

FRENDYのインストール方法について

概要

- インストール先 : Ubuntu (Ubuntu 20.04.1 LTS)
 - Windows subsystem for Linuxを用いて、WindowsにUbuntuをインストール
 - インストールしたUbuntu上でgcc等をインストールし、FRENDYをコンパイルする
- 本資料の説明は以下の通り
 - Windows subsystem for Linuxの導入
 - Ubuntuの設定
 - FRENDYのコンパイル

Windows subsystem for Linux の導入

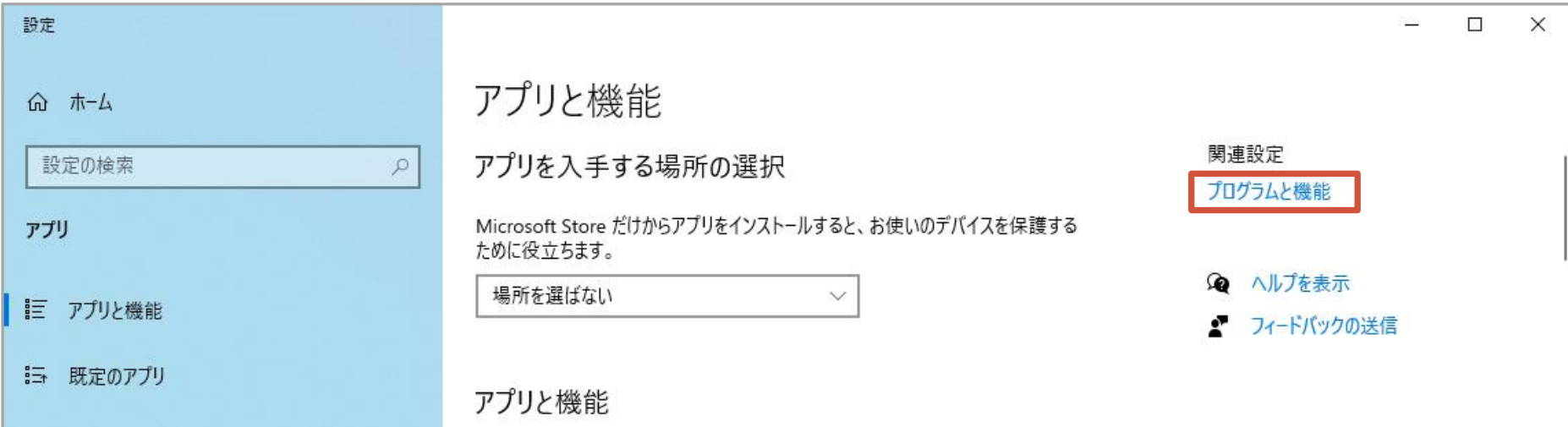
Windows subsystem for Linuxの導入 (1)

- Windowsアイコンを右クリック
- 『アプリと機能(F)』を起動



Windows subsystem for Linuxの導入 (2)

- 右上の関連設定の『プログラムと機能』をクリック

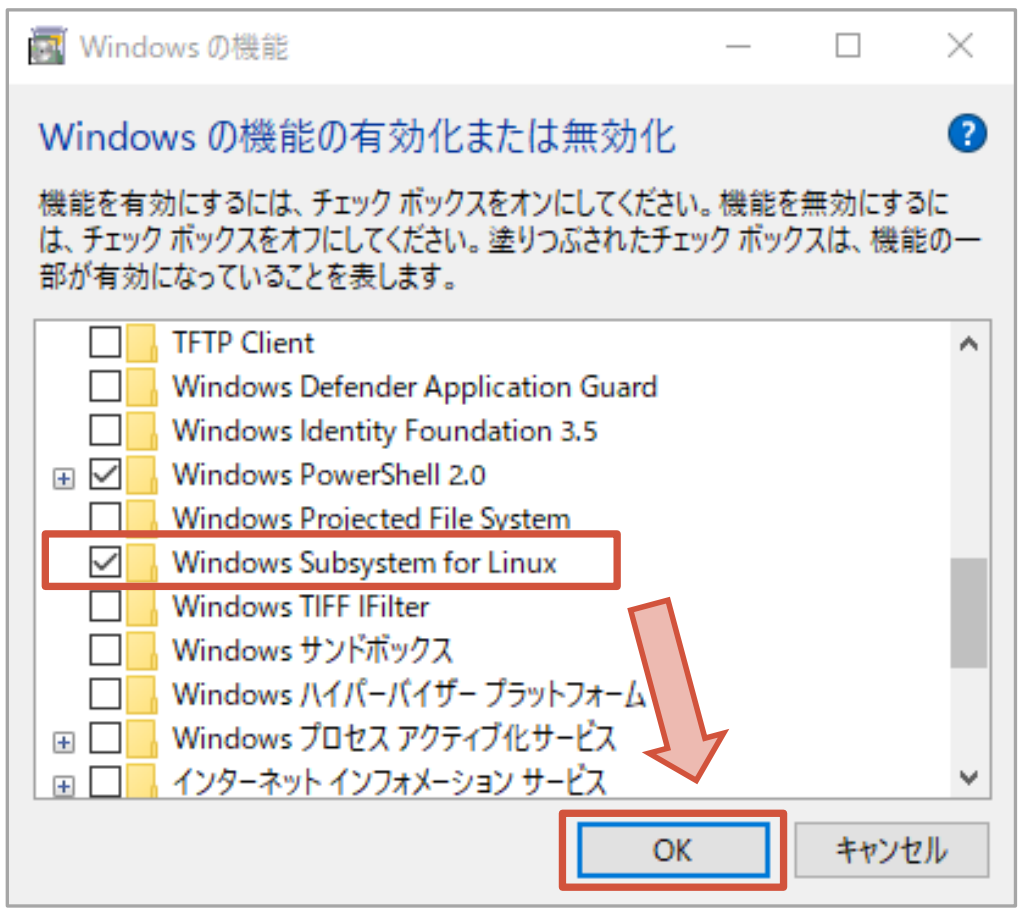


- 左の『Windows機能の有効化または無効化』をクリック



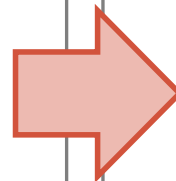
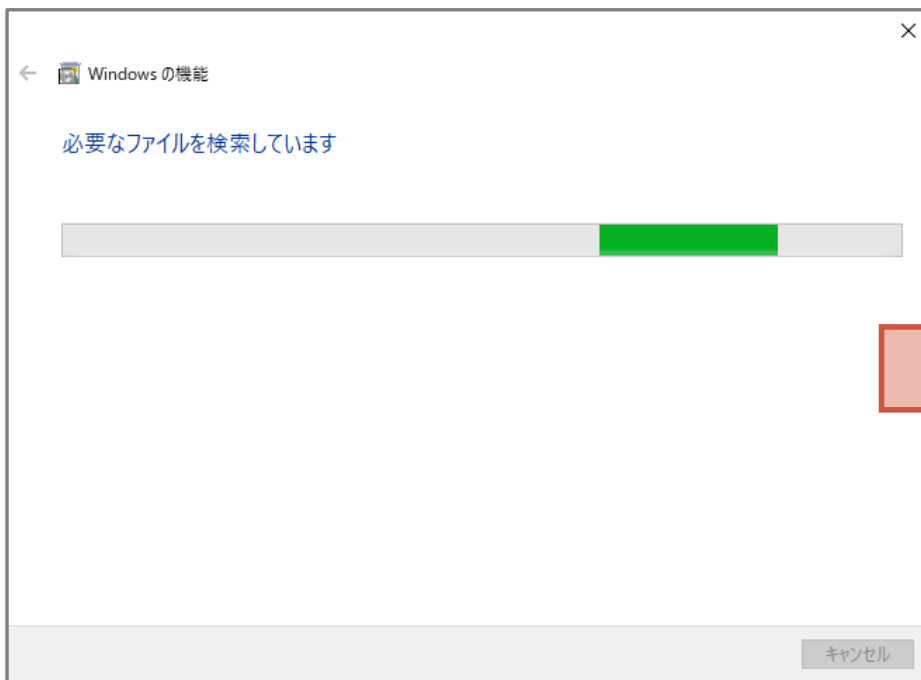
Windows subsystem for Linuxの導入 (3)

- 中段辺りにある『Windows Subsystem for Linux』のチェックボックスをオンにして、『OK』



Windows subsystem for Linuxの導入 (4)

- 必要なファイルを検索していますと表示されるので、しばらく待つ
- 変更完了後、再起動を求められるので、Windowsを再起動する



WSLへのUbuntuのインストール (1)

- Windowsを再起動後、Microsoft StoreでUbuntuを検索
- インストールしたいLinuxを選択

Microsoft Store

ホーム ゲーム エンターテイメント 仕事効率化 お買い得商品

検索

結果: Ubuntu

部門: 全ての部署

次で使用可能: パーソナル コンピューター

検索をクリックし、Ubuntuと入力

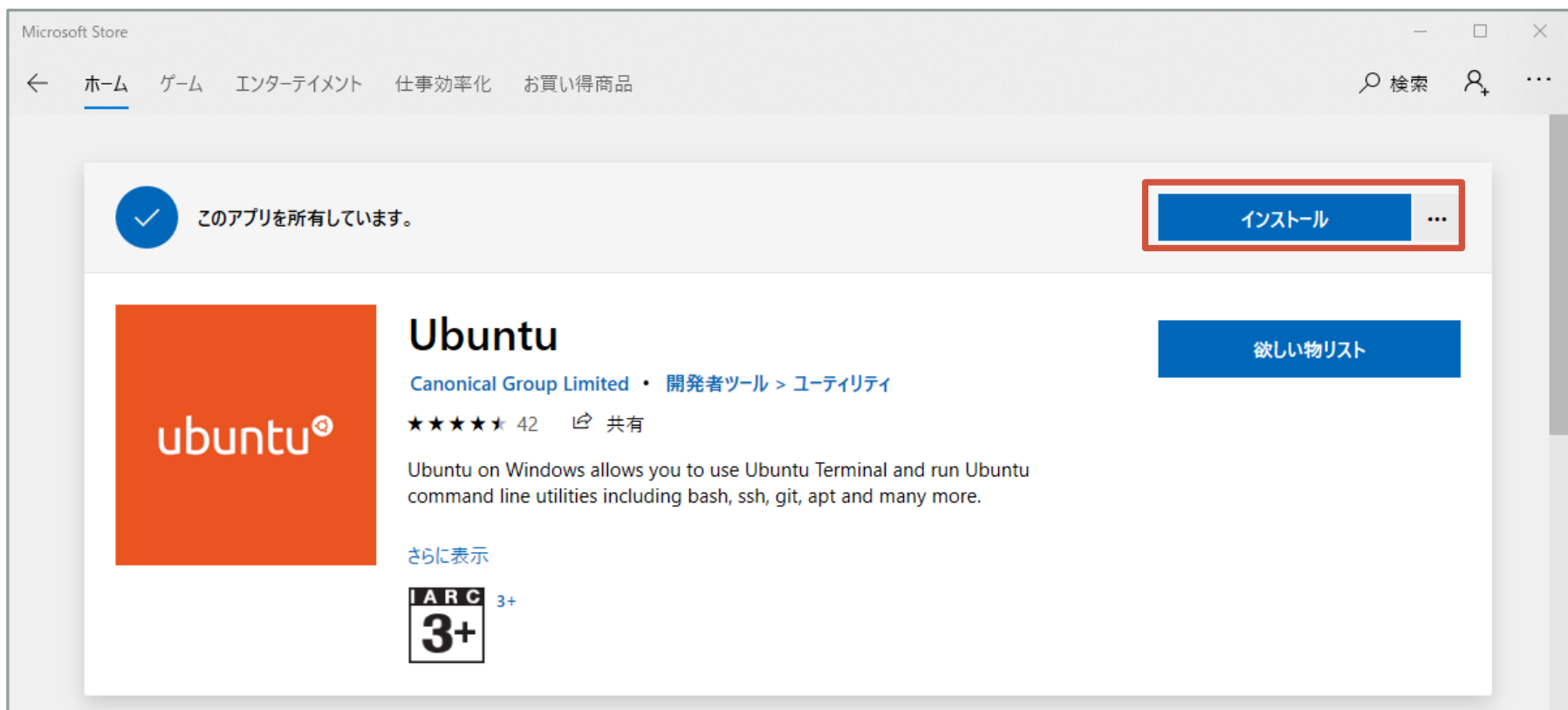
アプリ (12) すべて表示

好みが無いなら無印のUbuntuをお勧めします

アプリ名	評価	価格
Ubuntu	★★★★☆ 42	無料
Ubuntu 20.04 LTS	★★★★★ 4	無料
Ubuntu 18.04 LTS	★★★★★ 11	無料
XSERVER4WINDOWS10 X410	★★★★★ 8	¥4,700 割引 ¥5,050 ¥1,150
Raft WSL	★★★★★ 1	無料+
Linux Cheatsheet	☐☐	無料+
2buntu.com	☐☐	無料

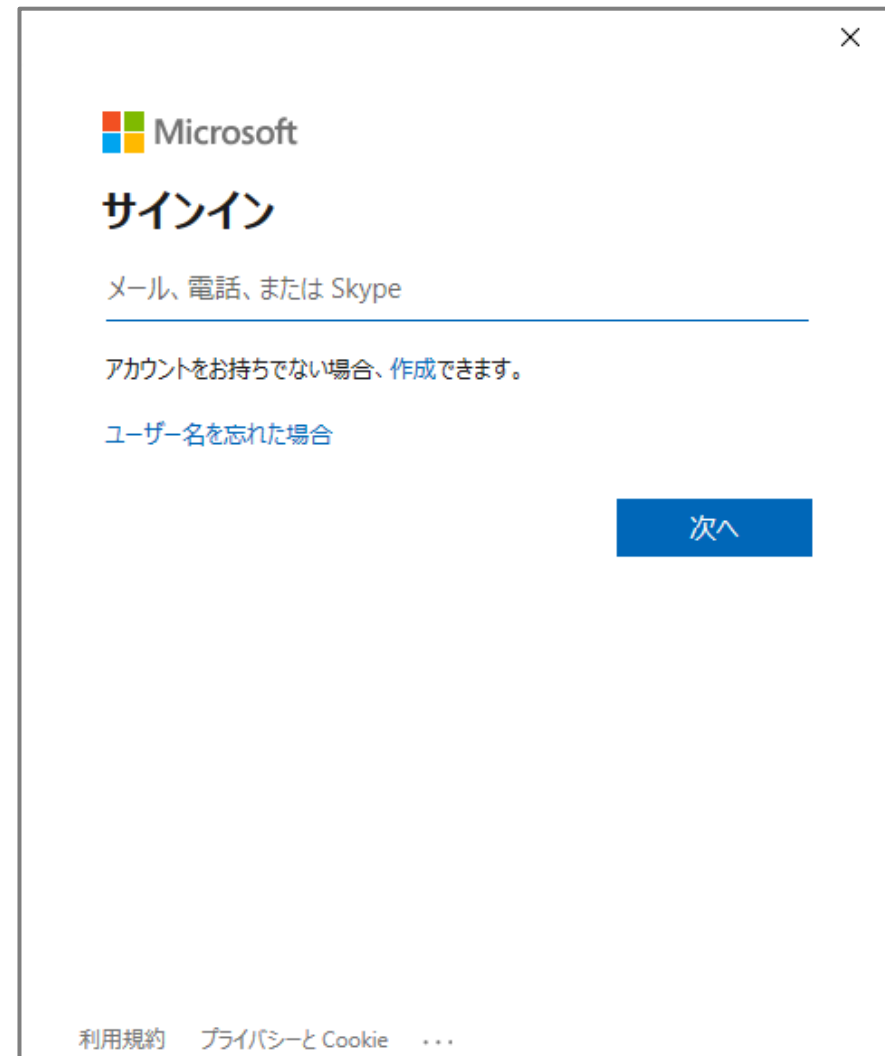
WSLへのUbuntuのインストール (2)

- アプリの画面が出てきたらインストールをクリック



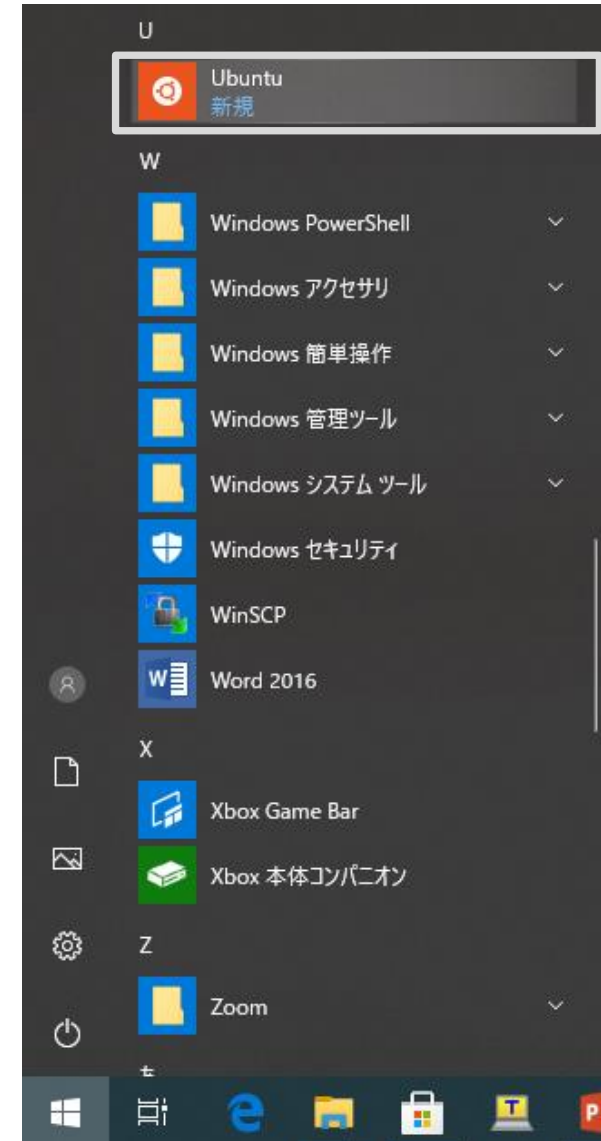
WSLへのUbuntuのインストール (3)

- 場合によってはサインインを求められるので、その場合はサインインを行う



WSLへのUbuntuのインストール (4)

- Ubuntuのインストールが完了すると、プログラム一覧にUbuntuが登録される
 - Ubuntuをクリックすれば、起動する



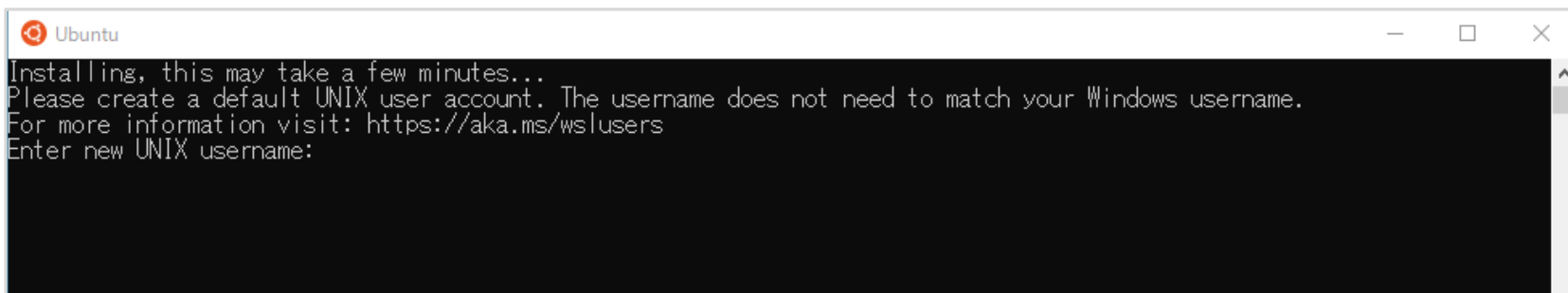
WSLへのUbuntuのインストール (5)

- Ubuntuを起動すると、図のようにターミナルが開き、インストール作業を行う
 - インストール作業が終わるまで数分～十数分程度かかります



WSLへのUbuntuのインストール (6)

- しばらくすると、Ubuntuで用いるユーザー名、パスワードを求められる
 - このユーザー名、パスワードはWindowsのものと同じである必要はない
 - パスワードはsudoコマンドなどを使う場合に必要となる
 - 本資料ではユーザー名を『tada』と設定



```
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:
```

WSLへのUbuntuのインストール (7)

- ユーザー名、パスワードを入力すると、Ubuntuを使う準備が完了する

```
tada@DESKTOP-8DR8GSO: ~
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 20.04.1 LTS (GNU/Linux 4.4.0-18362-Microsoft x86_64)

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

System information as of Thu Aug 27 18:37:41 JST 2020

System load:  0.52      Processes:            7
Usage of /home: unknown  Users logged in:     0
Memory usage: 36%      IPv4 address for eth1:
Swap usage:   0%

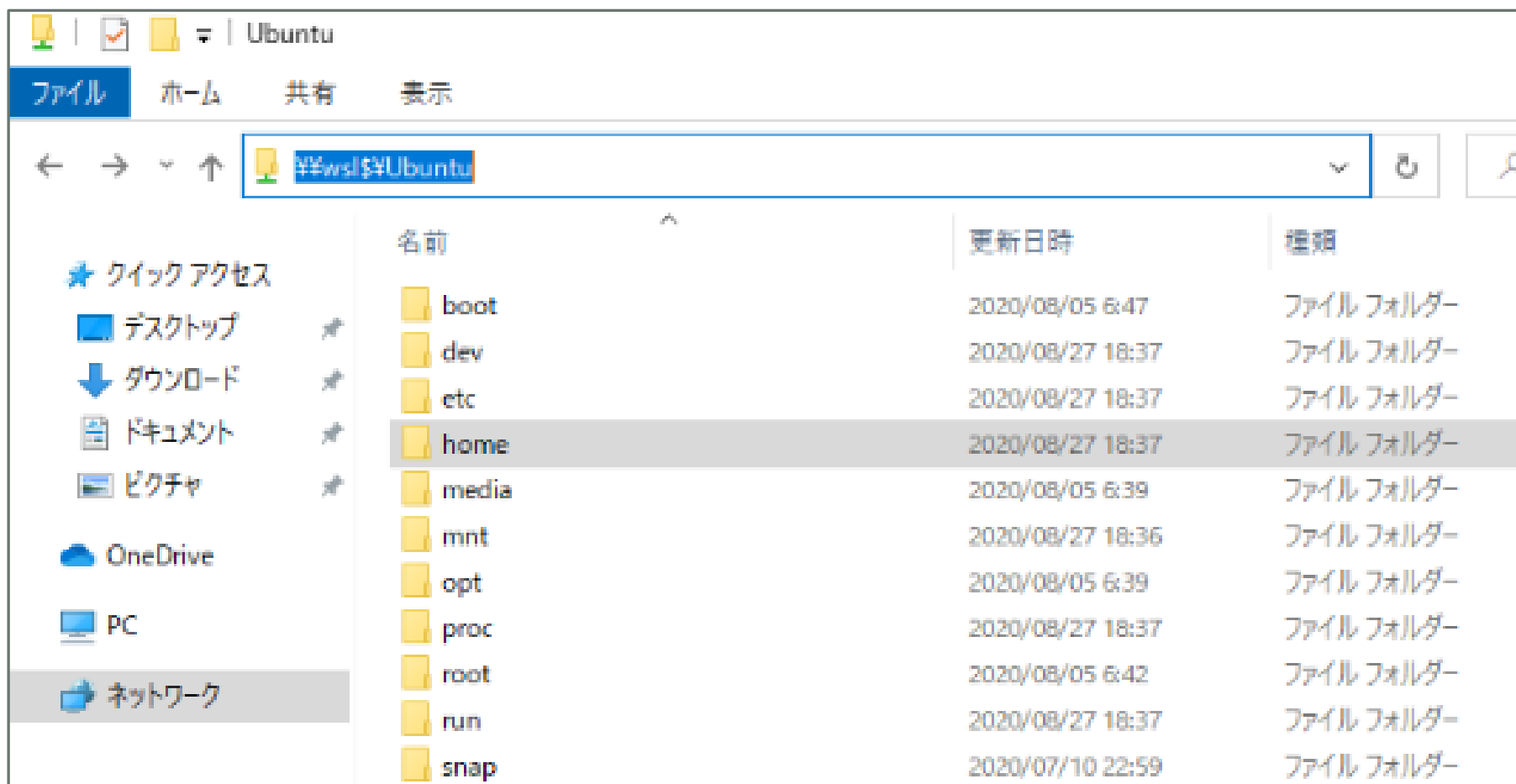
1 update can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

This message is shown once once a day. To disable it please create the
/home/tada/.hushlogin file.
tada@DESKTOP-8DR8GSO: ~$
```

Ubuntuのディレクトリ

- Ubuntuのディレクトリは以下の通り
 - [\\ws\Ubuntu](#)



Ubuntuを効率よく使うために

- デスクトップのディレクトリのリンクを貼ると便利
 - `ln -s /mnt/c/Users/ユーザー名/Desktop/ディレクトリ名`
 - デスクトップ上のディレクトリでUbuntuとやり取りが可能に

```
tada@DESKTOP-8DR8GSO: ~  
tada@DESKTOP-8DR8GSO:~$ ln -s /mnt/c/Users/ユーザー名/Desktop/wsl_data/ ./  
tada@DESKTOP-8DR8GSO:~$ ll  
total 44  
drwxr-xr-x 1 tada tada 4096 Sep 23 20:06 .  
drwxr-xr-x 1 root root 4096 Aug 27 18:37 ..  
-rw-r----- 1 tada tada 5502 Sep 23 20:03 .bash_history  
-rw-r----- 1 tada tada 220 Aug 27 18:37 .bash_logout  
-rw-r----- 1 tada tada 3771 Aug 27 18:37 .bashrc  
drwxr-xr-x 1 tada tada 4096 Sep 1 15:55 .cache/  
-rw-rw-rw- 1 tada tada 53 Aug 27 19:48 .gitconfig  
-rw-r----- 1 tada tada 21 Sep 1 15:54 .gnuplot_history  
drwxr-xr-x 1 tada tada 4096 Aug 27 18:37 .landscape/  
-rw-rw-rw- 1 tada tada 0 Sep 23 11:08 .motd_shown  
-rw-r----- 1 tada tada 807 Aug 27 18:37 .profile  
-rw-r----- 1 tada tada 0 Aug 27 18:57 .sudo_as_admin_successful  
-rw-r----- 1 tada tada 25972 Sep 1 18:13 .viminfo  
drwxrwxrwx 1 tada tada 4096 Sep 1 16:14 code/  
drwxrwxrwx 1 tada tada 4096 Sep 1 11:47 work/  
lrwxrwxrwx 1 tada tada 43 Sep 23 20:06 wsl_data -> /mnt/c/Users/ユーザー名/Desktop/wsl_data/
```


指定ディレクトリからのWSLの起動

- WSLではアクセスしたいディレクトリからの起動も可能
 - アクセスしたいディレクトリを開き、アドレスバーに『wsl』と入力するだけ
 - アクセスしたいディレクトリに『wsl』という名前のディレクトリ、ファイルがあるとうまく起動しないので要注意

The image shows a Windows File Explorer window with the address bar set to 'wsl'. A red box labeled 'アドレスバー' (Address Bar) points to the address bar. Below the address bar, a search dropdown shows 'wsl', 'wsl_data', and '"wsl" の検索'. A red box at the bottom right is labeled 'デスクトップから開始される' (Started from Desktop). The terminal output shows the WSL environment starting at the Desktop directory:

```
tada@DESKTOP-8DR8GSO: /mnt/c/Users/[redacted]/Desktop
tada@DESKTOP-8DR8GSO: /mnt/c/Users/[redacted]/Desktop$
```

コマンドプロンプトからのWSLの起動

- コマンドプロンプトからもWSLは起動可能
 - コマンドプロンプトで『wsl』と入力するだけ
 - プログラムから開いても大きな差異はない
 - コマンドプロンプトから起動するか、プログラムから起動するかは個人の好み

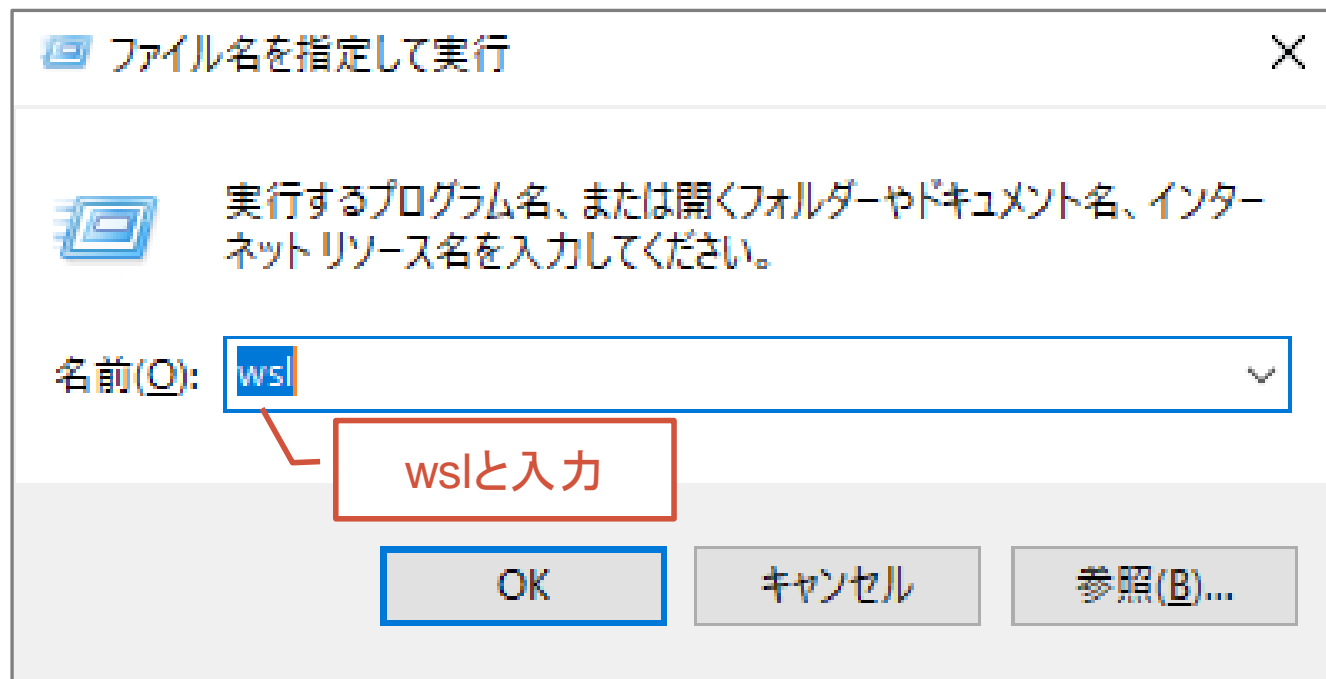
```
選択 tada@DESKTOP-8DR8GSO: /mnt/c/Users/ [redacted]  
Microsoft Windows [Version 10.0.18363.1082]  
(c) 2019 Microsoft Corporation. All rights reserved.  
C:\Users\ [redacted] > wsl  
tada@DESKTOP-8DR8GSO: /mnt/c/Users/ [redacted] $
```

wslと入力

自動的にUbuntuが立ち上がる

CUIでのWSLの起動

- マウスを使わず、キーボードのみでWSLを起動したい場合は以下の操作で起動できる
 - Windowsキー+rでファイル名を指定して実行を開く
 - 名前に『wsl』と入力し、エンターキーを押す



Ubuntuの設定

関連プログラムのインストール

- FRENDYをインストールするためにはいくつかのプログラムを事前にインストールする必要がある
 - tcsh: 実行シェル
 - gcc、make: コンパイラ
 - Boost、Lapack、Blas、Eigen: ライブラリ
- **サーバー管理者がいる場合、以降の操作は必ずサーバー管理者に確認し、サーバー管理者の監督の下で行って下さい**
- FedraやCentOSなど、他のLinuxをご利用の方はapt-getをyumなどに適宜読み替えて下さい

関連プログラムのインストール (1/3)

- インストール前にアップデートを行う
 - `sudo apt-get update`
 - `sudo apt-get upgrade`
- tcshのインストール
 - `sudo apt-get install tcsh`
 - CentOSの場合は`sudo yum install tcsh`

```
tada@DESKTOP-8DR8GSO:~$ sudo apt-get install tcsh
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  tcsh
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 427 kB of archives.
After this operation, 1363 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal/universe amd64 tcsh amd64 6.21.00-1 [427 kB]
Fetched 427 kB in 2s (259 kB/s)
Selecting previously unselected package tcsh.
(Reading database ... 31871 files and directories currently installed.)
Preparing to unpack .../tcsh_6.21.00-1_amd64.deb ...
Unpacking tcsh (6.21.00-1) ...
Setting up tcsh (6.21.00-1) ...
update-alternatives: using /bin/tcsh to provide /bin/csh (csh) in auto mode
Processing triggers for man-db (2.9.1-1) ...
tada@DESKTOP-8DR8GSO:~$
```

本資料の緑色の文字で記載した
コマンドを端末に入力すればOK

関連プログラムのインストール (2/3)

- gcc、makeのインストール
 - `sudo apt-get install build-essential`
 - インストールの有無を聞かれるので、『Y』を入力
 - インストールが完了するまで数分かかります
 - CentOSの場合は`sudo yum install gcc`
- Boostのインストール
 - `sudo apt-get install libboost-all-dev`
 - インストールの有無を聞かれるので、『Y』を入力
 - インストールが完了するまで数分～十数分かかります
 - CentOSの場合は`sudo yum install boost boost-devel`
- Lapack、Blasのインストール
 - `sudo apt-get install liblapack-dev libblas-dev`
 - インストールの有無を聞かれるので、『Y』を入力
 - CentOSの場合は`sudo yum install lapack-devel blas-devel`

関連プログラムのインストール (3/3)

- Eigenのインストール
 - この作業はランダムサンプリングツールをインストールしない場合は不要です
 - `sudo apt-get install libeigen3-dev`
 - インストールの有無を聞かれるので、『Y』を入力
- CentOSの場合はyumでインストールできないので、以下の手順を取る必要がある
 - ①以下のサイトからEigen3をダウンロード
 - http://eigen.tuxfamily.org/index.php?title=Main_Page#Download
 - ②適当なディレクトリで解凍
 - ③”eigen-3.X.X/Eigen”をfrendy_YYYYMMDDと同じディレクトリにコピーするかリンクを貼る

FRENDYのコンパイル

FRENDYの入手

- FRENDYはJAEAのHPから入手可能
 - https://rpg.jaea.go.jp/main/ja/program_frendy/

日本原子力研究開発機構 原子力基礎工学研究センター 核工学・炉工学ディビジョン

炉物理標準コード研究グループ

English 索引

炉物理標準コード研究グループ »

Table Of Contents

- プログラム等の利用について
- 核計算コード関連の解説資料
- 成果一覧
- 受賞
- スタッフ
- 連絡先

このページの目次

- FRENDY
 - 核データ処理コードFRENDY第1版のダウンロード
 - 最新版
 - 旧版
 - 引用すべき参考文献
 - ライセンス (2条項BSDライセンス準拠) 及び著作権
 - 日本語訳 (参考)
 - FRENDYの関連資料

FRENDY

核データ処理コードFRENDY第1版のダウンロード

FRENDY : FRom Evaluated Nuclear Data librarY to any application

最新版

- [README\(日本語版\)](#) (文字化けする場合はダウンロードしてください。)
- [Updatelist](#)
- [FRENDY\(1.02.003\)](#)

旧版

- [FRENDY\(1.01.001\)](#)
- [FRENDY\(1.00.005\)](#)

引用すべき参考文献

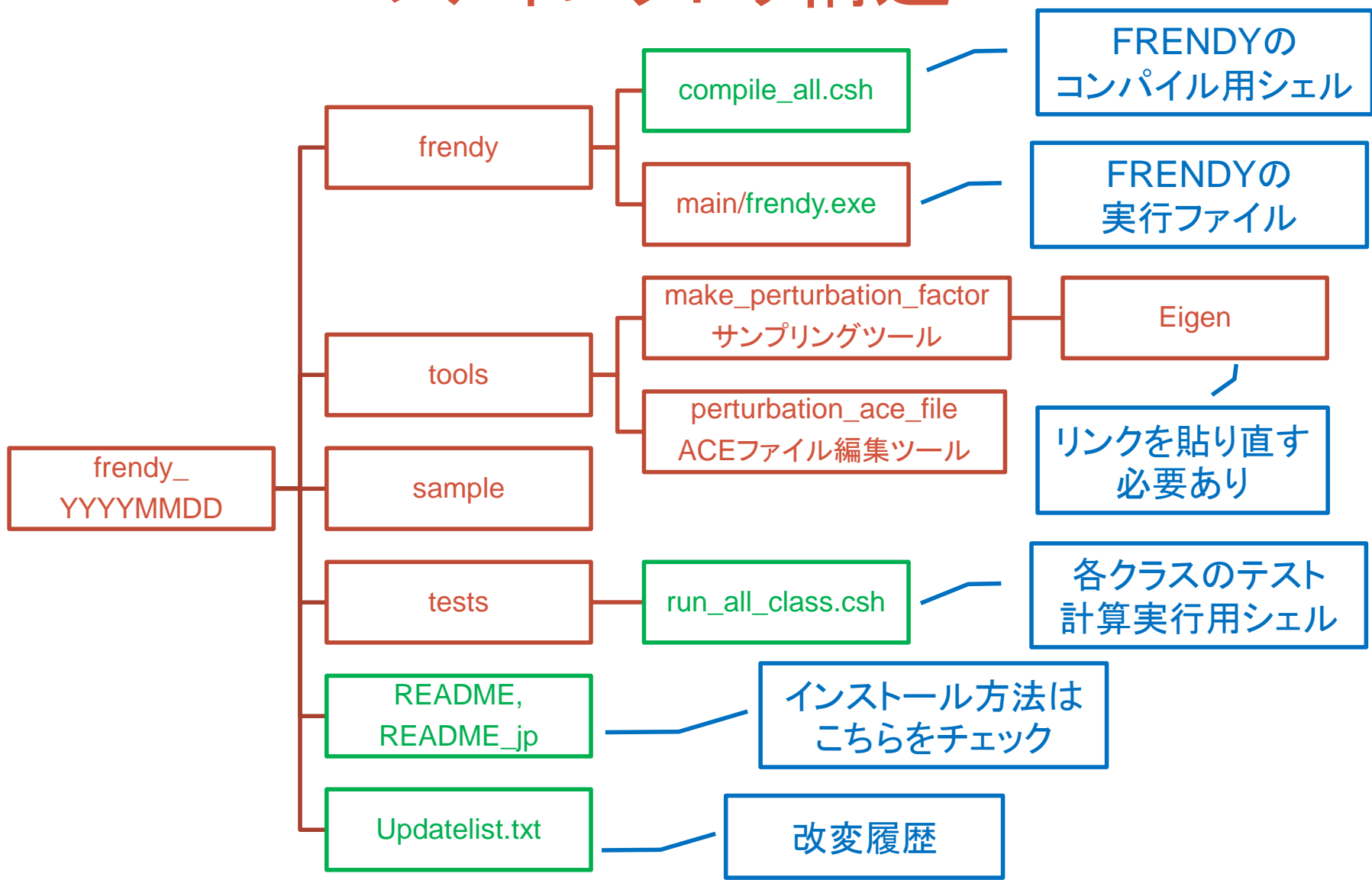
最新版はこちらから入手可能

以前のバージョンはこちらから

FRENDYの解凍

- FRENDY(frendy_YYYYMMDD.tar.gz)をダウンロードし、インストールするディレクトリにコピー
 - YYYYMMDDはリリース日
- FRENDYを解凍
 - `tar -xvzf frendy_YYYYMMDD.tar.gz`

FRENDYのディレクトリ構造



FRENDYのコンパイル

- frendy_YYYYMMDD/frendyのcompile_all.cshを実行
 - `cd frendy_YYYYMMDD/frendy`
 - `csh ./compile_all.csh`
 - コンパイルは10分程度かかります
 - 途中でWarningメッセージが出る場合あり
 - fatal errorが出なければ大きな問題はない
 - 画面上に『### Completed to make FRENDY ###』と表示され、frendy/mainに実行ファイル(frendy.exe)が作成されていればOK
- 上記のコマンドが使えない場合
 - `cd frendy_YYYYMMDD/frendy/main`
 - `make clean`
 - `make`

FRENDYのサンプルプログラムの実行

- frendy_YYYYMMDD/sample/runに移動
 - `cd sample/run`
- run_frendy.cshを実行
 - `csh ./run_frendy.csh`
- いくつかの核種、TSLを自動で処理
 - H001、C000、O016、Fe056、Xe135、Gd155、U235、U238、Pu238、Pu239
 - 軽水中のH、パラ水素、オルト水素、黒鉛(graphite)
 - 20～30分程度かかります
 - col_dataにj40a00fa、f40a00ta、xsdir.j40a00f、xsdir.j40a00tが出来ていればOK

ランダムサンプリングツールのコンパイル

- 先にFRENDYをコンパイルしておく必要がある
- frendy_YYYYMMDD/tools/make_perturbation_factorに移動
 - `cd tools/make_perturbation_factor`
- Eigenライブラリのリンクをmake_perturbation_factorの下に作成
 - Eigen3をapt-getでインストールした場合は以下の通り
 - `ln -s /usr/include/eigen3/Eigen/ ./`
 - 元々の”frendy_YYYYMMDD/tools/make_perturbation_factor/Eigen”は削除するか、上書きしてください
- 実行ファイルのコンパイル
 - `make`
- サンプルの実行
 - `cd sample`
 - `csh ./run_make_perturbation_factor.csh`
 - Warningメッセージが無く、output/1001に1001_0XXXの100個のファイルが生成されていればOK

ACEファイル編集ツールのコンパイル

- 先にFRENDYをコンパイルしておく必要がある
- frendy_YYYYMMDD/tools/perturbation_ace_fileに移動
 - `cd tools/perturbation_ace_file`
- 実行ファイルのコンパイル
 - `make`
- サンプルの実行
 - 先にランダムサンプリングツールのサンプル計算を実行しておく必要がある
 - `cd sample`
 - `csch ./run_perturbation_ace_file.csh`
 - Warningメッセージが無く、outputに0001～0100の100個のディレクトリが出来ていればOK

Appendix A

macOSでのFRENDYのコンパイル

macOSでのFRENDYのコンパイル

- macOSでもLinuxとコンパイル方法はほぼ同じ
 - ソースファイルをHPからダウンロード
 - <https://rpg.jaea.go.jp/download/frendy>
 - ダウンロードしたファイルを解凍
 - FRENDYのコンパイル
 - `cd frendy_YYYYMMDD/frendy`
 - `csh ./compile_all.csh`
 - 上記のコマンドが使えない場合
 - `cd frendy_YYYYMMDD/frendy/main`
 - `make clean`
 - `make`
- 異なるのはboostやlapackのインストール方法
 - macOSではHomebrewを使うとLinux同様に簡単にインストール可能
 - Homebrewを使ったBoost、Lapackのインストールは次ページに記載

Homebrewのインストール

- macOS上でFRENDYに必要なBoostやLapackを導入するため、Homebrewを利用する
 - Homebrew以外にはMacPortsなどもある
- Homebrewのインストールコマンドは下記の通り
 - `/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"`
 - HomebrewのHPのトップにインストールコマンドが記載されているのでそちらもご参考に
 - <https://brew.sh/>

Boost、Lapackライブラリのインストール

- Homebrewを用いてBoostとLapackをインストールする
- Boostのインストール
 - `brew install boost`
 - もしPython関係でインストールが失敗する場合は下記のようにPython supportを無効化してインストールする
 - `brew install boost --without-python`
- Lapackのインストール
 - `brew install lapack`
 - インクルードパスおよびリンクパスの環境変数を設定
 - `brew info lapack`
 - 上記のコマンドを打ち、“**For compilers to find lapack you may need to set:**”と書かれている部分を参考にシンボリックリンクを作成する。

Appendix B

VisualStudioを用いたFRENDYの コンパイル

VisualStudioを用いたFRENDYのコンパイル

- WindowsでのFRENDYのコンパイルでは、WSLを使ってFRENDYをコンパイルする方法を説明
- VisualStudioを使ってFRENDYをコンパイルすることも可能
 - ただし、かなりの手間がかかります
 - FRENDYを修正し、デバッグなどにVisualStudioを使いたい場合には本資料をご参照下さい

Boostのインストール (1/2)

- BoostのHPにアクセス
 - <https://www.boost.org/doc/>
- “Getting Started”をクリック
- ページ右下にある”Next: Getting Started on Microsoft Windows”をクリック
- Get Boostの項目にあるファイルをダウンロード
 - VisualStudioへのビルド方法が記載しているので、ダウンロードしてもページを閉じないこと!!
- ダウンロードしたファイルをboostをインストールするディレクトリに移動し、ファイルを展開

Boostのインストール (2/2)

- Boostをダウンロードしたページの”Simplified Build Form Source”を参考にBoostをインストール
 - Windowsメニュー→Visual Studio→x64 Native Tools Command Prompt for VSを選択し、コマンドプロンプトを開く
 - Boostを展開したディレクトリに移動
 - 以下のコマンドを実行
 - `bootstrap`
 - `./b2`
 - Boostのコンパイルには1時間程度の時間を要する場合もある
 - Boostを展開したディレクトリ内の`stage\lib`にライブラリが出来ていることを確認する

CLAPACKのインストール (1/3)

- CLAPACKのHPでCLAPACKのファイルをダウンロード
 - <http://icl.cs.utk.edu/lapack-for-windows/clapack/index.html#build>
- HP中のEasy Windows Buildを参考にインストール
 - CLAPACK用のCMAKEをダウンロード
 - <https://icl.utk.edu/lapack-for-windows/clapack/clapack-3.2.1-CMAKE.tgz>
 - ダウンロードしたファイルを展開
 - CMakeのHPでWindows版Cmakeをダウンロード
 - <https://cmake.org/download/>
 - Windows x64 Installerを使うと簡単にインストール可能

CLAPACKのインストール (2/3)

- インストールしたCmakeを起動
 - 以下の二つのディレクトリを指定
 - CLAPACKのソースコードを展開したディレクトリ
 - CLAPACKをビルドするディレクトリ
 - “Specify the generator for this project”でVisual Studioを選択し、“Finish”をクリック

CLAPACKのインストール (3/3)

- コンパイルが完了したら“Where to build the binaries”にCLAPACKをインストールするディレクトリを選択し、“Configure”をクリック
 - 例えばCLAPACKをビルドしたのが“D:¥clapack¥clapack-3.2.1-CMAKE”なら、“D:¥clapack¥build-3.2.1”のようにclapackをビルドしたディレクトリの近くにインストールした方が管理がしやすい
- “Generate”をクリック
- “Open Project”をクリックすると、Visual Studioのウィンドウが開く
- ソリューションエクスプローラーから“ALL_BUILD”を選択
 - ビルド→ALL_BUILDのビルドを実行
 - DebugとReleaseの両方とも実施する必要がある。

msdirent.hの入手

- msdirent.hを下記のHPからダウンロード
 - <http://svn.apache.org/repos/asf/avro/trunk/lang/c/tests/msdirent.h>
 - Visual Studioではdirent.hが使えないので、dirent.hの代替として、msdirent.hを使う
 - FRENDYのソースファイルが保存されている”frendy”の直下に”VisualStudio”ディレクトリを新たに作成し、msdirent.hを保存

FRENDYのコンパイル (1/5)

- VisualStudioを起動し、プロジェクトを作成する
 - “コードなしで続行”を選択する
- 以下の流れで『既存コードファイルからの新しいプロジェクトの作成』をウィンドウを立ち上げる
 - ファイル→新規作成→既存のコードからプロジェクトを作成
- 作成するプロジェクトの種類に“Visual C++”を選択し、“次へ”をクリック
- 下記のように『プロジェクトの場所とソースファイルの指定』を行い、“次へ”をクリック
 - 『プロジェクトファイルの場所』に、FRENDYをコンパイルするディレクトリまでの絶対パスを記入
 - 『次のフィルダーからプロジェクトにファイルを追加』にチェックを入れ、FRENDYのソースファイルが保存されているディレクトリを追加
 - 『プロジェクト名』にFrendyなど、名前を入力する
 - これが今後のプロジェクト名となる

FRENDYのコンパイル (2/5)

- 下記のように『プロジェクト設定の指定』を行い、“次へ”をクリック
 - プロジェクトのビルド方法として『VisualStudioを使用する』にチェックを入れる
 - プロジェクトの種類として『コンソール アプリケーション プロジェクト』を選択
- 『デバッグ校正の設定の指定』には何も入力せず、“次へ”をクリックし、“完了”をクリック
 - これでソースコードエディタが表示されるようになる

FRENDYのコンパイル (3/5)

- 左側のソースコードの一覧から下記のファイルを削除する
 - main.cpp、main_frendy_njoy_mode.cpp
 - 削除するには当該ファイルを右クリックし、「削除」→「クリア」or「削除」
- プロジェクト→”プロジェクト名”のプロパティで『Projectプロパティページ』のウィンドウを開く
 - 以降の作業はDebug、Releaseの双方で実施すること
 - プラットフォームがx64となっていることを確認すること
 - Win32を使う場合は、BoostやCLAPACKのインストールもWin32とする必要がある

FRENDYのコンパイル (4/5)

- インクルードディレクトリの追加
 - 左欄の構成プロパティ→C/C++→全般の”追加のインクルードディレクトリ”にboostをインストールしたディレクトリ、FRENDYのソースコードのディレクトリ、msdirent.hを保存したディレクトリ名を入力する。
 - 例: D:¥boost_1_75_0;D:¥frendy¥src;E:¥frendy¥src¥VisualStudio
- ライブラリディレクトリの追加
 - 左欄の構成プロパティ→リンカー→全般の”追加のライブラリディレクトリ”にboostとLAPACKのライブラリディレクトリを追加
 - LAPACKについては、LAPACK、BLAS、F2Cの各ディレクトリをそれぞれ追加のディレクトリとして指定する必要がある。
 - 場所が分からない場合はlibで検索して～.libというファイルがある場所を探すこと。
 - 例: D:¥boost_1_75_0¥lib; D:¥clapack¥lapack¥lib; D:¥clapack¥blas¥lib; D:¥clapack¥f2c¥lib;

FRENDYのコンパイル (5/5)

- 依存ファイルの追加
 - 左欄の構成プロパティ→リンカー→入力の”追加の依存ファイル”に以下の記載を追加
 - Releaseの場合
 - `lapack.lib;blas.lib;libf2c.lib;`
 - Debugの場合
 - `lapackd.lib;blasd.lib;libf2cd.lib;`
- 右下の”適用”をクリックし、『Project プロパティページ』のウィンドウを閉じる
- ソースエディタに戻ったらFRENDYをコンパイルする
 - ビルド→ソリューションのリビルド