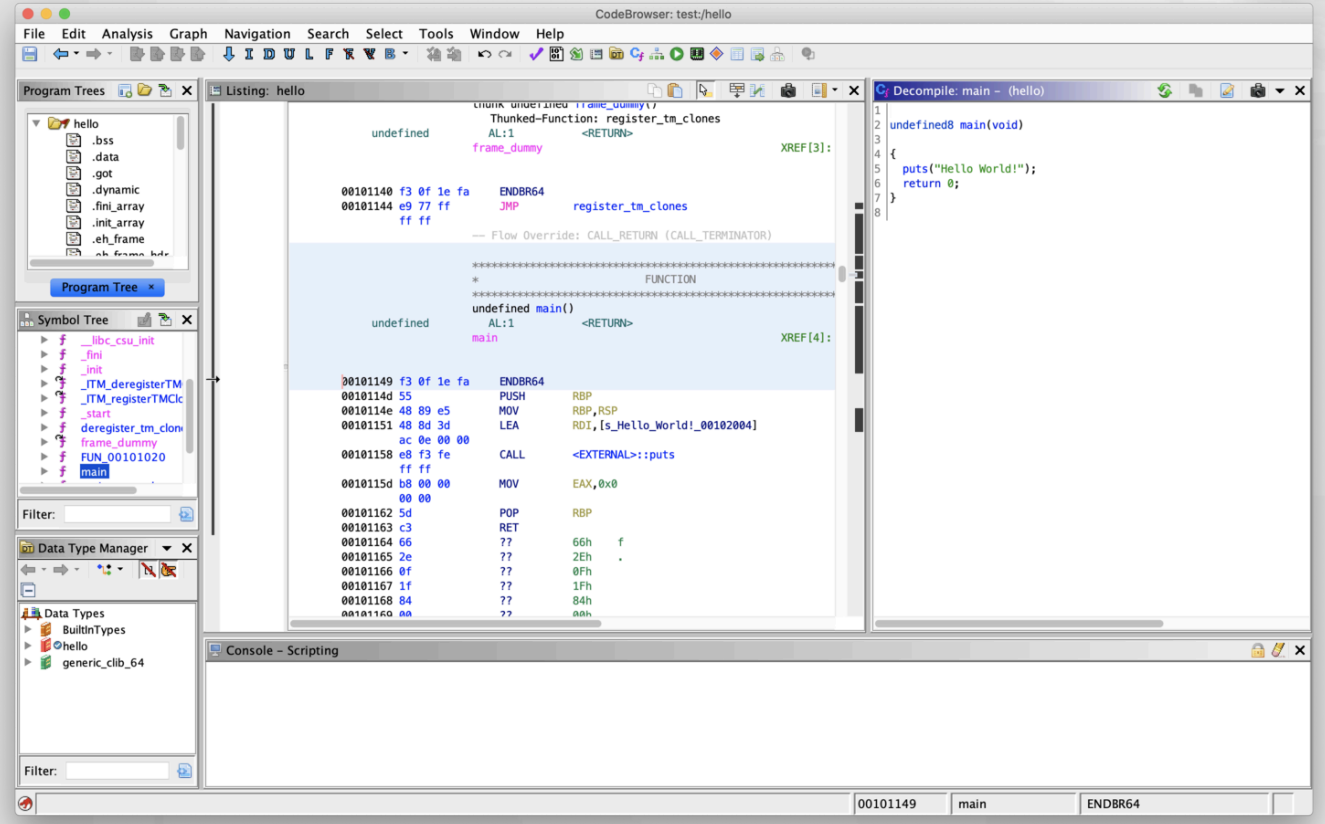
② x86-64のLinux実行環境を用意し，GDBとPedaをインストールし，指定のファイルを読み込む (スライドP.48~)

(注意)

このスライドで紹介するセットアップ方法はあくまで推奨ですので他の方法でも問題ありません
またこちらのセットアップ方法は必ずすべての環境で動作することを保証するものではないことをご了承ください
そしてこれらのセットアップ方法によって環境が動作しなくなった・動作が不安定になった場合の責任は負いません．環境に合わせて公式ドキュメント等を参考にしながらインストールしてください

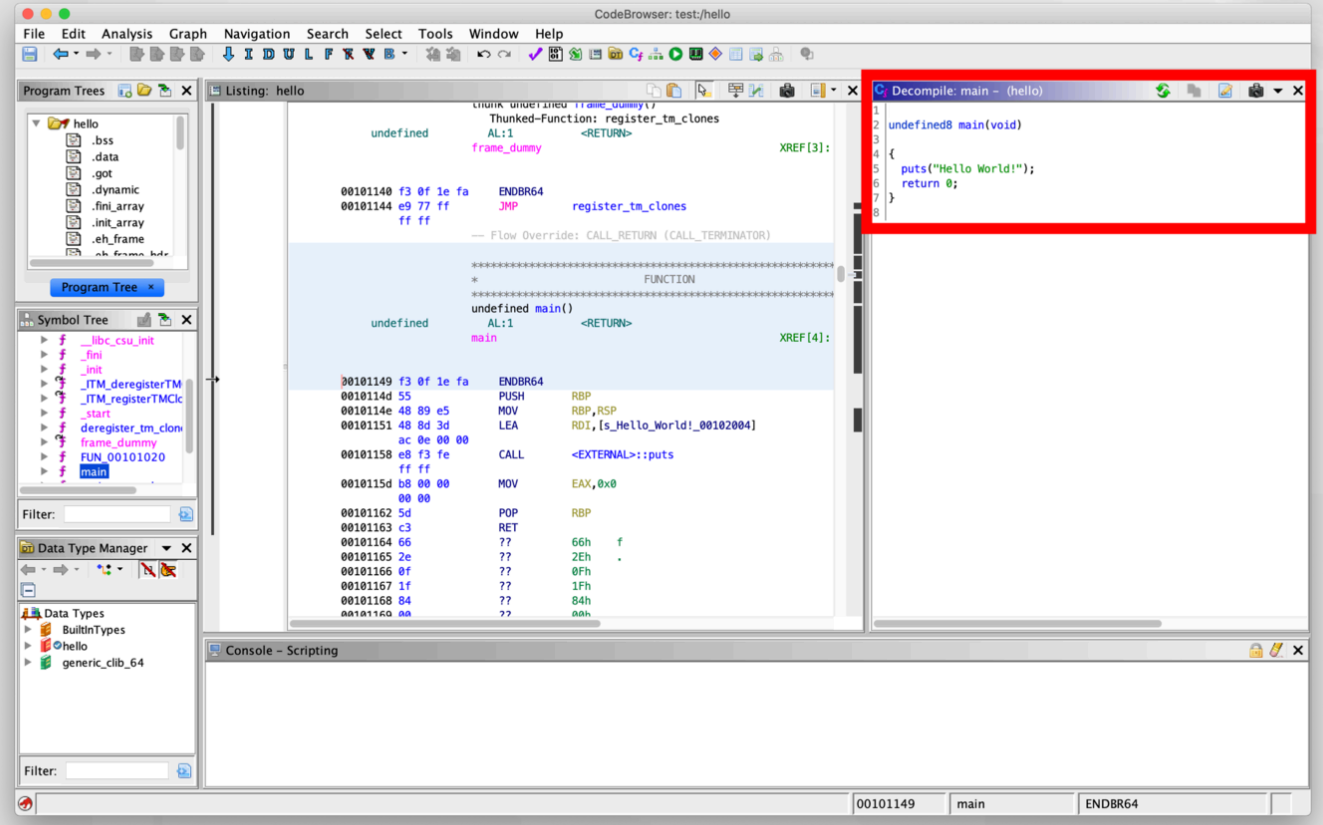
事前準備①のゴール

Ghidraをインストールし，指定のファイルを読み込み，一番右のパネル内に `puts("Hello World!");` が表示される



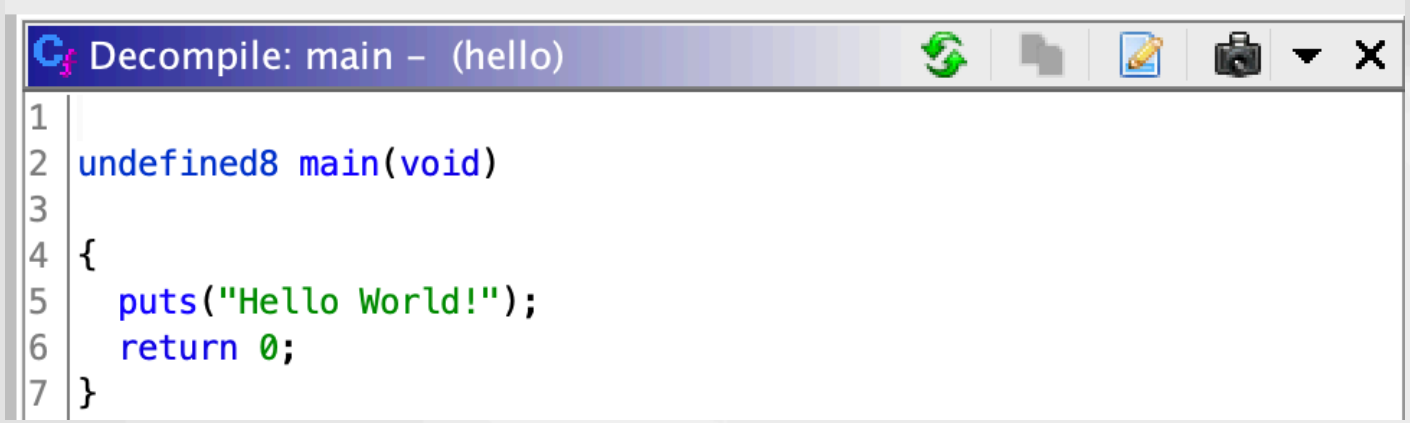
事前準備①のゴール

Ghidraをインストールし，指定のファイルを読み込み，一番右のパネル内に `puts("Hello World!");` が表示される



事前準備①のゴール

Ghidraをインストールし，指定のファイルを読み込み，一番右のパネル内に`puts("Hello World!");`が表示される



事前準備①の手順

1. Ghidraをインストール
   1-1. Mac (Intel & Arm)
   1-2. Windows
   1-3. Linux
2. 指定のファイルを読み込む
3. 一番右のパネル内の表示を確認する

1-1. MacでGhidraをインストール

1-1-1. Homebrewのインストール
1-1-2. teruminのインストール
1-1-3. Ghidraのインストール
1-1-5. Ghidraの起動

(Macについては2022年9月時点でArm搭載・Intel搭載どちらでも同様の方法でGhidraをインストール・実行できます．もし動作しない場合はGhidraの公式ドキュメント Installation Guide を参照してください)

1-1-1. Homebrewのインストール - Mac

https://brew.sh/index_ja の「インストール」よりコマンドをコピーし，ターミナルに貼り付けてエンターキーを押します

コマンド実行後，続けて実行する必要のあるコマンドがターミナルに表示されるので，案内に従ってそのコマンドをコピーし，ターミナルに貼り付けてエンターキーを押します

`brew --version` コマンドをターミナルで実行し，以下のような表示(バージョン番号や日付は異なることがある)が出ればインストール完了です

```
$ brew --version
Homebrew 4.4.17
```

1-1-2. teruminのインストール - Mac

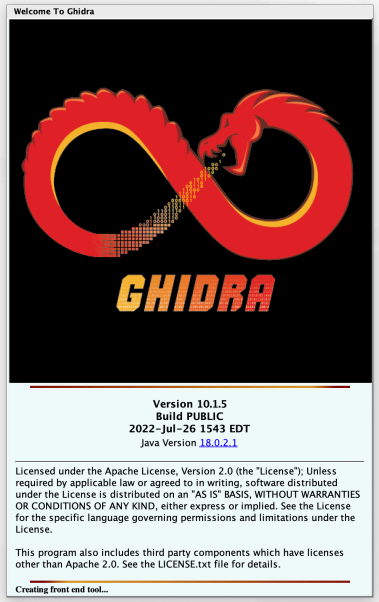
`brew install --cask temurin` コマンドをターミナルで実行し，teruminをインストールします

1-1-3. Ghidraのインストール - Mac

`brew install ghidra` コマンドをターミナルで実行し，ghidraをインストールします

1-1-5. Ghidraの起動 - Mac

`ghidraRun`で実行します

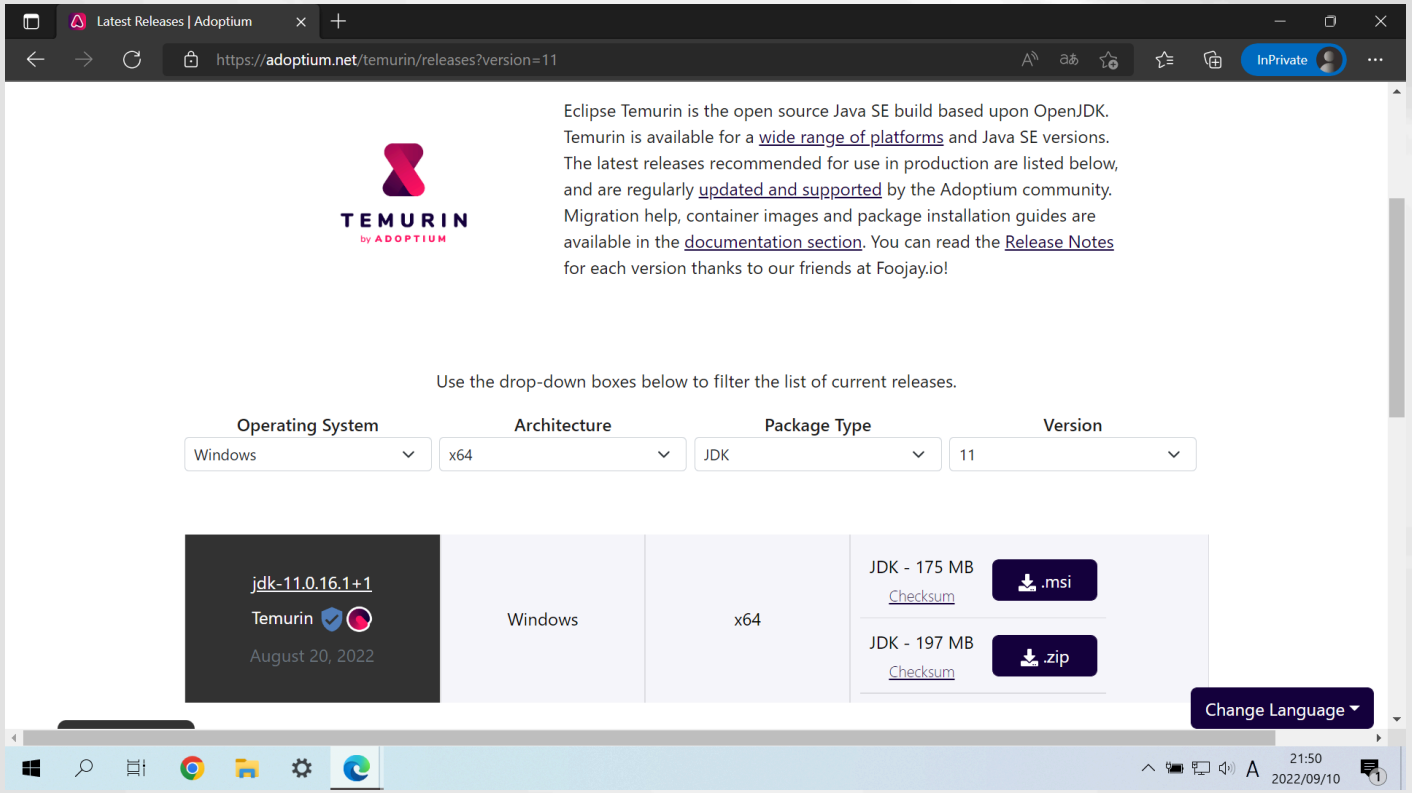


1-２. WindowsでGhidraをインストール

1-２-1. OpenJDKのインストール
1-２-2. Ghidraのダウンロード
1-２-3. Ghidraのファイルの解凍
1-２-4. Ghidraの起動

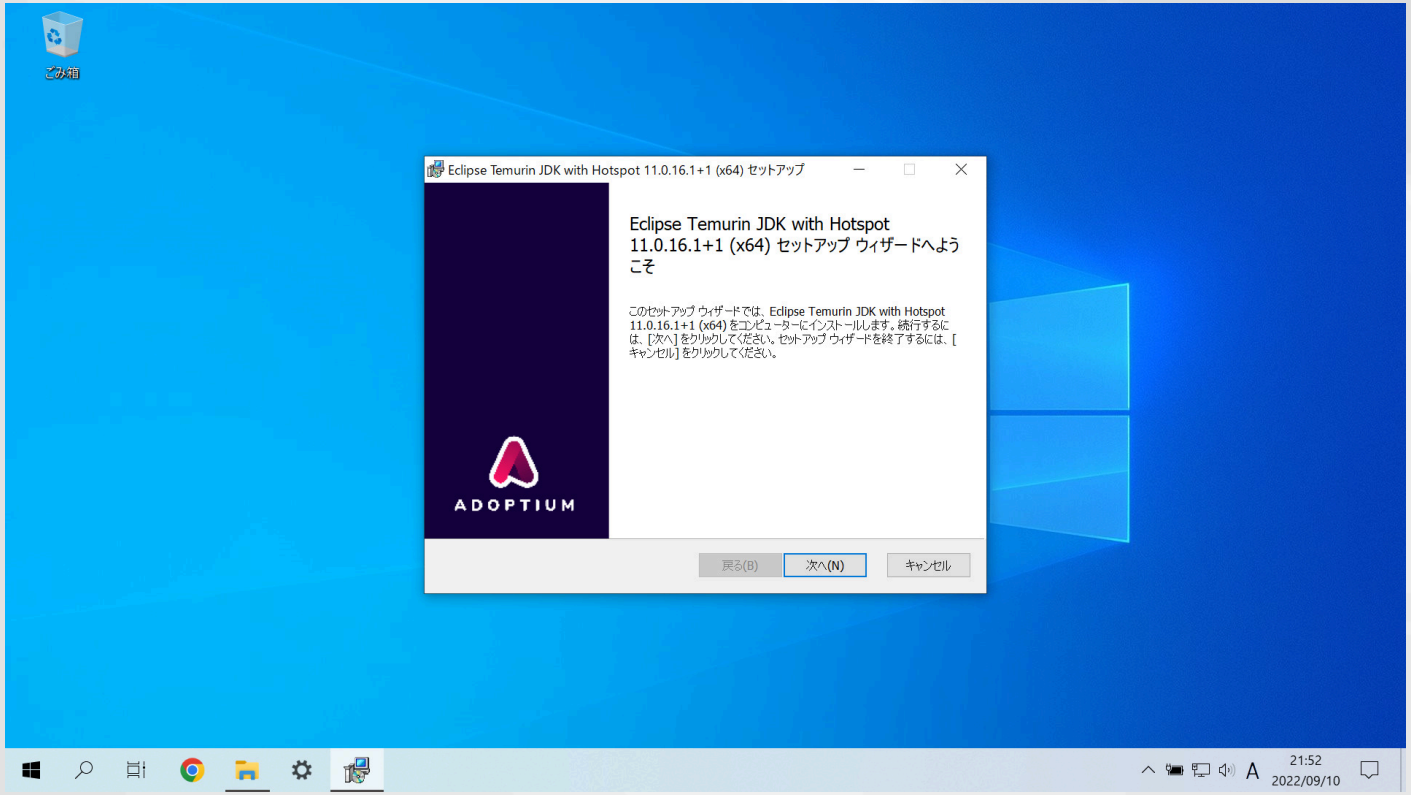
1-2-1. OpenJDKのインストール - Windows

https://adoptium.net/temurin/releases/ からWindows x64向けVersion 11のJDKのmsiインストーラーをダウンロードします



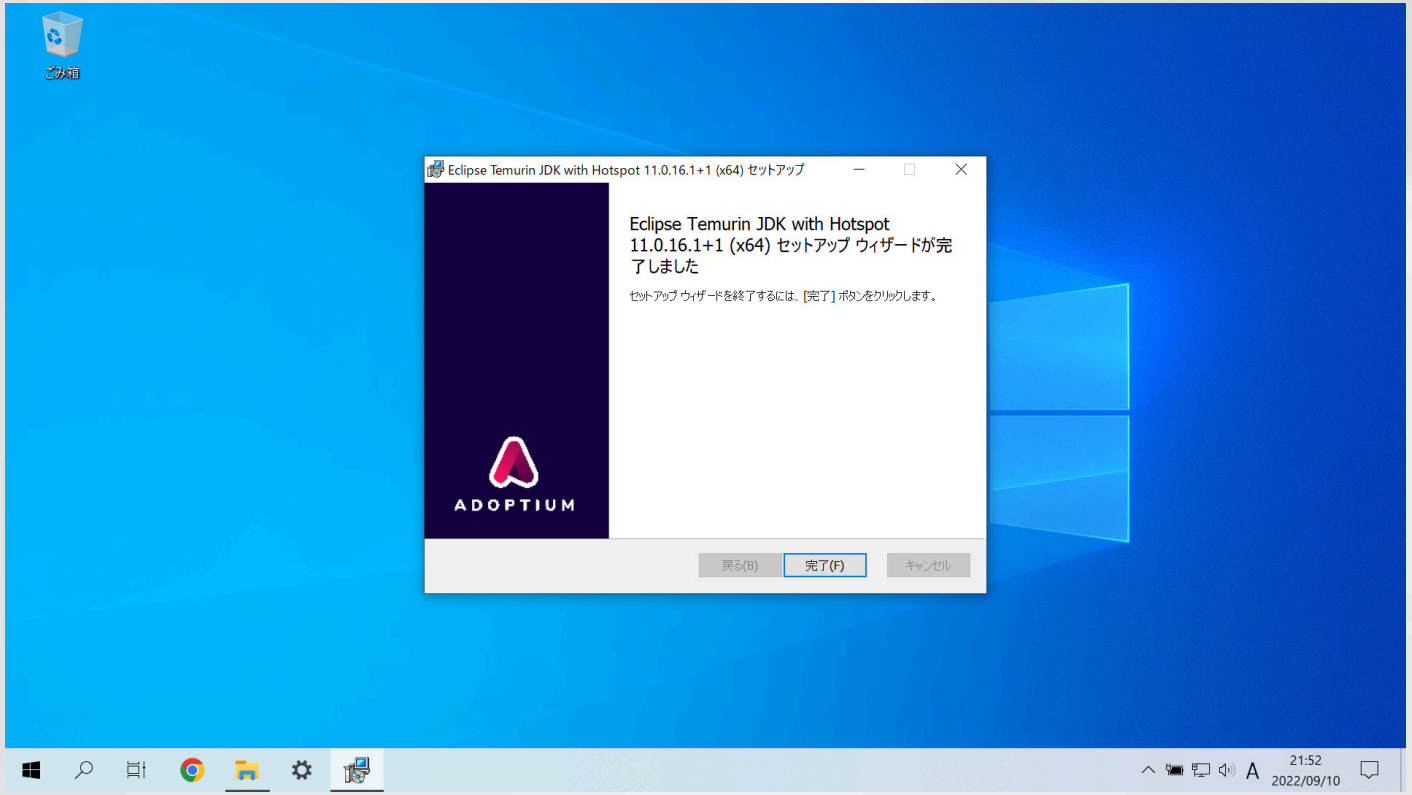
1-2-1. OpenJDKのインストール - Windows

ダウンロードしたmsiファイルをダブルクリックで起動し，インストールします



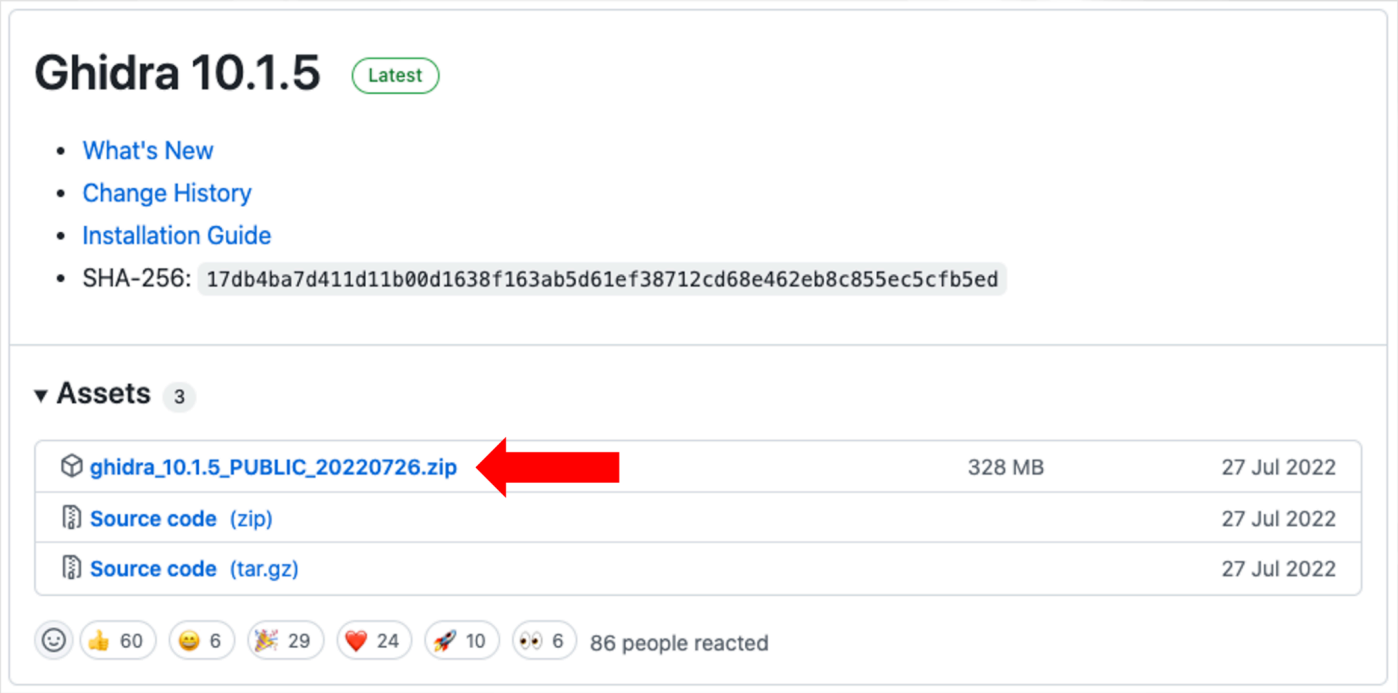
1-2-1. OpenJDKのインストール - Windows

これでJDKのインストールが完了しました



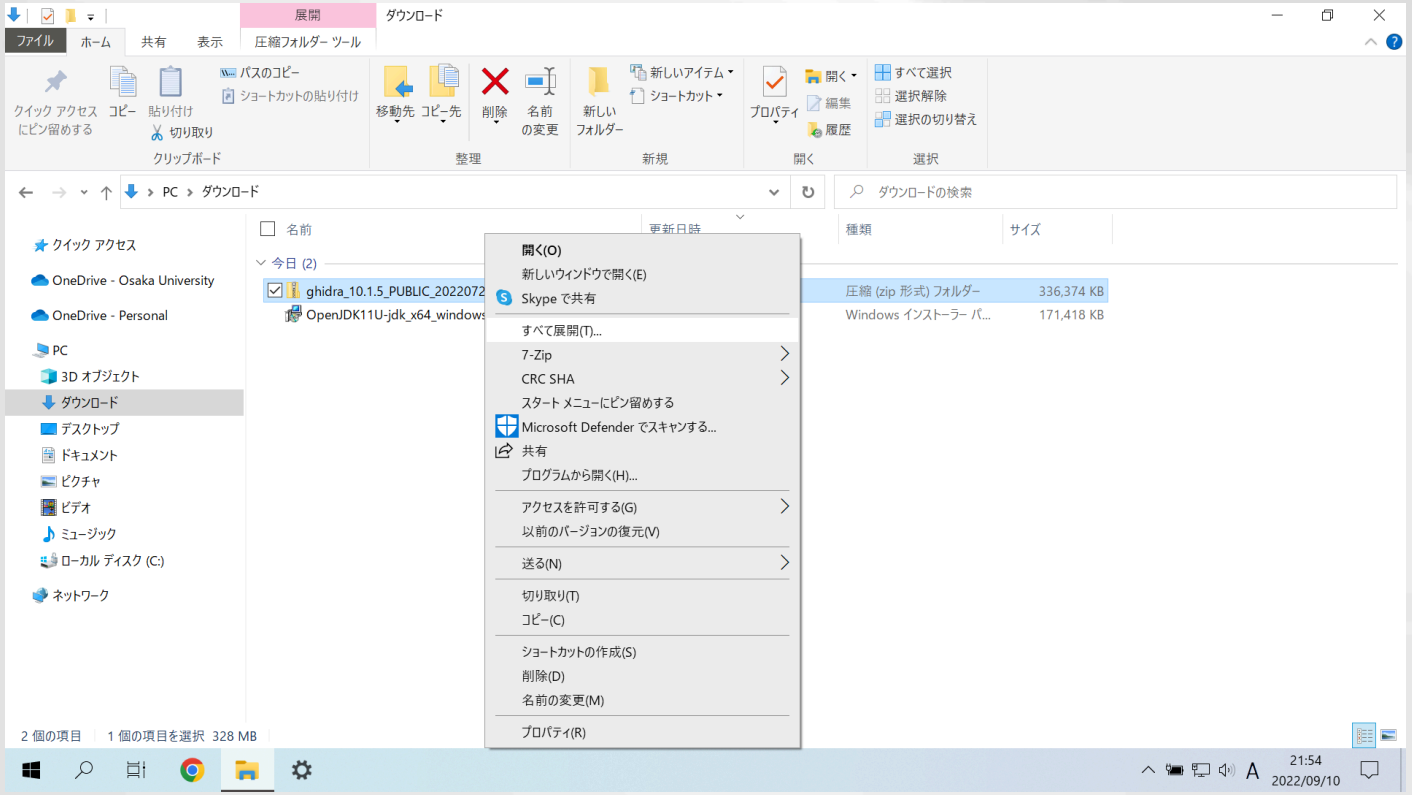
1-2-2. Ghidraのダウンロード - Windows

https://github.com/NationalSecurityAgency/ghidra/releases をブラウザで開き最新のバージョンのzipファイル `ghidra_<バージョン>_PUBLIC_<日付>.zip` をダウンロードする



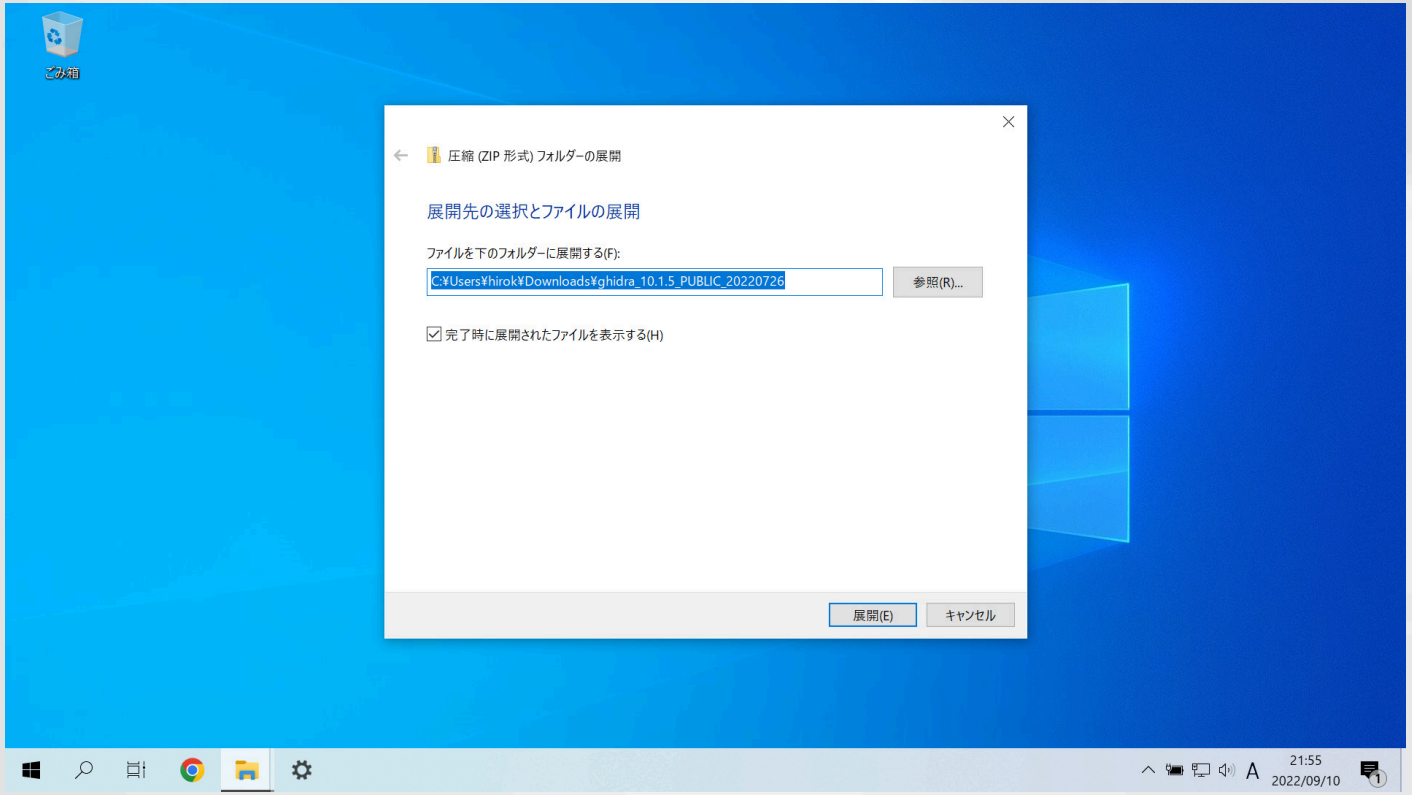
1-2-3. Ghidraのファイルの解凍 - Windows

ダウンロードしたファイルを右クリックし，「すべて展開」を選択します



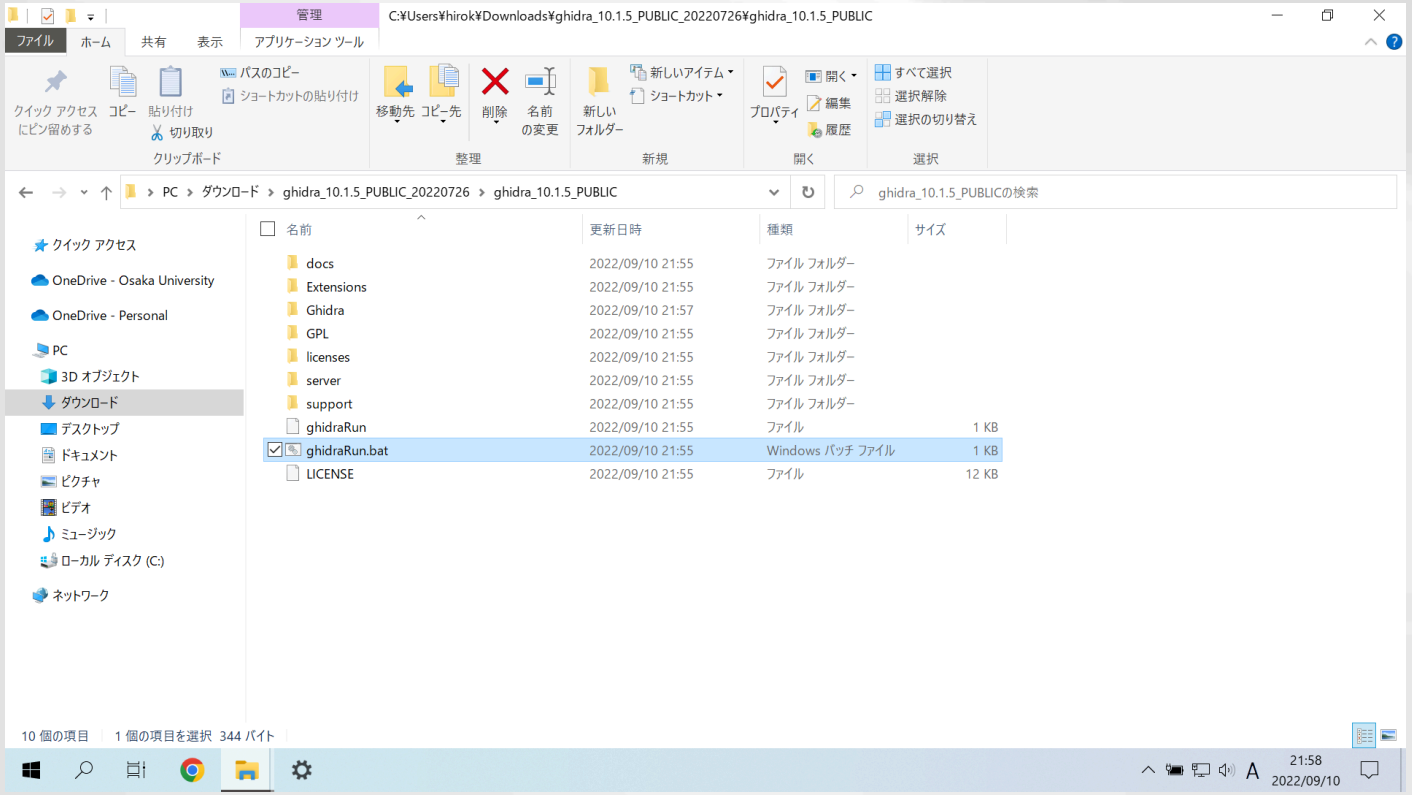
1-2-3. Ghidraのファイルの解凍 - Windows

「展開」をクリックします



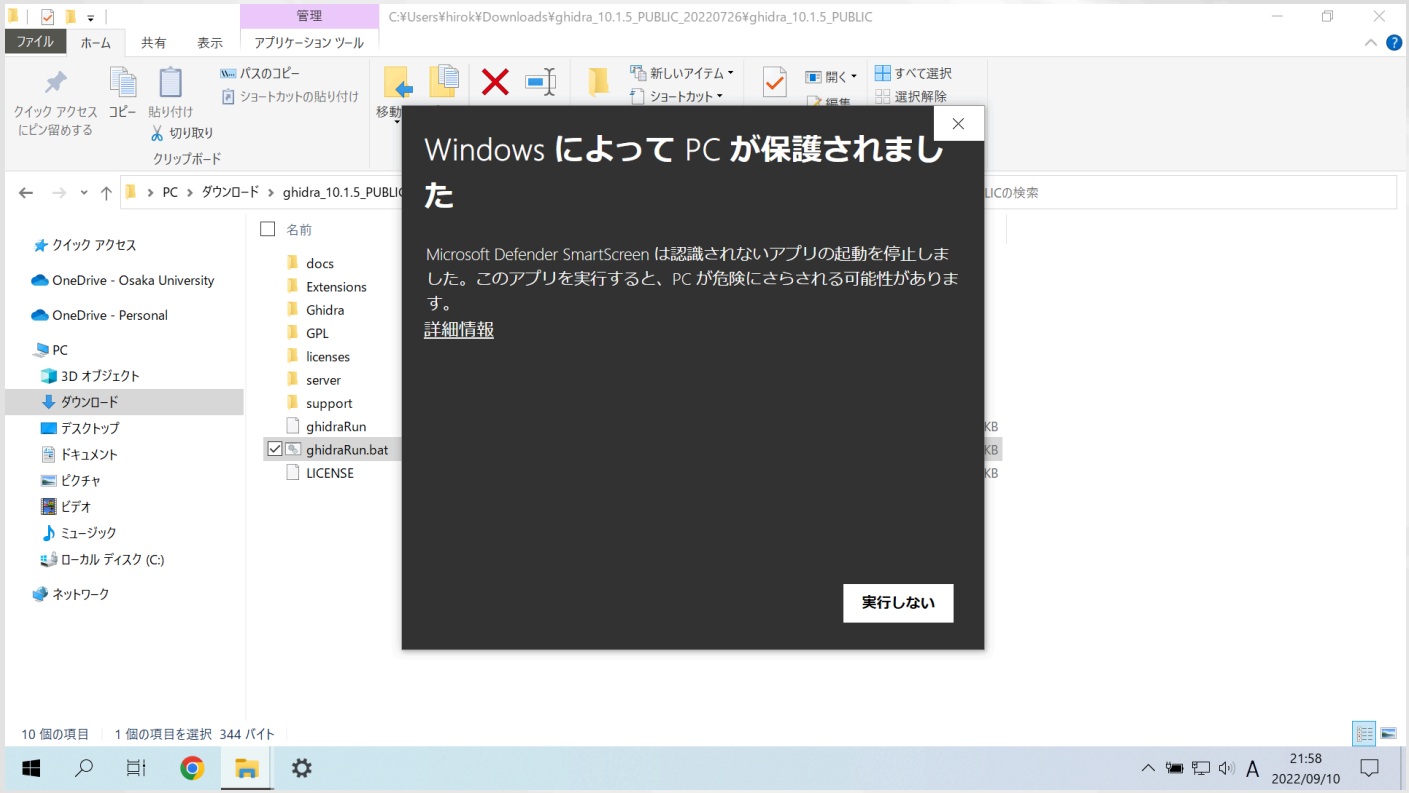
1-2-4. Ghidraの起動 - Windows

展開したフォルダの中の「ghidrarun.bat」をクリックしGhidraを起動します



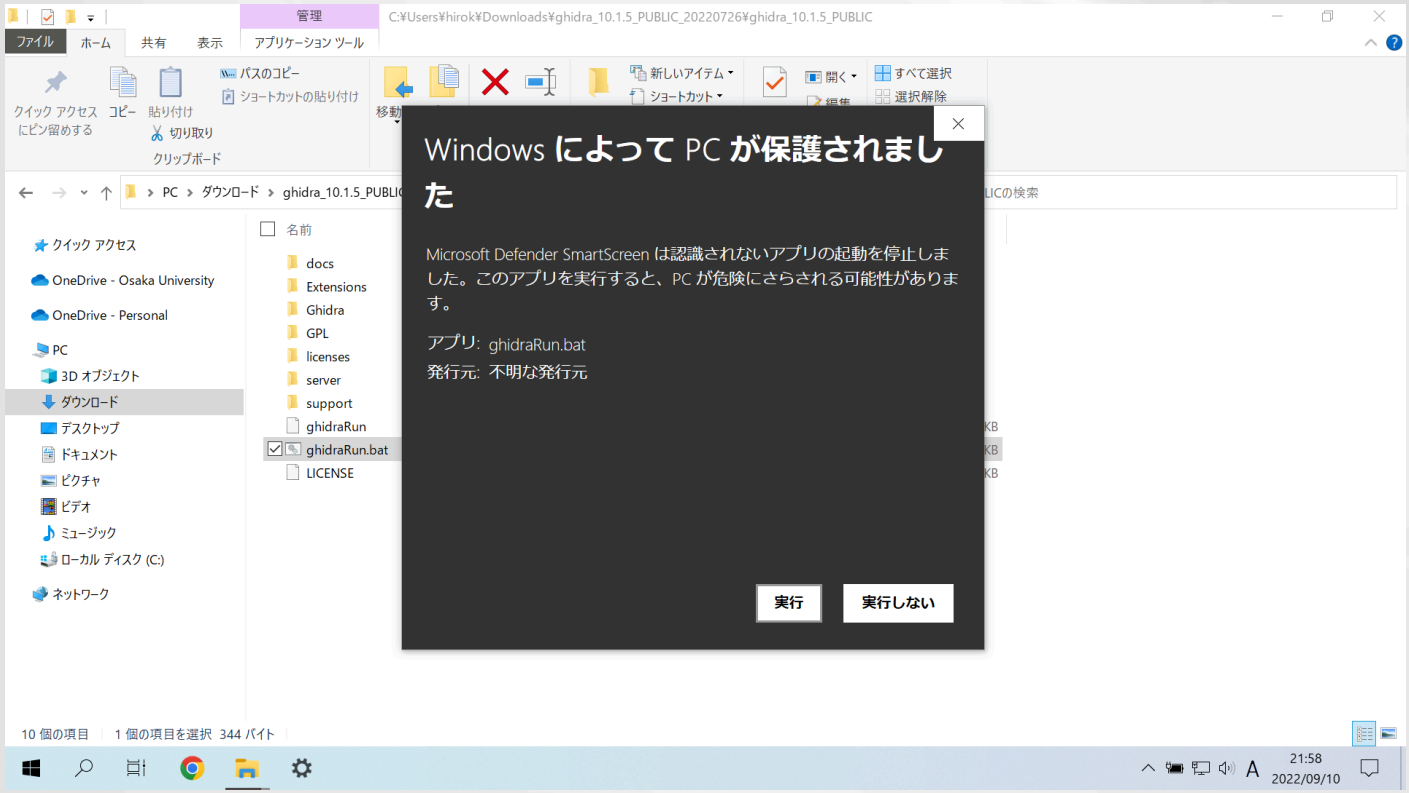
1-2-4. Ghidraの起動 - Windows

(注意) ここでWindows Defenderなどのアンチウイルスソフトが検知をする場合がありますが，製品の案内に従って実行を続行します



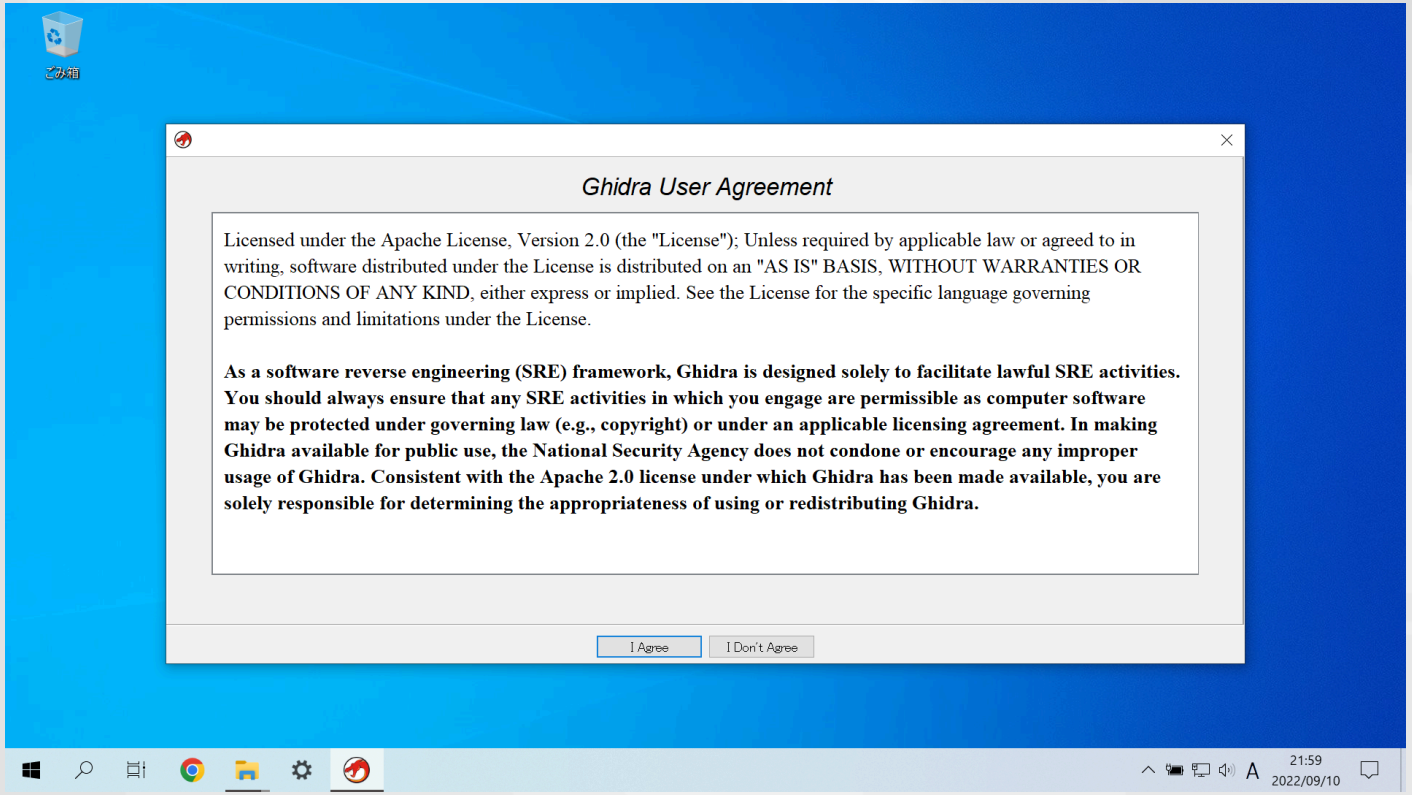
1-2-4. Ghidraの起動 - Windows

(Windows Defenderの場合，「詳細情報」をクリックすることで「実行」ボタンが出てくるのでこれをクリックします)



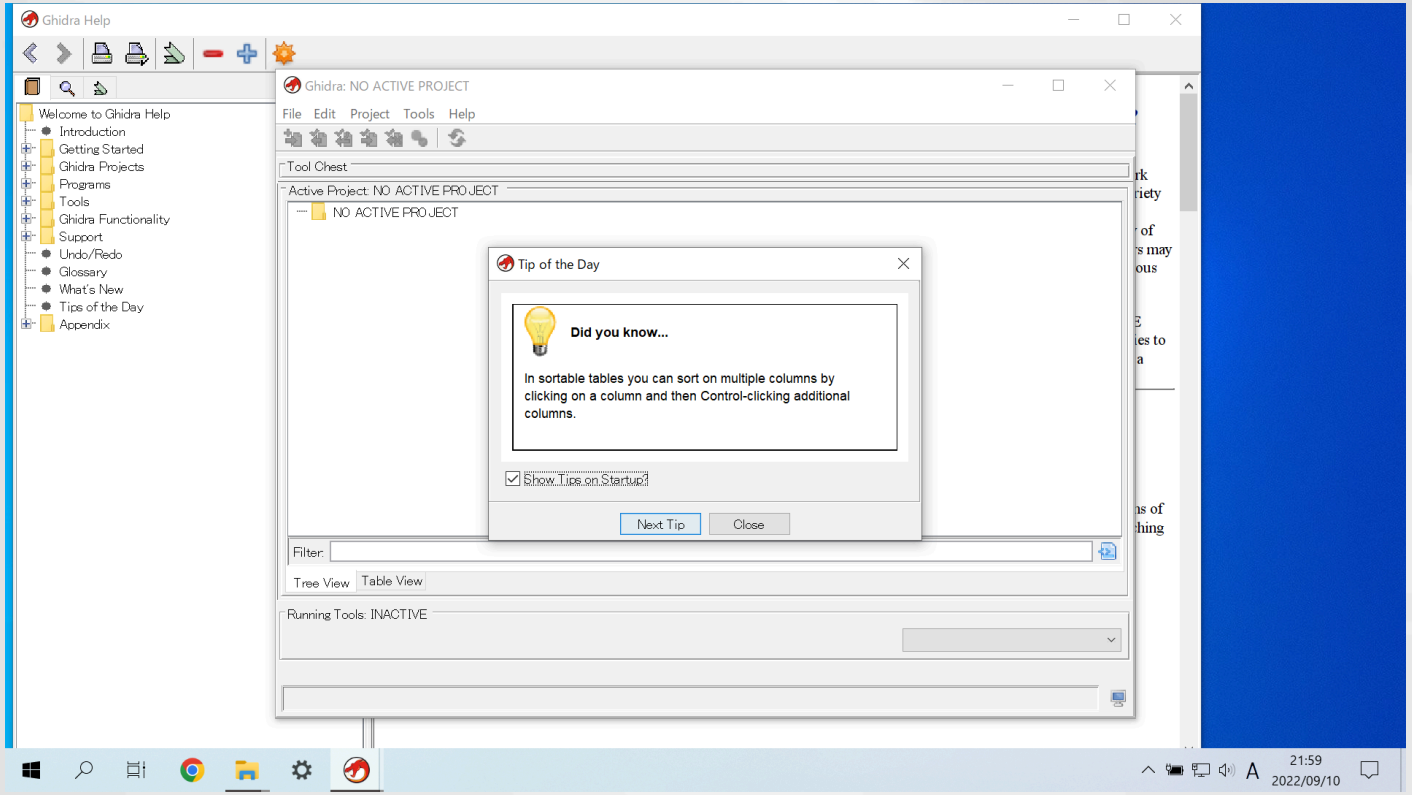
1-2-4. Ghidraの起動 - Windows

User Areementが表示されるので「I Agree」をクリックします



1-2-4. Ghidraの起動 - Windows

Ghidraが起動しました



1-3. LinuxでGhidraをインストール

1-3-1. OpenJDKのインストール
1-3-2. Ghidraのダウンロード
1-3-3. Ghidraのファイルの解凍
1-3-4. Ghidraの起動

1-3-1. OpenJDKのインストール - Linux

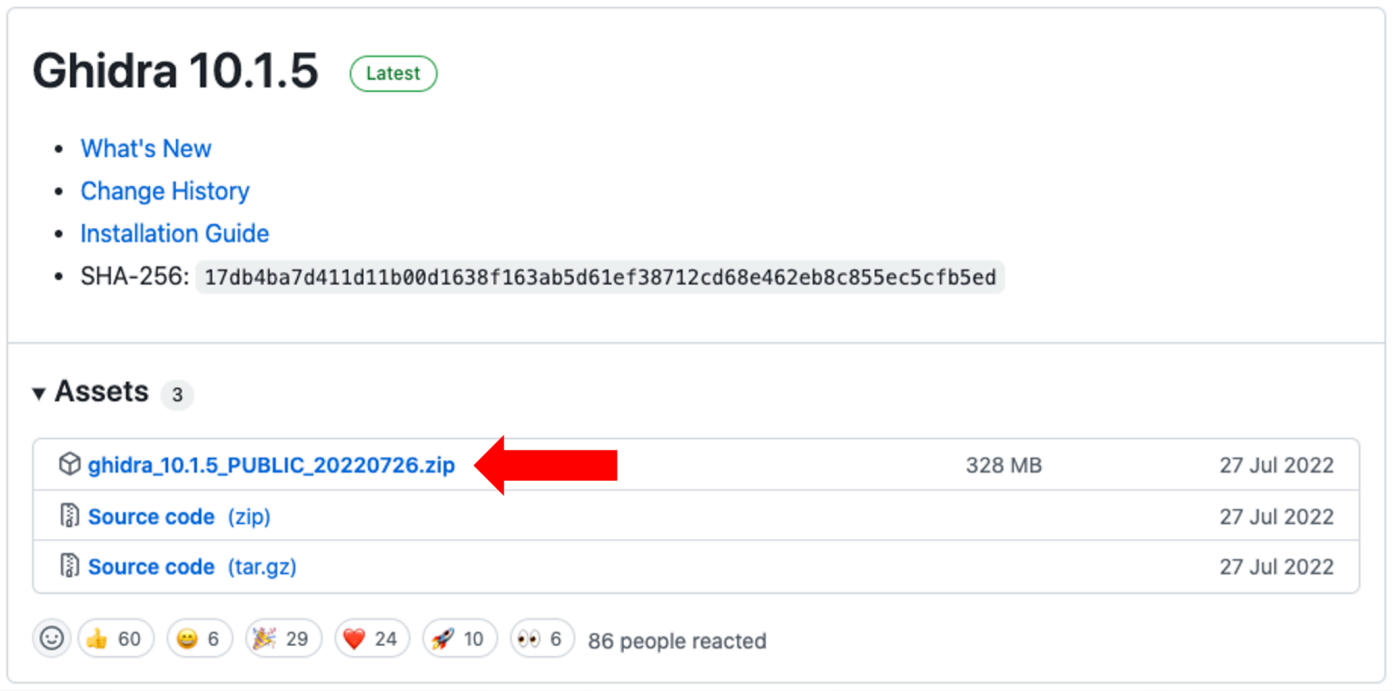
ターミナルで`sudo apt update` `sudo apt install -y default-jdk`コマンドをターミナルで実行し，OpenJDKをインストールします

`java --version`コマンドをターミナルで実行し，以下のような表示(バージョン番号や日付は異なることがある)が出ればインストール完了です

```
$ java --version
openjdk 11.0.16 2022-07-19
OpenJDK Runtime Environment (build 11.0.16+8-post-Ubuntu-0ubuntu120.04)
OpenJDK 64-Bit Server VM (build 11.0.16+8-post-Ubuntu-0ubuntu120.04, mixed mode)
```

1-3-2. Ghidraのダウンロード - Linux

https://github.com/NationalSecurityAgency/ghidra/releases をブラウザで開き最新のバージョンのzipファイル `ghidra_<バージョン>_PUBLIC_<日付>.zip` をダウンロードします



1-3-3. Ghidraのファイルの解凍 - Linux

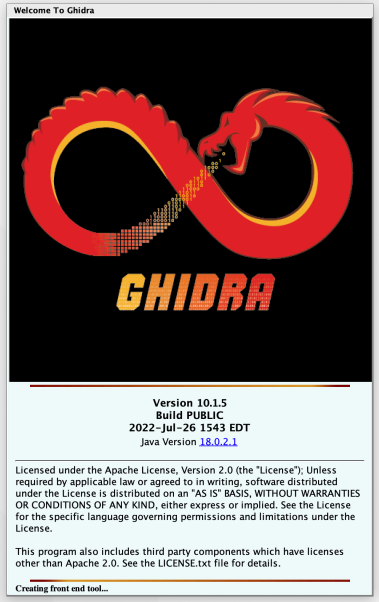
ターミナルでファイルを移動し，`unzip`コマンドで解凍します

```
mv ~/Downloads/ghidra_10.1.5_PUBLIC_20220726.zip ~/Documents
cd ~/Documents
unzip ghidra_10.1.5_PUBLIC_20220726.zip
```

1-3-4. Ghidraの起動 - Linux

ターミナルで解凍したディレクトリに移動し`./ghidraRun`で実行します

```
$ cd ~/Documents/
$ cd ghidra_10.1.5_PUBLIC/
$ ./ghidraRun
```



2. 指定のファイルを読み込む

2-1. ファイルのダウンロード
2-2. Ghidraでの新規プロジェクトの作成とファイルの読み込み

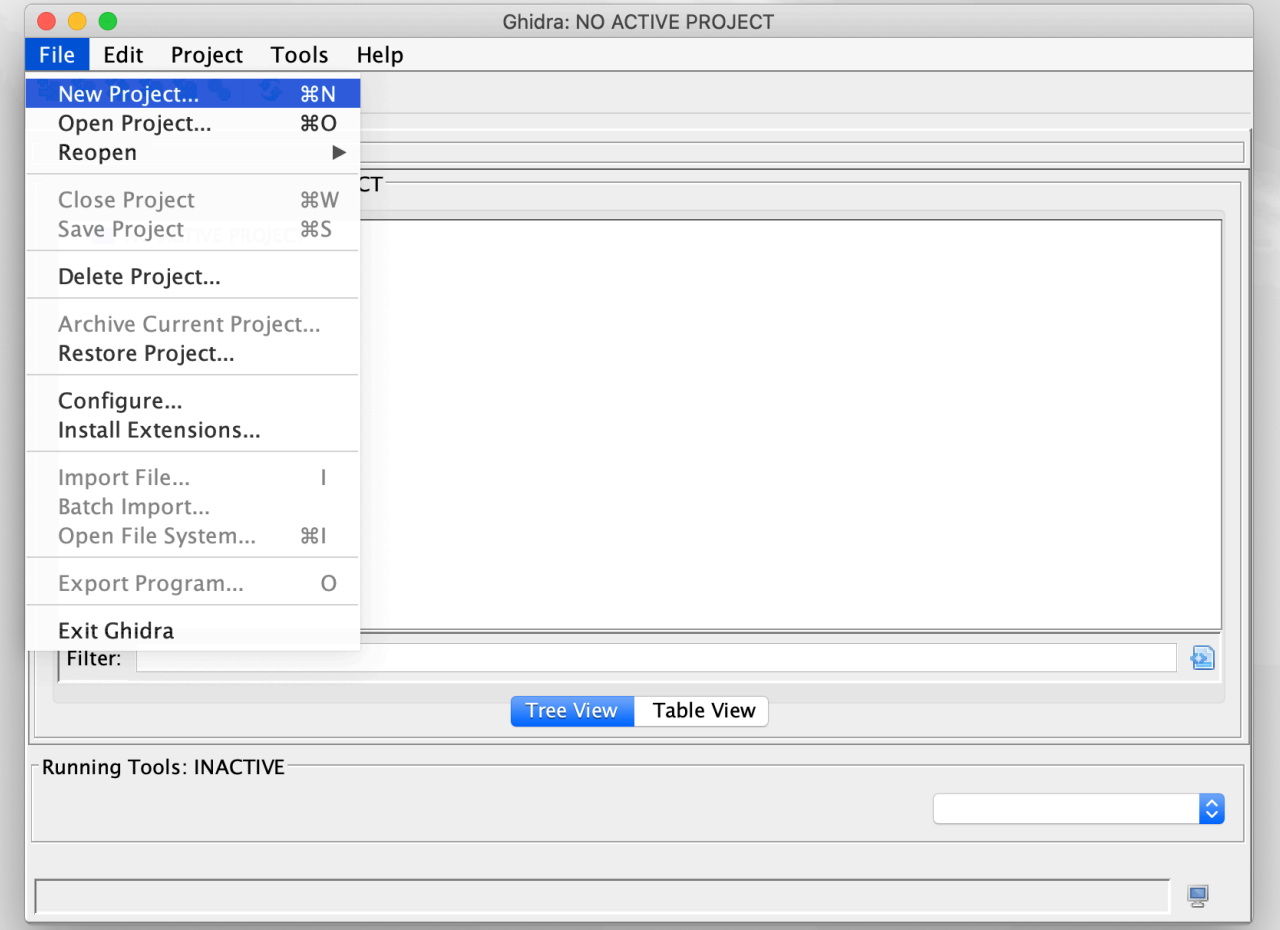
2-1. ファイルのダウンロード

https://github.com/hi120ki/ctf4b-2022-testbin/raw/main/hello よりバイナリをダウンロードします

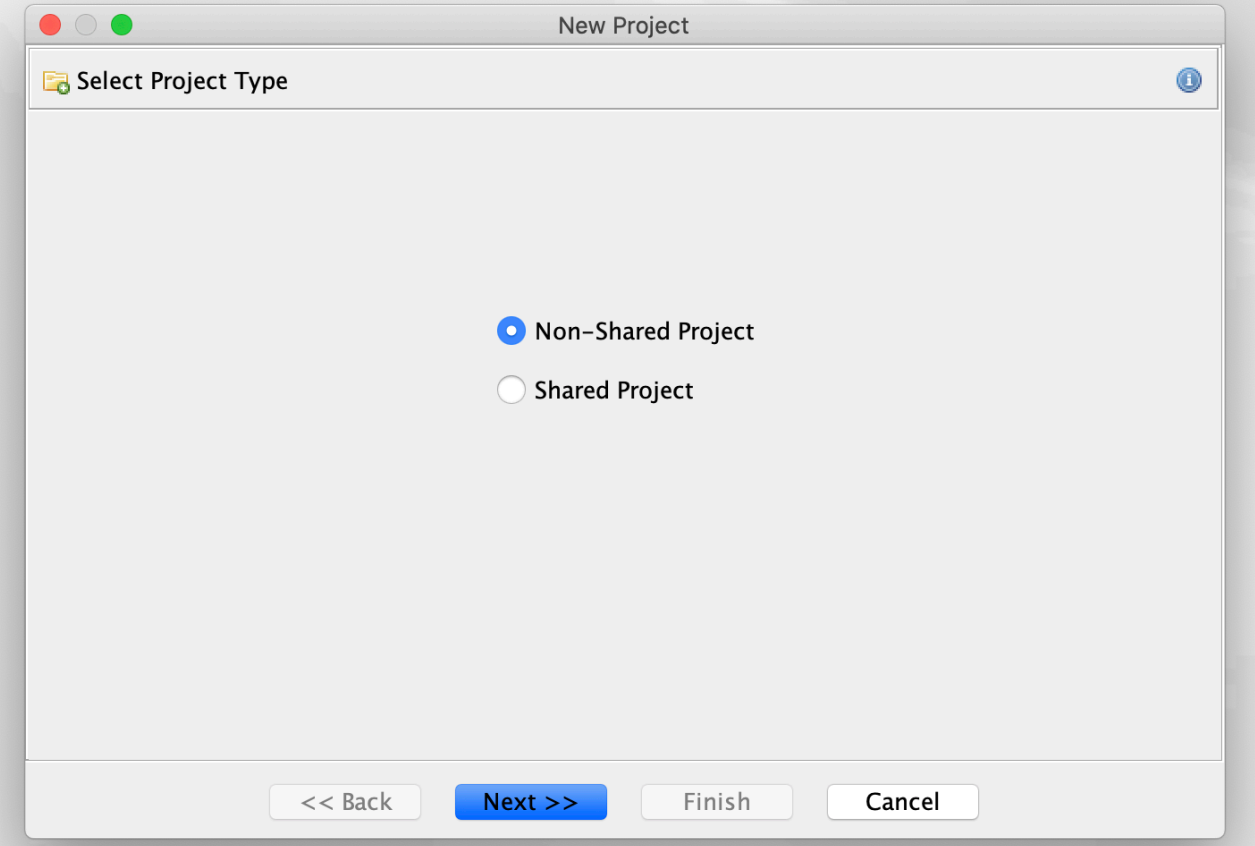
(`Hello World!`を表示するELFファイルです)

```
$ file hello
hello: ELF 64-bit LSB pie executable, x86-64,
version 1 (SYSV), dynamically linked,
interpreter /lib64/ld-linux-x86-64.so.2,
BuildID[sha1]=052027a0a045abf419a7c865f9f0224ee798365a,
for GNU/Linux 3.2.0, not stripped

$ md5sum hello
2ca3e453fb1aa01f28aa7aa8a0c97dda

$ sha256 hello
726958e877bfcb0f99dacbd83ccc595772eecba509211b37069ac39a0e316c95
```

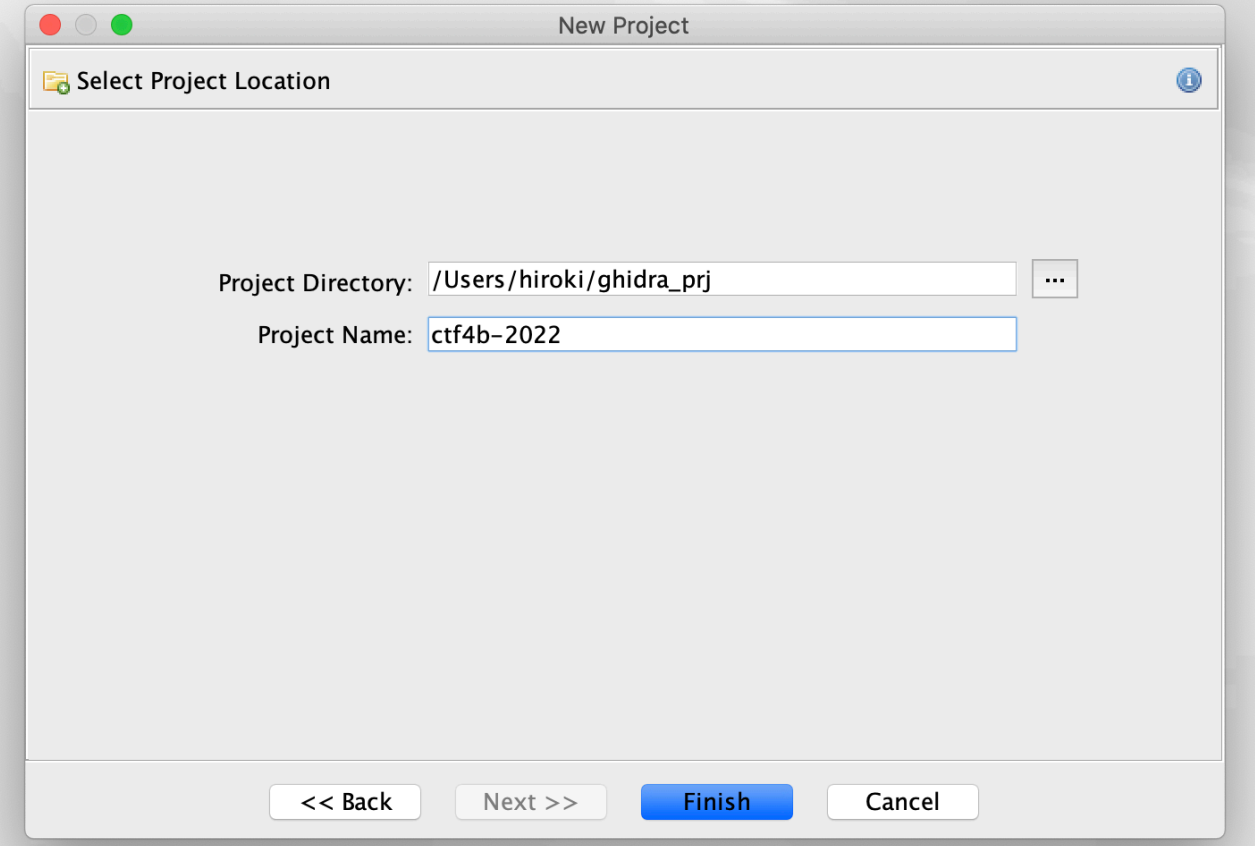
2-2-1. 起動後File→New Projectを選択



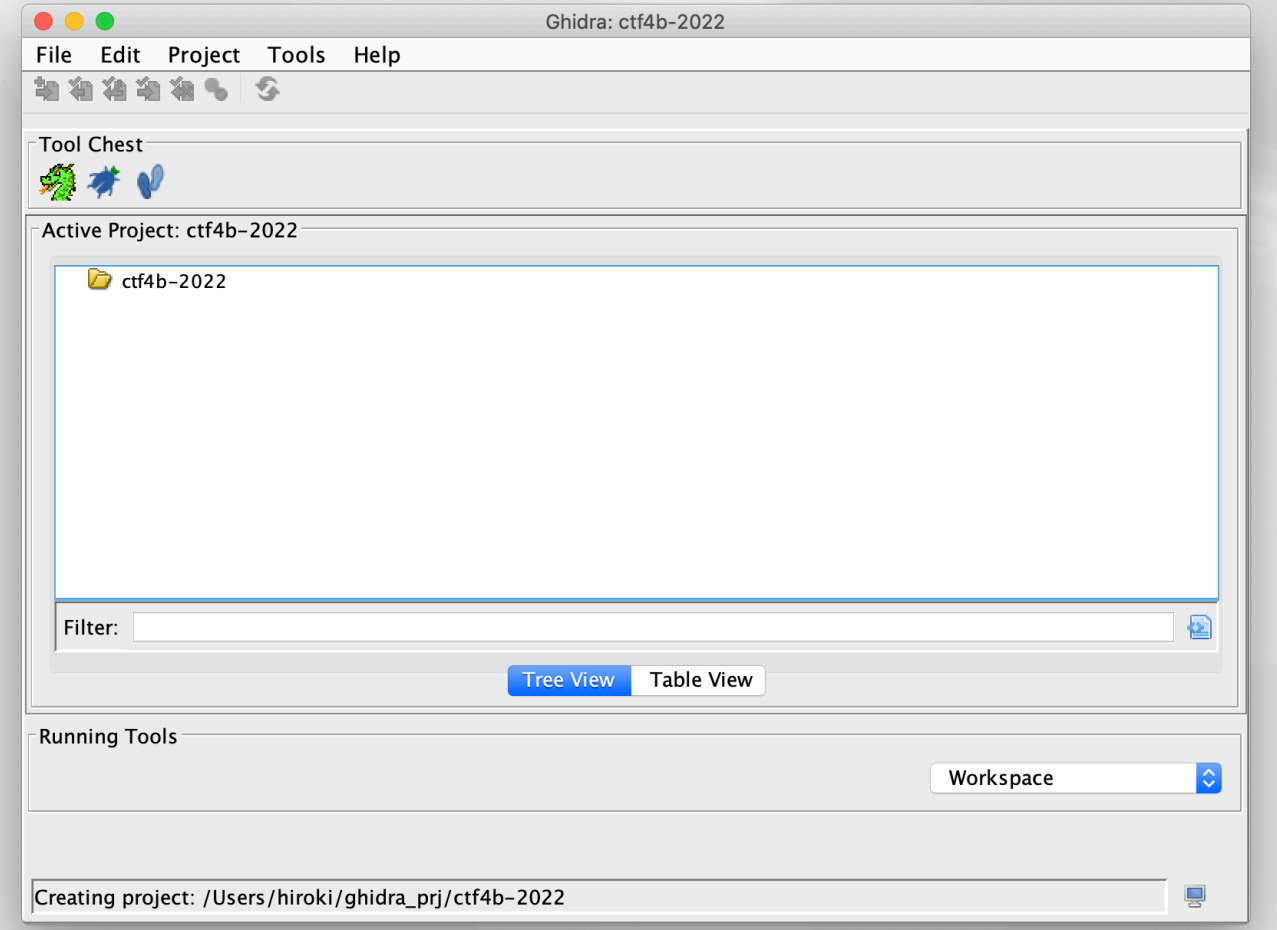
2-2-2. Non-shared projectを選択



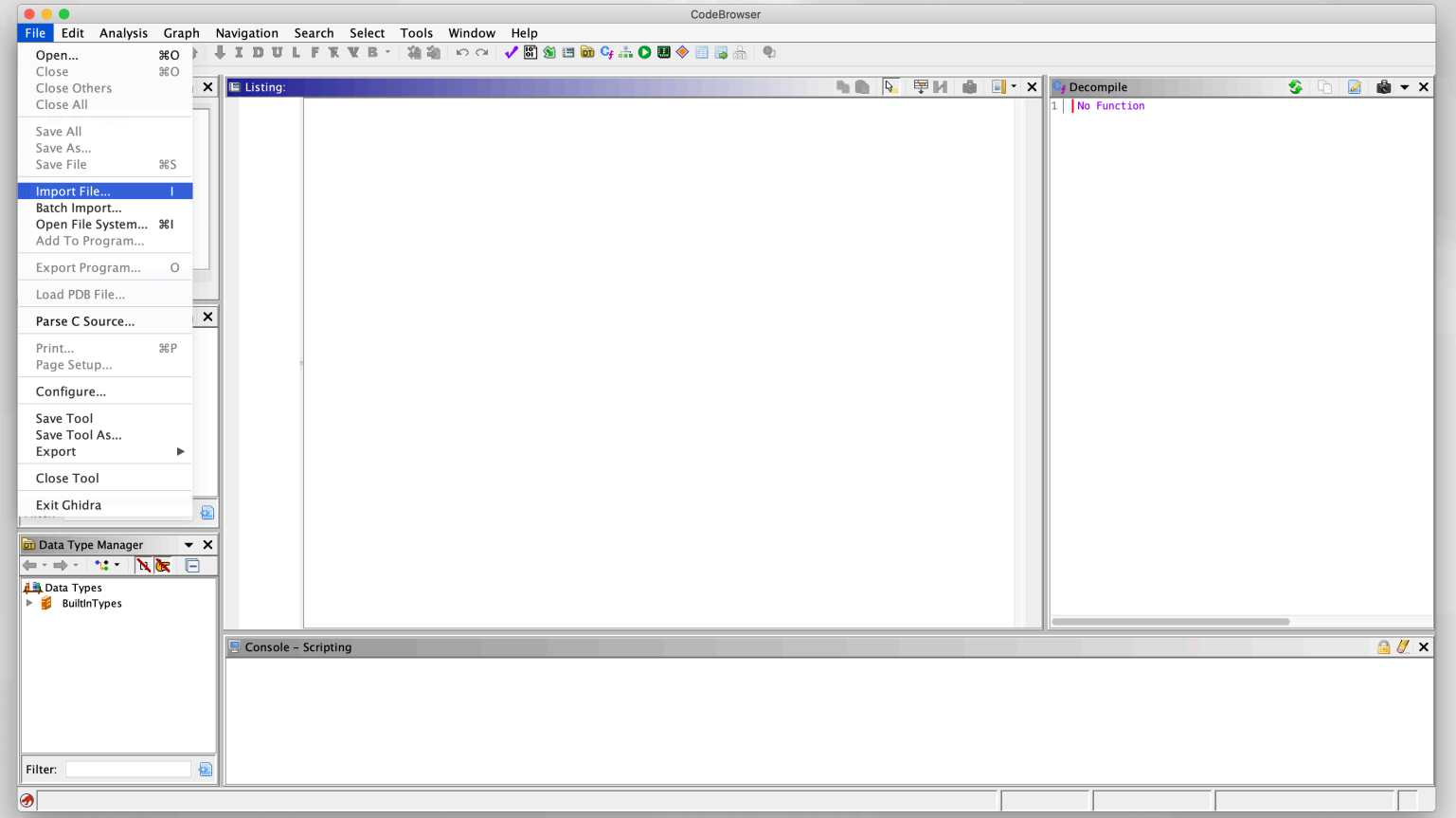
2-2-3. 適当なProject DirectoryとProject Nameを入力



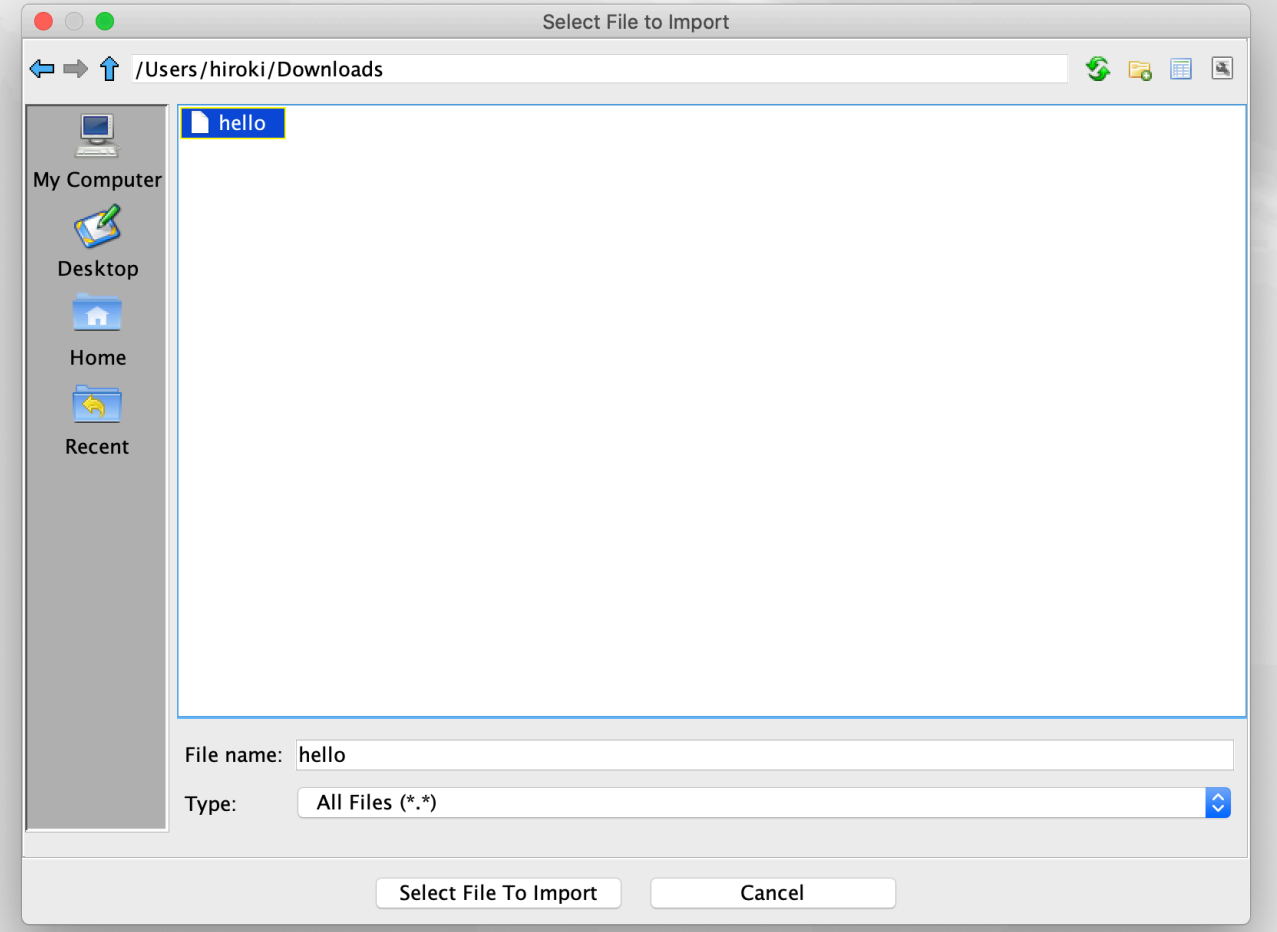
2-2-4. Tool Chestの緑のアイコンをクリック



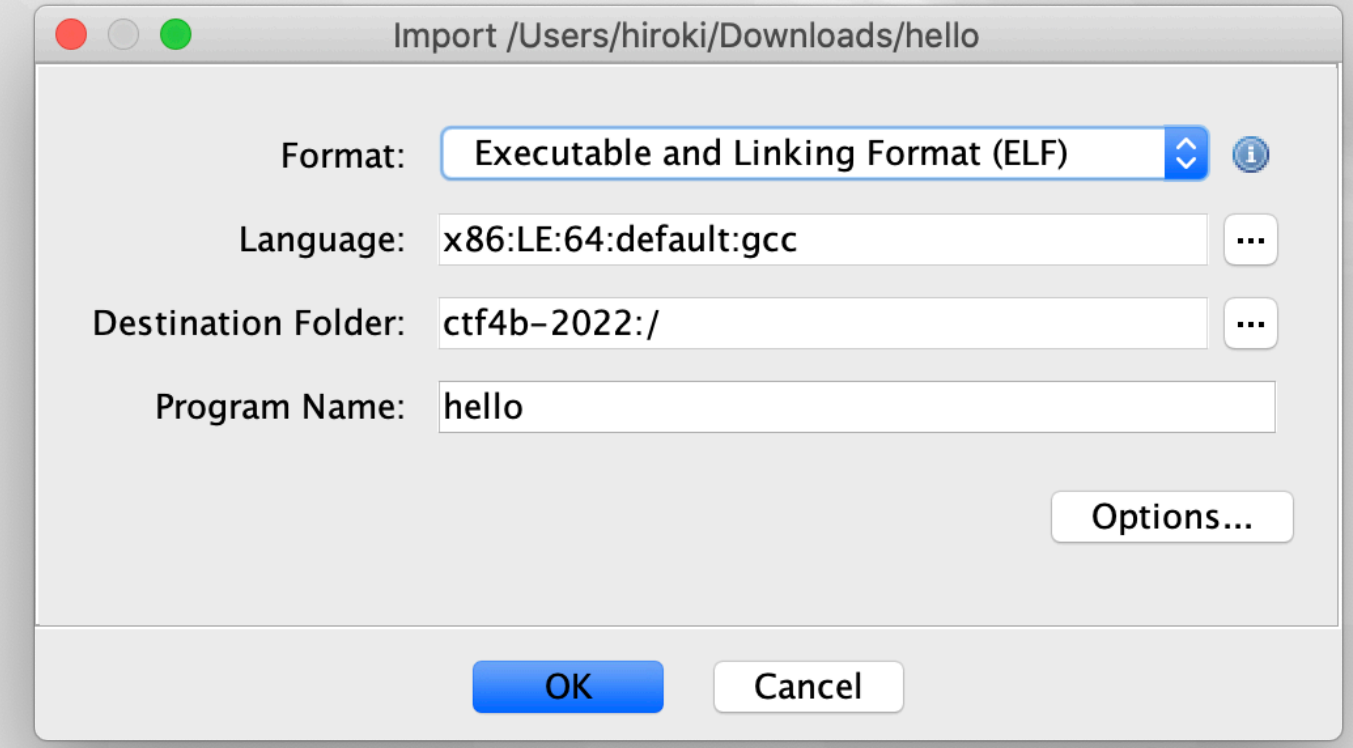
2-2-5. File→Import Fileから配布バイナリを選択



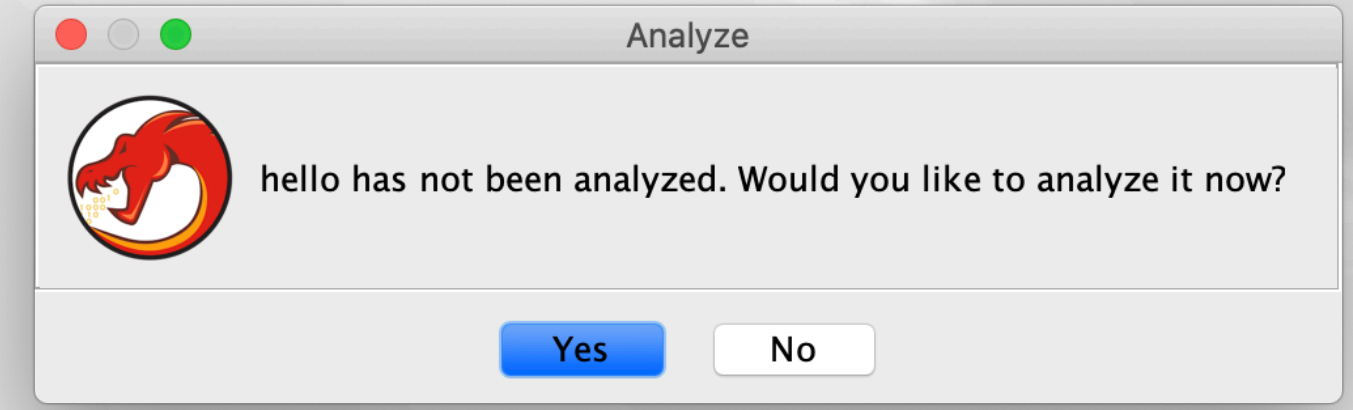
2-2-5. File→Import Fileから配布バイナリを選択



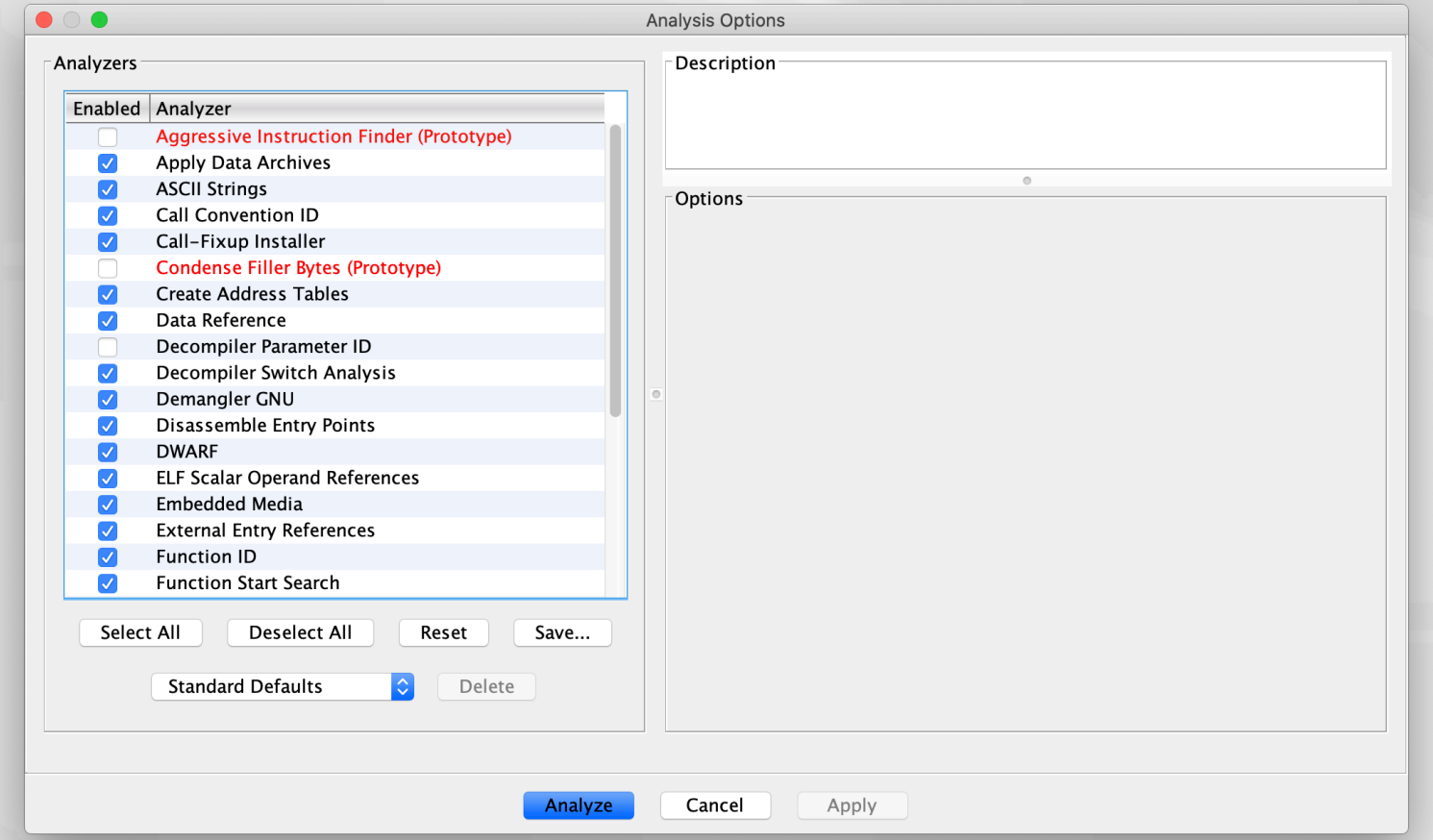
2-2-5. File→Import Fileから配布バイナリを選択



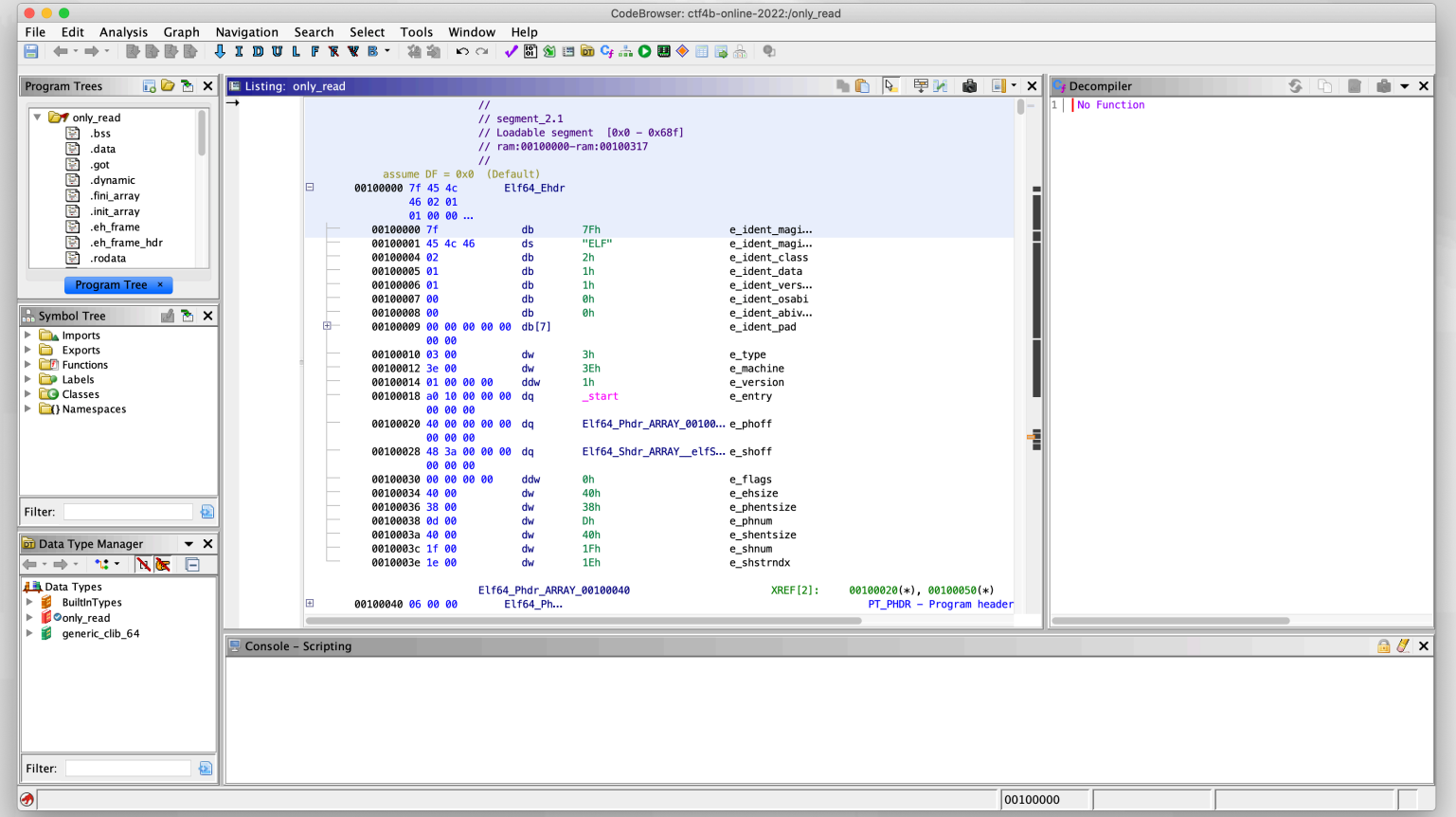
2-2-6. 「analyze now?」にYes



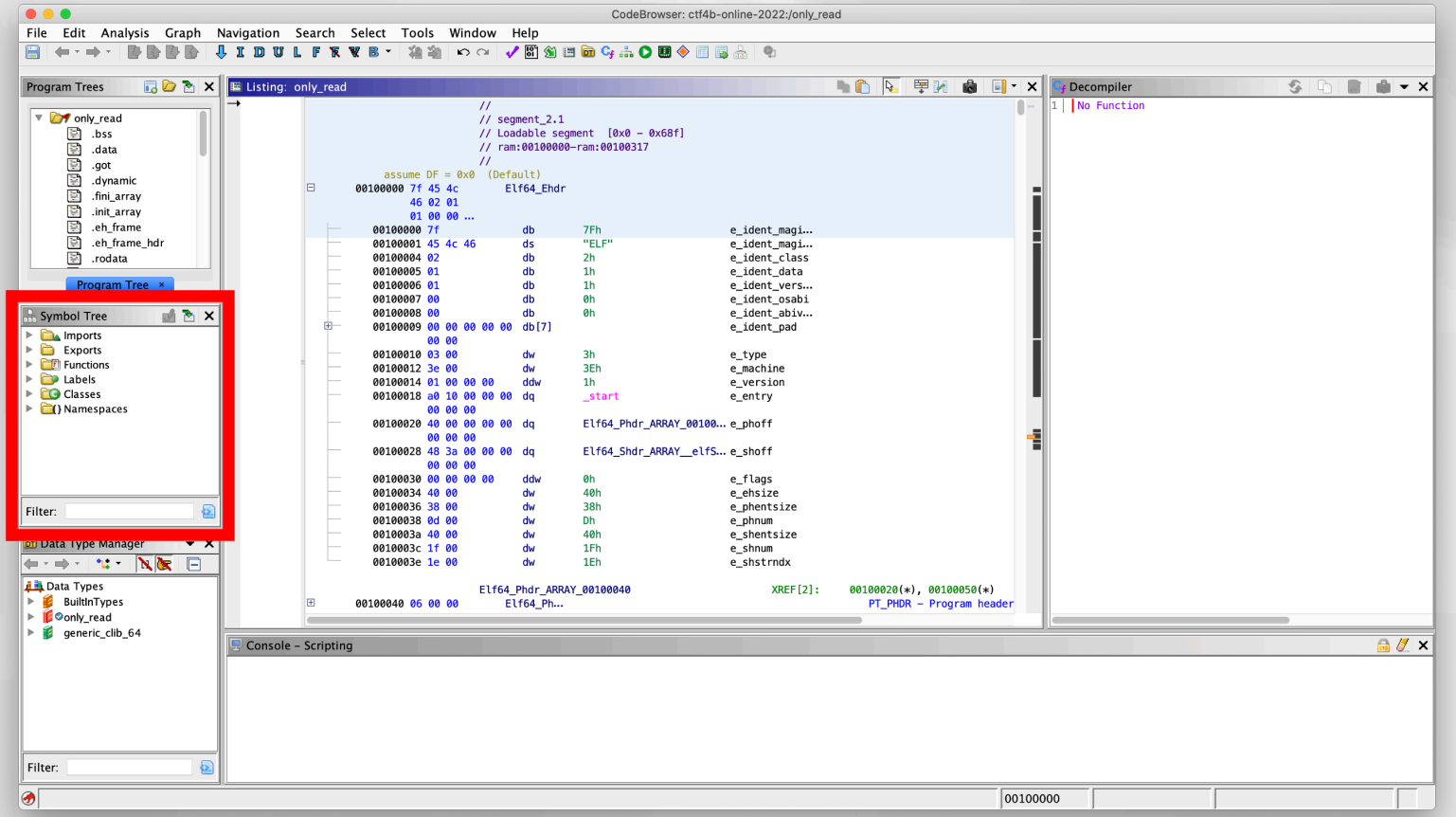
2-2-7. Analysis Optionはデフォルトで OK



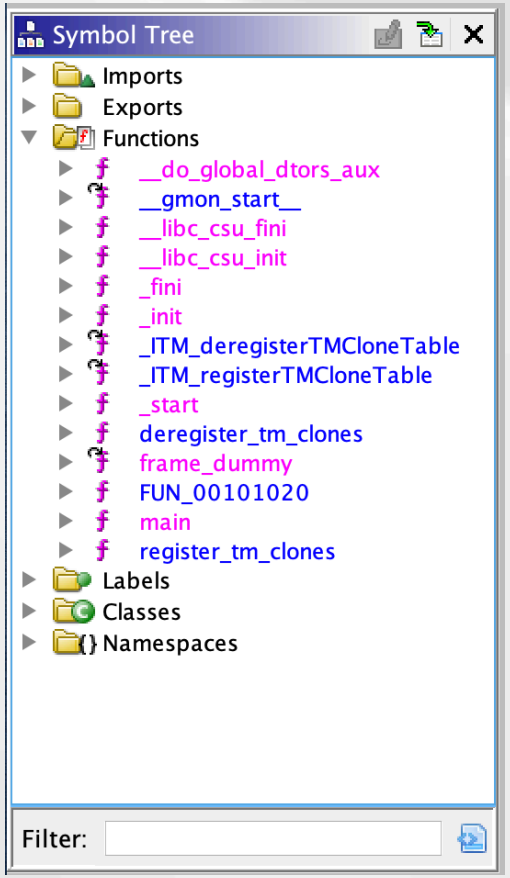
2-2-8. 左の真ん中のSymbol TreeのFunctionsを展開しmainを選択



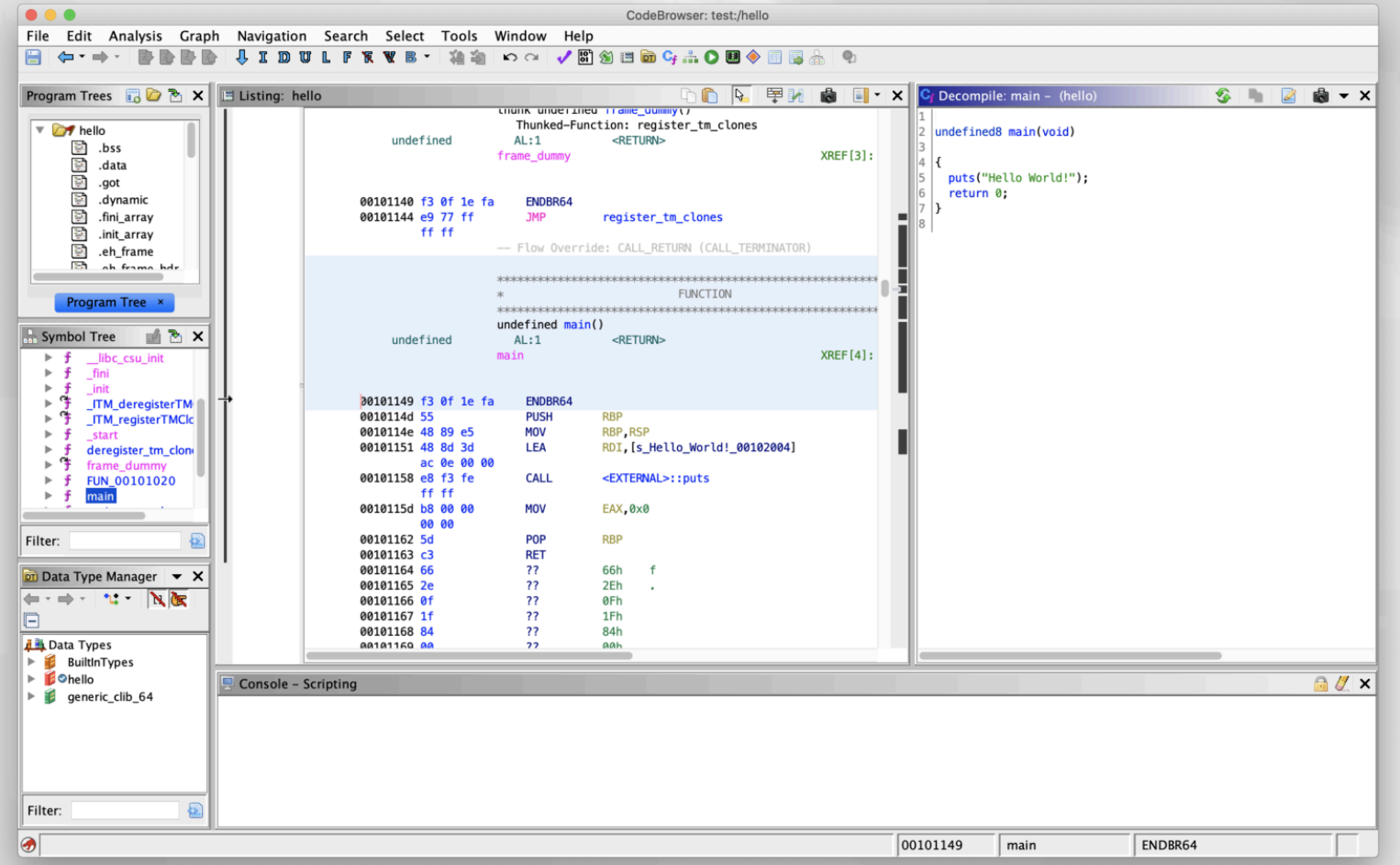
2-2-8. 左の真ん中のSymbol TreeのFunctionsを展開しmainを選択



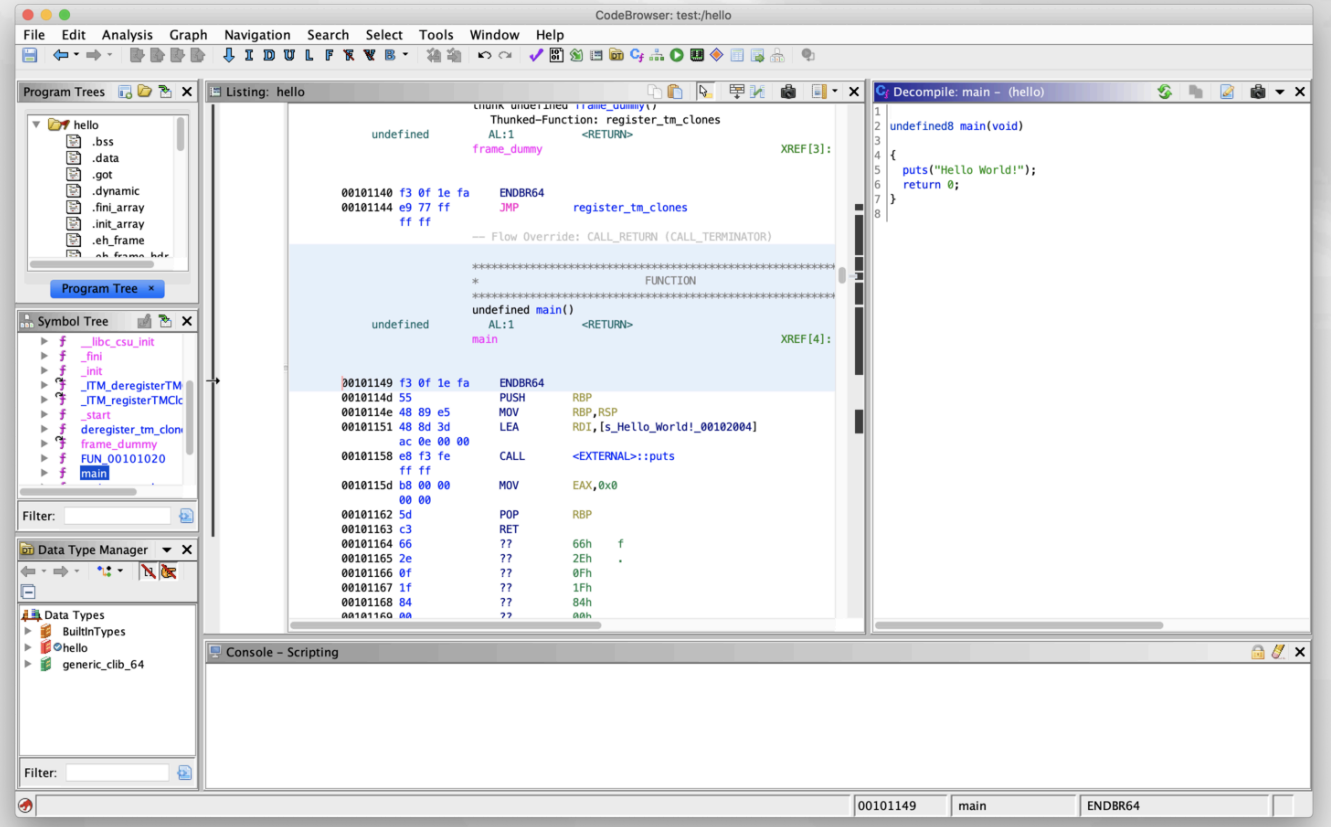
2-2-8. 左の真ん中のSymbol TreeのFunctionsを展開しmainを選択



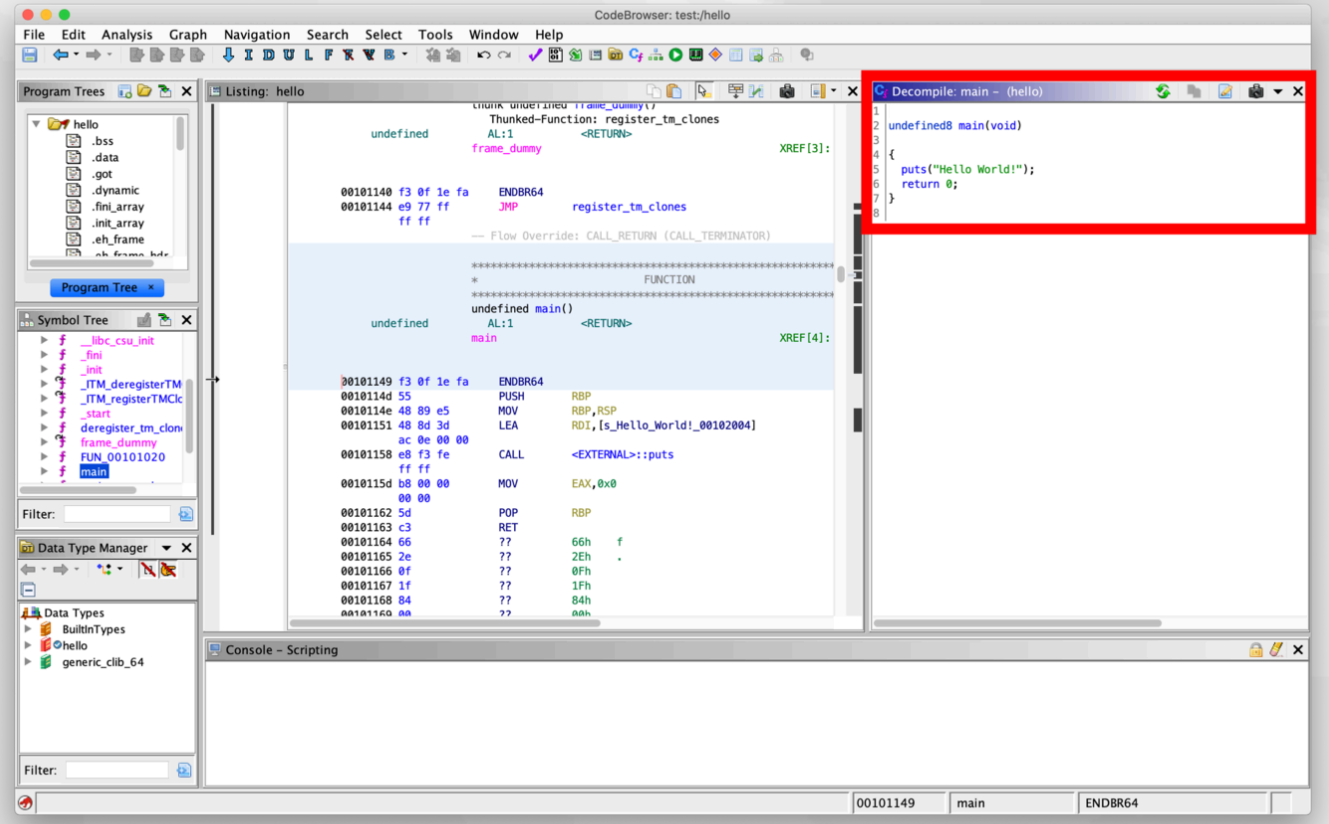
2-2-9. 真ん中に逆アセンブル結果，右に逆コンパイル結果が表示される



3. 一番右のパネル内の表示を確認する

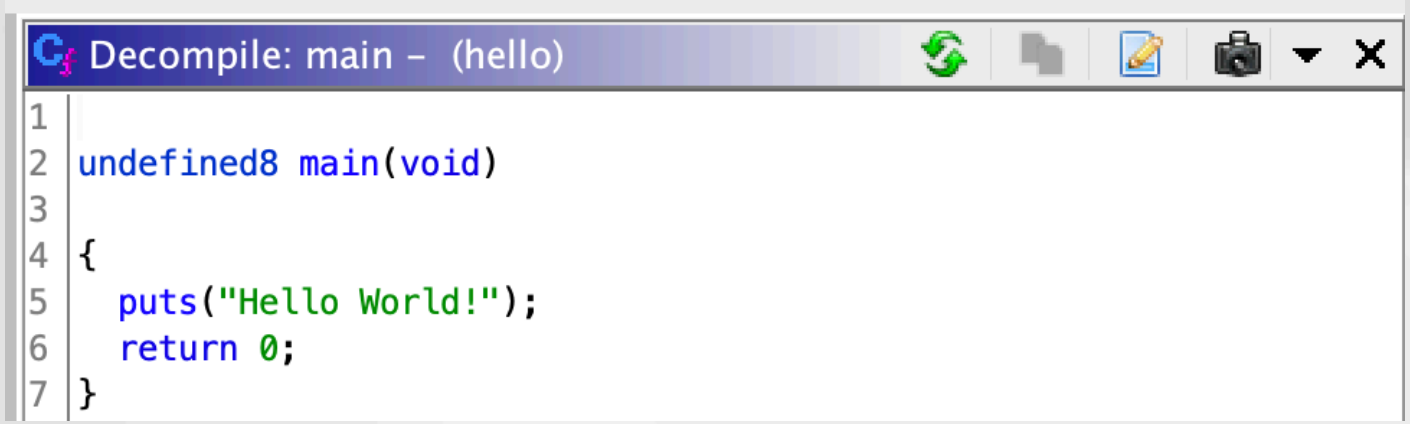


3. 一番右のパネル内の表示を確認する



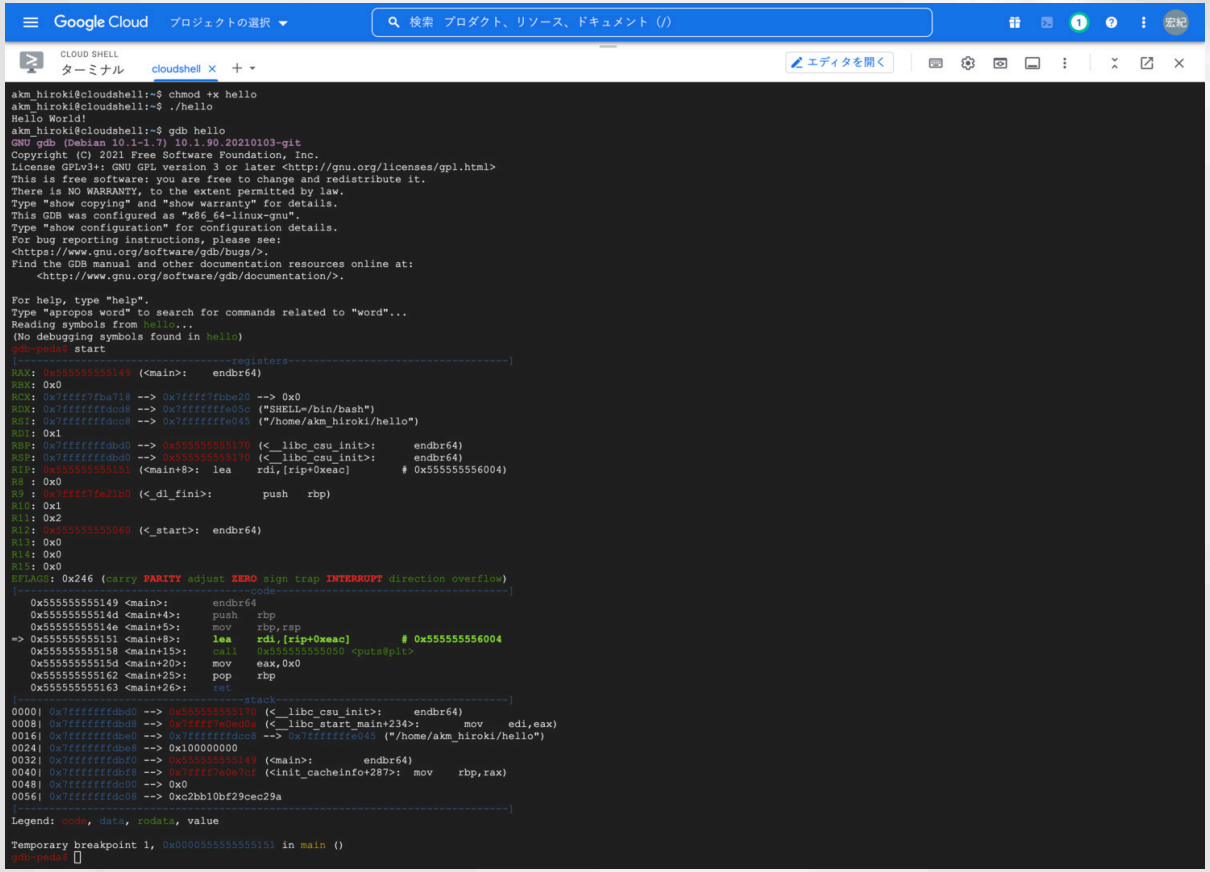
3. 一番右のパネル内の表示を確認する

`puts("Hello World!");` が表示されていれば事前準備①は完了です



事前準備②のゴール

x86-64のLinux実行環境を用意し，GDBとPedaをインストールし，指定のファイルを読み込み，以下のような画面が表示される



事前準備②の手順

1. Google CloudShell(x86-64のLinux実行環境)を用意する
2. Pedaのセットアップ
3. 指定のファイルを読み込む

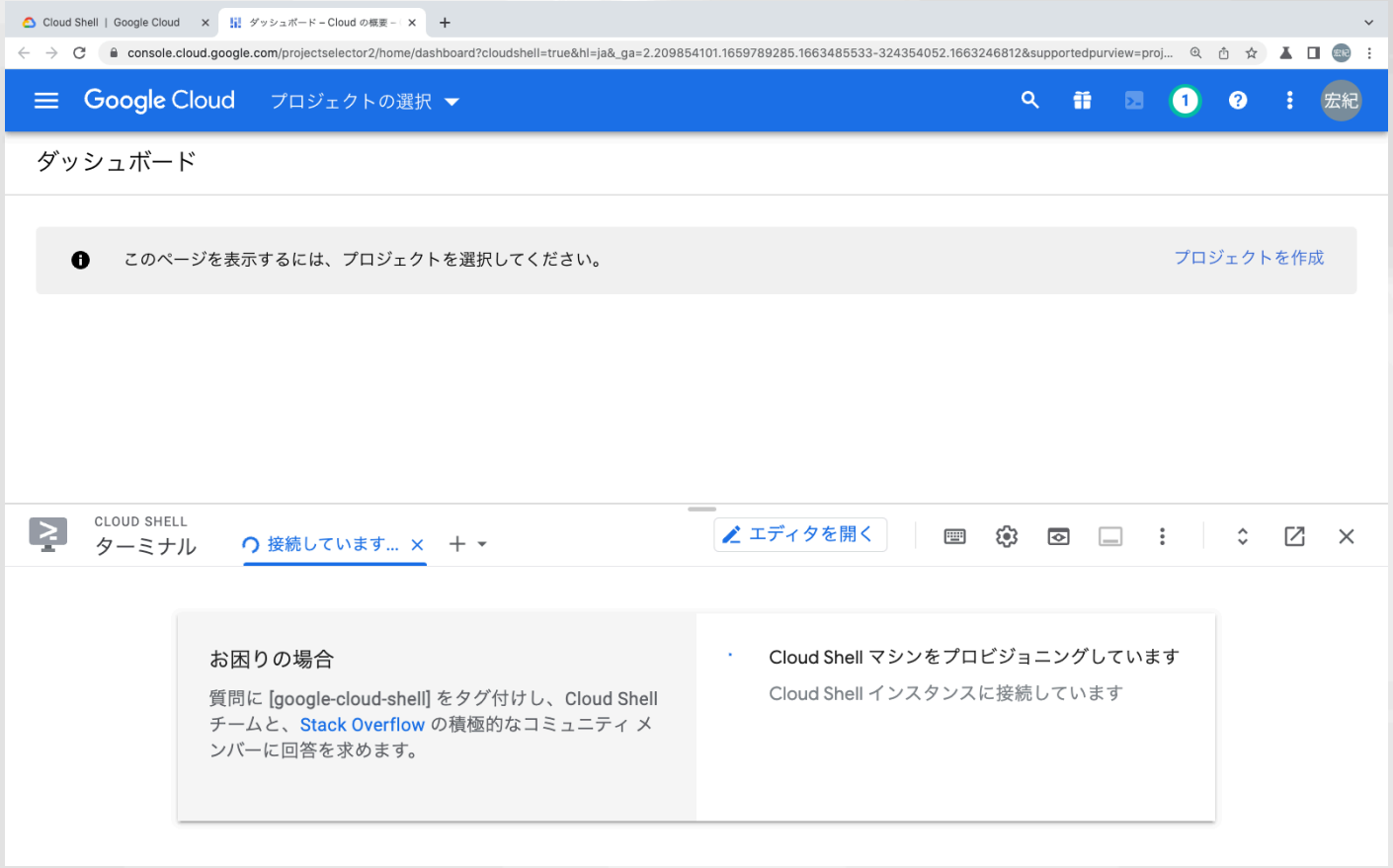
2. Google CloudShellを用意する

Google Cloud Shell(無料)を利用します

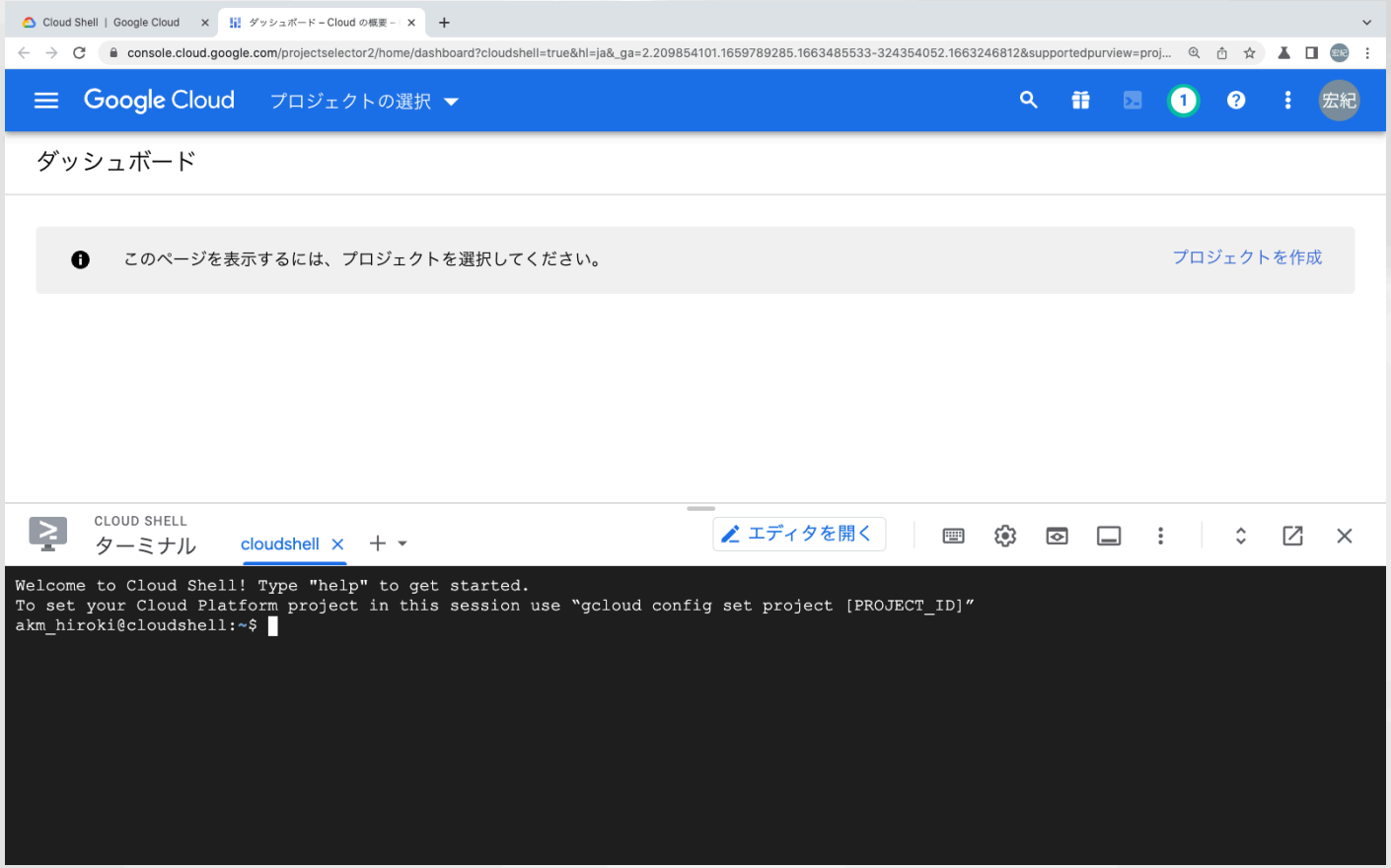
手元で環境を用意できる場合はCloud Shellでなくても良いです。

ブラウザでhttps://cloud.google.com/shellを開き，「コンソールを開く」をクリックします

画面の下半分でターミナルの起動を待ちます



ターミナルが起動し，x86-64のLinux実行環境を使えるようになります



2. Pedaのセットアップ

https://github.com/longld/peda の案内に従って以下の2つのコマンドを実行します

```
$ git clone https://github.com/n01e0/peda.git ~/peda
$ python3 -m pip install -U -r ~/peda/requirements.txt
$ echo "source ~/peda/peda.py" >> ~/.gdbinit
```

3. 指定のファイルを読み込む

`wget`コマンドで指定のファイルをダウンロードし，`chmod`コマンドで実行権限を付与して実行し，`Hello World!`が表示されることを確認します．

```
$ wget https://github.com/hi120ki/ctf4b-2022-testbin/raw/main/hello
$ chmod +x hello
$ ./hello
Hello World!
```

3. 指定のファイルを読み込む

その後，`gdb hello`コマンドを実行し，GDBを起動したあと`start`コマンドを実行し以下のような画面が表示されれば事前準備②は完了です

```
$ gdb hello
gdb-peda$ start
```