

#00b

# プログラミング環境の構築

2022 年度 / プログラミング及び実習 III

角川裕次

龍谷大学 先端理工学部

# プログラミング環境について

## C 言語の開発環境を使用します

### 実習・課題の進め方のイメージ

- テキストエディタでプログラムコードを入力/作成
- コンパイル・実行・デバッグ

### 主に想定するプログラミング環境

- BYOD Windows 上の WSL2
- BYOD Mac 上の xcode
- 情報実習室 PC 上の Ubuntu

### 避けてほしいプログラミング環境

- paiza.IO (オンライン)
- Visual Studio (Windows 上)
- いずれも一部の实習内容 (ファイル関係) が実施できないため

# 環境構築：概要

## 環境構築 (WSL2 または xcode) ができているひと

- 「プログラミング及び実習 II」(佐野先生)を受講済の人が該当：  
環境構築済みのはずなのでそれを使ってください
- WSL2：1つ前のバージョンの WSL (WSL1) でも OK
- Ubuntu のバージョンは 18.04LTS でも 20.04LTS でも OK
- xcode：バージョンはとくに気にしなくて多分 OK

## 環境構築がまだのひと

- Windows なら WSL2 を,  
Mac なら xcode をインストールしてください
- 実習室 PC の Ubuntu を使うのもありますが  
自宅等での学習のために BYOD 上にインストールのが良い

## 本講義・実習で使うソフトウェア

- C コンパイラ (cc)
- テキストエディタ (nano, emacs, vi, vs code, 等, どれかひとつ)
- ターミナル (端末, コンソール, コマンドライン, など呼び方は様々)

## C 言語開発環境の確認

開発環境のインストールができたと思ったら...  
以下のプログラムを入力・実行して確認して下さい

C 言語ソースコード hello.c

```
#include <stdio.h>

int main(void) {
    printf("HELLO WORLD\n");
    return 0;
}
```

実行 (ターミナル/端末上にて)

```
cc hello.c
./a.out
```

実行結果の出力

```
HELLO WORLD
```

# スクリーンショットのとり方メモ

演習の実行結果の提出の際に必要です

- (ただしソースコードはファイルそのままを提出)

スクリーンショット用のアプリの代表的な例を以下に紹介

- 使い方詳細は各自でググって下さい

Windows10 (WSL2) : アプリ Snipping Tool を使う例

- 「スタート」メニュー → 「Windows アクセサリ」 → 「Snipping Tool」をクリック
- モードにはウインドウを選択, WSL2 のウインドウを切り取り, ファイルの保存

macOS : ショートカットキーでできる

- 「Command + Shift + 4 + Caps Lock」を同時に押す
- キャプチャーする範囲を選択; 自動的にファイルに保存される

Ubuntu : アプリを使う

- 左上「アクティビティ」 → 検索窓で「スクリーンショット」を入力 → 該当のアプリが出てくるので起動

## WSL2 のインストール (Windows10)

# WSL2 インストール手順のあらすじ

1. (必要に応じて) Windows のバージョンをアップグレード :  
WSL2 が動作できるように

- 「設定」→「システム」→「詳細情報」にて調べる
- Windows10 はバージョン 1903 以上のこと
- バージョンが古い場合は Windows をアップグレード

2. WSL2 をインストール

3. WSL2 用 Ubuntu 20.04LTS をインストール

4. WSL2 の初期設定

5. 開発ツールのインストール

# Windows 機能の有効化

デスクトップ左下の検索窓で検索: キーワード「機能の有効化」

「Windows の機能の有効化または無効化」で 2 つの機能を有効化

- 「Linux 用 Windows サブシステム」にチェックを入れる
- 「仮想マシンプラットフォーム」にチェックを入れる

変更したら Windows を再起動



# WSL2 ダウンロード

## 1. Linux カーネル更新プログラムパッケージのダウンロードと実行

- ブラウザで <https://aka.ms/wsl2kernel> をアクセス
- 「手順 4 Linux カーネル更新プログラムパッケージをダウンロードする」から入手
- ダウンロードしたファイルを実行

## 2. WSL2 (WSL1 でなくて) を規定バージョンにするための設定手順

- 1 PowerShell を起動 (必ず管理者モードで)
- 2 PowerShell 上で以下を実行

```
wsl --set-default-version 2
```

## 3. Microsoft Store を開いて Ubuntu 20.04 LTS を入手・インストール

# WSL2 の設定

## 1. WSL2 の起動

- Windows10 メニュー (画面左下) に Ubuntu 20.04 LTS が登録済み
- それをクリックすれば WSL2 が起動

## 2. WSL2 でのユーザー名とパスワードの設定

- ユーザー名 (半角文字で)
- パスワード (半角文字で) 同じものを 2 回入力
- 忘れないように : WSL2 の使用のログインの際に入力します

## 3. Ubuntu の更新 & 日本語設定 & プログラム開発ツールのインストール

- WSL2 上で以下のコマンドラインを実行 (パスワード聞かれたら入力)

```
sudo apt-get update
sudo apt-get upgrade
sudo apt install language-pack-ja
sudo update-locale LANG=ja_JP.UTF-8
sudo apt install manpages-ja manpages-ja-dev
sudo apt install build-essential
```

# VS code (Visual Studio Code) のインストール

【Microsoft による開発環境; 実習でのプログラム入力に便利】

インストール作業は Windows10 上で行う

- Windows 版をダウンロード & インストール

「Windows10 VS code インストール」でググればやり方が分かる

- 以下の解説 Web ページを見てやってみる

「VScode のインストール手順@Windows10」

<https://qiita.com/Shi-nakaya/items/c43fb6c1e638d51bf1c8>

(2021/03/07 閲覧)

# VS Code に “Remote-WSL” 機能の追加

【Windows10 上の VS Code から WSL2 のファイル編集等ができる】

## 0. WSL2 を起動

### 1. Windows10 上で VS Code を起動

“Recommended Extension” (おすすめの拡張機能) のインストールですぐできる

- 手作業でのインストールは以下の手順

### 2. VS Code にて “Remote-WSL” をインストール

- 「表示」→「拡張機能」の検索窓で “remote wsl” → インストール
- 以下の手順解説 Web ページを見てやってみる

「Visual Studio Code の「Remote-WSL」がめっちゃくちゃ便利」  
<https://syobon.jp/blog/2019/11/04/visual-studio-code-remote-wsl/>  
(2021/03/07 閲覧)

# VS code メニューの日本語化

【英語メニューが日本語になってくれて便利】

1. VS Code を起動
2. 「Windows10 VS code 日本語化」でググる
  - 以下の手順解説 Web ページを見てやってみる  
「Visual Studio Code を日本語化する方法」  
<https://qiita.com/HiroCh/items/481adfa969dbe689f566>  
(2020/08/04 閲覧)
3. VS Code にて “Japanese Language Pack” をインストール
4. VS Code を再起動

# WSL2 + VS Code :

## 実習でのプログラミング作業のフロー例

1. Windows10 上で WSL2 を起動
2. WSL2 のコマンドラインで VS Code を起動

```
code
```

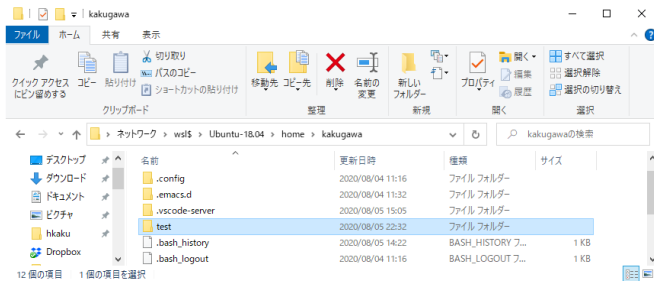
3. VS Code でプログラムを入力・修正
  - そのたびにファイルを「保存」
4. コンパイル&実行 : WSL2 のコマンドラインで行う
  - cc コマンド (C コンパイラ) など

```
cc test.c
```

# Windows10 ⇔ WSL2 間のファイルのやり取り (1/2)

WSL2 上で完成したソースコードを manaba へ提出する時など

- 1 WSL2 を起動しておくのを忘れずに
- 2 エクスプローラを開いてパス窓に \\wsl\$ を入力
- 3 Ubuntu-20.04 → home → ユーザ名
- 4 WSL ログイン時のファイルが見える (ホームディレクトリ)
- 5 あとはいつものようにファイルを移動/コピー



(この画像はちょっと古い Ubuntu-18.04 での画像です. すみません)

## Windows10 ⇔ WSL2 間のファイルのやり取り (2/2)

Q. プログラムコードなどを manaba に提出するにはどうする？

A. 以下の手順です

- 1 エクスプローラーで WSL2 から Windows10 へファイルをコピー
- 2 Windows10 上の Web ブラウザで manaba にログイン
- 3 コピーしたファイルを Web ブラウザで manaba へ提出

「WSL ファイル共有」に関するほかの情報

- 「wsl2 でよく使う、windows と linux 間のファイル移動やコマンド実行.」  
<https://qiita.com/Uchitaso/items/6e0a7859e87bb8bdb527>  
(2021/03/16 閲覧) けっこう詳しく書いてある
- 「WSL と windows 間のファイル連携」  
<https://qiita.com/quzq/items/1096c638c0d86795be13>  
(2020/08/05 閲覧)



# WSL2 あるある

「ソースファイルを作ったのに見つかりません」

- Windows 上で作ってないですか？
- WSL2 内でファイルを作ってください

「提出するファイルがエクスプローラーで探しても見つからないです」

- WSL2 を起動していないのでは？

xcode のインストール (macOS)

# 開発環境のインストール

Mac を持っていないので実際に試せてません. すみません  
以下の方法で C コンパイラ等の開発環境が整うはず

1. ターミナルを起動
2. 開発環境のインストール

```
xcode-select --install
```

## 有益な情報源のリスト

BYOD (ノート PC 必携化) サポートページ

<https://byod.st.ryukoku.ac.jp/byod/>

佐野先生の分かりやすい手順説明

- 不明点があるときの質問は角川&TA へ

「Windows10 への WSL2 の導入」

<https://www-slab.math.ryukoku.ac.jp/lecture/prog2/wsl2>

「Windows10 WSL2 の C 言語開発環境」

<https://www-slab.math.ryukoku.ac.jp/lecture/prog2/wsl2cc>

「WSL のための Visual Studio Code の設定」

<https://www-slab.math.ryukoku.ac.jp/lecture/prog2/vsc4wsl>

「WSL と Windows とのファイル共有」

<https://www-slab.math.ryukoku.ac.jp/lecture/prog2/wsl2share>

「macOS の C 言語開発環境」

<https://www-slab.math.ryukoku.ac.jp/lecture/prog2/maccc>

おわり