

# **FreeBSD 14.3**

## [ インストール & 活用ガイド ]

2026-02-04 第11版

茂木ネットワークセンター

<https://moginwc.sakura.ne.jp/>

# はじめに

概要：

- FreeBSD（フリー・ビーエスディー）は、フリー／オープンソースの UNIX 系 OS（オペレーティング・システム）です。
- このガイドでは、① FreeBSD をパソコンにインストール、②初期設定、③日本語入力が行える状態にするまでの手順を説明します。
- 使い勝手は、サン・マイクロシステムズ（当時）の SunOS/Solaris の SunView や OpenWindows、見た目は CDE などを目指します。
- UNIX 経験者向けです（vim必須）。

# はじめに（続き）

詳細：

- パソコンの実機に対し、ストレージの中身をフォーマットして、インストールを行います。  
(VMWareでの利用は、14-5. をご覧ください)
- 英語キーボードの使用を想定しており、日本語入力の ON/OFF は、[CapsLock]キーで行えるように設定します。（※日本語キーボード設定の記載もあります）
- 有線 LAN を使用します。
- ウィンドウシステムは、fvwm3 を使用します。
- 日本語入力関連は、uim+anthy を使用します。
- 端末エミュレータは、xfce4-terminal を使用します（mltermの記載もあります）。
- シェルは、tcsh を使用します。
- ディスクは、暗号化します。よって、起動ごとにパスワードを入力する必要があります。  
(※暗号化しない設定の記載もあります)
- BIOS/UEFI で表示される CMOS クロックは、ローカルタイム（日本標準時JST）に設定されていることを前提とします。

# はじめに（続き）

## 注意・免責事項：

- · · · ·
- 当インストール&活用ガイドは、当方の趣味の知識で記載されています。誤り・勘違い等あるかもしれません、ご容赦いただけた幸いです。
  - 記載された内容を実行した結果、いかなる問題・損害が発生しても、責任は負えませんので、あらかじめご了承願います。

## 補足：

- 「appx.便利な使い方」で紹介されている内容を応用することにより、ウェブブラウザ専用PC、デジタルサイネージ専用PC、radiko専用PCなど、○○専用PCを構築することができると思います。
- また「appx.便利な使い方」では、FreeBSDの普段使いとまではいかないが、個人的にはサブマシンとして活用できそうなレベルの内容を盛り込んでみました。参考になれば幸いです。
- 当インストール&活用ガイドに掲載のソースコードや参考情報は、<https://github.com/moginwc/freebsd143>にもあります（一部）。

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

- 1. インストーラーのダウンロード
- 2. インストール
- 2. インストール (キーボード設定)
- 2. インストール (ホスト名設定)
- 2. インストール (ディストリビューション選択)
- 2. インストール (ディスク設定)
- 2. インストール (rootパスワードの設定)
- 2. インストール (ネットワーク設定)
- 2. インストール (タイムゾーン設定)
- 2. インストール (日付・時刻の設定)
- 2. インストール (一般ユーザーの追加)
- 3. 初期設定
- 3. 初期設定 (システム共通)
- 3. 初期設定 (sudo)
- 3. 初期設定 (ntpd)
- 3. 初期設定 (powerd)
- 3. 初期設定 (グラフィックドライバー)
- 3. 初期設定 (ファイヤーウォール)
- 3. 初期設定 (vimエディタ)
- 3. 初期設定 (シェルスクリプト)
- 3. 初期設定 (ウインドウ関連1~3)
- 3. 初期設定 (フォント設定)
- 3. 初期設定 (端末エミュレータ)
- 3. 初期設定 (日本語入力1~3)
- 3. 初期設定 (端末エミュレータ設定)
- 3. 初期設定 (Firefox、その他)
- 3. 初期設定 (音量キー設定)
- 4. FreeBSDの起動～終了

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

appx. 便利な使い方

### 5. システム設定

- 5-1. 自動的にログインできるようにしたい
- 5-2. ログイン後、自動的にウインドウマネージャーを起動したい
- 5-3. 特定のコマンドは、パスワードなしでsudoを実行したい
- 5-4. ログインした際のメッセージを非表示にさせたい
- 5-5. 起動時のブートメニュー や メッセージをできるだけ表示させない
- 5-6. IPアドレスを固定化したい (IPV4)
- 5-7. ウィンドウシステム終了後、自動的に電源が切れるようにしたい
- 5-8. 起動時に、/tmp フォルダーをクリアしたい
- 5-9. 無線LANアクセスポイントを使いたい

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

appx. 便利な使い方

### 5. システム設定（続き）

5-10. pkg（パッケージ）関連コマンド

5-11. IPv6で接続したい

5-12. ログインまで進めなくなった場合の対処方法（UFS）

5-13. ログインまで進めなくなった場合の対処方法（暗号化+ZFS）

5-14. システムログを画面上に表示させておきたい

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

appx. 便利な使い方

### 6. ハードウェア管理

6-1. AC動作か、バッテリー動作かを確認したい。残量を確認したい

6-2. ノートパソコンの液晶バックライトの輝度を変更したい・確認したい

6-3. マスターボリューム（音量）を確認・変更・ミュート／解除したい

6-4. CPU動作周波数の確認をしたい

6-5. ネットワークに接続したプリンタから印刷したい

汎用ドライバ（PPDファイル）を使用する例

Windows用プリンタドライバーのPPDファイルを使用する例

6-6. スピーカーやイヤホン端子から音が出るようにしたい

6-7. NumLockを効かせたい

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

### appx. 便利な使い方

#### 7. ネットワーク転送／リモートアクセス

- 7-1. リモートからログイン、ファイル転送したい
- 7-2. 公開鍵認証でSSH接続したい
- 7-3. WindowsやMacとファイル共有したい (SMB)
- 7-4. 他のパソコンからリモートデスクトップでFreeBSDに接続したい
- 7-5. FreeBSDから、Windowsにリモートデスクトップで接続したい
- 7-6. シフトJIS表示の機器類にTelnetでログインしたい
- 7-7. FreeBSDから、MacにVNC接続したい

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

### appx. 便利な使い方

#### 7. ネットワーク転送／リモートアクセス（続き）

7-8. テスト用FTPサーバーを立てたい

7-9. FTPにてFreeBSDとサーバー間のファイルを同期したい

ワイルドカードを用いてファイルをサーバーへアップロードしたい

7-10. 家庭内FTPサーバーを立てたい（SFTP編）

7-11. 家庭内Webサーバーを立てたい（nginx）

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

appx. 便利な使い方

### 8. ソフトウェアとアプリケーション

- 8-1. Firefoxで、ローカルファイルをブラウズしたい
- 8-2. ハングル文字や簡体字・繁体字、絵文字を表示させたい
- 8-3. Firefoxで、ダウンロードフォルダーを「~/Downloads」に変更したい
- 8-4. 付箋アプリを使いたい
- 8-5. Firefoxを起動し、radikoでラジオNIKKEI第1を自動的に再生したい  
ラジオNIKKEI第2に切り替えたい

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

appx. 便利な使い方

8. ソフトウェアとアプリケーション（続き）

8-6. Chromium（ウェブブラウザ）を使用したい

座標、サイズを指定して Chromium を起動したい

Chromiumを2画面横並びで起動したい

Chromiumの初回起動前にブックマークを設定しておきたい

Chromiumの初回起動前に初期設定をしておきたい

デフォルトブラウザの確認をさせたくない

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

### appx. 便利な使い方

#### 8. ソフトウェアとアプリケーション（続き）

8-7. Macのユーザー辞書をインポートしたい

8-8. 日本語を含むPostScriptファイルを、PDFファイルに変換したい

8-9. 画面スライドショーをしたい

8-10. Firefoxの初期設定を起動せずに行いたい

8-11. GIMPを使いたい

ダークモードを解除したい

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

### appx. 便利な使い方

#### 8. ソフトウェアとアプリケーション（続き）

##### 8-12. GIMPで画像編集を行いたい

縦横比を維持したままトリミングしたい

画像を拡大・縮小したい（リサイズ・スケール変更）

画像を少しクッキリさせたい

編集した画像の保存

画像に余白を加えたい

画像の一部にモザイクを入れたい

画像の一部を修復・補修したい

画像をスライドフィルム風にしたい

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

### appx. 便利な使い方

#### 8. ソフトウェアとアプリケーション（続き）

8-13. OpenSCADで通信鉄塔をモデリングしたい

8-14. サムネイル一覧から画像を選択して表示したい

表示中の画像を外部コマンドに渡して加工したい

表示中画像のExif情報を見たい

8-15. システム情報を表示したい（conky）

8-16. マイク端子から録音をしたい

録音日時付きで録音したい

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

appx. 便利な使い方

8. ソフトウェアとアプリケーション（続き）

8-17. QGIS（地理空間情報の閲覧、編集、分析）を使いたい

座標表示を変更したい

国土地理院の地図を使いたい

CSVファイルからポイント（座標）データを読み込み、表示させたい

CSVファイルからラインデータを読み込み、2点間の線を表示させたい

GPS口ガードのデータ（NMEA0183形式）を読み込みたい

GPS口ガードの特定の地点を表示させたい

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

appx. 便利な使い方

### 8. ソフトウェアとアプリケーション（続き）

#### 8-18. 閲覧専用でメールを使いたい

Macのメールアプリから、データをインポートしたい

フォントを変更したい

スレッド表示をやめたい

メールを検索したい

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

appx. 便利な使い方

8. ソフトウェアとアプリケーション（続き）

8-19. webカメラを利用したい

8-20. 外付けカメラを利用したい

外付けカメラの画像をキャプチャしたい

外付けカメラの映像を録画したい

動きを検出して映像を録画したい（動体検知）

外付けカメラの映像をそのまま見たい

外付けカメラの映像をYouTubeで配信したい

外付けカメラの映像をYouTubeで配信したい（日付時刻入り）

8-21. Anthyの辞書を強化したい

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

appx. 便利な使い方

8. ソフトウェアとアプリケーション（続き）

8-22. DOS/V環境を構築したい

LHAやFILMTNを使用できるようにしたい

8-23. ファイル管理ソフト Thunar を使いたい

使いやすくするための設定（表示回り）

使いやすくするための設定（ファイルタイプに応じた起動プログラムの指定）

使いやすくするための設定（複数ファイルの一括処理）

使いやすくするための設定（右クリックの「送る」の活用）

使いやすくするための設定（ファイルタイプ表示名の変更）

使いやすくするための設定（vimを右ダブルクリックで終了）

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

### appx. 便利な使い方

#### 8. ソフトウェアとアプリケーション（続き）

8-24. フォントファイルの中身を閲覧したい

8-25. mltermを使いたい（再コンパイル）

8-26. 軽量画像ビューア nsxiv をカスタマイズして使いたい

8-27. 画面録画したい

8-28. 端末を閉じても、作業を続けられるようにしたい（tmux）

8-29. 端末エミュレータを画面分割して使用したい（tmux）

8-30. xfce4-terminalの色を標準16色にしたい

8-31. （メモ書き） mltermとxfce4-terminalの違い

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

### appx. 便利な使い方

#### 8. ソフトウェアとアプリケーション（続き）

8-32. gnuplotでいい感じの棒グラフを作成したい

8-33. PlantUMLにて組織図を作成したい

8-34. PDFにて、ページの入れ替え、追加、削除などを行いたい

8-35. PDFにて、複数ファイルの結合や、偶数/奇数ページ毎の処理を行いたい

8-36. コマンドラインにて、PDFファイルの結合・分割処理などを行いたい

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

appx. 便利な使い方

### 9. デスクトップのカスタマイズ

9-1. デスクトップに、アプリを起動するランチャーを表示させたい

9-2. ランチャーに、システム負荷やバッテリー状況を表示させたい

9-3. ファンクションキーの押下で、アプリが起動するようにしたい

(補足：画面キャプチャソフト scrot のよく使用されるオプション)

9-4. キーボードのボリュームキーにて音量調整をしたい

9-5. Windowsのように、ALT+F4 キーで、アプリを終了させたい

9-6. タイトルバーのダブルクリックで、ウィンドウシェードになるようにしたい

9-7. タイトルバーをダブルクリックした際は、縦方向にのみ最大化したい

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

appx. 便利な使い方

9. デスクトップのカスタマイズ（続き）

9-8. 画面表示スケールを変更したい

9-9. フォントを追加したい

9-10. クリップボード関連の不具合を解決したい

9-11. 一部のフォントを無効にしたい

9-12. CTRL+ALT+DELで画面ロックしたい

9-13. GTK系アプリのデフォルトフォントを変更したい

9-14. フォントのアンチエイリアスをグレースケール方式にしたい（GTK2系）

9-15. デュアルモニター化（2画面表示）したい

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

appx. 便利な使い方

### 9. デスクトップのカスタマイズ（続き）

- 9-16. Firefoxから画像をドラッグ&ドロップしたい
- 9-17. IPA系フォントのアンチエイリアスを改善したい（GTK2系）
- 9-18. ウィンドウのクラス名を取得したい
- 9-19. 音量調整時に、画面上に音量・ミュート状態を表示したい
- 9-20. （メモ書き）ルートメニューのデフォルトフォント検証
- 9-21. （メモ書き）フォントのアンチエイリアスについて
- 9-22. （メモ書き）.fvwm2rcの解説

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

appx. 便利な使い方

10. FreeBSD上でWindowsのアプリを使いたい (Wine)

10-1. FreeBSD上でWindowsのアプリを使いたい (Wine)

インストーラー形式のWindowsアプリのインストールと起動

zip形式のWindowsアプリのインストールと起動

10-2. Wine (Ver 10) + uim-anthy,mozc でのアプリ動作状況・課題一覧

10-3. 秀丸エディターを使用したい

10-4. VB6のランタイムを必要とするソフトを起動させたい

10-5. Wineを完全消去したい

10-6. Wineで起動したWindowsアプリが暴走した場合の対処方法

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

### appx. 便利な使い方

10. FreeBSD上でWindowsのアプリを使いたい（Wine）（続き）

10-7. 拡張子が .lzh のファイルを展開したい

10-8. 文字化けしているファイルやディレクトリを削除したい

10-9. 日本語ファイル名を含むWindowsのzipファイルを展開したい

10-10. ランチャー（FvwmButtons）にWindowsアプリを追加したい

10-11. 拡張子が .cab のファイルを展開したい

10-12. Wine起動時のメッセージを抑制したい

10-13. Windowsアプリから、FreeBSD上のプログラムを起動したい

10-14. Wineで使えるいい感じの等幅フォントを生成したい

10-15. Thunarの右クリックで、Binary Editor Bzを起動したい

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

appx. 便利な使い方

11. mozcを使いたい

11-1. mozcのインストールと設定

11-2. Macのユーザー辞書をMozcにインポートしたい

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

appx. 便利な使い方

### 12. ディスクの管理

12-1. UFSファイルシステムを使用したい

12-2. ZFSスナップショットを使用したい

  ZFSスナップショットを取得したい

  現在のディスクの状況を見る

  zroot/home/pcuserのスナップショットを取得したい

  取得したスナップショットを参照したい

  スナップショットの一覧を見たい

  スナップショットの詳細を見たい

  既存のファイルとスナップショットとのファイルを比較したい

  ロールバックしたい

  スナップショットを削除したい

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

appx. 便利な使い方

### 12. ディスクの管理（続き）

スナップショット運用上の注意点

- 12-3. Windowsからもスナップショットを見たい
- 12-4. 外付けHDDをフォーマットしたい
- 12-5. 外付けHDDにバックアップを取りたい
- 12-6. ファイル・ディレクトリを指定してバックアップを取りたい
- 12-7. ディレクトリの中身を同期（ミラーリング）させたい
- 12-8. スナップショット領域をバックアップしたい
- 12-9. 外付けHDDを、完全消去したい
- 12-10. 外付けHDDをumount後、安全に取り外したい

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

appx. 便利な使い方

12. ディスクの管理（続き）

12-11. SSDやHDDの型番、シリアル番号を見たい

12-12. CrystalDiskInfoのような情報を見たい

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

appx. 便利な使い方

### 13. その他

- 13-1. .isoファイルをマウントしたい
- 13-2. 毎日23:00に、自動的にパソコンをシャットダウンしたい
- 13-3. Linux等のrpmパッケージの中身を確認したい
- 13-4. 暗号化イメージを作成し、必要な時にマウントしたい
- 13-5. デジカメなどの画像データを、撮影日（年/月/日）別に整理したい
- 13-6. Macで作成した .dmg / .sparseimage ファイルの中身を参照したい
- 13-7. Macのpkgファイルの中身を確認・展開したい
- 13-8. アプリがGTK系かQt系かを見分けたい

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

appx. 便利な使い方

13. その他（続き）

13-9. イメージファイルをUSBメモリーに書き込みたい

13-10. Windowsで使用しているUSBメモリーを読み取りたい

13-11.（メモ書き）簡単なデータベースソフトを作りたい・使いたい

13-12. 自作のmanページを作成したい

13-13. Windowsの自己解凍型.exeファイルを展開したい

13-14.（メモ書き）日本語入力時の表示と未確定文字の扱い

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

appx. 便利な使い方

### 14. その他・上級編

14-1. chroot環境を作つてGUIアプリを動作させたい

chroot環境のFirefoxを起動する

chroot環境を完全消去する

14-2. 仮想環境を使って、FreeBSD内にLinuxをインストールしたい

仮想環境でUbuntuを起動する

14-3. 過去のパッケージ群を利用したい

リポジトリを指定して、パッケージのインストールを行いたい

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

appx. 便利な使い方

14. その他・上級編（続き）

14-4. インストール済みのパッケージをバックアップしたい

すべてのパッケージを削除したい

バックアップしたパッケージから再インストールしたい

14-5. VMWareでの使用方法

14-6. Jail（軽量コンテナ）を用いたWebサーバーを構築したい

14-7. サービスをすぐに起動したい

# FreeBSD 14.3

## [ インストール & 活用ガイド ]

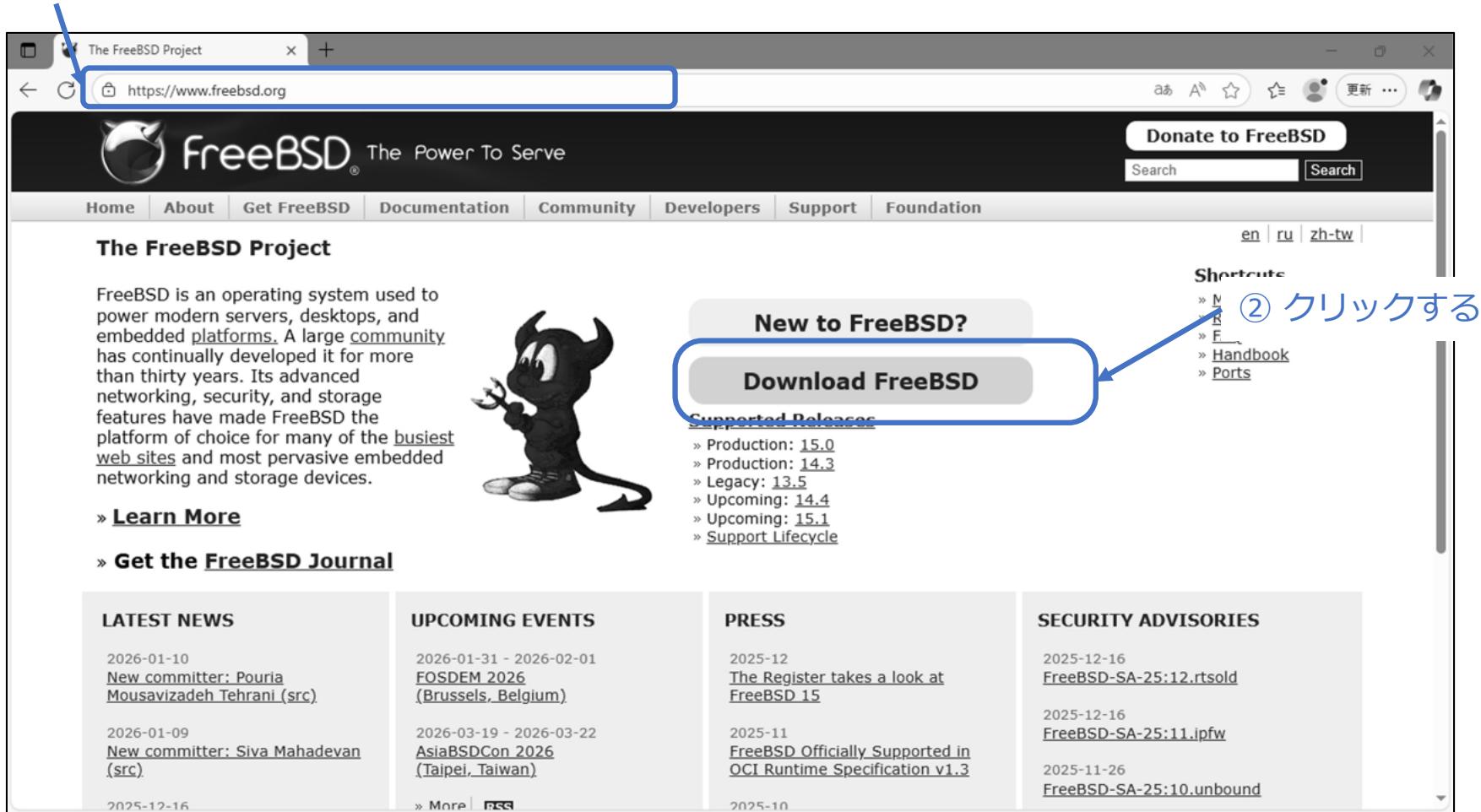
appx. 便利な使い方

### 15. プログラミング+a

- 15-1. 簡単なデータベースwebアプリを作りたい
- 15-2. 駅発車標表示専用パソコンにしたい
- 15-3. デジタルサイネージ（デジタルフォトフレーム）専用パソコンにしたい
- 15-4. ドット絵を描いて動かしたい
- 15-5. オリジナルのビットマップフォントを作りたい
- 15-6. 色付き丸ボタンのクリックでプログラムを起動できるようにしたい
- 15-7. ファイル作成をトリガに二重起動を防いで処理を実行したい
- 15-8. PDFファイルに文書番号・ページ番号などを付与したい

# 1. インストーラーのダウンロード

① <https://www.freebsd.org/> へアクセスする



# 続き | 1. インストーラーのダウンロード

① FreeBSD 14.3-RELEASE  
であることを確認する

② クリックする

FreeBSD 14.3-RELEASE

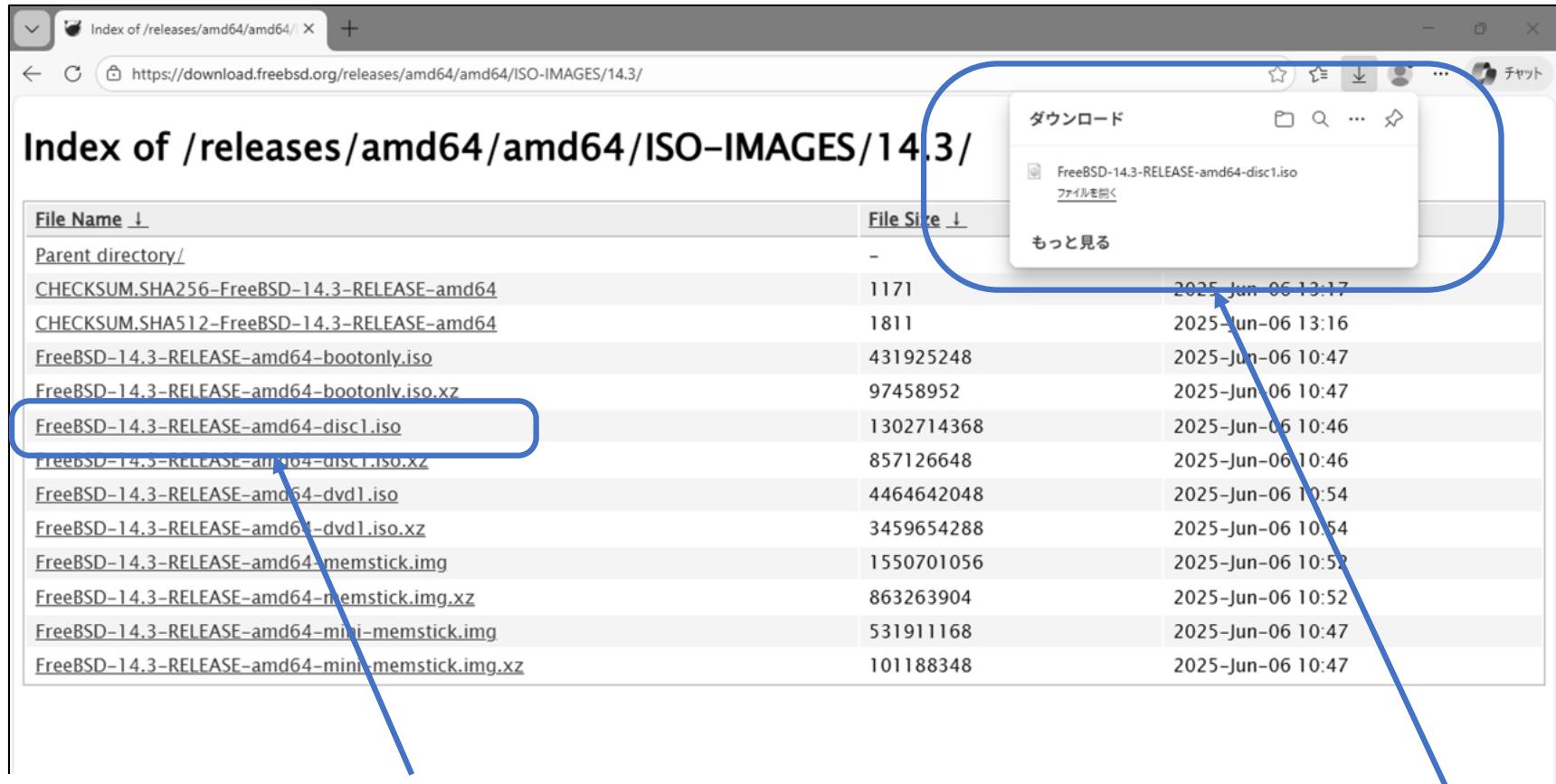
Installer	VM	SD Card	Documentation
<a href="#">amd64</a>	<a href="#">README</a> <a href="#">amd64</a> <a href="#">i386</a> <a href="#">aarch64</a> <a href="#">armv7</a> <a href="#">powerpc</a> <a href="#">powerpc64</a> <a href="#">powerpc64le</a> <a href="#">powerpcspe</a> <a href="#">riscv64</a>	<a href="#">aarch64</a> <a href="#">PINE64</a> <a href="#">PINE64-LTS</a> <a href="#">ROCK64</a> <a href="#">ROCKPRO64</a> <a href="#">RPI (3/4)</a> <a href="#">armv7</a> <a href="#">GENERICSD</a> <a href="#">riscv64</a> <a href="#">GENERICSD</a>	<a href="#">Readme</a> <a href="#">Hardware Compatibility List</a> <a href="#">Installation Instructions</a> <a href="#">Errata</a> <a href="#">Signed Checksums</a>

**Production Quality (legacy)**

Some FreeBSD RELEASEs are supported but are older or built from older STABLE branches; these are considered "legacy" and are probably only of interest to people maintaining existing systems on FreeBSD.

**FreeBSD 13.5-RELEASE**

# 続き | 1. インストーラーのダウンロード



① FreeBSD-14.3-RELEASE-amd64-disc1.isoクリックする

② ダウンロードが終わると、このように表示されることを確認する

# 続き | 1. インストーラーのダウンロード

---

① ダウンロードした FreeBSD-14.3-RELEASE-amd64-disc1.iso を DVD に書き込む。

Rufesなどを使って、USBメモリーに書き込むことも可。

## 2. インストール

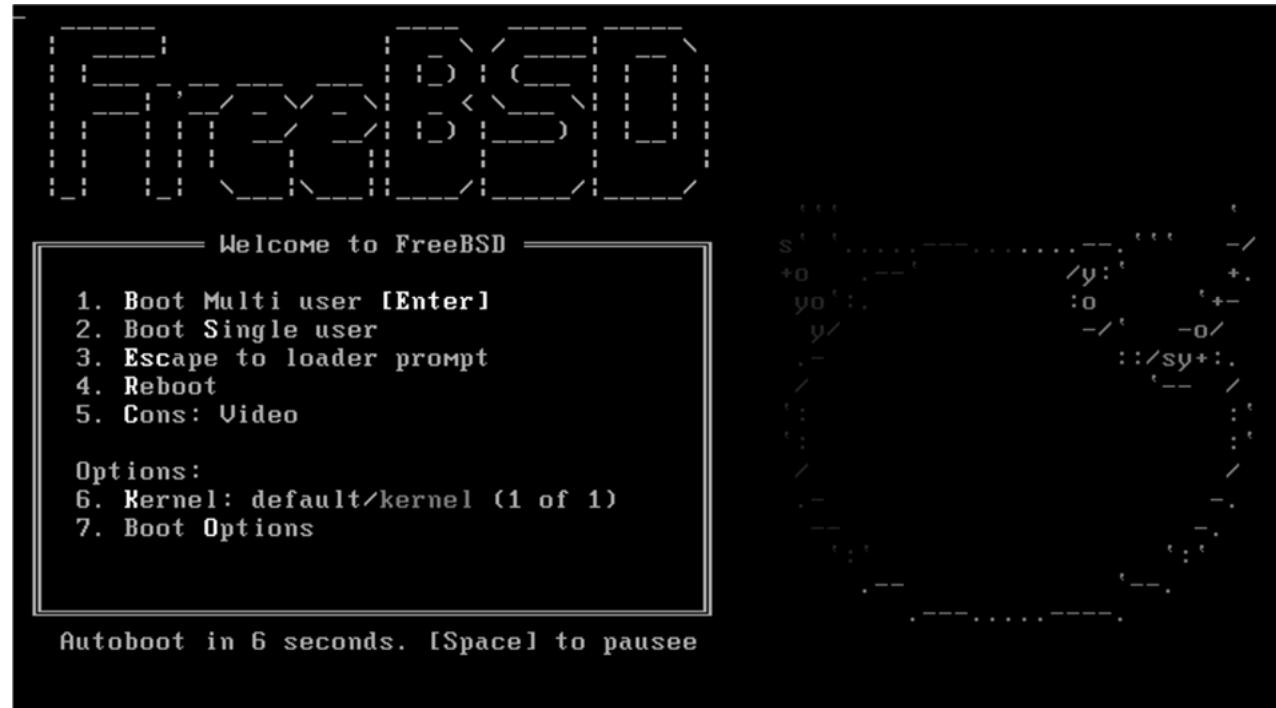
---

① FreeBSDをインストールするパソコンで、DVDを使って起動する。

USBメモリーに書き込んだ場合は、USBメモリーから起動する。

## 続き | 2. インストール

① Enterキーを押す。



このメニューはお使いのパソコンの起動方法によって若干異なります（以降同じ）。

## 続き | 2. インストール

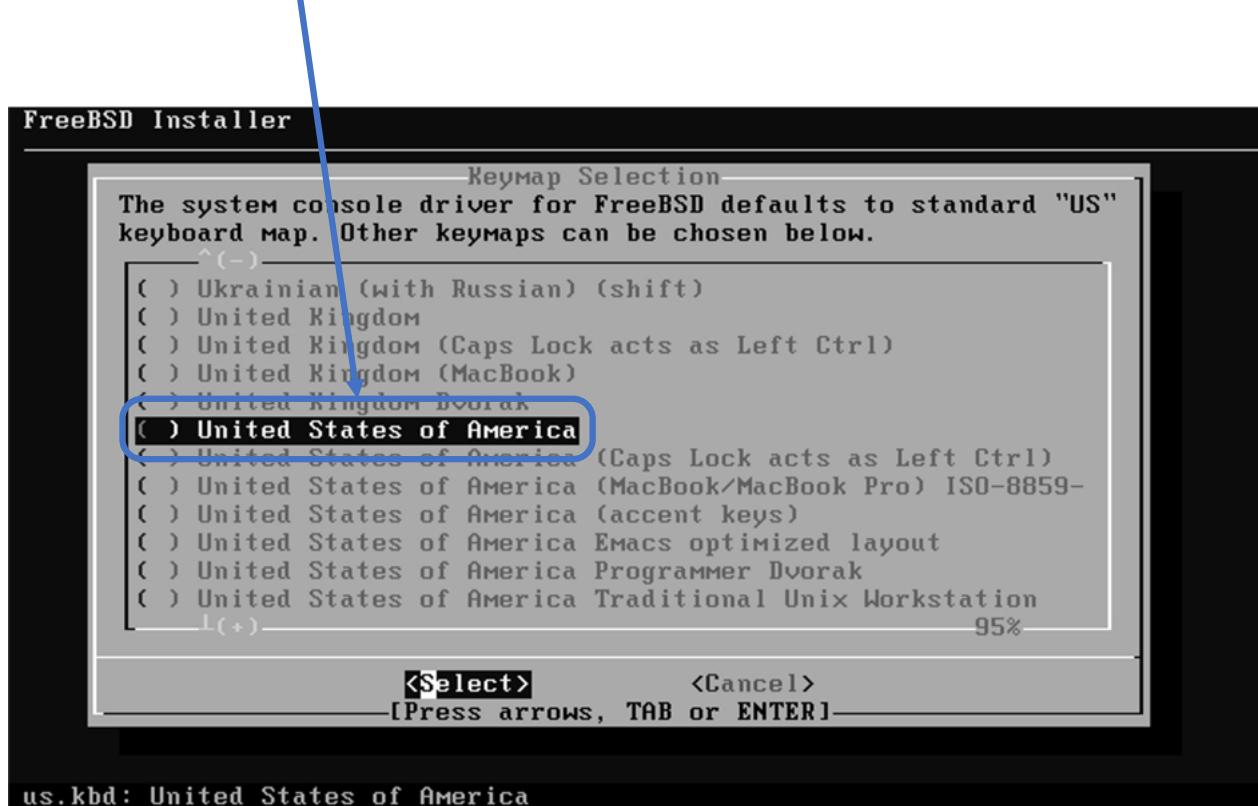
---

① Enterキーを押す。



## 2. インストール（キーボード設定）

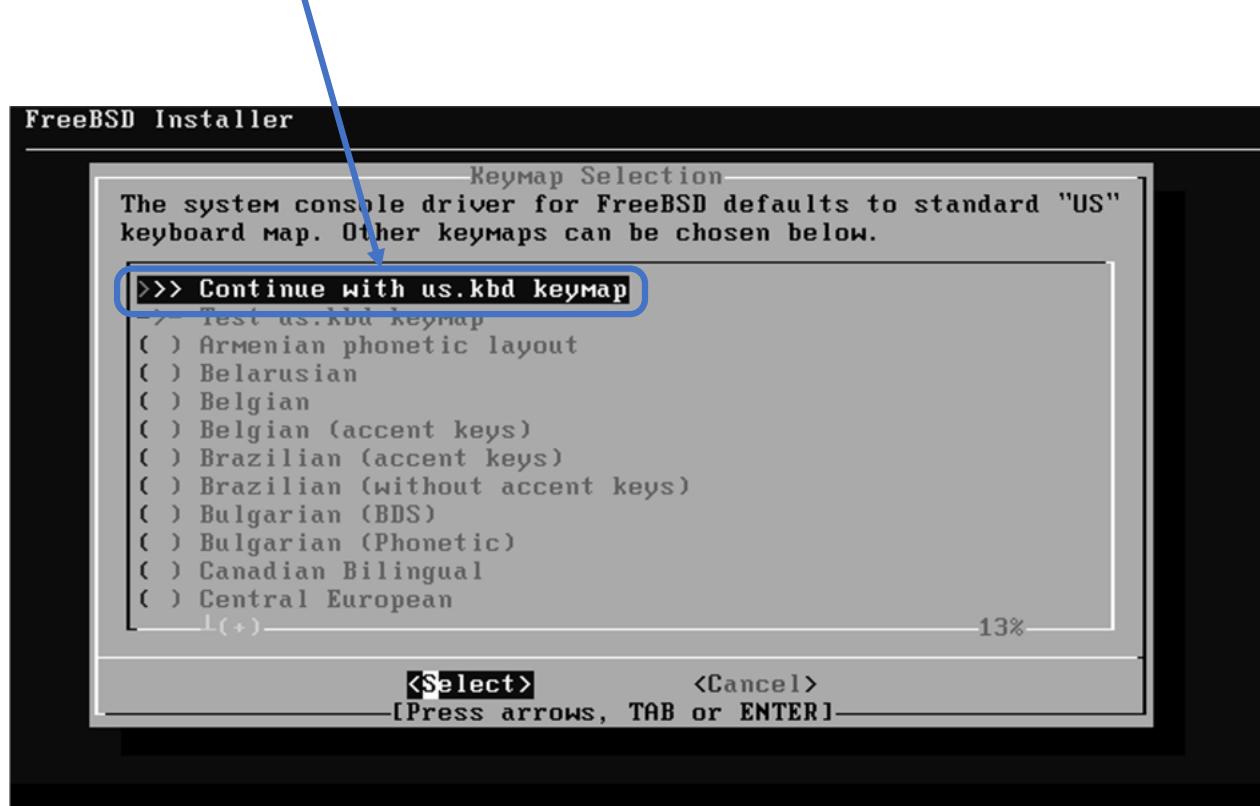
- ① ↑↓キーで、United States of Americaを選択し、Enterキーを押す。



※ 日本語キーボードの場合は、Japanese 106 を選択し、Enterキーを押す。

## 続き | 2. インストール（キーボード設定）

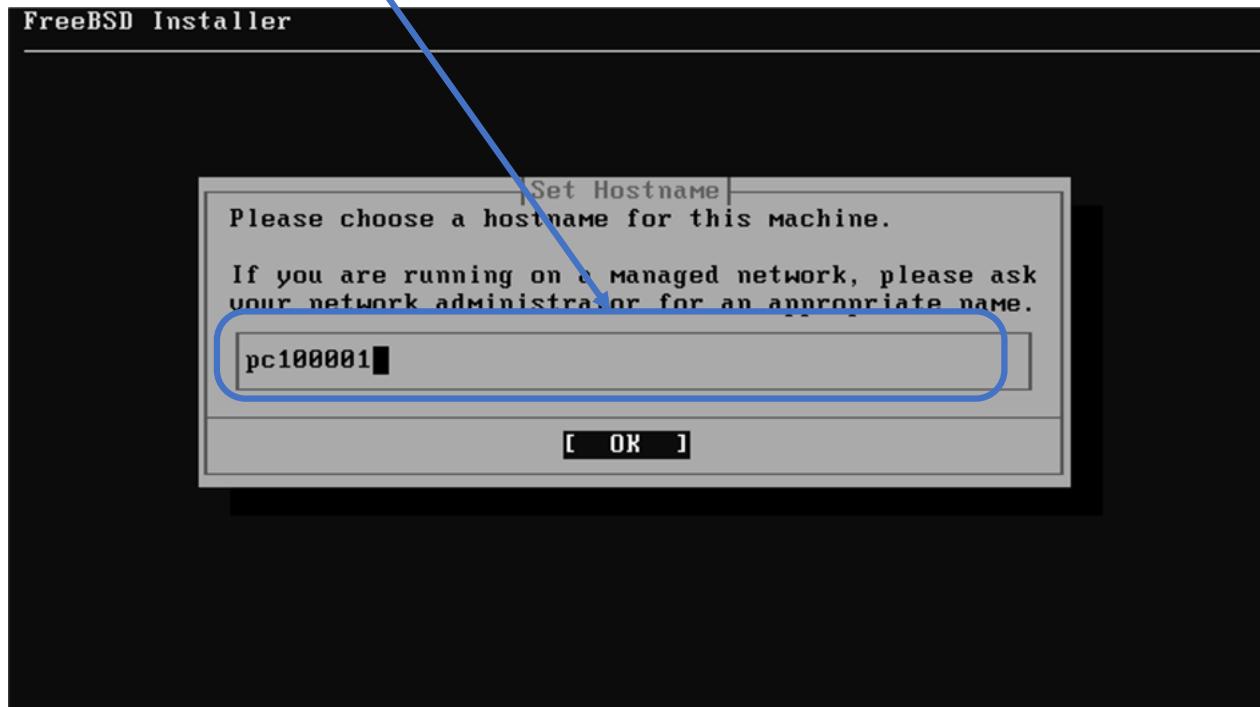
① ↑キーで、Continue with us.kbd keymapを選択し、Enterキーを押す。



※ 日本語キーボードの場合は、↑キーで、Continue with jp.kbd keymapを選択し、Enterキーを押す。

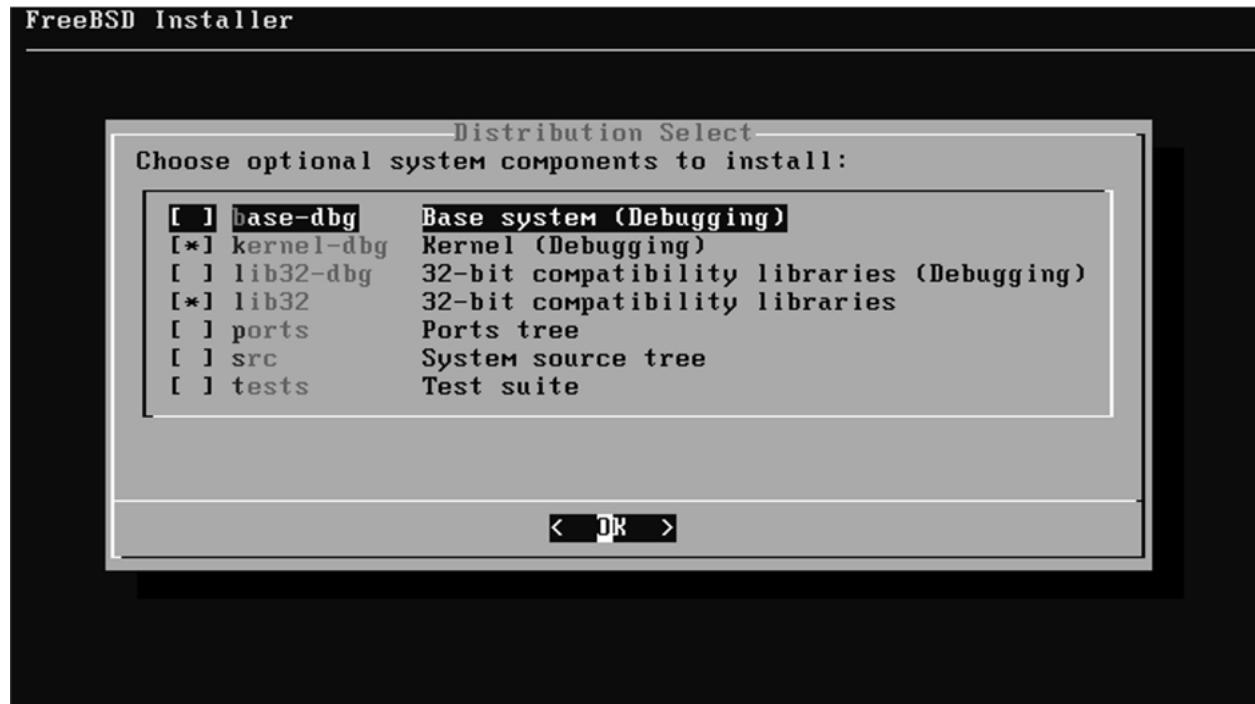
## 2. インストール（ホスト名設定）

① ここでは、ホスト名として pc100001 を入力し、Enterキーを押す。



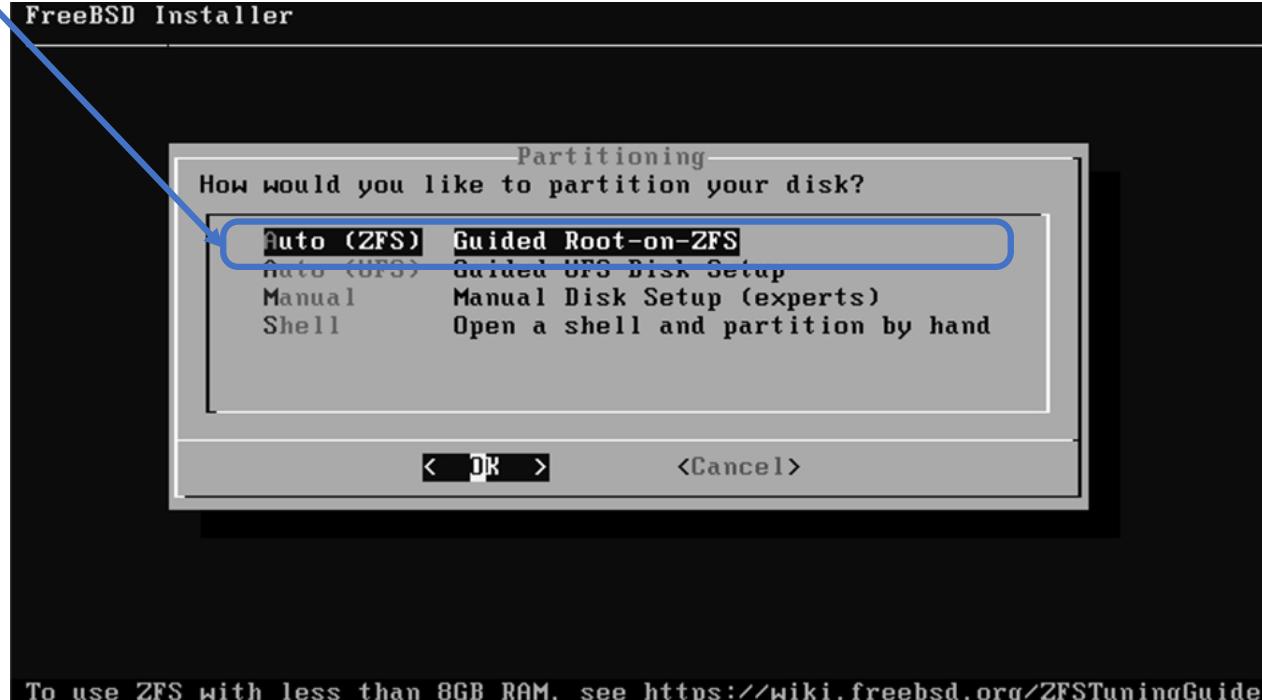
## 2. インストール（ディストリビューション選択）

① Enterキーを押す。



## 2. インストール（ディスク設定）

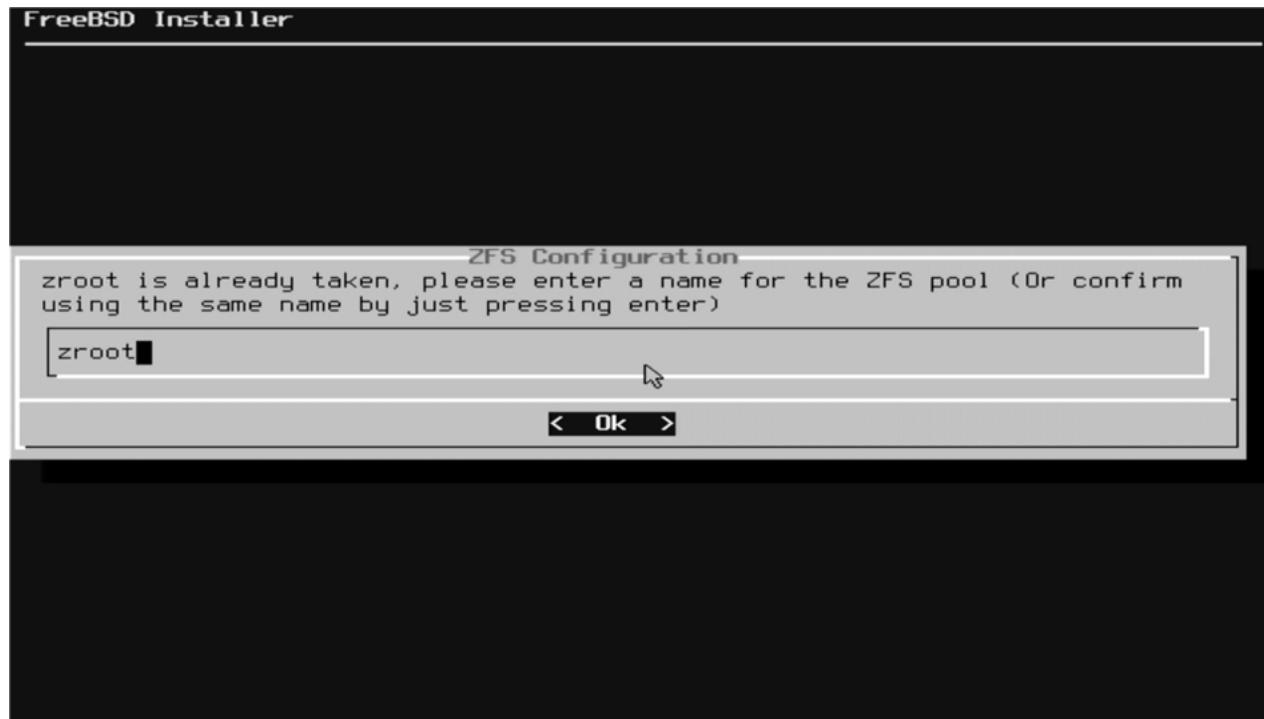
① Auto (ZFS) を選択し、Enterキーを押す。



UFSを使いたい場合は、「12-1. 従来通り、UFSファイルシステムを使用したい」を参照。

## 続き | 2. インストール (ディスク設定)

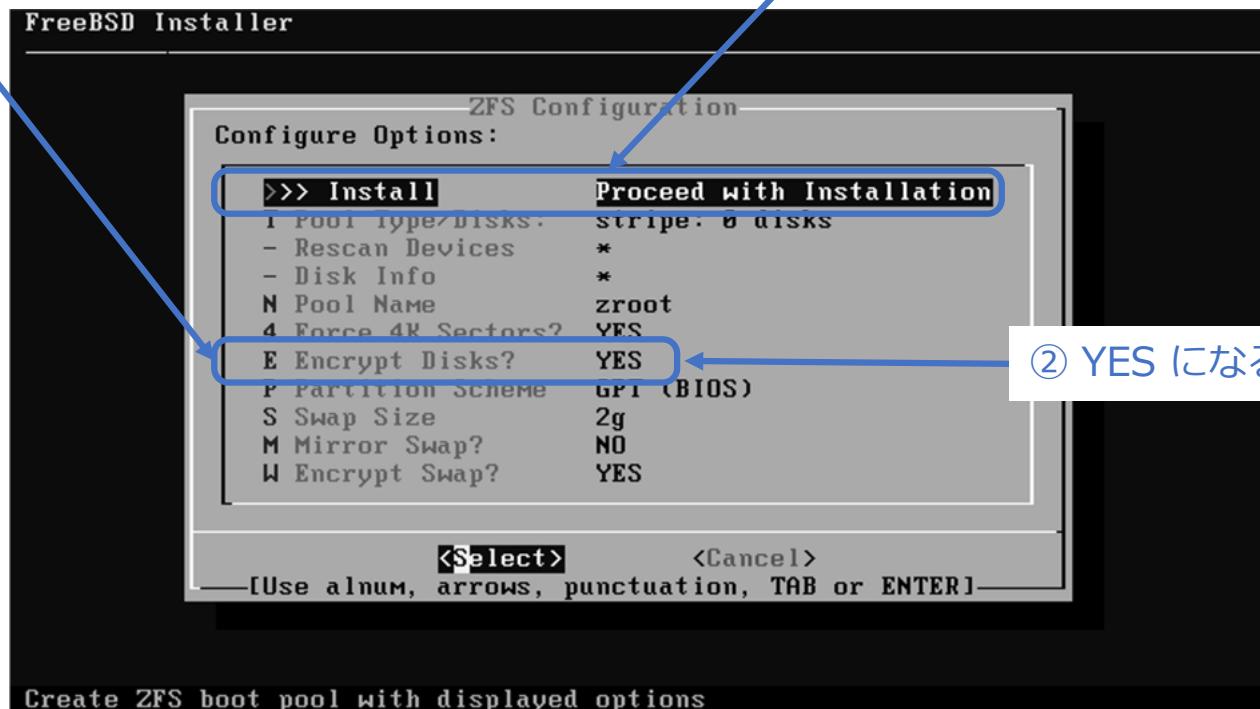
- ① もし、下記の画面が表示された場合は、そのままEnterキーを押す。



## 続き | 2. インストール（ディスク設定）

① Encrypt Disks? を選択し、スペースキーを押す。

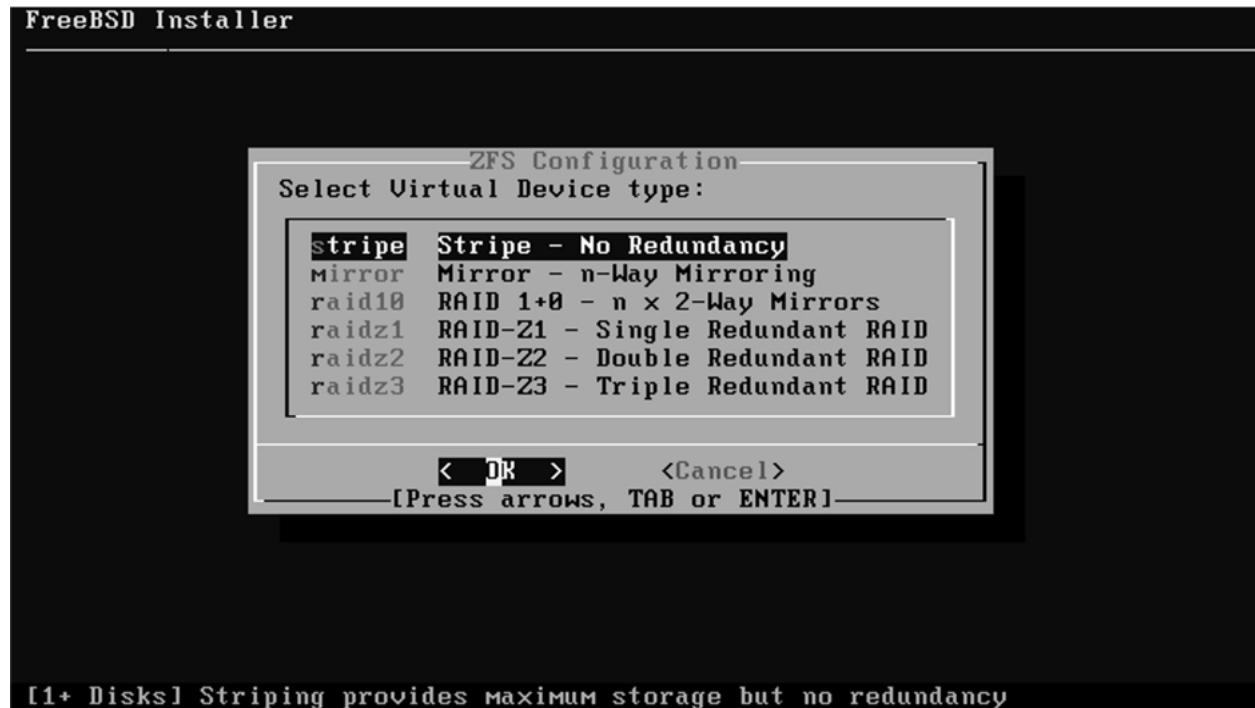
③ Install を選択し、Enter キーを押す。



このEncrypt Diskの暗号化は、ZFSの暗号化ではなく、FreeBSD標準のGELI暗号化である。

## 続き | 2. インストール（ディスク設定）

① Enter キーを押す。

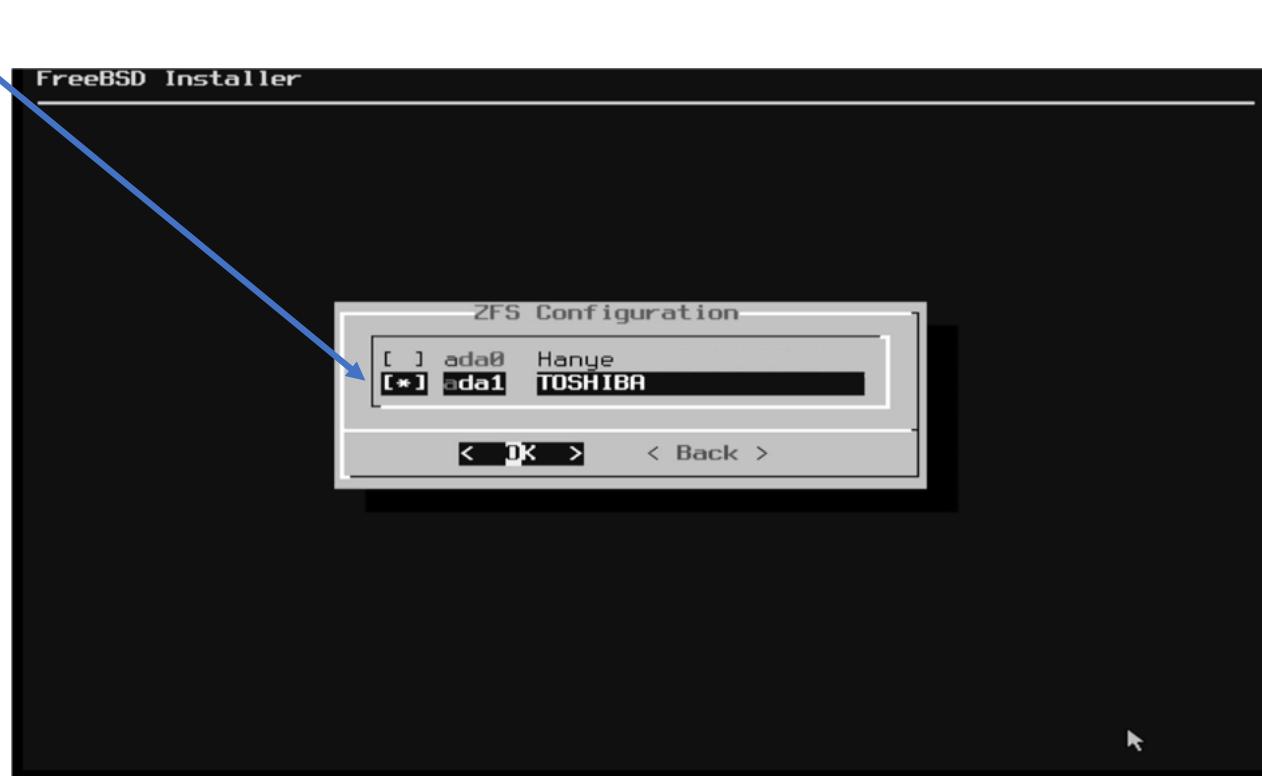


## 続き | 2. インストール（ディスク設定）

① FreeBSDをインストールするディスクを↑↓で選択し、スペースキー を押す。

※間違えないように！

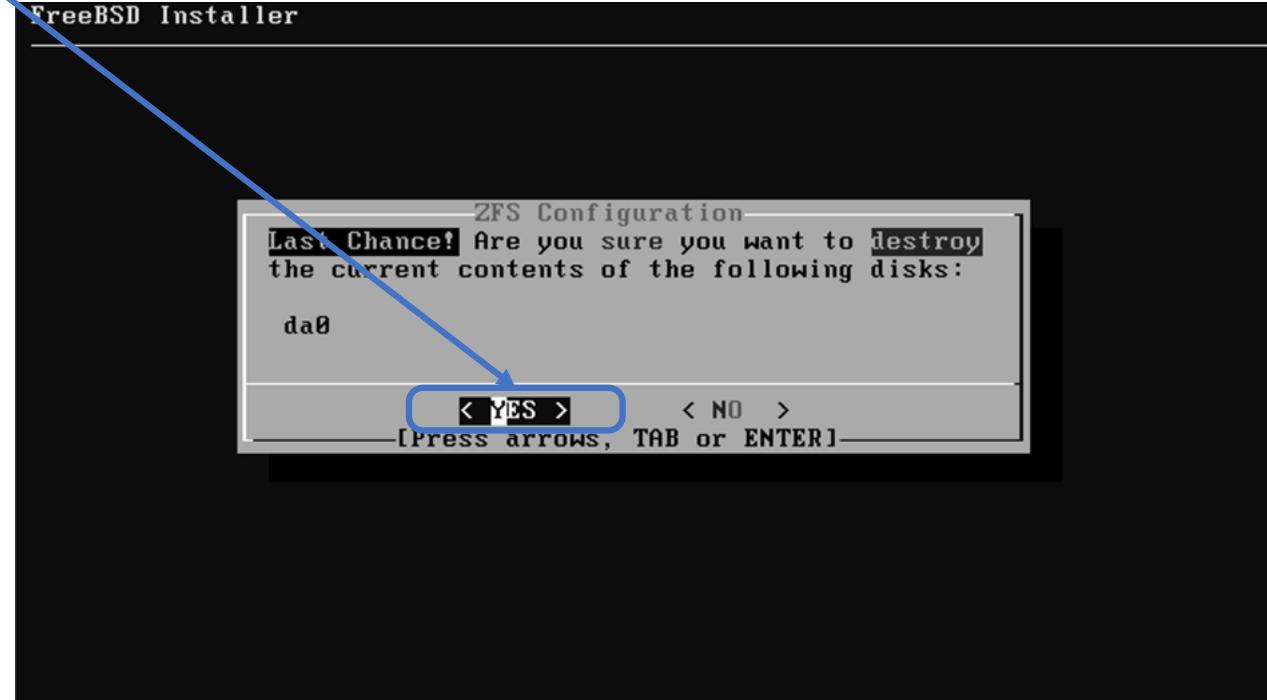
② \*印がつくことを確認する。



③ Enter キーを押す。

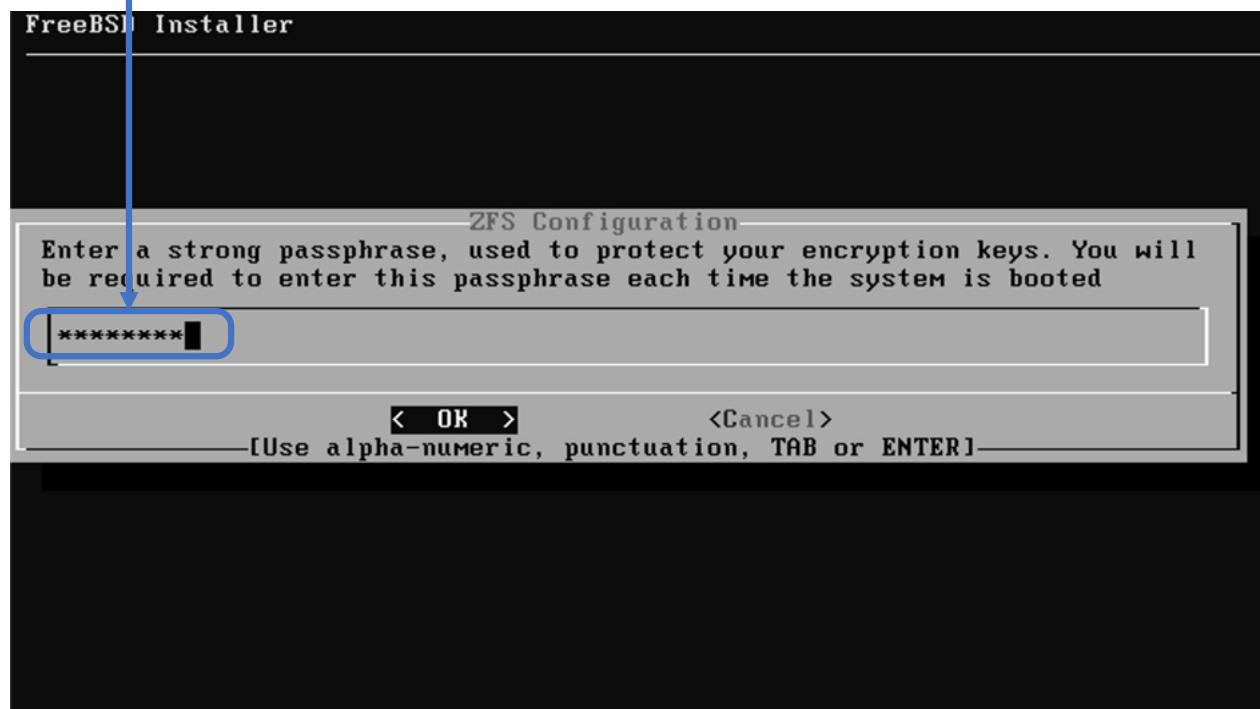
## 続き | 2. インストール（ディスク設定）

① Yes を選択し、Enter キーを押す。



## 続き | 2. インストール（ディスク設定）

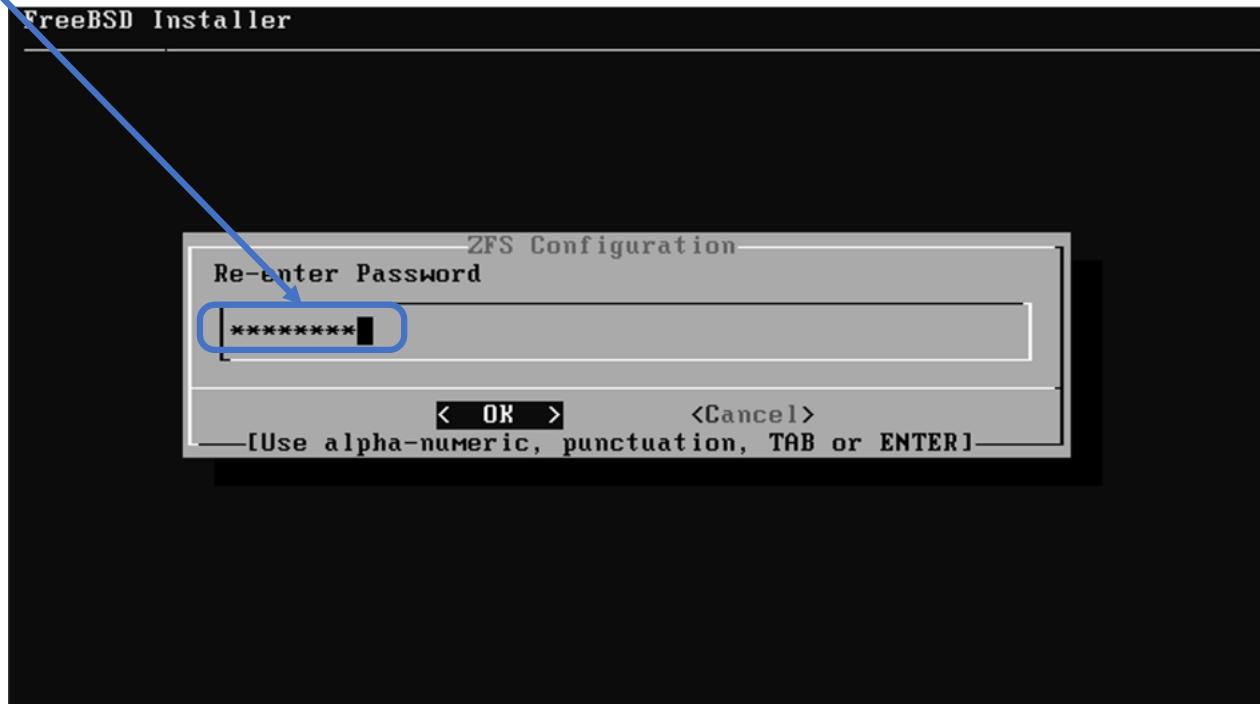
① ディスク暗号化のパスワードを入力し、Enterキーを押す。



※ちなみにこの暗号化設定は、zfsとは無関係で、GELIという暗号化を使用している。

## 続き | 2. インストール（ディスク設定）

① 再度パスワードを入力し、Enterキーを押す。

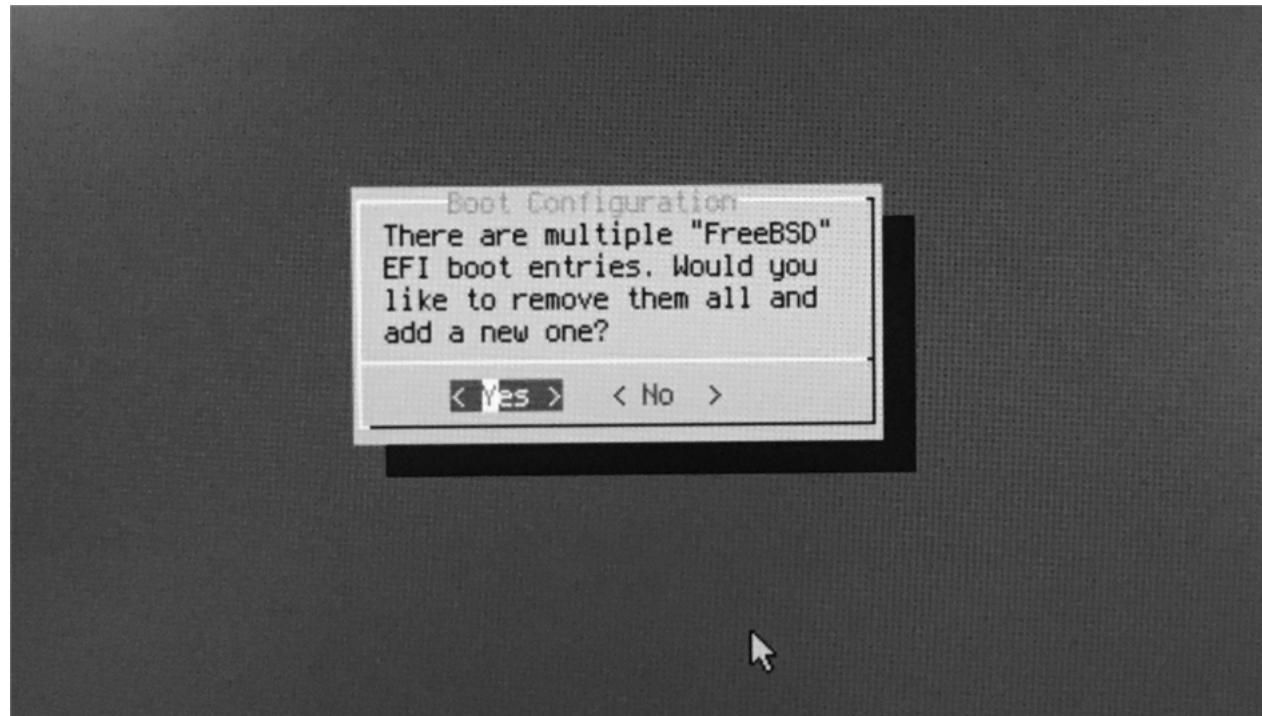


② しばらく待つ。

## 続き | 2. インストール（ディスク設定）

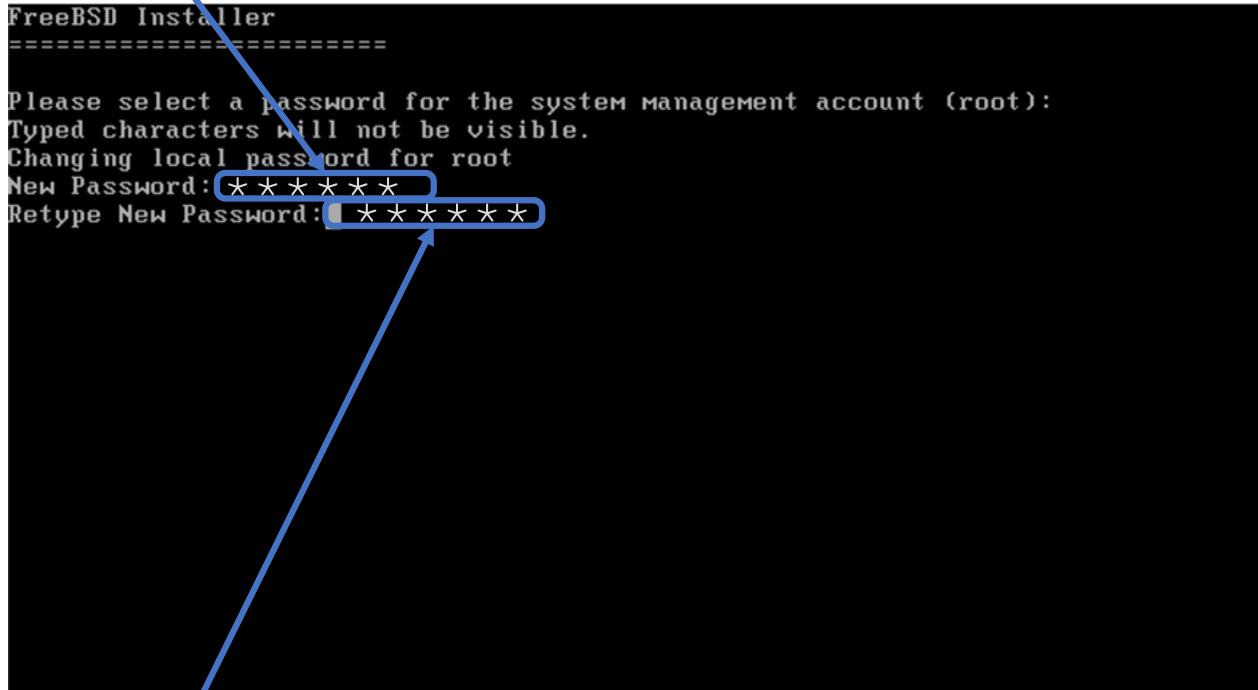
---

- ① もし、下記の画面が表示された場合は、そのままEnterキーを押す。



## 2. インストール (rootパスワードの設定)

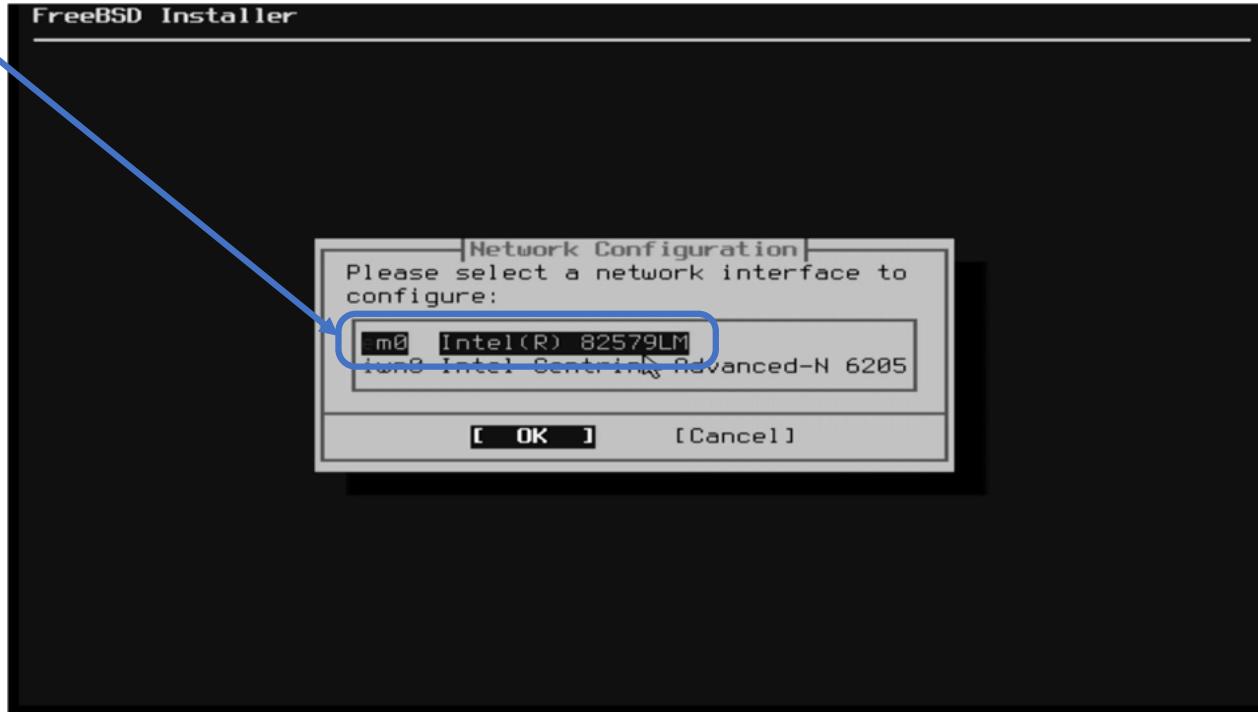
① rootのパスワードを入力して、Enterキーを押す。



② 再度、rootのパスワードを入力し、Enterキーを押す。

## 2. インストール（ネットワーク設定）

- ① 有線LANのインターフェースを選択し、Enterキーを押す。



本インストール&活用ガイドでは、このインターフェース名が「em0」であることを前提に記載していますので、適宜読み替えてください。

## 続き | 2. インストール（ネットワーク設定）

---

① Enterキーを押す。



## 続き | 2. インストール（ネットワーク設定）

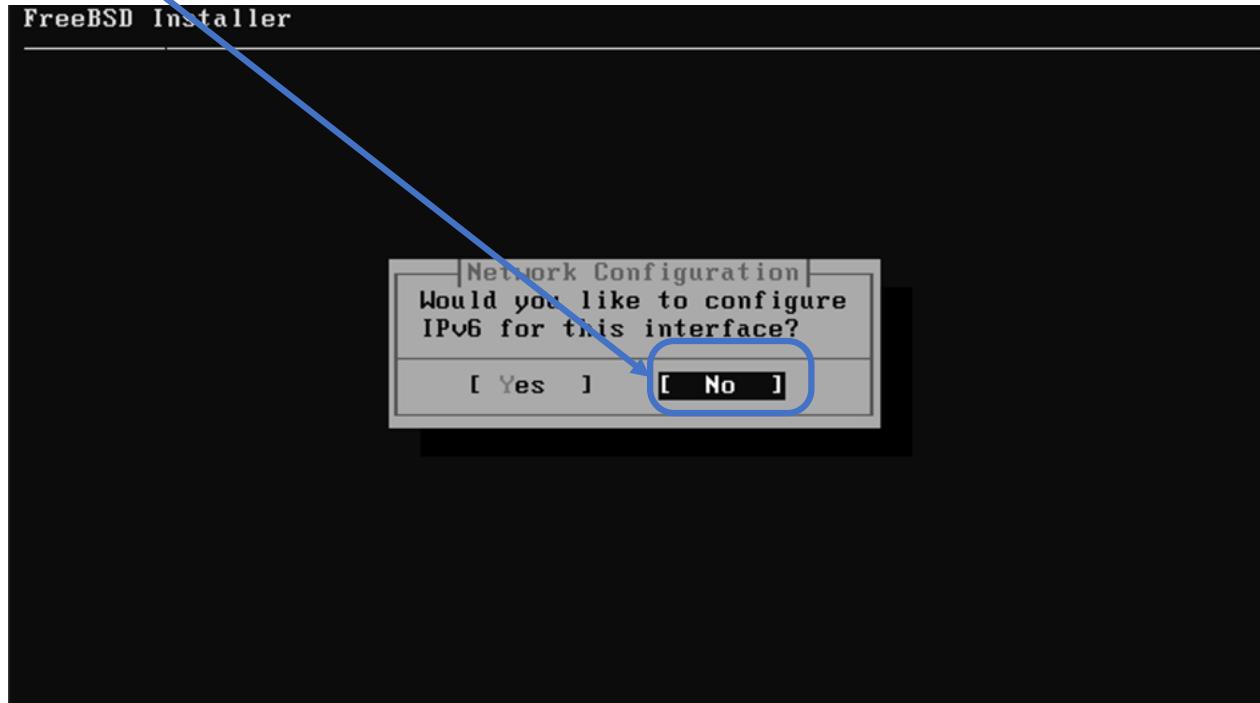
① Enterキーを押す。



② 少し待つ。

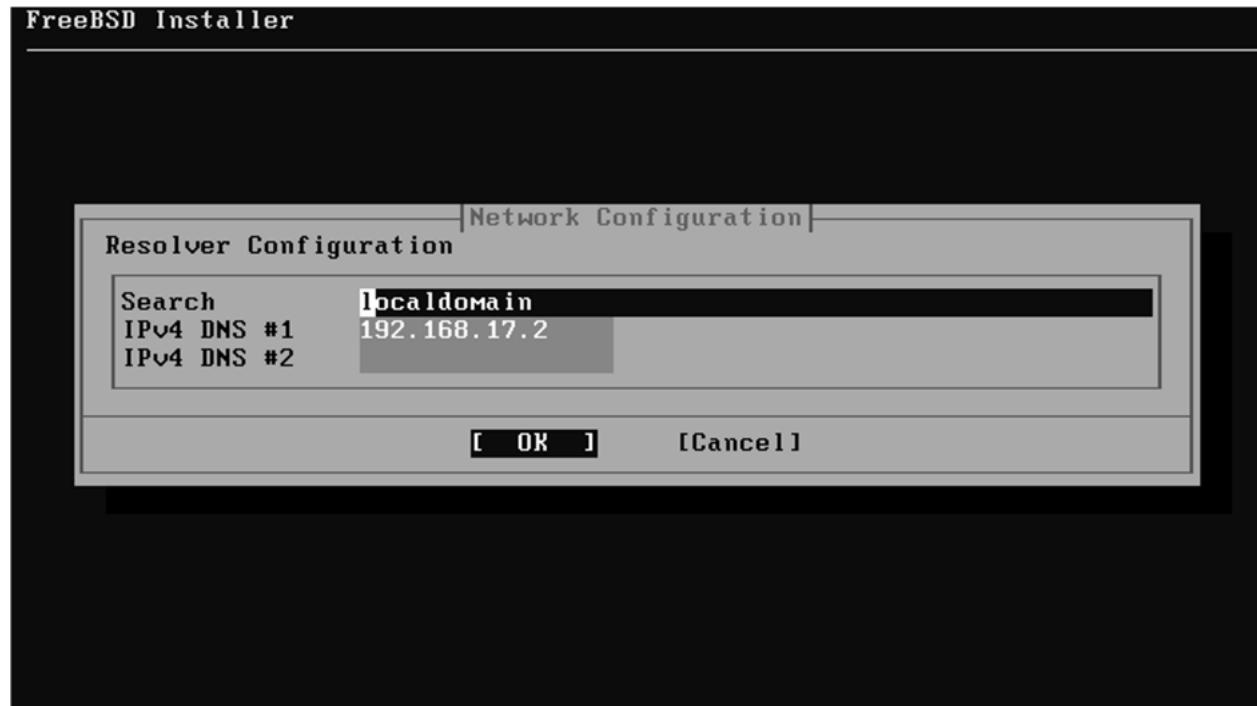
## 続き | 2. インストール（ネットワーク設定）

① [No] を選択し、Enterキーを押す。



## 続き | 2. インストール（ネットワーク設定）

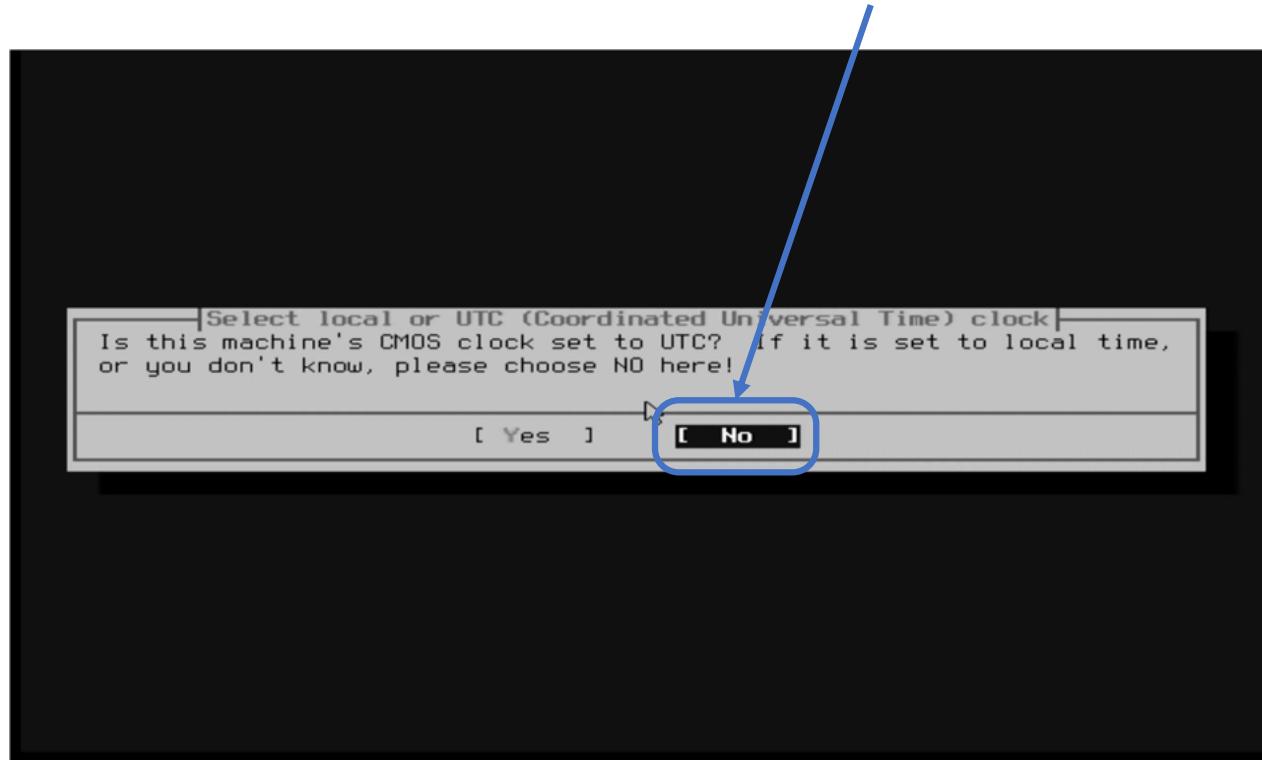
① Enterキーを押す。



※「localdomain」と表示されない場合もあるが、そのままEnterキーを押す。

## 2. インストール（タイムゾーン設定）

- ① もし、パソコンのCMOSクロックに関する画面が表示された場合はNoを選択し、Enterキーを押す。



※この画面は、実機にインストールする場合に表示され、VMware Workstation Player などにインストールする場合は表示されない傾向です。

※FreeBSDは、パソコンに保持されているCMOSクロック（BIOS/UEFIで表示される時刻）が協定世界時（UTC）であることを前提としているが、ここで<No>を設定することにより、ローカルタイムであることをOSに通知している（/etc/wall\_cmos\_clock が作成される）。

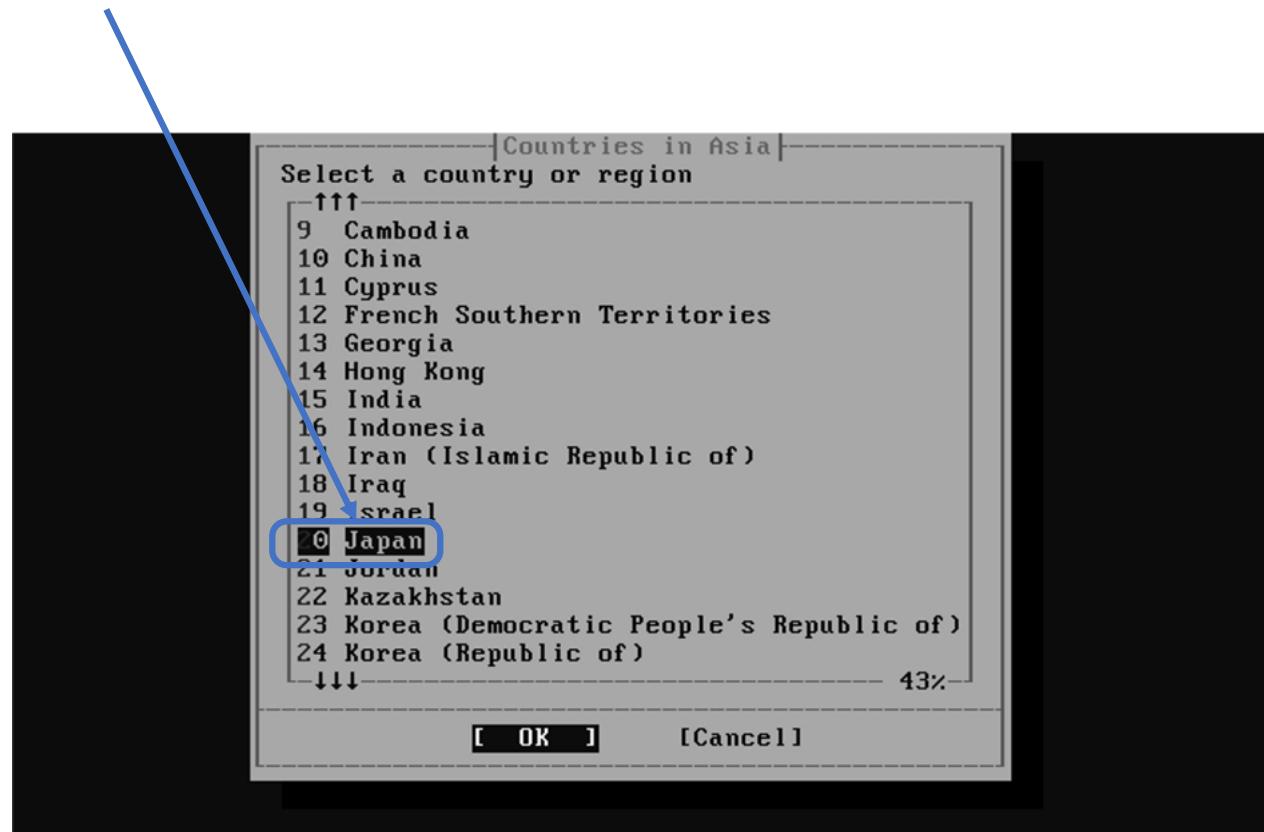
## 続き | 2. インストール（タイムゾーン設定）

① ↑↓キーで、Asiaを選択し、Enterキーを押す。



## 続き | 2. インストール（タイムゾーン設定）

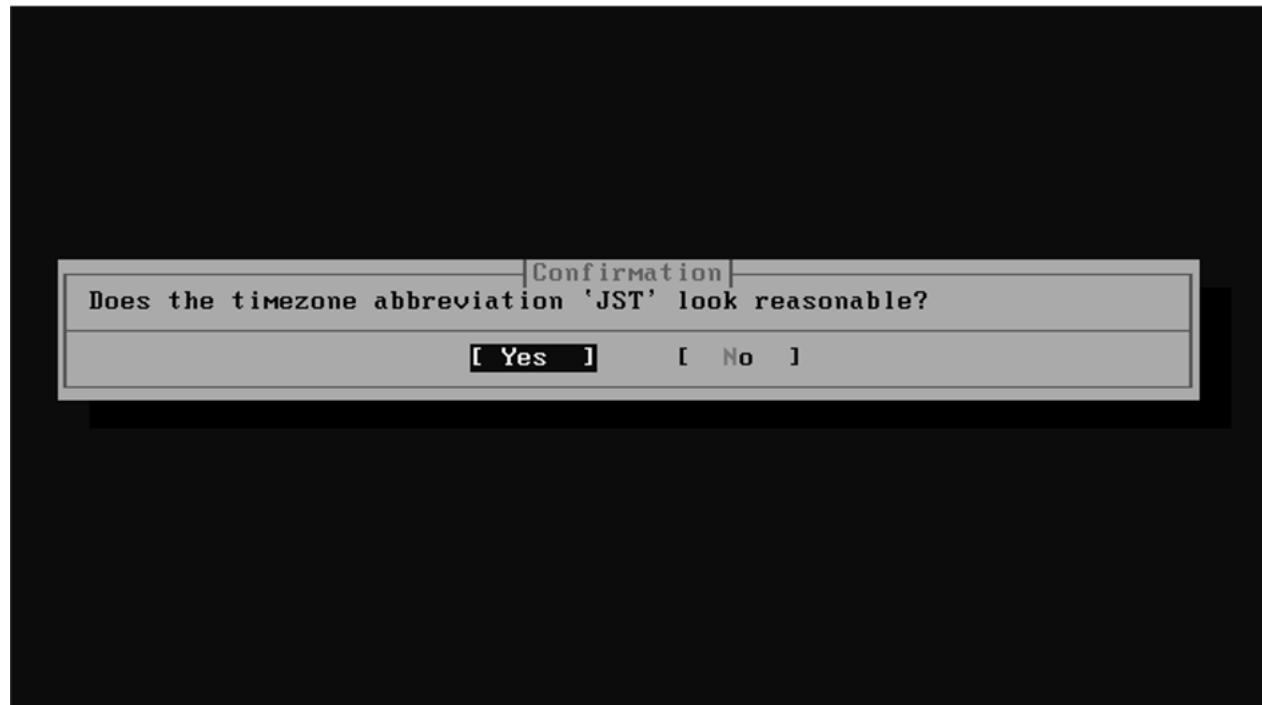
① ↑↓キーで、Japanを選択し、Enterキーを押す。



## 続き | 2. インストール（タイムゾーン設定）

---

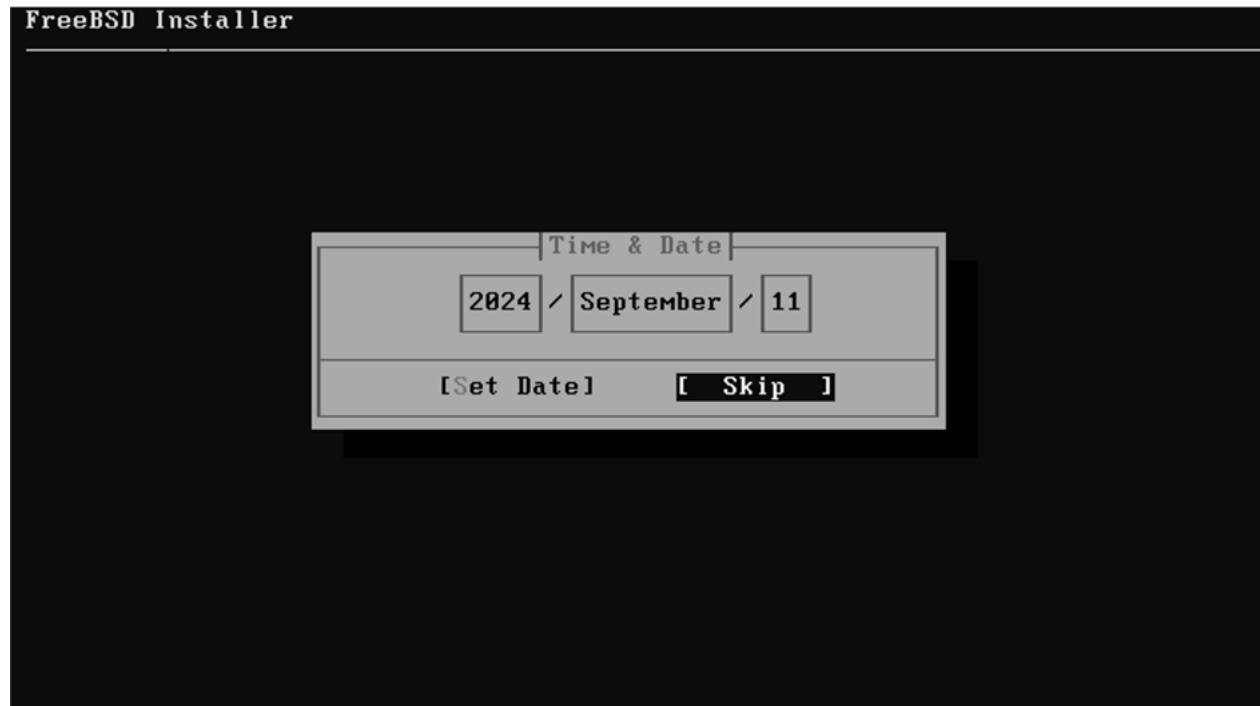
① Enterキーを押す。



## 2. インストール (日付・時刻の設定)

---

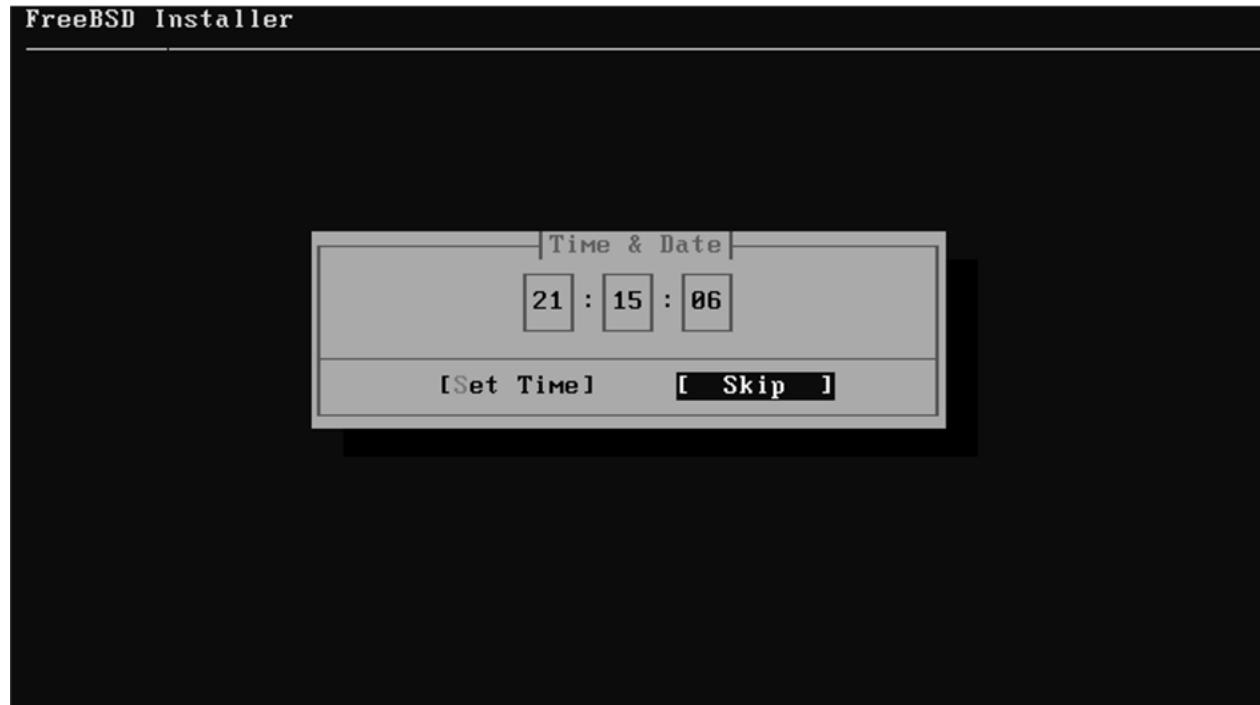
① Enterキーを押す。



## 続き | 2. インストール（日付・時刻の設定）

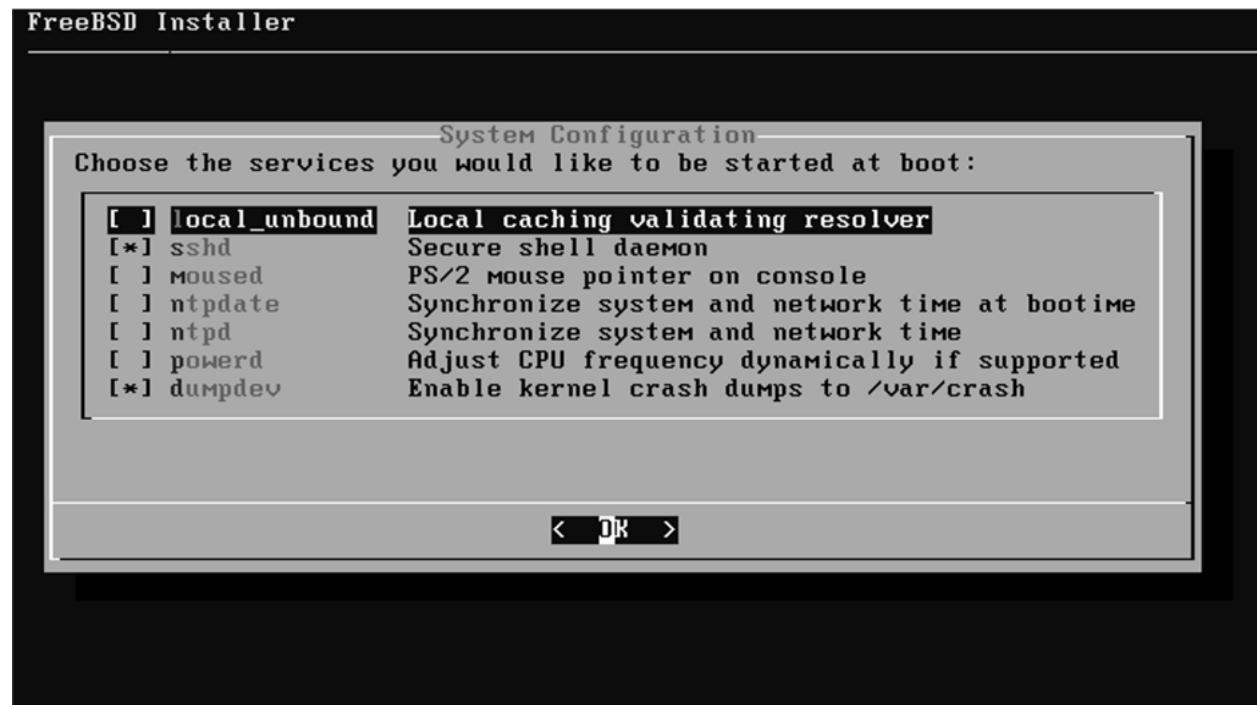
---

- ① Enterキーを押す。（時刻が異なっていても必ずSkipする）



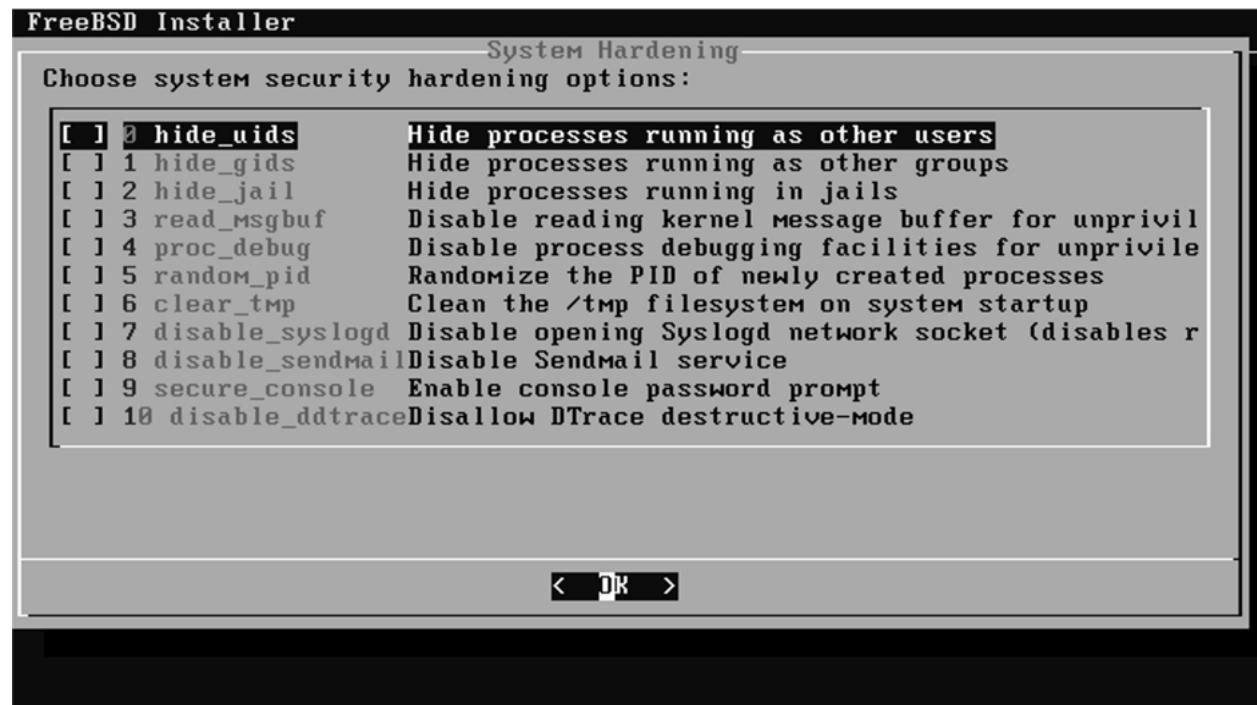
## 続き | 2. インストール

① Enterキーを押す。



## 続き | 2. インストール

① Enterキーを押す。



## 続き | 2. インストール

- ① もし、FreeBSD Firmware Installationの画面が表示された場合は、OKを選択し、Enterキーを押す。



※ カウントダウンのような画面が表示された場合は、そのまま待つ。

## 2. インストール（一般ユーザの追加）

---

① Enterキーを押す。



## 続き | 2. インストール (一般ユーザの追加)

```
FreeBSD Installer
=====
Add Users

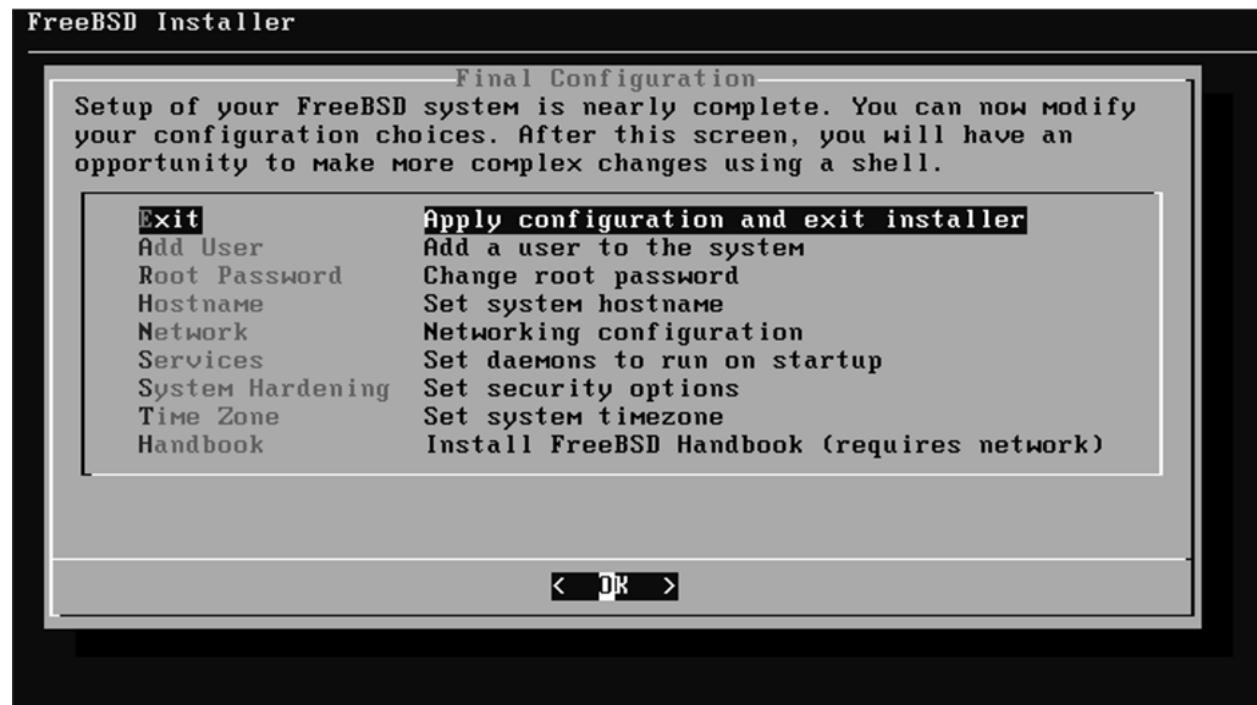
Username: pcuser          ① ここでは pcuser と入力し、Enterキーを押す。
Full name:                ② Enterキーを押す。
Uid (Leave empty for default):    ③ Enterキーを押す。
Login group [pcuser]: wheel      ④ wheel と入力し、キーを押す。
Login group is wheel. Invite pcuser into other groups? []: ⑤ Enterキーを押す。
Login class [default]:          ⑥ Enterキーを押す。
Shell (sh csh tcsh nologin) [sh]: tcsh ⑦ tcsh と入力し、キーを押す。
Home directory [/home/pcuser]:   ⑧ Enterキーを押す。
Home directory permissions (Leave empty for default): ⑨ Enterキーを押す。
Enable ZFS encryption? (yes/no) [no]: ⑩ Enterキーを押す。
Use password-based authentication? (yes): ⑪ Enterキーを押す。
Use an empty password? (yes/no) [no]: ⑫ Enterキーを押す。
Use a random password? (yes/no) [no]: ⑬ Enterキーを押す。
Enter password: *****        ⑭ パスワードを入力し、Enterキーを押す。
Enter password again: *****  ⑮ 再度パスワードを入力し、Enterキーを押す。
Lock out the account after creation? [no]: ⑯ Enterキーを押す。
```

## 続き | 2. インストール（一般ユーザの追加）

```
Username: pcuser
Full name:
Uid (Leave empty for default):
Login group [pcuser]: wheel
Login group is wheel. Invite pcuser into other groups? []:
Login class [default]:
Shell (sh csh tcsh nologin) [sh]: tcsh
Home directory [/home/pcuser]:
Home directory permissions (Leave empty for default):
Enable ZFS encryption? (yes/no) [no]:
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no] :
Username      : pcuser
Password      : *****
Full Name     :
Uid           : 1001
ZFS dataset   : zroot/home/pcuser
Class          :
Groups         : wheel
Home           : /home/pcuser
Home Mode     :
Shell          : /bin/tcsh
Locked         : no
OK? (yes/no) [yes]   ① Enterキーを押す。
adduser: INFO: Successfully added (pcuser) to the user database.
Add another user? (yes/no) [no] :   ② Enterキーを押す。
```

## 続き | 2. インストール

① Enterキーを押す。



## 続き | 2. インストール

---

- ① Enterキーを押す。



## 続き | 2. インストール

---

① Enterキーを押す。



② DVDをドライブから取り出す。

またはUSBを抜く。

### 3. 初期設定

---

- ① ディスク暗号化のパスワードを入力し、  
Enter キーを押す。



# 続き | 3. 初期設定

① Enterキーを押す。



## 続き | 3. 初期設定

---

- root でログインする

```
login: root  
Password: *****
```

### 3. 初期設定（システム共通）

---

- セキュリティパッチ・マイナーアップデートを実行する

```
# freebsd-update fetch
:
:           ←何か表示され、待ち状態になっている場合は、何度かQキーを押す
# freebsd-update install
```

- 再起動する

```
# shutdown -r now
```

## 続き | 3. 初期設定

---

ディスク暗号化のパスワードを入力し、  
Enter キーを押す。

```
BIOS drive C: is disk0
GELI Passphrase for disk0p3: _
```



## 続き | 3. 初期設定

Enterキーを押す。



## 続き | 3. 初期設定

---

- root でログインする

```
login: root  
Password: *****
```

# 続き | 3. 初期設定（システム共通）

## ■ 安定パッケージの設定

```
# vi /etc/pkg/FreeBSD.conf  
:  
url: "pkg+https://pkg.FreeBSD.org/${ABI}/release_3",  
:
```

release\_3に変更する

本作業は、インストール作業を確実に行うため、2025年6月にFreeBSD14.3がリリースされた時点のパッケージ群を使用する宣言をしています。なお、この操作を行うと、パッケージの更新は一切行われなくなります。

## 続き | 3. 初期設定（システム共通）

---

### ■ pkgをインストールする

```
# pkg  
• • [y/N] : y
```

もし、pkg: not enough arguments

:

For more information · · ·

と表示された場合は、インストール済みなので、次のページへ進んでください。

### 3. 初期設定 (sudo)

---

- sudo をインストールする

```
# pkg install -y sudo
```

- sudo を設定する

```
# visudo
```

```
%wheel  ALL=(ALL:ALL)  ALL ← 133行目前後 先頭の#を削除する
```

ここでは、sudoを実行できるグループを指定している。  
72ページの一般ユーザの追加で設定したLogin groupとも関連している。

### 3. 初期設定 (ntpd)

- システム起動時に ntpd が起動するよう設定する

```
# vi /etc/rc.conf
```

```
ntpdate_enable="YES"
```

← 最終行に追加する

- 時刻の同期先を設定する

```
# vi /etc/ntp.conf
```

```
server ntp1.jst.mfeed.ad.jp  
server ntp2.jst.mfeed.ad.jp  
server ntp3.jst.mfeed.ad.jp
```

]} 最終行に追加する

### 3. 初期設定 (powerd)

---

- 省エネ動作の設定を行う

```
# vi /etc/rc.conf  
  
powerd_enable="YES"
```

← 最終行に追加する

### 3. 初期設定（グラフィックドライバー）

※インテル内蔵GPUの場合は、下記の設定を行なってください。

#### ■ ドライバーインストール

```
# pkg install -y drm-515-kmod
```

下記表を目安にしてください。

CPU世代	内蔵GPU	ドライバ名
3	HD Graphics 2500/4000	drm-515-kmod
4	HD Graphics 4200～5200	drm-515-kmod
5	HD Graphics 5300～6300	drm-515-kmod
6	HD Graphics 510～580	drm-515-kmod
7	UHD Graphics 610～630	drm-61-kmod
8 / 9	UHD Graphics 610～630	drm-61-kmod
10	Iris Plus Graphics	drm-61-kmod
11	Iris Xe Graphics	drm-61-kmod
12	UHD Graphics 710/730/770	drm-61-kmod
13	UHD Graphics 730/770	drm-61-kmod
14	Arc iGPU	drm-61-kmod

※AMDのGPUの場合は、[drm-515-kmod をインストールしてください。](#)

## 続き | 3. 初期設定 (グラフィックドライバー)

---

### ■ ドライバー設定

```
# vi /etc/rc.conf
```

```
kld_list="i915kms" ← 最終行に追加する
```

※AMDのGPUの場合は、`kld_list="amdgpu"` と記載してください。

### ■ ユーザー設定

```
# pw groupmod video -m pcuser
```

### 3. 初期設定（ファイアウォール）

#### ■ ファイアウォールの設定を行う

```
# vi /etc/rc.conf
```

```
pf_enable="YES" ← 最終行に追加する  
pf_rules="/etc/pf.conf" ←
```

```
# vi /etc/pf.conf
```

```
set block-policy drop ← パケット拒否時の挙動を「無視」に設定する  
set skip on lo ← ループバック (localhost、127.0.0.1) の通信をチェックしない
```

```
block in all ← 外部から自分のPCへの通信を全て拒否する
```

```
pass out all keep state ← 自分のPCから外部への通信を全て許可する
```

```
pass in on em0 proto tcp to port { 22 } keep state
```

新規に  
追加する

LANのインターフェース名 (57ページ参照)

外部から自分のPCへの通信において、TCPの22番 (SSH) のみを通す

## 続き | 3. 初期設定

---

### ■ 再起動する

```
# shutdown -r now
```

## 続き | 3. 初期設定

---

ディスク暗号化のパスワードを入力し、  
Enter キーを押す。

```
BIOS drive C: is disk0
GELI Passphrase for disk0p3: _
```



## 続き | 3. 初期設定

Enterキーを押す。



## 続き | 3. 初期設定

---

### ■ 一般ユーザーでログインする

login: **pcuser**

Password: **\*\*\*\*\***

### 3. 初期設定 (vimエディタ)

- vim エディターをインストールする

```
% sudo pkg install -y vim
```

```
Password: *****
```

pcuserのパスワードを入力する

(※以降、sudoコマンドで「Password:」、または「パスワード:」と表示された場合は、pcuser のパスワードを入力する)

- ~/.vimrc に以下の内容を追加する

```
% vi ~/.vimrc
```

```
set tabstop=4
```

タブ幅を4に設定する

```
set ambiwidth=double
```

★や■などの記号を全角扱いする

```
set ignorecase
```

検索時に大文字と小文字の区別をしない

```
set ruler
```

ルーラ（カーソル位置）を表示する

```
syntax on
```

プログラムコードなどの構文を色付けする

### 3. 初期設定（シェルスクリプト）

- `~/.cshrc` に以下の内容を追加する

```
% vi ~/.cshrc

{
alias ll      ls -laF

alias l       'ls -l | more -e'
alias ll     'ls -la | more -e'
alias vi      vim
alias rm     'rm -i'

}
```

既存の alias 直下に追加する

ファイル削除時に確認する

(次ページに続きます)

## 続き | 3. 初期設定（シェルスクリプト）

(続き) `~/.cshrc` に以下の変更を行う

```
        }  
→ set path = (/sbin /bin /usr/sbin /usr/bin /usr/local/sbin  
  /usr/local/bin $HOME/bin)  
        }  
→ umask 27
```

先頭のコメント#を削除し、22を27に変更する

先頭のコメント#を削除する

## 続き | 3. 初期設定（シェルスクリプト）

- `~/.login` を編集する

```
% vi ~/.login

()

# if ( -x /usr/bin/fortune ) /usr/bin/fortune freebsd-tips
```

最終行をコメントアウトする

### 3. 初期設定（ウィンドウ関連1）

---

- X Window System をインストールする

```
% sudo pkg install -y xorg
```

## 続き | 3. 初期設定（ウィンドウ関連1）

- FVWM2 の一部のアイコンファイルが欲しいので、いったんインストールする

```
% sudo pkg install -y fvwm
```

- アイコンファイルを抜き出す

```
% mkdir ~/icons  
% cp /usr/local/share/fvwm/pixmaps/programs.xpm ~/icons  
% cp /usr/local/share/fvwm/pixmaps/xterm-sol.xpm ~/icons
```

- アイコンファイルを.png形式にするためのツールをインストールする

```
% sudo pkg install -y ImageMagick7
```

- アイコンファイルを.png形式にする

```
% cd ~/icons  
% magick programs.xpm -trim +repage -scale 200% programs.png  
% magick xterm-sol.xpm -crop 44x34+5+2 xterm-sol.png  
% cd
```

アイコンの空白部分をトリミングし、2倍の大きさにしている。

少し画像サイズ小さくなるよう、トリミングしている。

## 続き | 3. 初期設定（ウィンドウ関連1）

---

- FVWM3をインストールする

```
% sudo pkg install -y fvwm3
```

- 日本語フォントをインストールする

```
% sudo pkg install -y ja-font-ipa
```

### 3. 初期設定（ウィンドウ関連2）

- `~/.xinitrc` に以下の内容を新規に追加する

```
% vi ~/.xinitrc

#!/bin/sh

export LANG=ja_JP.UTF-8           ← 日本語環境にする
export LC_ALL=ja_JP.UTF-8         ←

xset r rate 250 40              ← キーリピートの設定
xset m 5 3                       ← マウスの動きの設定

xsetroot -solid ¶#5b468f          ← 背景色の設定
xclock -update 1 -d -strftime '%Y-%m-%d(%a) %T' -geometry -1+1 -bg ¶#f8f8f7 &
xterm -geometry 109x5+1+1 -e tail -f /var/log/messages &
xfce4-terminal --geometry=80x24+1+97 &

exec fvwm3                         ← ウィンドウマネージャの起動
```

※ 日本語キーボードの場合は、この位置に `setxkbmap jp` を追加する。

### 3. 初期設定（ウィンドウ関連3）

---

- fvwm3の設定ファイルをダウンロードする

```
% cd  
% fetch ¥  
https://raw.githubusercontent.com/moginwc/freebsd143/refs/heads/main/.fvwm2rc
```

### 3. 初期設定（フォント設定）

#### ■デフォルトフォントの設定を行います

##### □GTK2系

```
% vi ~/.gtkrc-2.0  
  
gtk-font-name="IPAPGothic" ← 新規に追加する
```

##### □GTK3系

```
% mkdir -p ~/.config/gtk-3.0  
% vi ~/.config/gtk-3.0/settings.ini  
  
[Settings] } 新規に追加する  
gtk-font-name=IPAPGothic
```

## 続き | 3. 初期設定（フォント設定）

- フォントのアンチエイリアスをグレースケール方式（GTK2系）にします。また、若干なめらかにする設定です（GTK2,3系）。

```
% vi ~/.Xresources
```

```
Xft.rgba: none ← 新規に追加する  
Xft.hintstyle: hintslight ←
```

```
% vi ~/.xinitrc
```

```
{  
xset m 5 3  
xrdb -merge ~/.Xresources ← 追加する  
}
```

※この設定を行わないと、この後インストールする xfce4-terminal において、アンダーバーが表示されない場合があります。

### 3. 初期設定（端末エミュレータ）

---

- xfce4-terminal をインストールする

```
⇒ sudo pkg install -y xfce4-terminal
```

### 3. 初期設定（日本語入力1）

#### ■ uim-anthy をインストールする

```
% sudo pkg install -y ja-uim-anthy  
uim-gtk2 uim-gtk3 uim-qt5
```

#### ■ ~/.xinitrc に以下の内容を追加する

```
% vi ~/.xinitrc

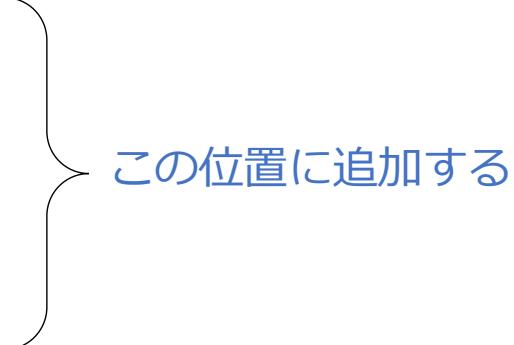
{

export LC_ALL=ja_JP.UTF-8

export GTK_IM_MODULE=uim
export QT_IM_MODULE=uim
export XMODIFIERS=@im=uim
export XIM=uim
uim-xim &
uim-toolbar-gtk3 -1-1 &

}

xset r rate 250 40
```



この位置に追加する

## 続き | 3. 初期設定

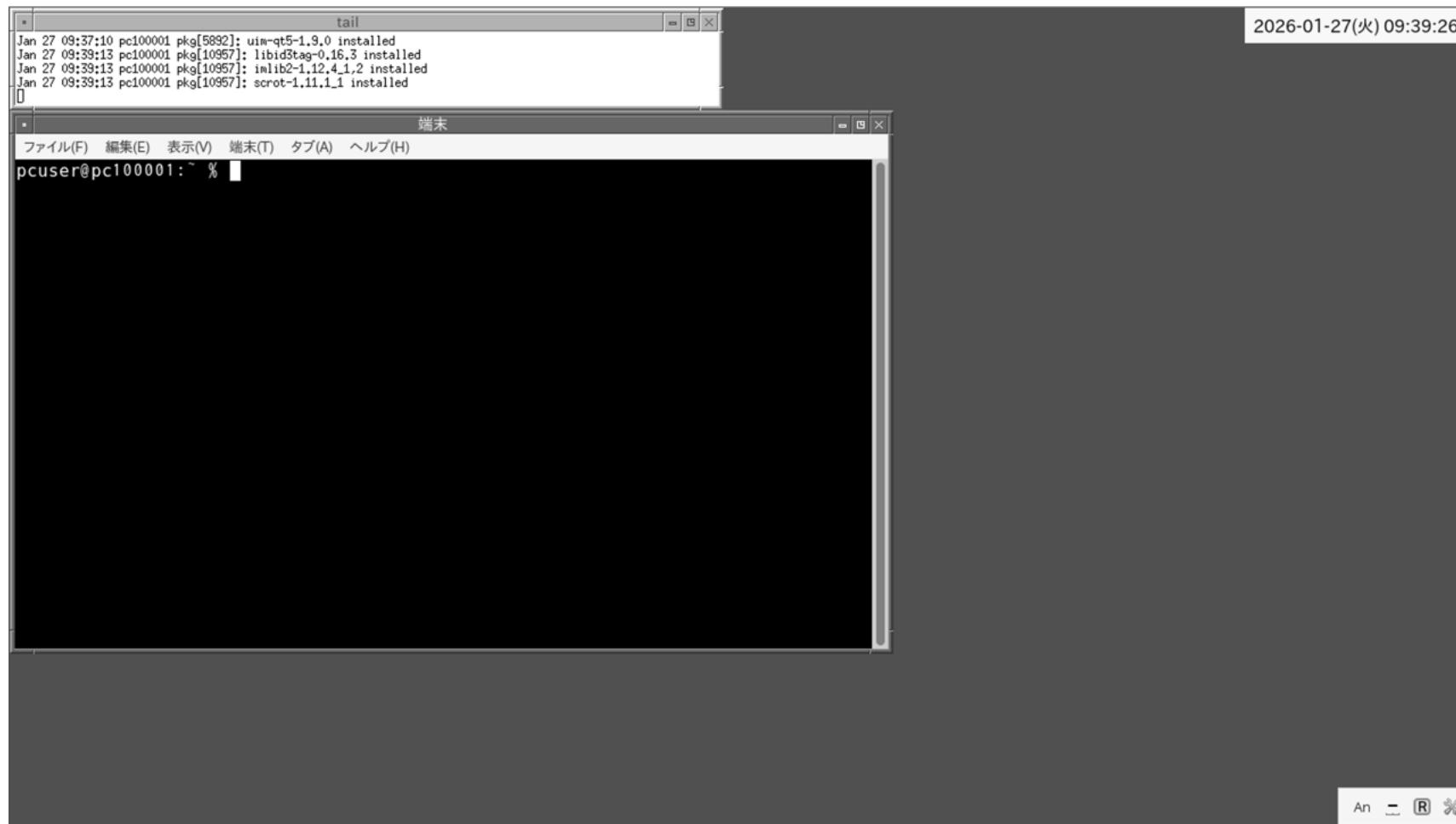
---

- ウィンドウシステムを起動する

```
% startx
```

## 続き | 3. 初期設定

下記画面が表示される



### 3. 初期設定（端末エミュレータ設定）

編集(E)→設定(E)をクリックする



### 続き | 3. 初期設定（端末エミュレータ設定）



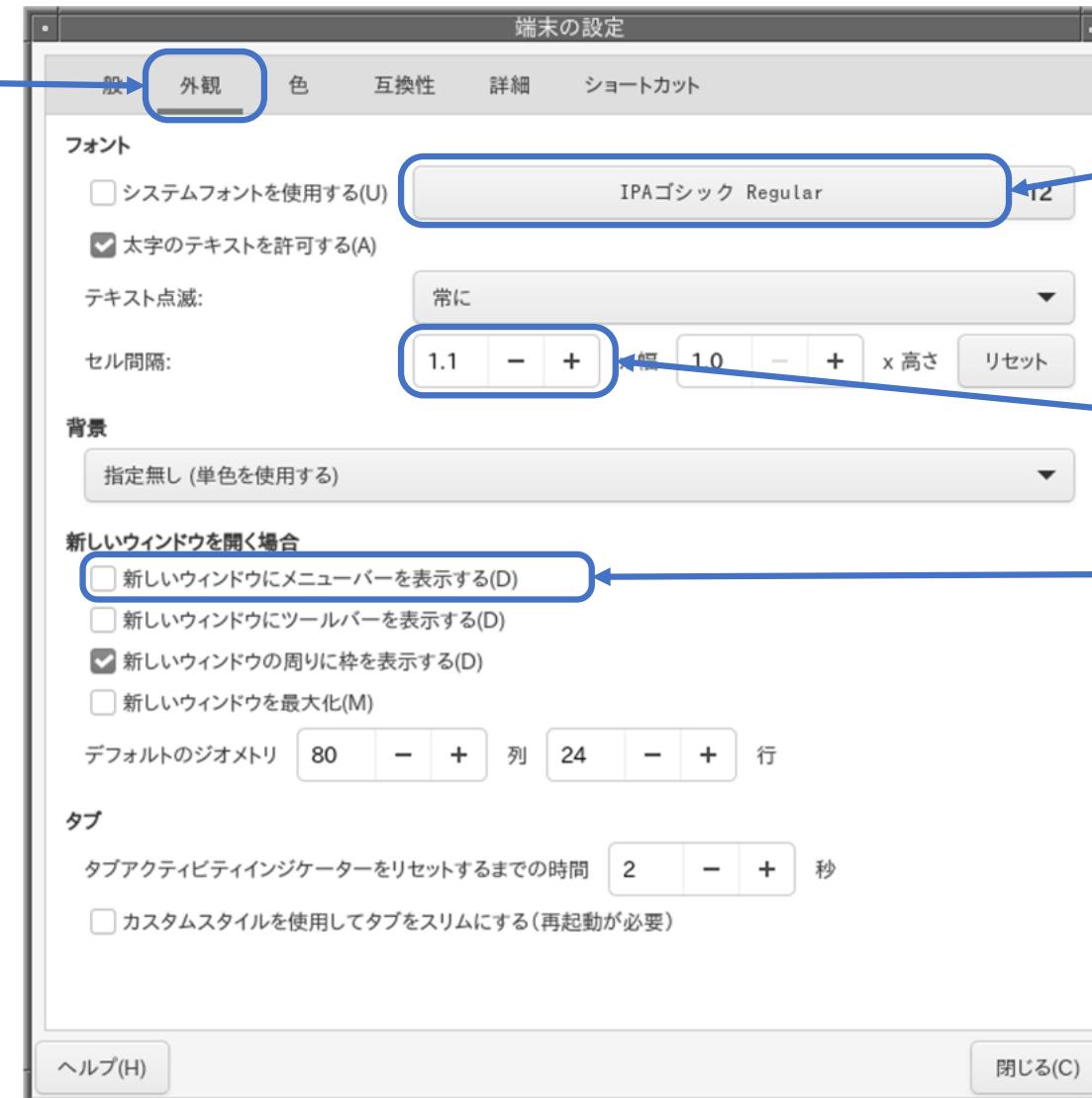
①チェックする

②チェックする

③チェックを  
外す

### 続き | 3. 初期設定 (端末エミュレータ設定)

①クリックする



②IPAゴシック  
Regularを選択  
する

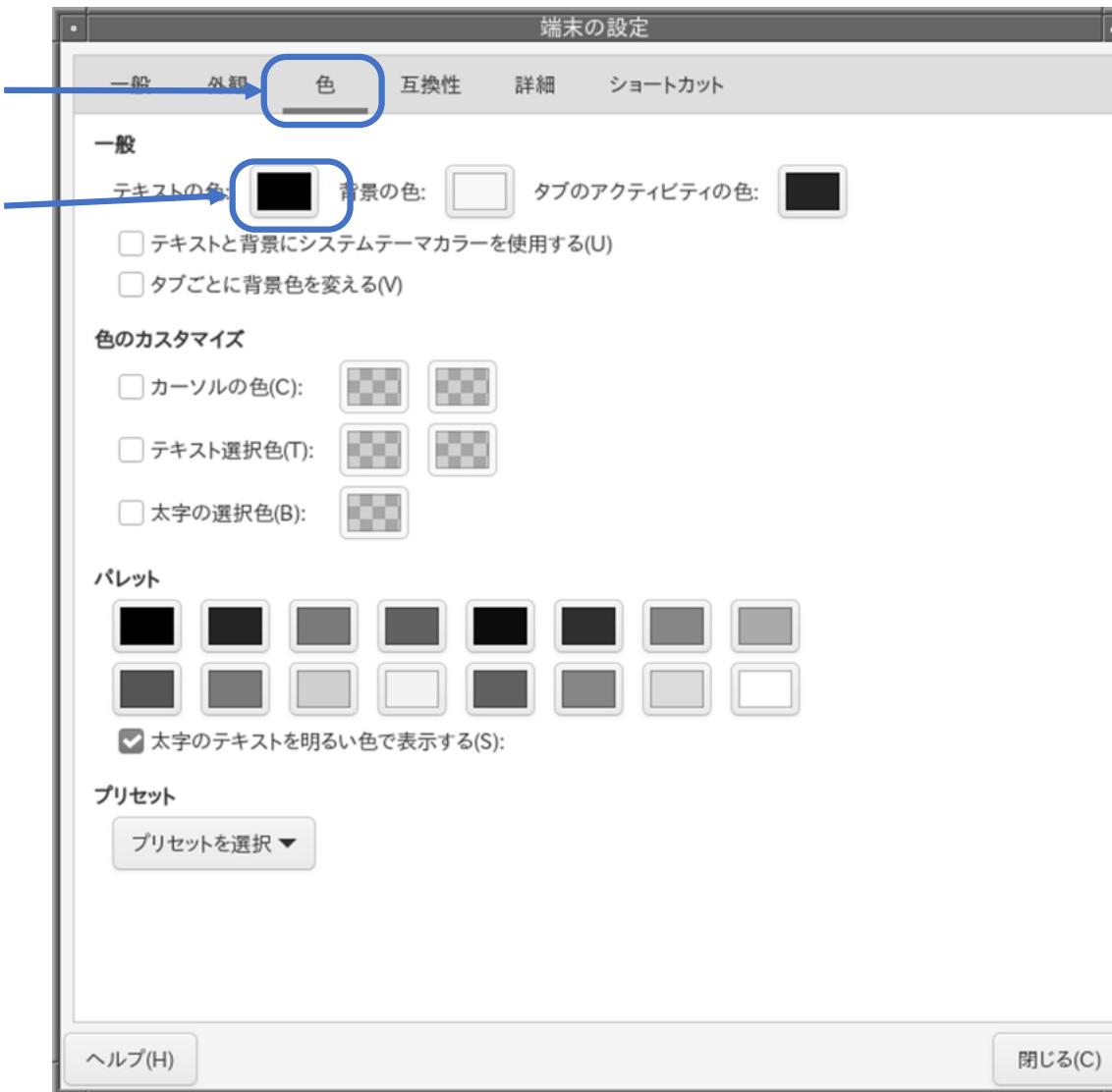
③1.1に設定する

④チェックを  
外す

## 続き | 3. 初期設定（端末エミュレータ設定）

①クリックする

②クリックする



### 続き | 3. 初期設定（端末エミュレータ設定）



## 続き | 3. 初期設定 (端末エミュレータ設定)

①クリックする



### 続き | 3. 初期設定（端末エミュレータ設定）



### 続き | 3. 初期設定（端末エミュレータ設定）



### 続き | 3. 初期設定（端末エミュレータ設定）

①クリックする



②全角にする

### 続き | 3. 初期設定（端末エミュレータ設定）

①クリックする



### 3. 初期設定（日本語入力2）

- [CapsLock]キーを[半角/全角]キーに割り当てる（その1）

※ 日本語キーボードの場合は、設定不要

```
% mkdir -p ~/.xkb/symbols
```

```
% vim ~/.xkb/symbols/myuser
```

```
partial modifier_keys
xkb_symbols "userkeys" {
    replace key <CAPS> { [ Zenkaku_Hankaku ] } ;
};
```

新規に追加する

## 続き | 3. 初期設定（日本語入力2）

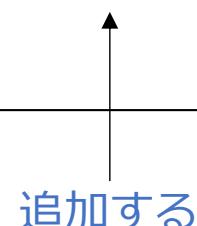
[CapsLock]キーを[半角/全角]キーに割り当てる（その2）

※ 日本語キーボードの場合は、設定不要

```
% mkdir ~/.xkb/keymap  
% setxkbmap -print > ~/.xkb/keymap/mykbd
```

```
% vim ~/.xkb/keymap/mykbd
```

```
xkb_keymap {  
    xkb_keycodes { include "evdev+aliases(qwerty)" } ;  
    xkb_types { include "complete" } ;  
    xkb_compat { include "complete" } ;  
    xkb_symbols { include "pc+us+inet(evdev) +myuser(userkeys)" } ;  
    xkb_geometry { include "pc(pc105)" } ;  
};
```



## 続き | 3. 初期設定（日本語入力2）

前々ページ、前ページの設定を有効にする

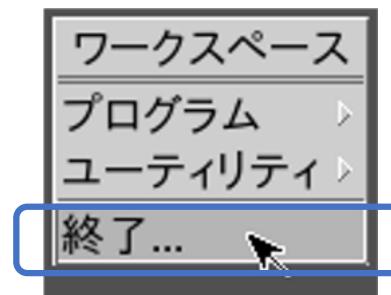
※ 日本語キーボードの場合は、設定不要

```
% vim ~/.xinitrc
{
    xrdb -merge ~/.Xresources
    xkbcomp -I$HOME/.xkb ~/.xkb/keymap/mykbd $DISPLAY
}
```

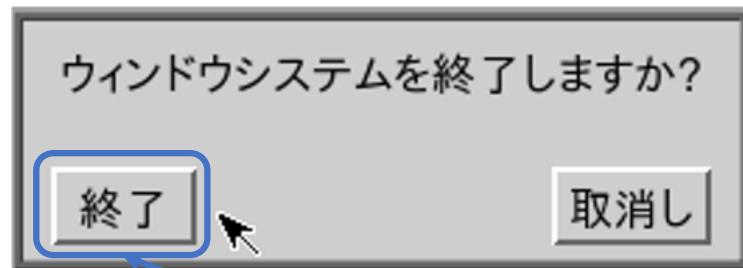
この位置に追加する

## 続き | 3. 初期設定

### ■ いったんウインドウシステムを終了する



デスクトップ上にて、  
・マウスを右クリック  
・終了をクリックする



終了をクリックする

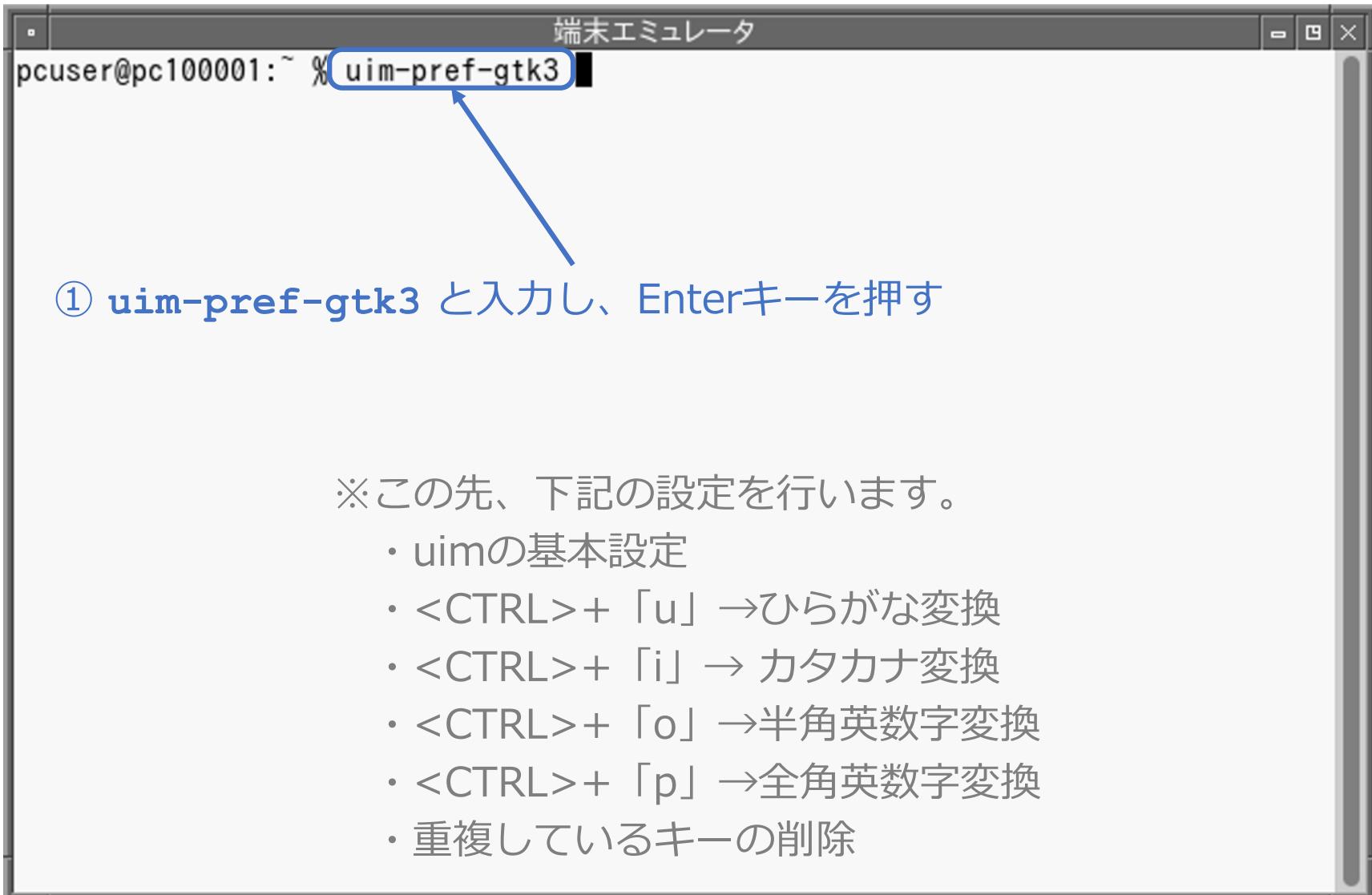
## 続き | 3. 初期設定

---

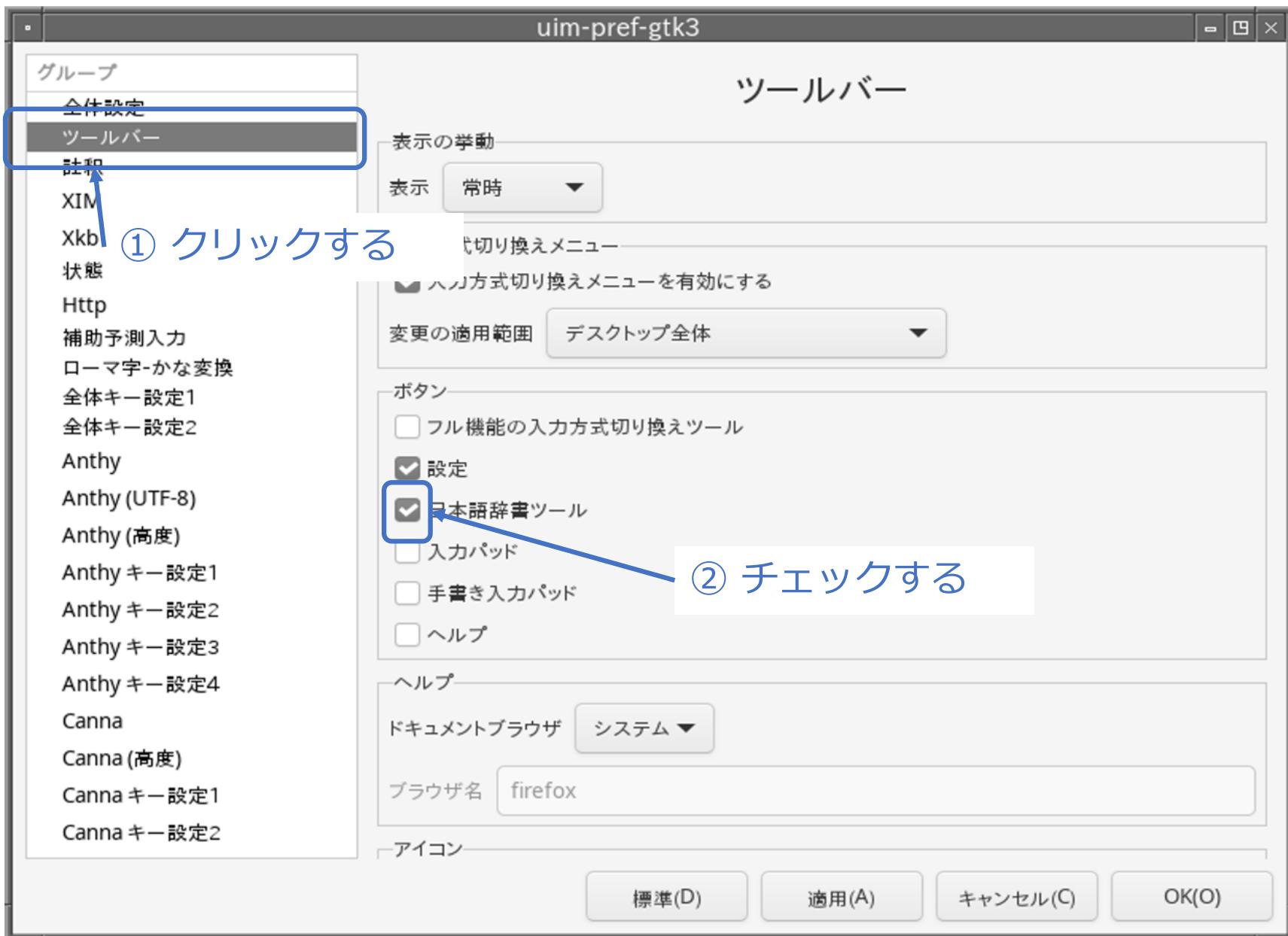
- 再度ウインドウシステムを起動する

```
% startx
```

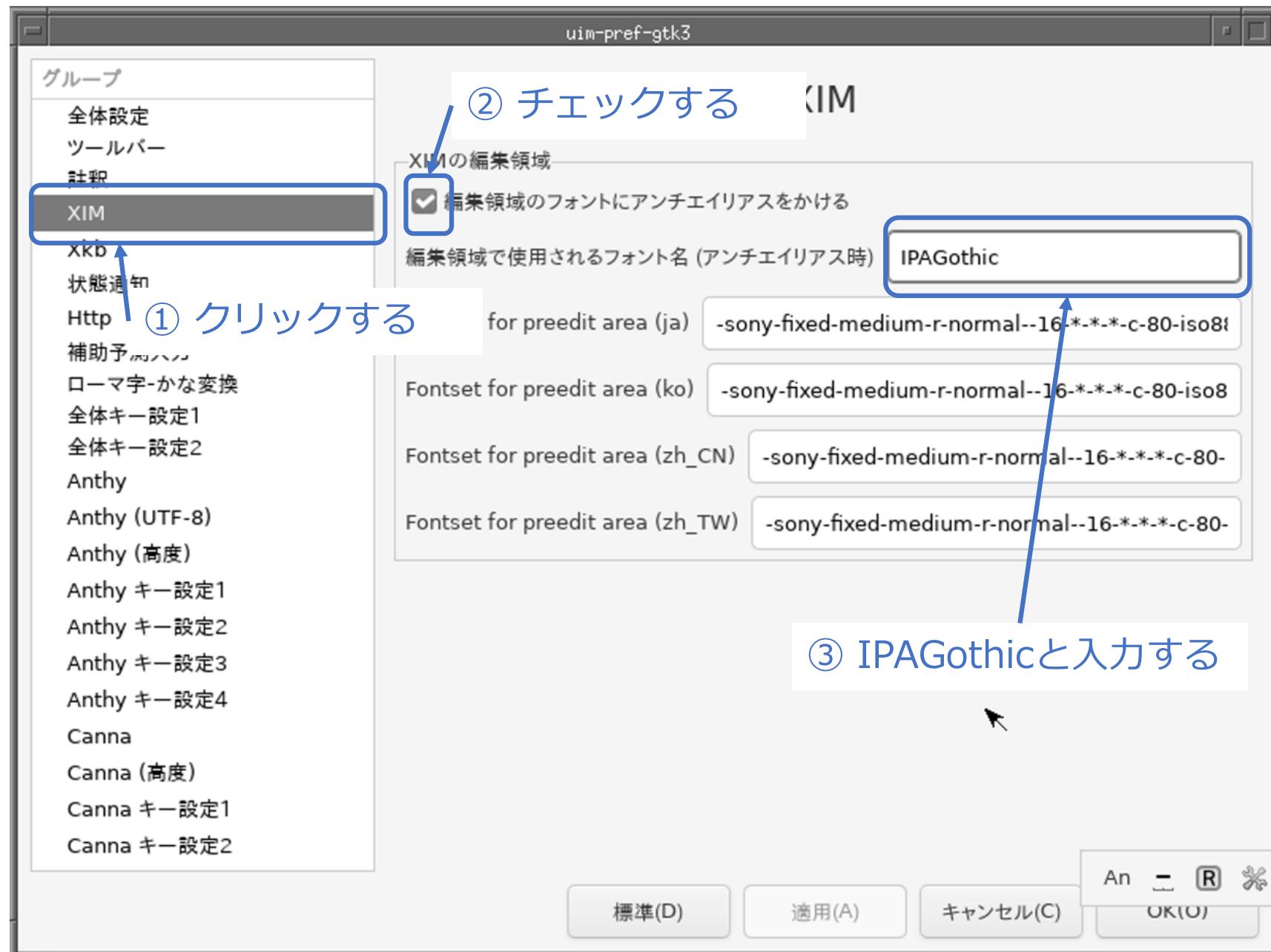
### 3. 初期設定（日本語入力3）



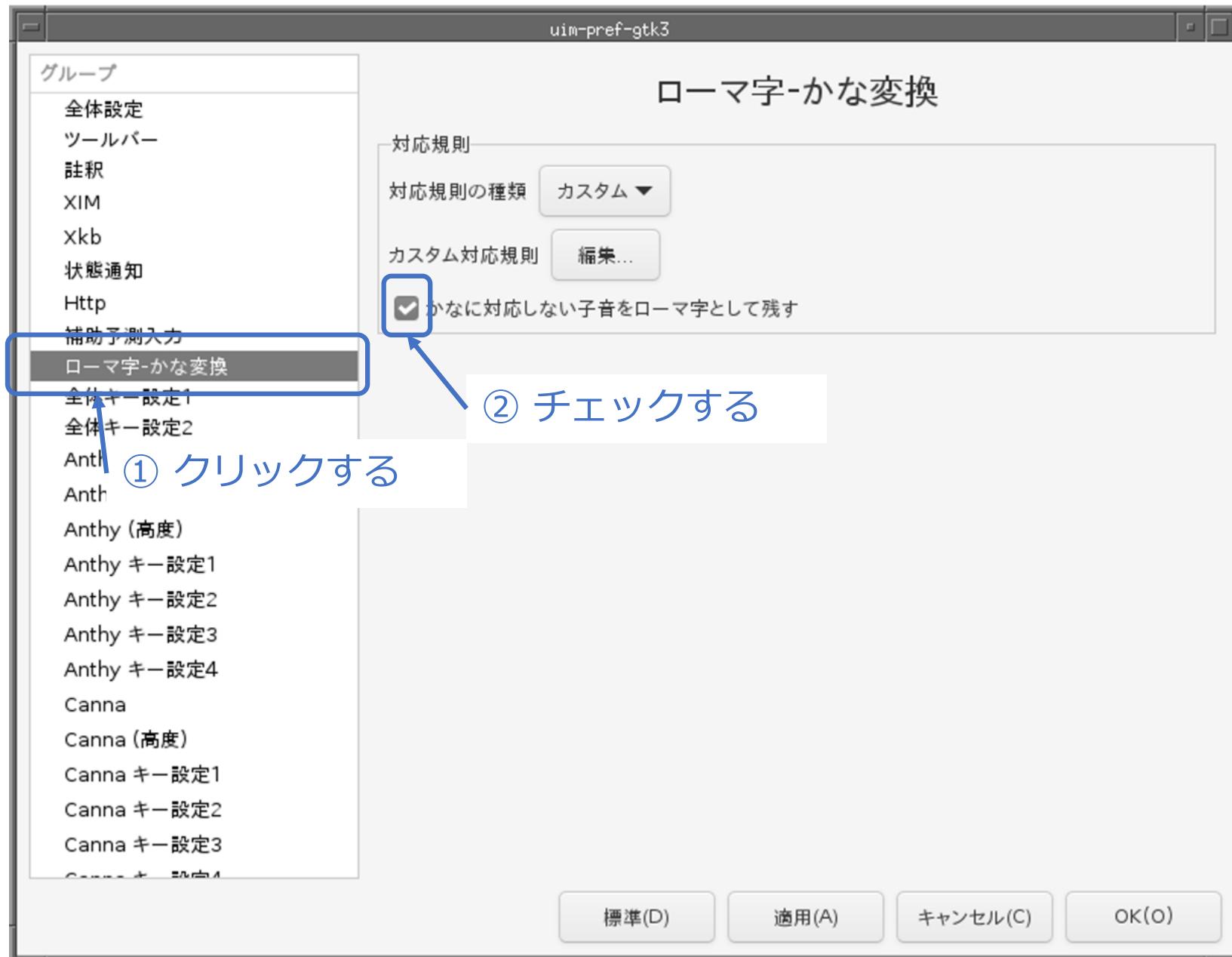
## 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定（日本語入力3）



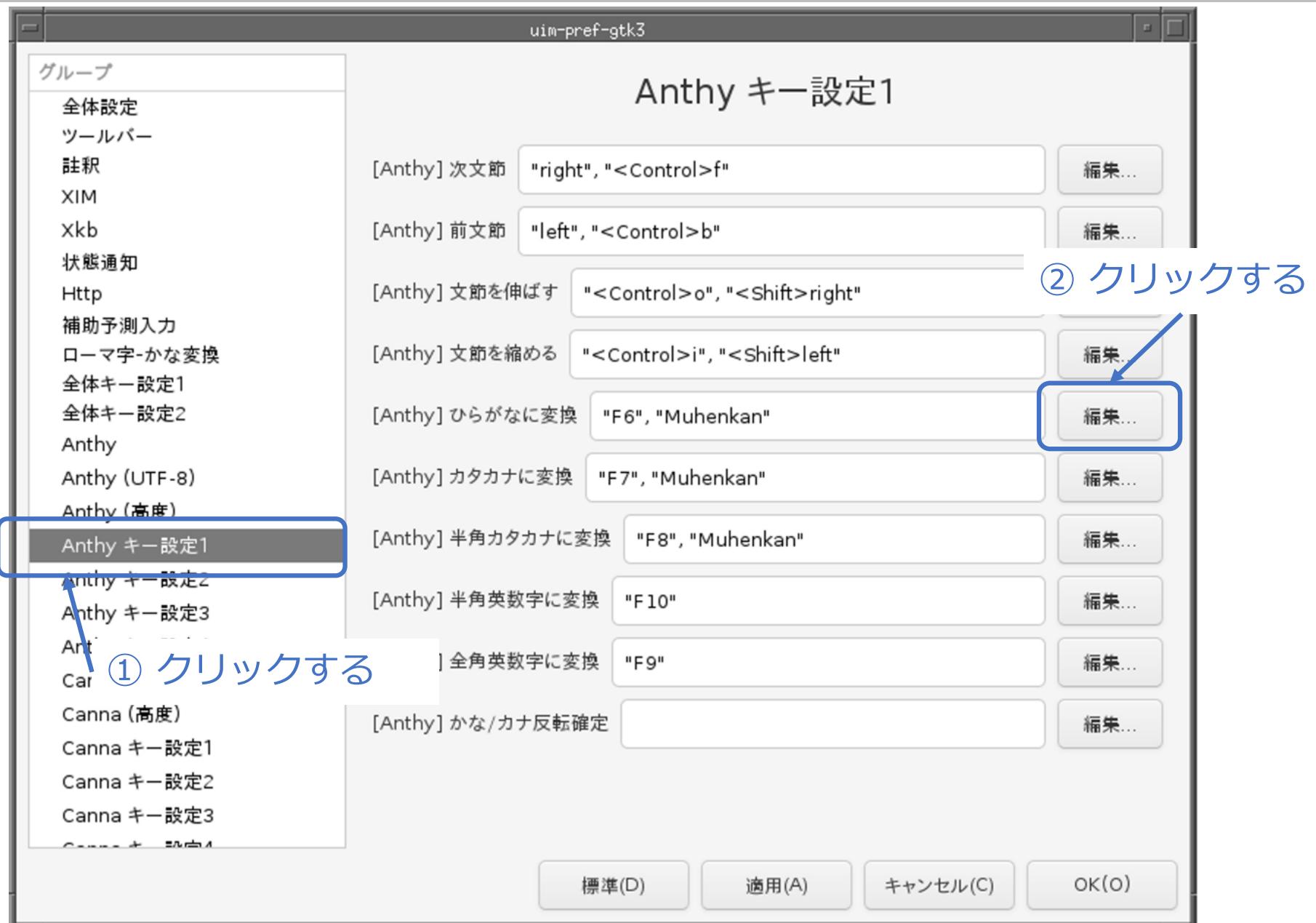
### 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定 (日本語入力3)



## 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定（日本語入力3）



### 続き | 3. 初期設定（日本語入力3）



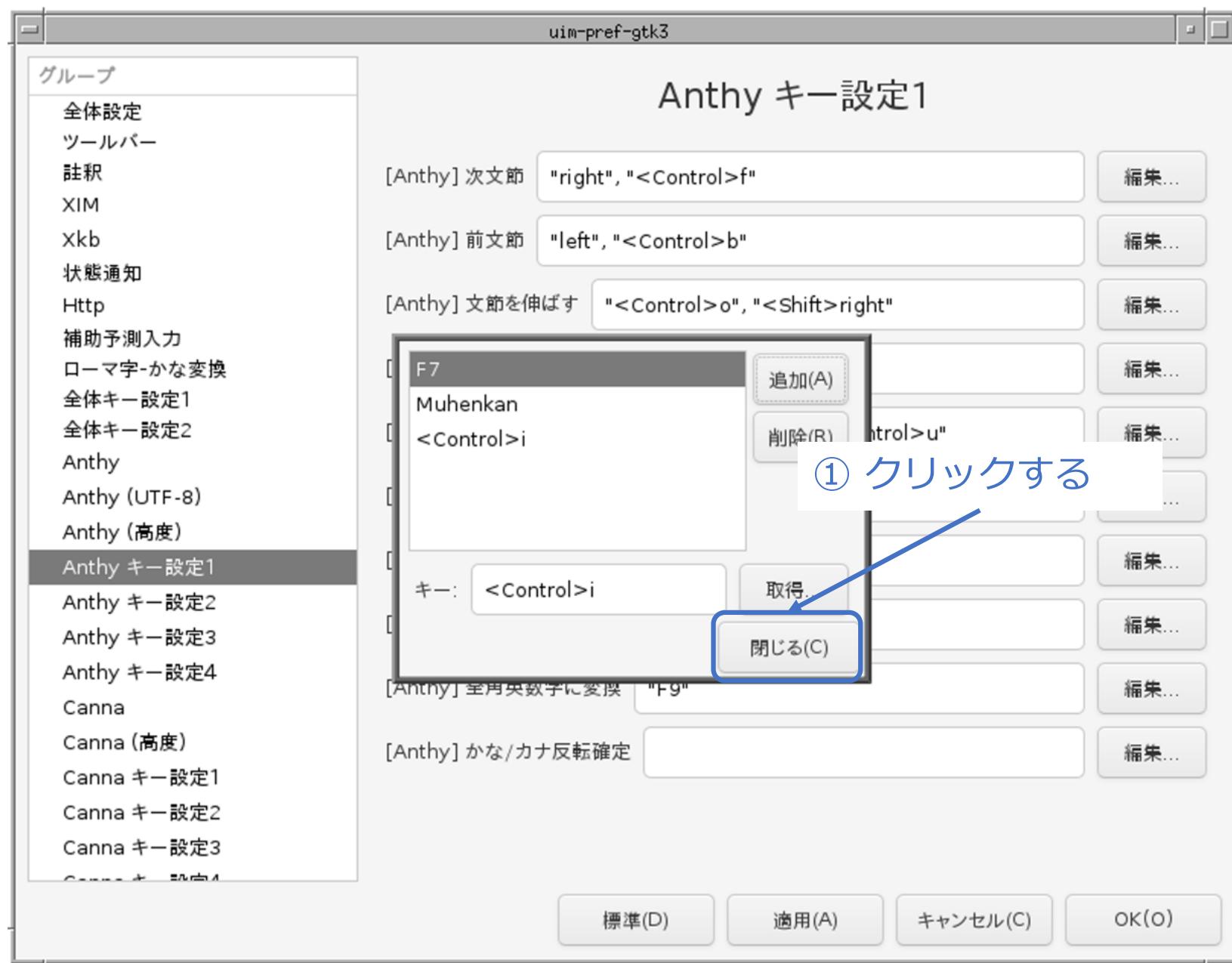
## 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定 (日本語入力3)



## 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定（日本語入力3）



# 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定（日本語入力3）



# 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定 (日本語入力3)



## 続き | 3. 初期設定（日本語入力3）



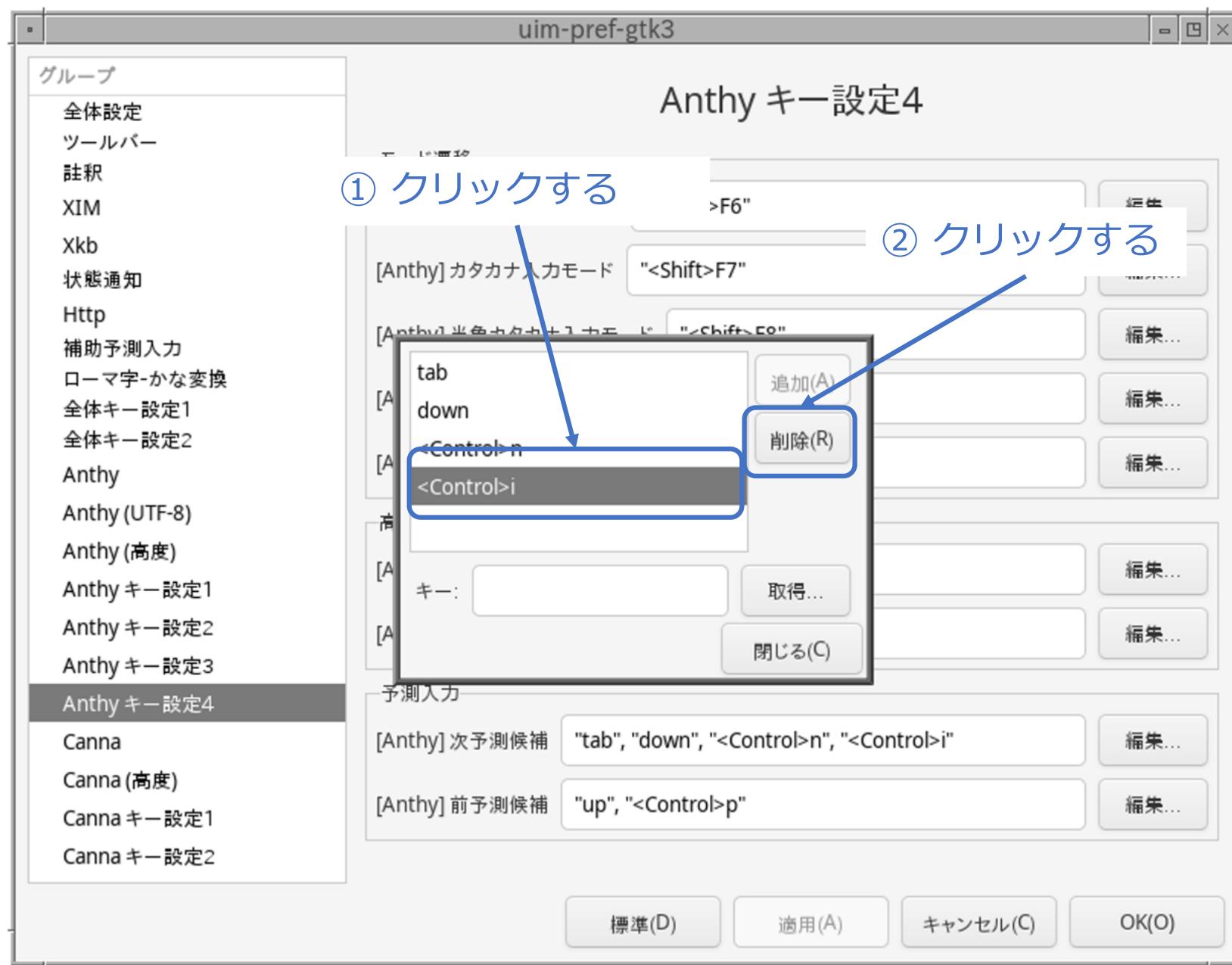
## 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定（日本語入力3）



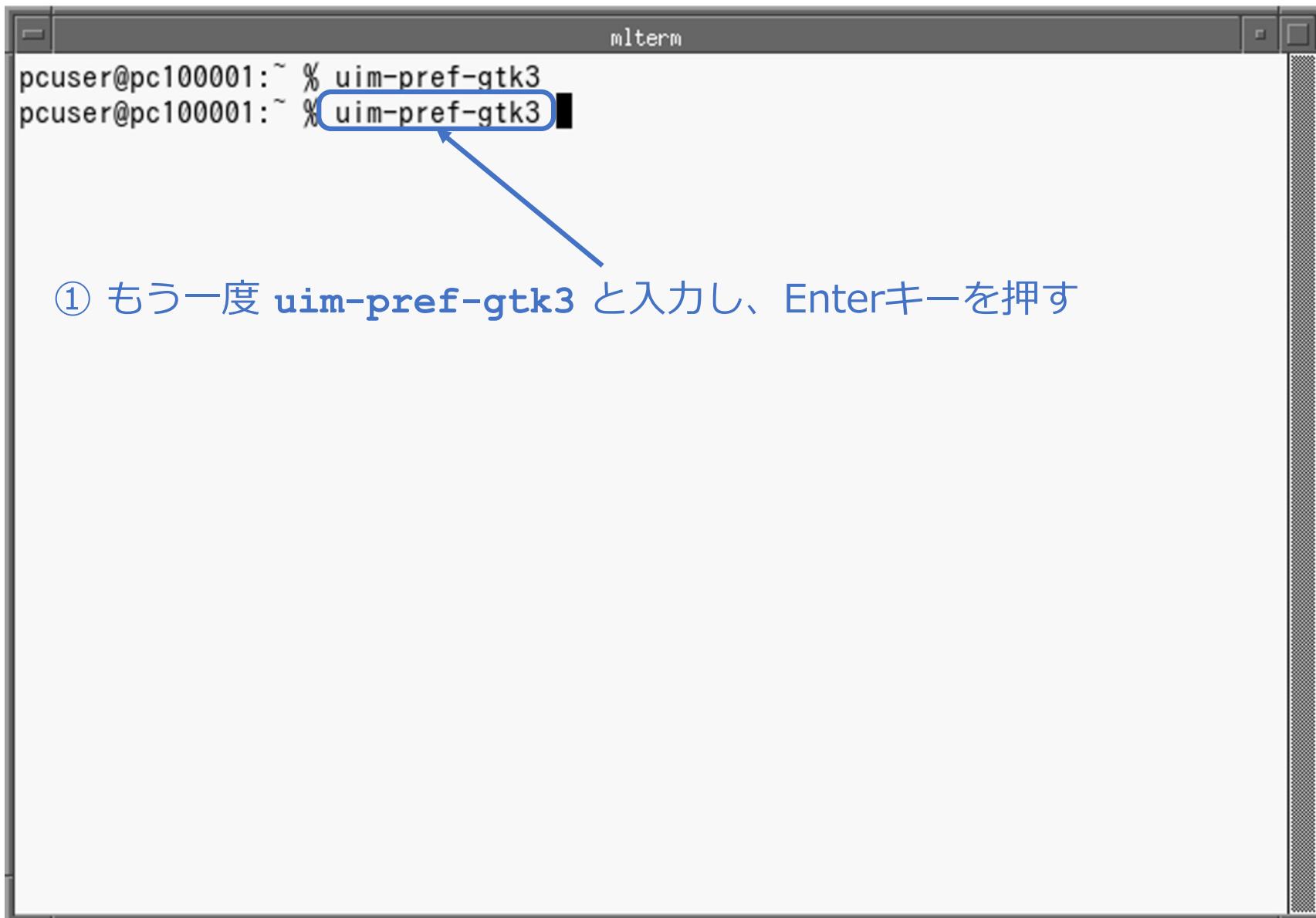
## 続き | 3. 初期設定（日本語入力3）



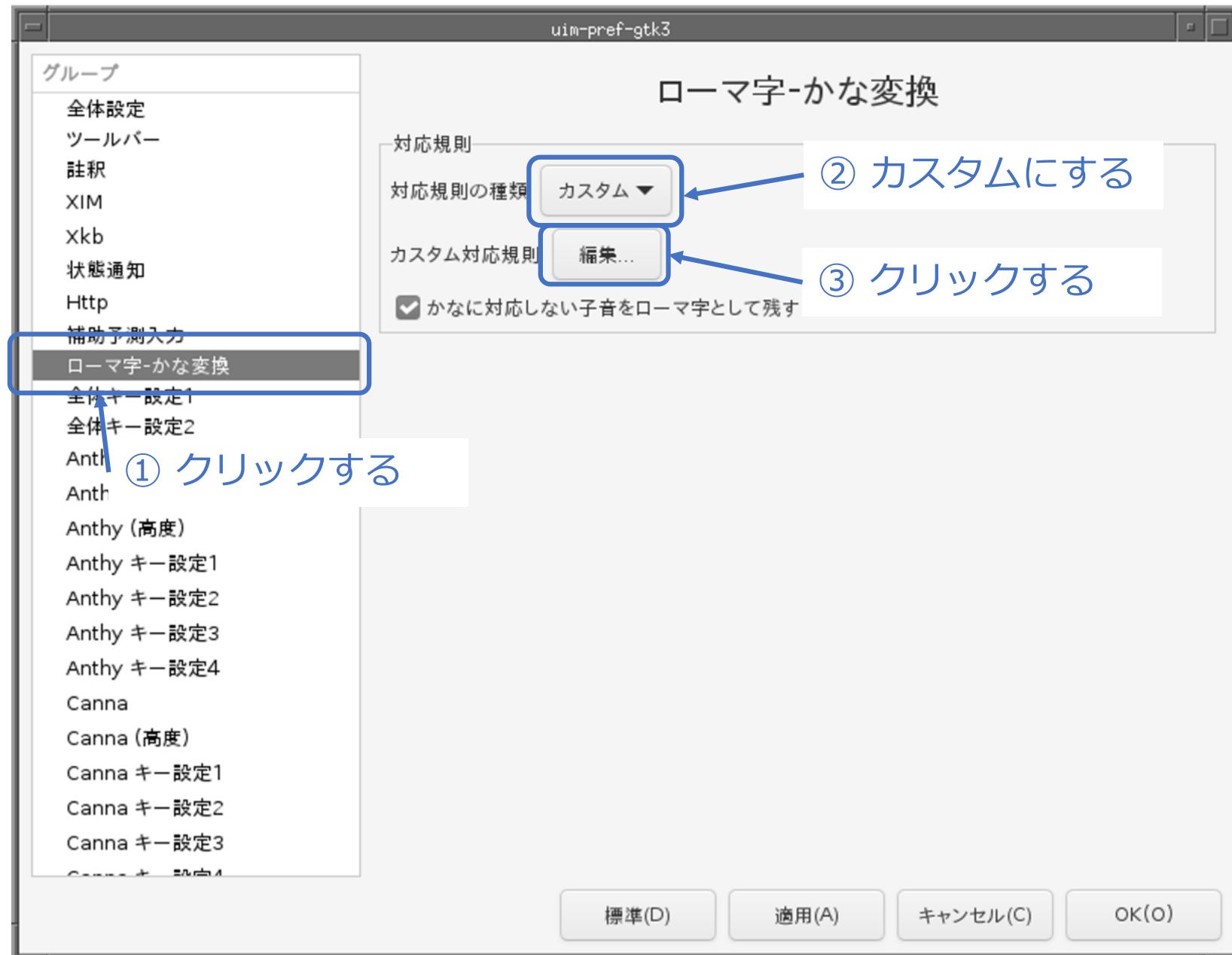
## 続き | 3. 初期設定（日本語入力3）



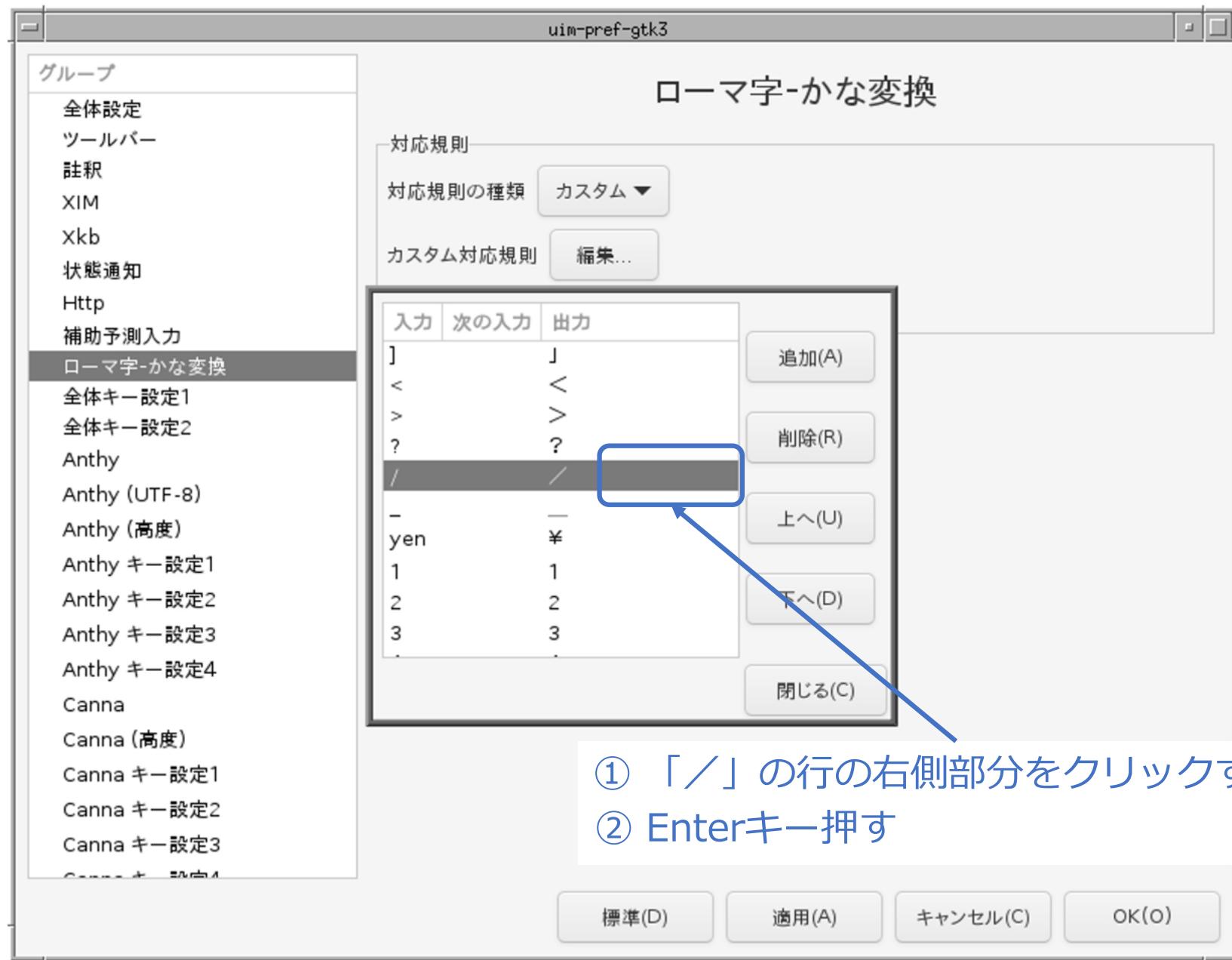
## 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定（日本語入力3）



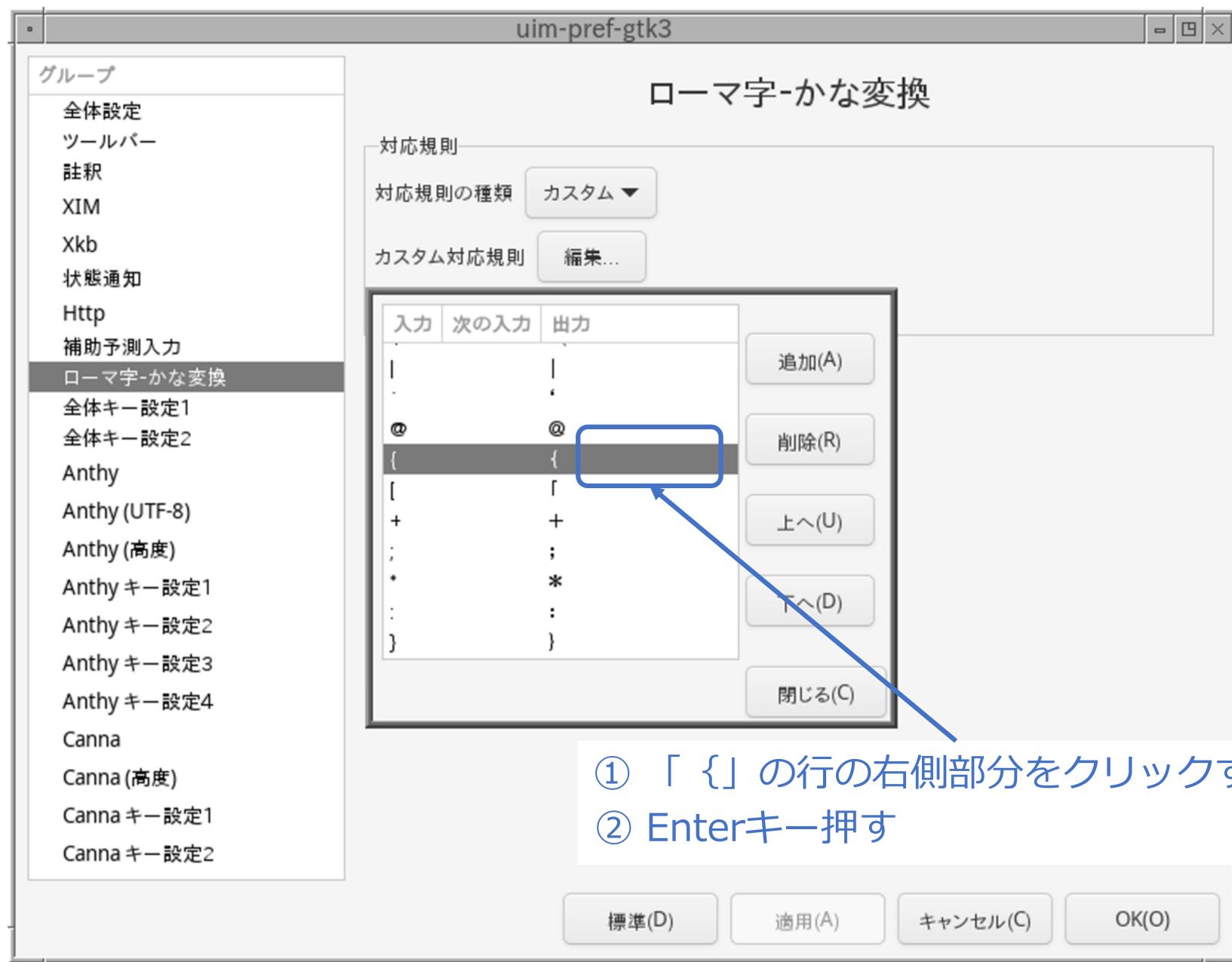
## 続き | 3. 初期設定（日本語入力3）



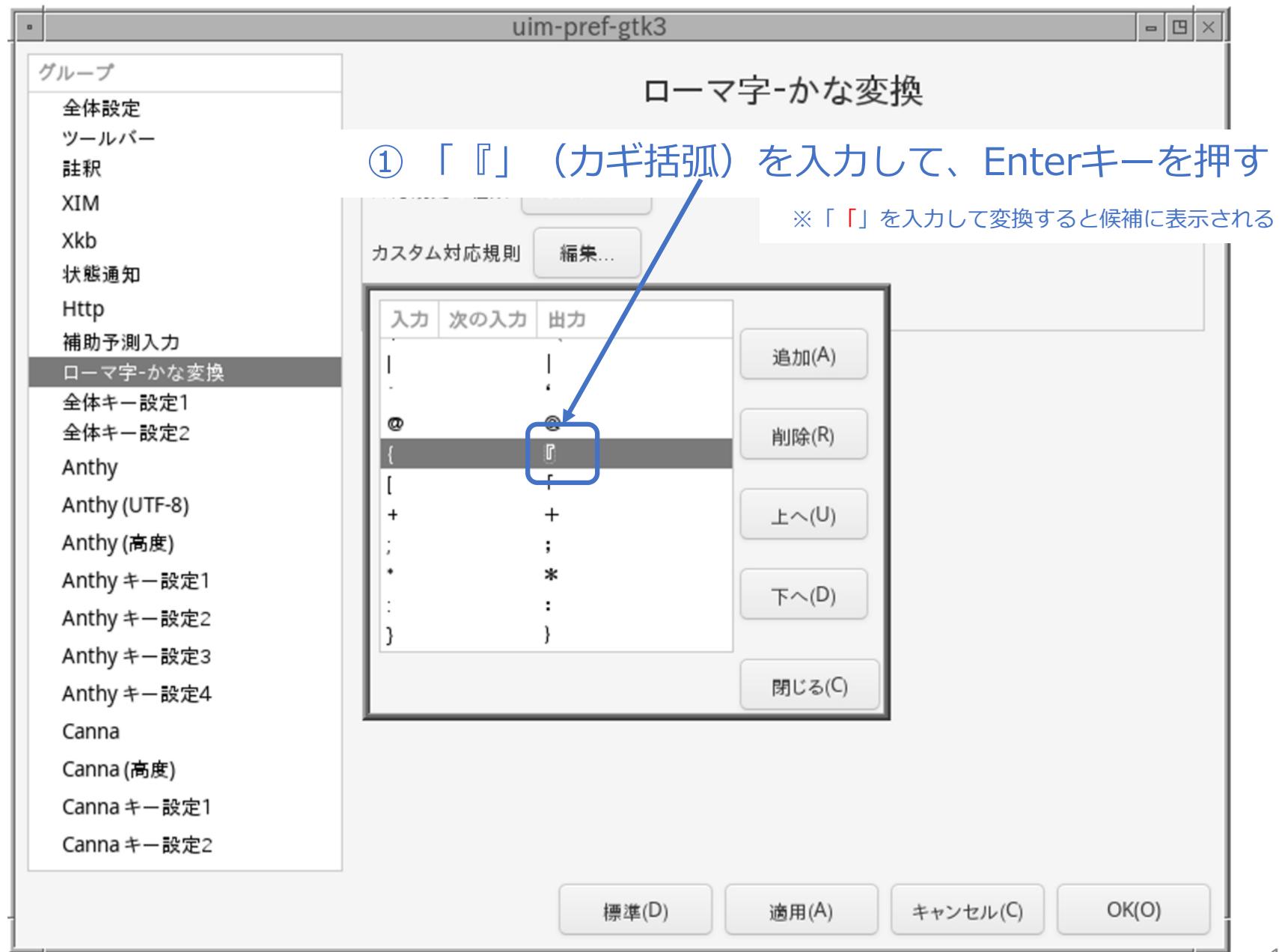
## 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定（日本語入力3）



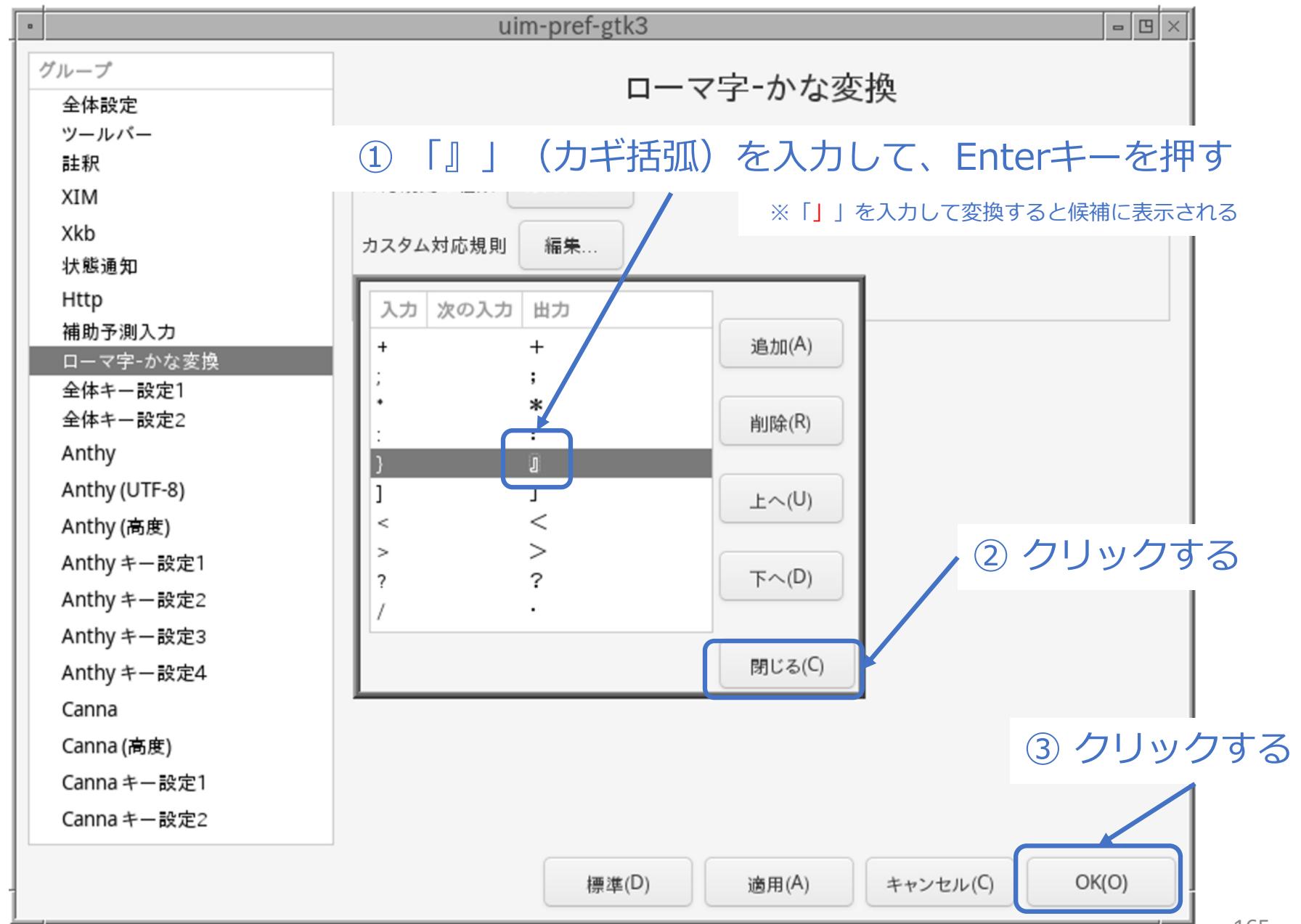
## 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定（日本語入力3）



## 続き | 3. 初期設定（日本語入力3）

---

- ユーザー辞書ファイルを作成する

```
% touch ~/.anthy/private_words_default
```

### 3. 初期設定 (Firefox、その他)

---

- Firefox (Webブラウザ) をインストールする

```
% sudo pkg install -y firefox-esr
```

- scrot (スクリーンショット) をインストールする

```
% sudo pkg install -y scrot
```

- xlock (画面ロック) をインストールする

```
% sudo pkg install -y xlockmore
```

- lupe (拡大鏡) をインストールする

```
% sudo pkg install -y lupe
```

- xpad (付箋) をインストールする

```
% sudo pkg install -y xpad3
```

### 3. 初期設定（音量キー設定）

- キーボード上のボリュームキーにて音量調節をするように設定します  
(ノートパソコンなど、キーボード上に音量キーが存在する場合)

コメント#を外す

```
% vim ~/.fvwm2rc

Key XF86AudioRaiseVolume A A Exec mixer vol.volume=+0.05
Key XF86AudioLowerVolume A A Exec mixer vol.volume=-0.05
Key XF86AudioMute          A A Exec mixer vol.mute=^
```

## 続き | 3. 初期設定（音量キー設定）

(デスクトップパソコンなど、101や106キーボードの場合)

コメント#を外す

```
% vim ~/.fvwm2rc
{
    Key Up          A CM Exec mixer vol.volume=+0.05
    Key Down        A CM Exec mixer vol.volume=-0.05
    Key M           A CM Exec mixer vol.mute=^
```

CTRL+ALT+↑ … 音量アップ  
CTRL+ALT+↓ … 音量ダウン  
CTRL+ALT+M … ミュート／ミュート解除

再起動後も音が出ない場合は、6-6.を参照

## 続き | 3. 初期設定

---

### ■ 再起動する

```
% sudo shutdown -r now
```

以上でインストール、初期設定は終わりです。

## 4. FreeBSDの起動～終了

---

ここからは、FreeBSDの起動から  
終了までの一通りの流れを説明します。

## 続き | 4. FreeBSDの起動～終了

---

ディスク暗号化のパスワードを入力し、  
Enter キーを押す。

```
BIOS drive C: is disk0
GELI Passphrase for disk0p3: _
```



## 続き | 4. FreeBSDの起動～終了

Enterキーを押す。



# 続き | 4. FreeBSDの起動～終了

---

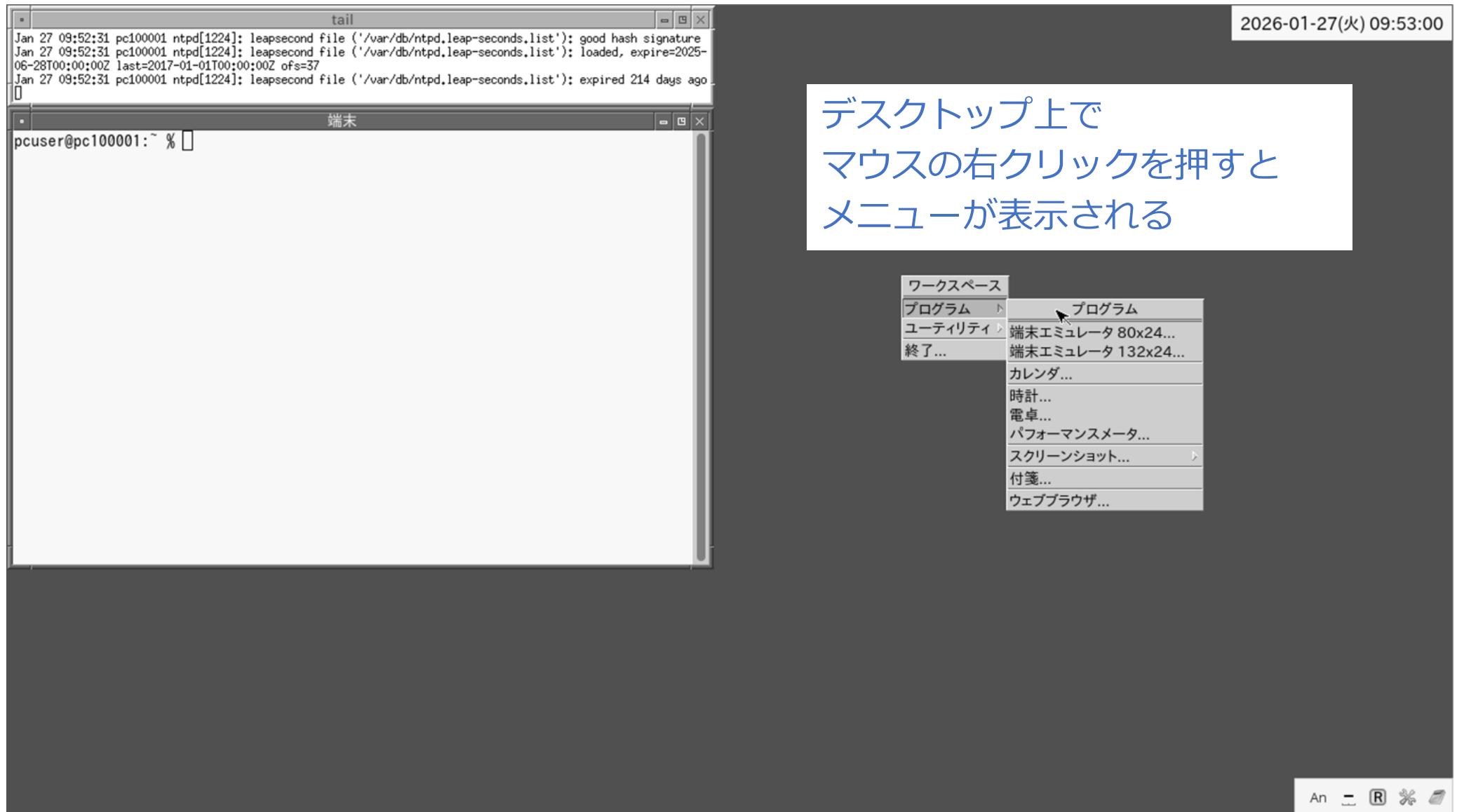
- 一般ユーザーでログインする

```
login: pcuser  
Password: *****
```

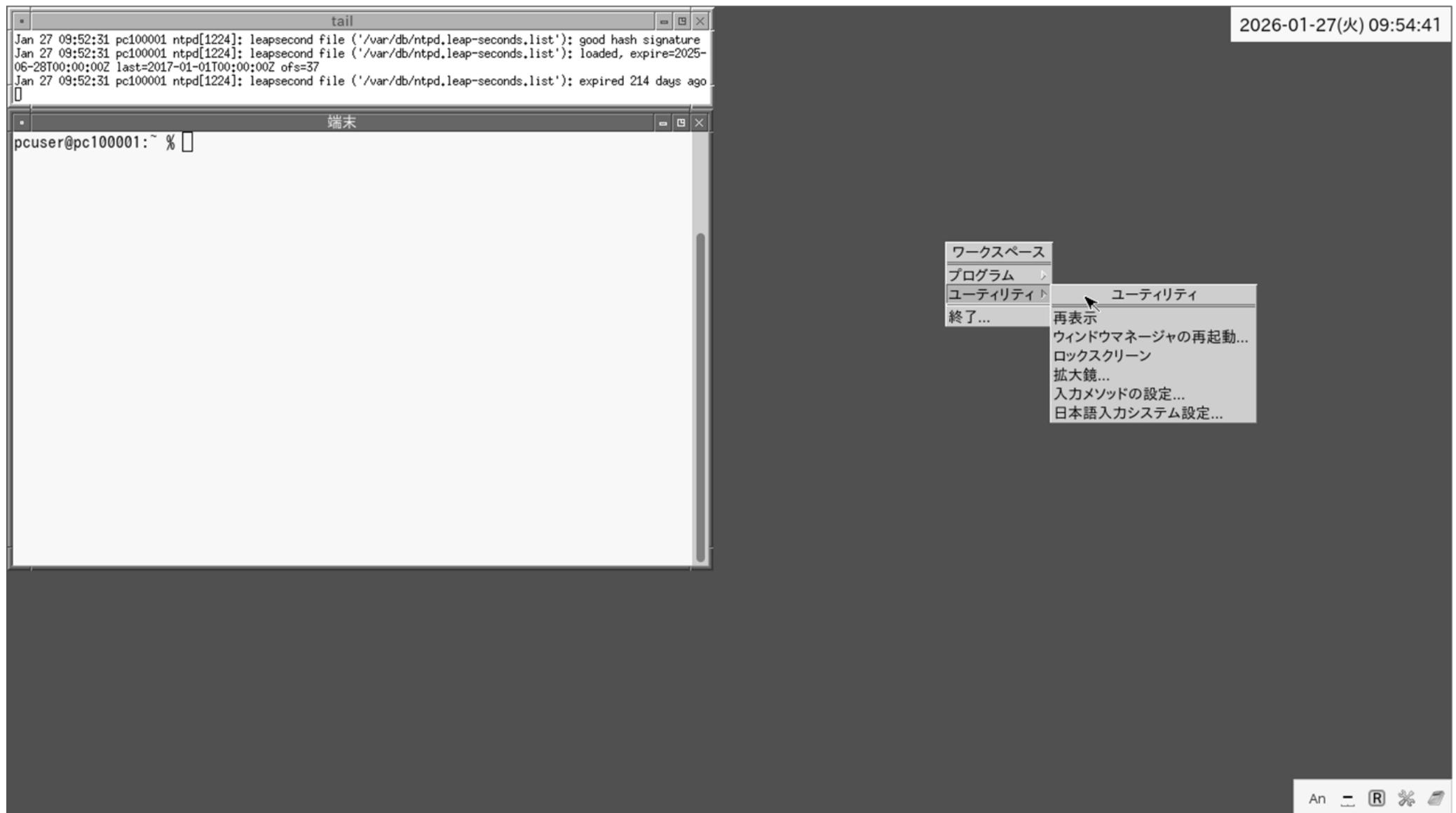
- ウィンドウシステムを起動する

```
% startx
```

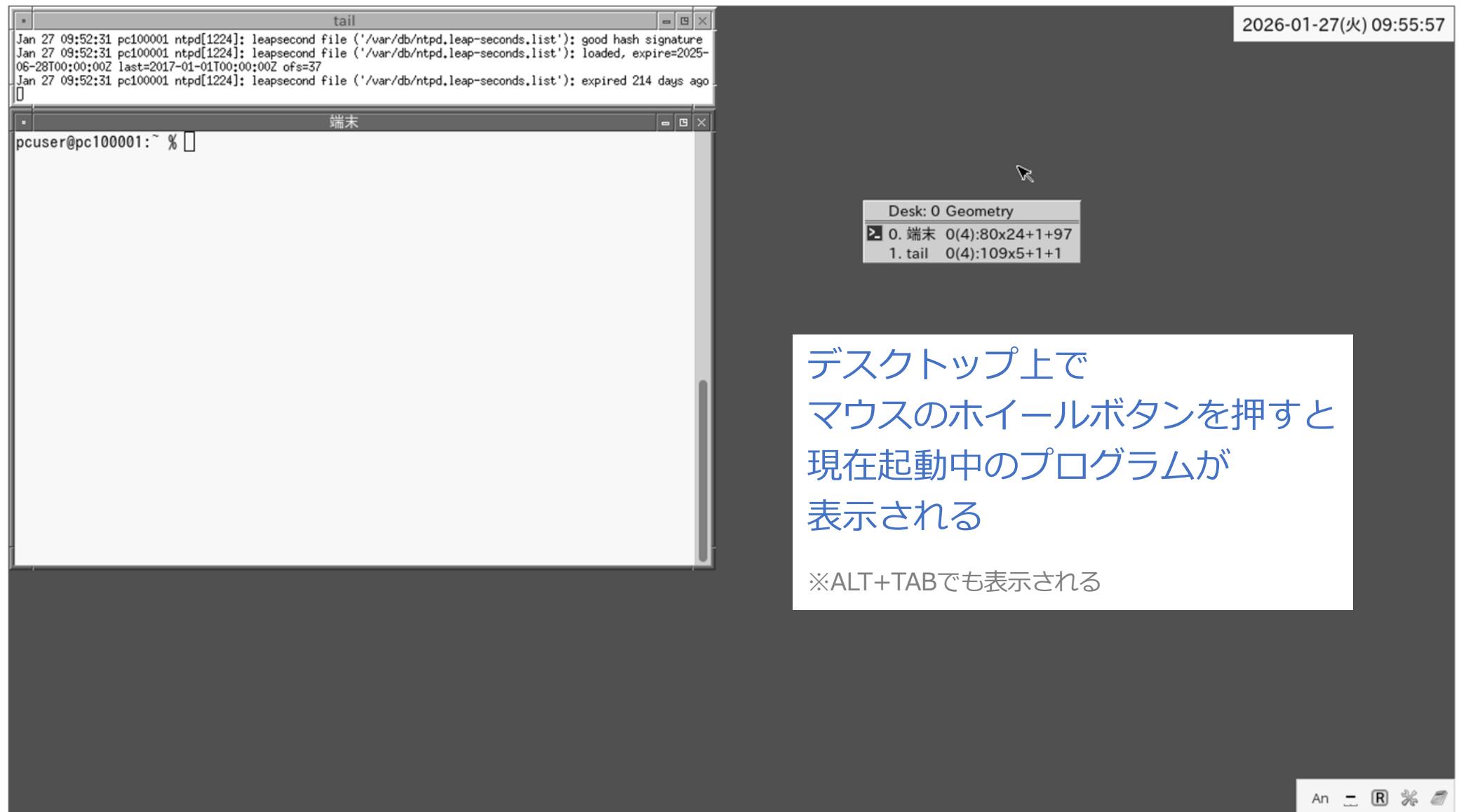
## 続き | 4. FreeBSDの起動～終了



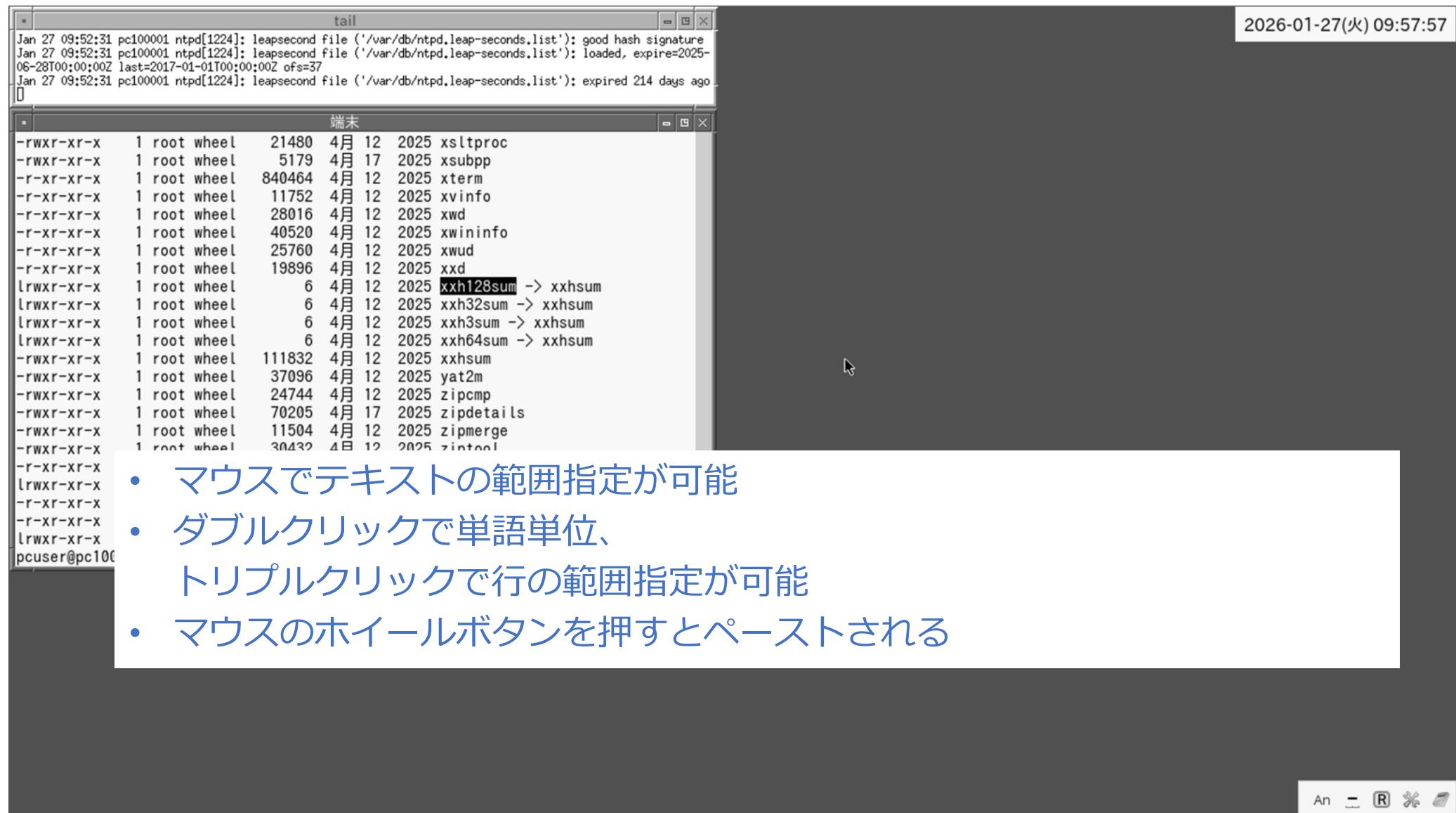
# 続き | 4. FreeBSDの起動～終了



## 続き | 4. FreeBSDの起動～終了

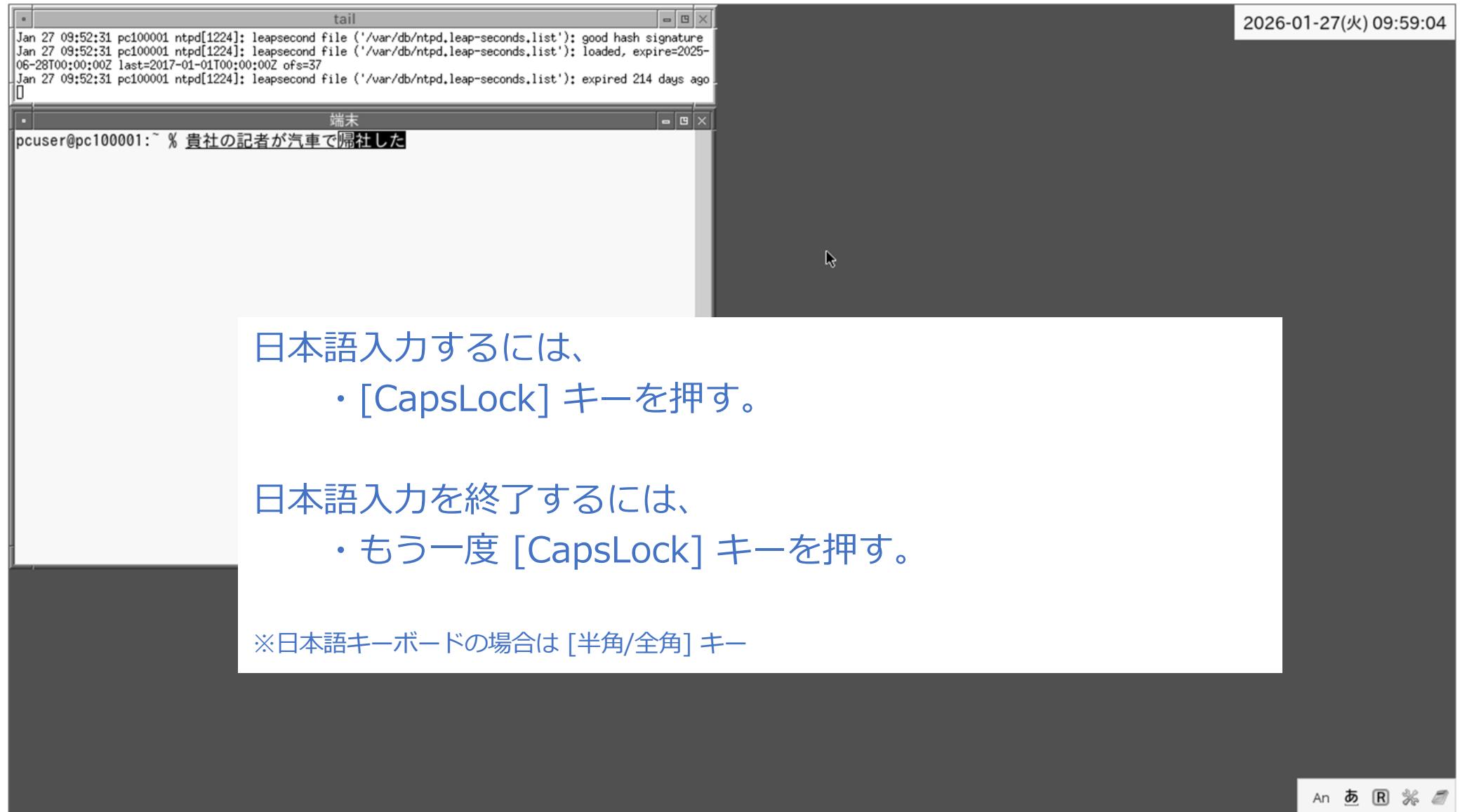


# 続き | 4. FreeBSDの起動～終了

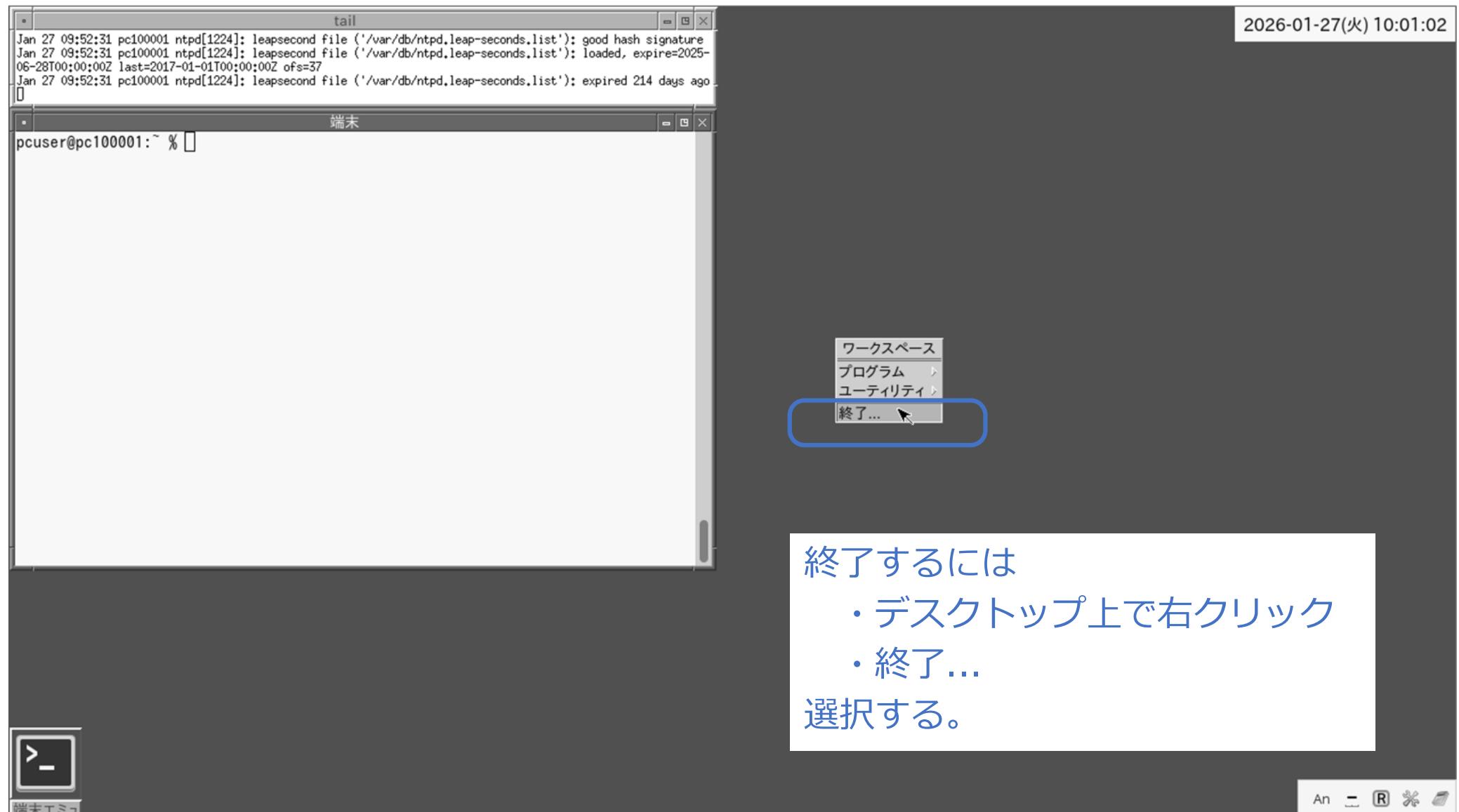


- マウスでテキストの範囲指定が可能
- ダブルクリックで単語単位、トリプルクリックで行の範囲指定が可能
- マウスのホイールボタンを押すとペーストされる

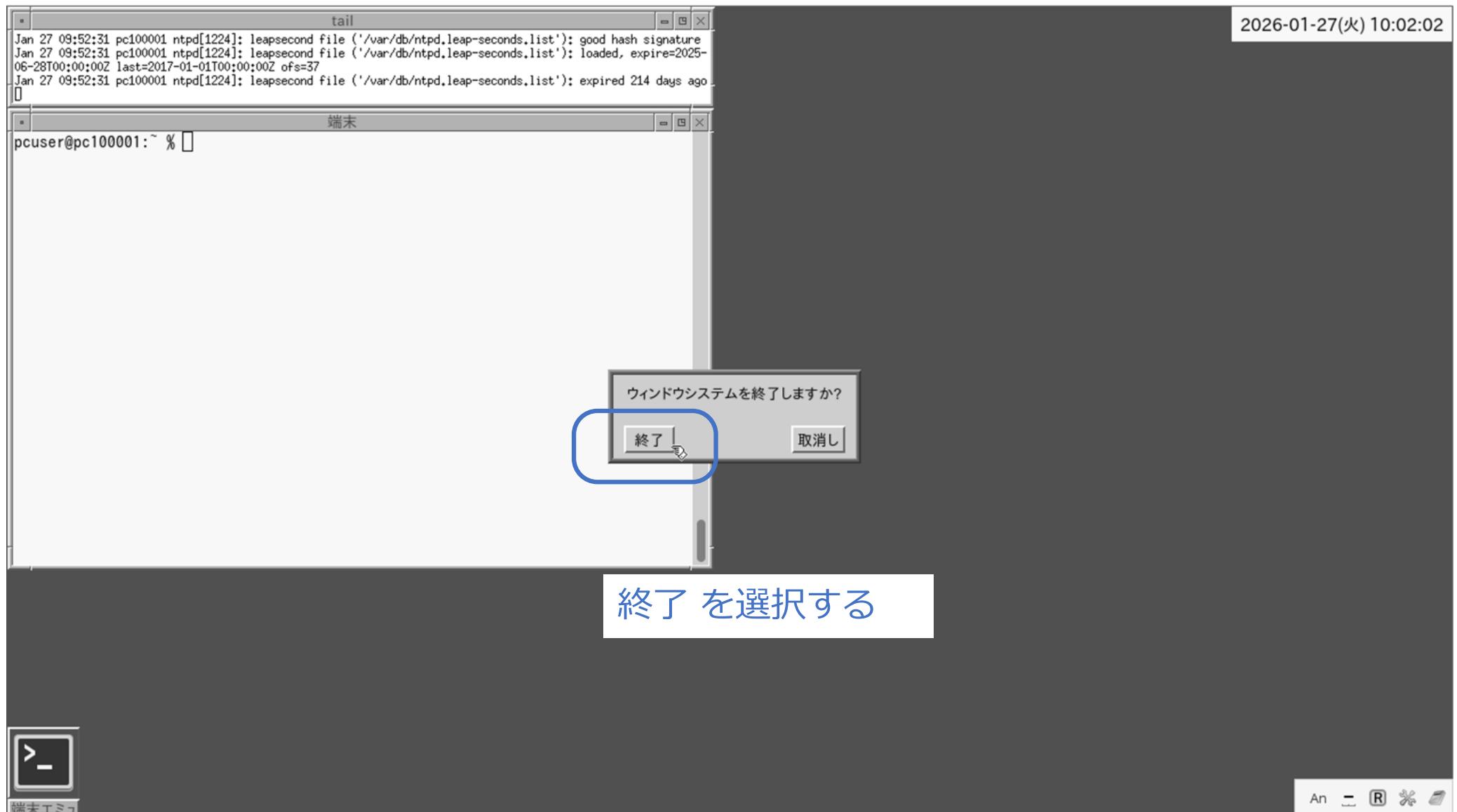
# 続き | 4. FreeBSDの起動～終了



## 続き | 4. FreeBSDの起動～終了



## 続き | 4. FreeBSDの起動～終了



## 続き | 4. FreeBSDの起動～終了

---

- FreeBSDを終了し、電源を切る

```
% sudo shutdown -p now  
Password: *****
```

以上で説明は終わりです。

## appendix. 今後の活用方法

1. デスクトップを右クリックすると表示されるメニューは、`~/.fvwm2rc` に記載されています。この部分をお好みの形に編集してみてください。
2. 画面のフォントについては `~/.fvwm2rc` を編集してみてください。
3. xfce4-terminal のカスタマイズは、端末ターミナル上で 右クリック で行えます。
4. `~/.xinitrc` に ウィンドウマネージャ起動時に実行されるプログラムが記載されています。この部分を編集して、どのように変化するかを確認してみてください。
5. 本ページ以降、「便利な使い方」を数多く紹介していますので、試してみてください。
6. ここまでインストール手順を自動化するシェルスクリプトを作つてみましょう。
7. このメモでは、細かいパラメーターなどについて説明を省いている部分があります。webや ChatGPTなどの情報を参考に理解を深めることをお勧めします。

## 5-1. 自動的にログインできるようにしたい

- 自動的にログインできるようにしたい (pcuserアカウントにて)

```
% sudo vi /etc/gettytab
```

```
pcuserautologin:¥  
:al=pcuser:tc=Pc:
```

タブ

最終行に追加する

```
% sudo vi /etc/ttys
```

```
ttyv0 "/usr/libexec/getty pcuserautologin" xterm on secure
```

ttyv0の行をこのように変更する

```
% sudo shutdown -r now
```

※ただし、ログアウトしても即ログインしてしまうという欠点がある。

再起動後、有効になる

## 5-2. ログイン後、自動的にウィンドウマネージャーを起動したい

```
% vim ~/.login  
  
if ( $?SSH_CLIENT == 0 && $?SSH_TTY == 0 && $?DISPLAY == 0 ) then  
    startx  
endif
```

最終行に追加する

## 5-3. 特定のコマンドは、パスワードなしでsudoを実行したい

```
% sudo visudo
```

```
pcuser ALL=NOPASSWD: /sbin/shutdown
```

← 最終行に追加する

ここではログイン名 pcuser に対して、shutdown コマンドを sudo のパスワードなしで実行できるように定義している。

```
% sudo shutdown -r now
```

この時点ですでに有効になっているはず

## 5-4. ログインした際のメッセージを、表示させない

- ログインした際のメッセージを、全て表示させない

```
% touch ~/.hushlogin
```

- ログインした際のメッセージを、Last login以外、表示させない

```
% sudo mv /etc/motd.template /etc/motd.template.org  
% sudo touch /etc/motd.template
```

```
% sudo shutdown -r now
```

再起動後有効になる

## 5-5. 起動時のブートメニュー やメッセージをできるだけ表示させない

以下、※BIOS/UEFIのブート設定がBIOSの場合のみ有効と思われます。

```
% sudo vi /boot.config  
-n mq } 新規に追加する
```

```
% sudo vi /boot/loader.conf  
autoboot_delay="-1" } 新規に追加する
```

```
% sudo shutdown -r now
```

再起動後有効になる

## 続き | 5-5. 起動時のブートメニュー やメッセージをできるだけ表示させない

※前ページまでの設定で「ある程度」表示されなくなるが、さらに抑制したい場合は下記設定を行う（劇的な効果はない）。

```
% sudo vi /etc/rc.conf
```

```
rc_startmsgs="NO"
```

} 追加する

## 5-6. IPアドレスを固定化したい (IPv4)

```
% sudo vi /etc/rc.conf
```

コメントアウトする

```
#ifconfig_em0="DHCP"
```

LANのインターフェース名

```
ifconfig_em0="inet 192.168.1.8/24"  
defaultrouter="192.168.1.1"
```

} 追加する

※上記の例では、IPアドレスを 192.168.1.8 に設定している

```
% sudo shutdown -r now
```

再起動後有効になる

## 5-7. ウィンドウシステム終了後、自動的に電源が切れるようにしたい

事前に、「5-3. 特定のコマンドは、パスワードなしでsudoを実行したい」を例の通りに設定しておく

```
% vim ~/.xinitrc
{
fvwm3
sudo shutdown -p now
```

先頭の `exec` を削除する

追加する

## 5-8. 起動時に、/tmp フォルダーをクリアーしたい

```
% sudo vi /etc/rc.conf
```

```
clear_tmp_enable="YES" ← 最終行に追加する
```

```
% sudo shutdown -r now ←
```

再起動後有効になる

## 5-9. 無線LANアクセスポイントを使いたい

```
% sudo vi /etc/rc.conf
```

```
#ifconfig_em0="DHCP"  
#ifconfig_em0="inet 192.168.1.8/24"  
#defaultrouter="192.168.1.1"
```

```
wlans_iwn0="wlan0"  
ifconfig_wlan0="country JP WPA SYNCDHCP"
```

} 最終行に追加する

有線LAN関連の設定をコメントアウトする。

## 続き | 5-9. 無線LANアクセスポイントを使いたい

```
% sudo vi /etc/wpa_supplicant.conf
```

```
network={  
    ssid="ここにSSIDを記述"  
    psk="ここにWi-Fiパスワードを記述"  
}
```

新規に追加する

```
% sudo chmod 600 /etc/wpa_supplicant.conf
```

パスワードが平文で見えててしまうので、root以外参照させないようにする。

```
% sudo shutdown -r now
```

再起動後有効になる

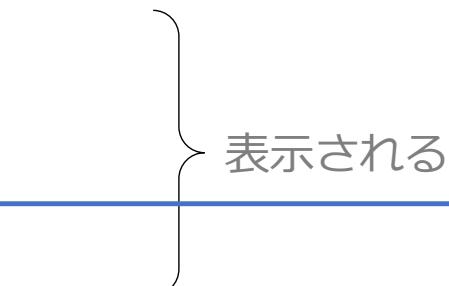
※必要に応じて、ファイヤーウォール (/etc/pf.conf) の設定も必要となります。

## 続き | 5-9. 無線LANアクセスポイントを使いたい

前ページにて、パスワードがそのまま見えてしまうのを防ぐ方法

% **wpa\_passphrase SSIDを記述 Wi-Fiパスワードを記述**

```
network={  
    ssid="SSID"  
    #psk="Wi-Fiパスワード"  
    psk=64文字のパスフレーズ  
}
```



表示される

% **sudo vi /etc/wpa\_supplicant.conf**

```
network={  
    ssid="SSID"  
    psk=64文字のパスフレーズ ←  
}
```

修正する  
(ダブルクオーテーション不要)

% **sudo shutdown -r now** ←

再起動後有効になる

## 5-10. パッケージの管理

- インストールしたパッケージを全て表示する

```
% pkg info
```

- 手動でインストールしたパッケージのみを表示する

```
% pkg query -e '%a = 0' '%o'
```

- pkg install 後に表示されるメッセージを再表示したい

```
% pkg info -D <パッケージ名>
```

- パッケージの依存関係を表示する

```
% pkg info -d -r <パッケージ名>
```

↑  
逆依存関係（このパッケージを必要としているもの）

↑  
依存関係（このパッケージを動作させるために必要なもの）

# 続き | 5-10. パッケージの管理

## ■ パッケージの削除

```
% sudo pkg delete <パッケージ名>
```

## ■ どのパッケージからも参照されていないパッケージを削除する

```
% sudo pkg autoremove
```

## ■ コマンド名から、どのパッケージに含まれるかを調べる（すでにインストールされている場合）

```
% pkg which `which <コマンド名>`
```

※パッケージの詳細（上級編）については、14-3. 14-4.などもご覧ください。

## 5-11. IPv6で接続したい

- IPv6の設定

```
% sudo vi /etc/rc.conf
```

↑ LANのインターフェース名

```
ifconfig_em0_ipv6="inet6 accept_rtadv" ← 最終行に追加する
```

- ファイアウォールの設定

```
% sudo vi /etc/pf.conf
```

:

```
pass inet6 proto icmp6 all keep state  
pass out inet6 all keep state
```

} 最終行に追加する

- 再起動

```
% sudo shutdown -r now
```

← 再起動後有効になる

## 5-12. ログインまで進めなくなった場合の対処方法（UFS）

① 起動直後のFreeBSDブートローダー画面において、「2. Boot Single user」を選択する。

② 「Enter full pathname or shell or RETURN for /bin/sh:」と表示されたら、Enterキーを押す。

③ ファイルシステムをマウントする

```
# mount -u /
# mount -a
```

④ 問題を調査する（例） ※日本語キーボードの場合、事前に `kbdcontrol -l jp.kbd` を実行する

```
# vi /etc/rc.conf
```

⑤ 再起動する

```
# shutdown -r now
```

## 続き | 5-12. ログインまで進めなくなつた場合の対処方法 (UFS)

- ・ブートローダーを非表示にしてしまつた場合 (UFSファイルシステム)

① FreeBSD14.3のインストールDVDなどで起動する。

② 「Welcome」表示画面にて、「Shell」を選択する。

③ ディスクの確認

※日本語キーボードの場合、事前に `kbdcontrol -l jp.kbd` を実行する

```
# gpart show | more
```

=>	0	123456789	ada0s1	BSD	(xxG)	} このような行を探す
	0	123456789	1	freebsd-ufs	(xxG)	
123456789	3456789	2	freebsd-swap	(x.xG)		
123456789	456789	-	free	-	(xxxM)	

④ マウントする

```
# mount /dev/ada0s1 /mnt
```

## 続き | 5-12. ログインまで進めなくなつた場合の対処方法 (UFS)

### ⑤ 問題を調査する（例）

```
# vi /mnt/etc/rc.conf
```

### ⑥ 再起動する

```
# shutdown -r now
```

## 5-13. ログインまで進めなくなった場合の対処方法（暗号化+ZFS）

① パソコン起動後、ディスク暗号化のパスワードを入力する。

② FreeBSDブートローダー画面において、「2. Boot Single user」を選択する。

③ 「Enter full pathname or shell or RETURN for /bin/sh:」と表示されたら、Enterキーを押す。

④ ファイルシステムを読み書き可能でマウントする

```
# mount -u -w /
```

⑤ 問題を調査する（例）

※日本語キーボードの場合、事前に `kbdcontrol -l jp.kbd` を実行する

```
# vi /etc/rc.conf
```

⑥ 再起動する

```
# shutdown -r now
```

- ・ブートローダーを非表示にしてしまつた場合（暗号化+ZFS）

- ① FreeBSD14.3のインストールDVDなどで起動する。
- ② ディスク暗号化のパスワードを入力するとともに、デバイス名をメモしておく。
- ③ FreeBSDのブートローダー画面では「Enter」キーを押す。
- ④ 「Welcome」表示画面にて、「Shell」を選択する。

disk?p3ならば  
ada0p3や  
nda0p3など

- ⑤ ディスク復号化

```
# geli attach /dev/ada0p3
Enter passphrase: *****
```

- ⑥ マウント

```
# zpool import -f -R /tmp/mnt zroot
# zfs mount zroot/ROOT/default
```

## 続き | 5-13. ログインまで進めなくなつた場合の対処方法（暗号化+ZFS）

- ⑦ 問題を調査する（例） ※日本語キーボードの場合、事前に `kbdcontrol -l jp.kbd` を実行する

```
# vi /tmp/mnt/etc/rc.conf
```

- ⑧ 再起動する

```
# shutdown -r now
```

## 5-14. システムログを画面上に表示させておきたい

SunViewやOpenWindowsのように、画面左上にコンソール画面を常時表示させておきたい場合があるかと思いますが、ここではシステムログの状況を常時表示させる設定を紹介します（すでに設定済みです）。

```
% vim ~/.xinitrc
:
xterm -geometry 109x5+1+1 -title コンソール -e tail -f /var/log/messages &
xfce4-terminal --geometry=80x24+1+97 --title="端末エミュレータ" &
:
```

ウィンドウシステムをいったん終了する



ウィンドウシステムを起動する

```
% startx
```

## 6-1. AC動作か、バッテリー動作かを確認したい。残量を確認したい

### ■ AC動作か、バッテリー動作かを確認したい

```
% sysctl hw.acpi.acline
```

```
hw.acpi.acline: 0
```

0であれば、バッテリー動作  
1であれば、AC動作

### ■ バッテリーの残量を確認したい

```
% sysctl hw.acpi.battery.life
```

```
hw.acpi.battery.life: 64
```

※仮想環境やデスクトップパソコンではエラーとなります

## 6-2. ノートパソコンの液晶バックライトの輝度を変更したい・確認したい

```
% backlight 16  
% backlight incr 2  
% backlight decr 2  
% backlight
```

※仮想環境やデスクトップパソコンではエラーとなります

## 6-3. マスター音量（音量）を確認・変更・ミュート／解除したい

```
% mixer  
pcm0:mixer: <Realtek ALC269 (Analog 2.0+HP/2.0)> on hdac0 (play/rec)  
(default)  
    vol      = 0.85:0.85      pbk  
    pcm      = 1.00:1.00      pbk  
    speaker  = 0.74:0.74      rec  
    mic      = 0.67:0.67      rec  
    mix      = 0.74:0.74      rec  
    rec      = 0.37:0.37      pbk  
    igain    = 0.00:0.00      pbk  
    ogain    = 1.00:1.00      pbk  
    monitor  = 0.67:0.67      rec src  
  
% mixer vol.volume=1.00          ← ボリュームを 1.00 にする  
% mixer vol.volume+=0.05        ← ボリュームを 0.05 上げる  
% mixer vol.volume=-0.05        ← ボリュームを 0.05 下げる  
% mixer vol.mute=^             ← ミュートする／解除する（トグル）  
% mixer vol.mute=0             ← ミュート解除  
% mixer vol.mute=1             ← ミュート  
  
% beep -D 1000                  ← 確認用ビープ音
```

※ appendix. 便利な使い方「9-4. キーボード上のボリュームキーにて音量調節をしたい」もご覧ください。

※ 音が出ない場合は、6-6.もご覧ください。

## 6-4. CPU動作周波数の確認をしたい

- ・今現在の動作クロックを確認する

```
% sysctl dev.cpu | grep freq:
```

## 6-5. ネットワークに接続したプリンタから印刷したい

ここでは、ネットワークに接続されたプリンタを使用する手順を紹介します。具体的に、A4モノクロページプリンタである「NEC MultiWriter 5350 (PR-L5350)」の例で説明します\*1。プリンタのIPアドレスは、192.168.1.201とします。

- ・CUPSをインストールする

↑  
プリンタのIPアドレスは固定しておいてください。

```
% sudo pkg install -y cups cups-filters
```

- ・起動できるように設定する

```
% sudo vi /etc/rc.conf
```

:

```
cupsd_enable="YES" ← 最終行に追記する
```

- ・再起動する

```
% sudo shutdown -r now
```

pcuserでログインし、stratxする。

\*1 有線LANネットワークに直接接続できる業務用ページ（レーザー）プリンタであれば、本設定を参考にすることで印刷できる可能性が高いです。安価なプリンターでは、そもそもFreeBSDから印刷できない可能性があります。

## 続き | 6-5. ネットワークに接続したプリンタから印刷したい

### ■汎用ドライバ（PPDファイル）を使用する例

- ドライバを登録する

```
% lpadmin -p mw5350_gen ¥ ← プリンター名  
      -E ¥ ← 有効化  
      -v socket://192.168.1.201 ¥ ← プリンタ接続URI  
      -m drv:///sample.drv/generic.ppd ← プリンタドライバー設定  
                                (CPUS付属汎用ドライバー)
```

- 設定されたことを確認する

```
% lpstat -p mw5350_gen
```

プリンター mw5350\_gen は待機中です。火 08/26 12:00:00 2025 以来有効です

```
% lpstat -v mw5350_gen
```

mw5350\_gen のデバイス: socket://192.168.1.201

## 続き | 6-5. ネットワークに接続したプリンタから印刷したい

- ・テスト印刷する

```
% /usr/local/bin/lp -d mw5350_gen /etc/hosts
```

↓  
プリンタ名

/usr/local/binのlpがcups用のコマンド（/usr/binにFreeBSD標準のlpコマンドがあるので注意！）。

- ・印刷されるかどうかを確認する。
- ・FireFoxなどのWebブラウザからもテスト印刷する。
- ・印刷されるかどうか、そして印刷品質を確認する。



文字がギザギザになっていると思います。  
次ページ以降で対策します。

## 続き | 6-5. ネットワークに接続したプリンタから印刷したい

### ■Windows用プリンタドライバーのPPDファイルを使用する例 \*1

- PPDファイルの入手（操作概要を説明）
- NEC のサイトから、MultiWriter 5350 の Windows 10用のドライバー（標準ドライバー・ユーティリティソフトウェアを含む）をダウンロードする（Ver D040）
- .zip ファイルを展開する
- start.exe を 7-Zip で展開する（13-13.を参照）
- 展開された、Install/Msi/neph15a.msi を、7-Zip でさらに展開する（13-13.を参照）
- Program Files/NEC/DriverTemp/HL15A/PS/64/JPN/neph5350.pp\_ を、さらに 7-Zip で展開する（13-13.を参照）
- neph5350.ppd ファイルが作成されるのを確認する
- ドライバ（PPDファイル）のコピー

```
% mkdir -p ~/.cups/model  
% cp neph5350.ppd ~/.cups/model/
```

\*1 他のプリンタでこの方法が使えるかどうかは不明です。また、Linux用CUPSプリンタドライバが rpm 形式提供されている場合があります。13-3.を参考に rpm ファイルを展開すると、.ppdファイルが存在する可能性があります。そのファイルを使用することで、設定ができるかもしれません。

# 続き | 6-5. ネットワークに接続したプリンタから印刷したい

- PPDファイルの書式確認

```
% cupstestppd ~/.cups/model/neph5350.ppd  
  
/home/pcuser/.cups/model/neph5350.ppd: 失敗  
-- **失敗** 不正な ShortNickName - 31 文字を超えてます。  
      参照: 64-65 ページ、セクション 5.3。  
-- 警告 非 Windows PPD ファイルは、CR LF でなく LF のみを行末に使うべき です。  
WARN サイズ "A5L" は Adobe 標準名称 "A5" であるべきです。  
WARN サイズ "MexicanLegal" は Adobe 標準名称 "215.9x339.73mm" である べきです。  
WARN サイズ "IndiaLegal" は Adobe 標準名称 "214.84x345.02mm" であるべきです。
```

- PPDファイルの改行をLFに変更する

```
% sudo pkg install -y ja-nkf  
% nkf -w -Lu ~/.cups/model/neph5350.ppd > ~/.cups/model/neph5350.ppd.new
```

- PPDファイルを修正する

```
% vim ~/.cups/model/neph5350.ppd.new  
:  
*ShortNickName: "NEC MultiWriter 5350 PS3 emu" ← emulateをemuに変更する  
:
```

## 続き | 6-5. ネットワークに接続したプリンタから印刷したい

- PPDファイル名を変更する

```
% mv ~/.cups/model/neph5350.ppd ~/.cups/model/neph5350.ppd.org  
% mv ~/.cups/model/neph5350.ppd.new ~/.cups/model/neph5350.ppd
```

- ドライバを登録する

```
% lpadmin -p mw5350  
      -E  
      -v socket://192.168.1.201  
      -P ~/.cups/model/neph5350.ppd
```

- 設定されたことを確認する

```
% lpstat -p mw5350
```

プリンター mw5350 は待機中です。火 08/26 12:00:00 2025 以来有効です

```
% lpstat -v mw5350
```

mw5350 のデバイス: socket://192.168.1.201

## 続き | 6-5. ネットワークに接続したプリンタから印刷したい

- FireFoxなどのWebブラウザからテスト印刷する。
- 印刷されるかどうか、そして印刷品質を確認する。



文字が綺麗になったと思います。

## 続き | 6-5. ネットワークに接続したプリンタから印刷したい

□よく使われるコマンド、他

- ・デフォルトプリンタとして設定する

```
% lpadmin -d mw5350
```

- ・プリンタ設定を削除する

```
% lpadmin -x mw5350_gen
```

- ・lpコマンドは /usr/local/bin を参照するようにする

```
% vim ~/.cshrc
:
alias lp    /usr/local/bin/lp  ← 追加する
:
```

## 6-6. スピーカーやイヤホン端子から音が出るようにしたい

FreeBSD では、Firefoxなどのブラウザからは特に設定を行わなくても音声は出力されますが、デフォルトの出力先が意図しないデバイスになっていることがあります。ここでは、その出力先を変更する方法について紹介します。（以下は、音声出力端子や内蔵スピーカーなど、複数の音声出力デバイスを持つデスクトップパソコンを想定した設定例です。）

### 音声デバイスの確認

```
% cat /dev/sndstat
```

Installed devices:

pcm0: <ATI R6xx (HDMI)> (play)

pcm1: <ATI R6xx (HDMI)> (play)

pcm2: <ATI R6xx (HDMI)> (play)

pcm3: <ATI R6xx (HDMI)> (play)

pcm4: <ATI R6xx (HDMI)> (play)

pcm5: <ATI R6xx (HDMI)> (play)

pcm6: <Realtek ALC662 rev3 (Rear Analog)> (play/rec) default ← デフォルト

pcm7: <Realtek ALC662 rev3 (Analog)> (play/rec)

pcm8: <Realtek ALC662 rev3 (Front Analog Headphones)> (play)

No devices installed from userspace.

本体背面（Rear Analog）側の出力端子がデフォルトになっています。

## 続き | 6-6. スピーカーやイヤホン端子から音が出るようにしたい

本体内蔵のスピーカー (pcm7) に出力先を変更

```
% sudo sysctl hw.snd.default_unit=7  
% beep
```

これでビープ音が本体内蔵スピーカーから出ると思います。

ただし、OSを再起動すると、設定が元に戻ってしまいます。下記のように固定します。

設定を固定

```
% sudo vi /etc/sysctl.conf  
  
hw.snd.default_unit=7 ← 最終行に追加する
```

再起動

```
% sudo shutdown -r now
```

## 6-7. NumLockを効かせたい

デスクトップパソコンにおいては、FreeBSDが起動すると、NumLockが外れてしまう場合があります。ウインドウマネージャ起動以降にNumLockを有効にするには下記のように設定します。

### インストール

```
% sudo pkg install -y numlockx
```

### ウインドウマネージャ起動時に、NumLockをオンに設定する

```
% vim ~/.xinitrc
```

:

```
numlockx ←
```

この位置に追加する

```
exec fvwm3
```

## 続き | 6-7. NumLockを効かせたい

ウィンドウシステムをいったん終了する



ウィンドウシステムを起動する

```
% startx
```

NumLockが効いていると思います。

## 7-1. リモートからログイン、ファイル転送したい (SSH、SCP)

このページでは、FreeBSDをインストールしたパソコン以外からの操作を説明しています。また、FreeBSDをインストールしたパソコンのIPアドレスは、192.168.1.8 としています。

### ■ SSHでログインしたい

```
remotehost% ssh pcuser@192.168.1.8
(pcuser@192.168.1.8) Password for pcuser@pc100001:*****
```

### ■ ファイルを受信したい (get)

```
remotehost% scp pcuser@192.168.1.8:filename.txt .
(pcuser@192.168.1.8) Password for pcuser@pc100001:*****
```

### ■ ディレクトリを受信したい (get)

```
remotehost% scp -r pcuser@192.168.1.8:dirname .
(pcuser@192.168.1.8) Password for pcuser@pc100001:*****
```

## 続き | 7-1. リモートからログイン、ファイル転送したい (SSH、SCP)

このページでは、FreeBSDをインストールしたパソコン以外からの操作を説明しています。また、FreeBSDをインストールしたパソコンのIPアドレスは、192.168.1.8 としています。

### ■ ファイルを送信したい (put)

```
remotehost% scp filename.txt pcuser@192.168.1.8:./
(pcuser@192.168.1.8) Password for pcuser@pc100001:*****
```

### ■ ディレクトリを送信したい (put)

```
remotehost% scp -r dirname pcuser@192.168.1.8:./
(pcuser@192.168.1.8) Password for pcuser@pc100001:*****
```

※このページ、および前ページのコマンドは、/etc/ssh/sshd\_config の PasswordAuthentication の値が no に設定されない限り利用できる。

## 7-2. 公開鍵認証で、SSH接続したい

ここでは Mac から FreeBSD に公開鍵認証でSSH接続する例となります。

### ① FreeBSD側の作業（公開鍵と秘密鍵を生成する）

```
pcuser@pc100001% ssh-keygen
Enter file in which to save the key (/home/pcuser/.ssh/id_rsa): ← Enterを押す
Created directory '/home/pcuser/.ssh'.
Enter passphrase (empty for no passphrase): ← Enterを押す
Enter same passphrase again: ← Enterを押す
Your identification has been saved in /home/pcuser/.ssh/id_rsa
Your public key has been saved in /home/pcuser/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:***** pcuser@pc100001
The key's randomart image is:
+---[RSA 3072]---+
:
+---[SHA256]---+
pcuser@pc100001% cd ~/.ssh
pcuser@pc100001% cat id_rsa.pub >> authorized_keys
```

## 続き | 7-2. 公開鍵認証で、SSH接続したい

### ② Mac側の作業 (FreeBSD側で生成された秘密鍵をMac側にコピーする)

```
mac@mac100001% mkdir ~/.ssh
mac@mac100001% chmod 700 ~/.ssh
mac@mac100001% scp pcuser@192.168.1.8:/home/pcuser/.ssh/id_rsa ~/.ssh/
The authenticity of host '192.168.1.8 (192.168.1.8)' can't be established.
ED25519 key fingerprint is SHA256:*****.
This key is not known by any other names

Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.8' (ED25519) to the list of known hosts.
(pcuser@192.168.1.8) Password for pcuser@pc100001: *****
```

ここでは FreeBSD がインストールされたパソコンのIPアドレスを 192.168.1.8 としています。

## 続き | 7-2. 公開鍵認証で、SSH接続したい

### ③ FreeBSD側の作業（公開鍵認証にしたので、パスワード認証を禁止にする）

```
pcuser@pc100001% sudo vi /etc/ssh/sshd_config
```

```
PasswordAuthentication no  
KbdInteractiveAuthentication no
```

} 最終行に追加する

### ④ 再起動する

```
pcuser@pc100001% sudo shutdown -r now
```

↑  
再起動後有効になる

## 続き | 7-2. 公開鍵認証で、SSH接続したい

### ⑤ MacからFreeBSDにSSH接続できることを確認する

```
mac@mac100001% ssh pcuser@192.168.1.8
Last login: Tue Aug 26 08:26:00 2026 from :0
FreeBSD 14.3-RELEASE-p7 (GENERIC)
pcuser@pc100001:~ %
```

ここでは FreeBSD がインストールされたパソコンのIPアドレスを 192.168.1.8 としています。

## 7-3. WindowsやMacとファイル共有したい (SMB)

### ① インストール

```
% sudo pkg install -y samba416
```

### ② パソコン起動時の設定

```
% sudo vi /etc/rc.conf
```

  
samba\_server\_enable="YES"

← 最終行に追加する

### ③ 共有フォルダーの作成

```
% mkdir ~/share
```

## 続き | 7-3. WindowsやMacとファイル共有したい (SMB)

### ④ 設定ファイルの新規作成

```
% sudo vi /usr/local/etc/smb4.conf
```

```
[global]
```

```
security = user
```

```
[pcuser_share]
```

```
path = /home/pcuser/share
```

```
browsable = Yes
```

```
read only = No
```

この名前が共有名となる

新規に追加する

共有フォルダー

↑  
スペース

### ⑤ smbアカウントの設定

```
% sudo pdbedit -a -u pcuser
```

```
new password: *****
```

```
retype new password: *****
```

ここでは共有専用のパスワードを  
設定している

## 続き | 7-3. WindowsやMacとファイル共有したい (SMB)

### ⑥ ファイアウォールの設定

```
% sudo vi /etc/pf.conf
:
pass in on em0 proto tcp to port { 22, 445 } keep state
```

追記する (445番がSMB)

### ⑦ 再起動

```
% sudo shutdown -r now
```

再起動後有効になる

## 続き | 7-3. WindowsやMacとファイル共有したい (SMB)

- Windowsから共有サーバーに接続する

- ① エクスプローラーを起動する
- ② アドレス欄に ¥¥192.168.1.8 と入力する
- ③ ネットワーク資格情報の入力画面にて、ユーザー名に「pcuser」、パスワードにパスワードを入力し、OKを押す。

このパスワードは pdbedit で入力したものに入力する

- Macから共有サーバーに接続する

- ① Finder メニュー「移動」⇒「サーバへ接続...」を選択する。
- ② アドレス欄に「smb://192.168.1.8」と入力し、「接続」を押す。
- ③ サーバ"192.168.1.8"に接続しようとしていると表示された場合、「接続」をクリックする。
- ④ 名前:欄に「pcuser」、パスワード:欄にパスワードを入力し、「接続」を押す。

※MacのFinderから、ドットファイルを参照するには、「shift」+「command」+「.」を押してください。

※FreeBSDをインストールしたパソコンのIPアドレスは、192.168.1.8としています

## 7-4. 他のパソコンからリモートデスクトップでFreeBSDに接続したい

- ① xrdp（リモートデスクトップ）をインストールする。

```
% sudo pkg install -y xrdp
```

- ② パソコン起動時の設定

```
% sudo vi /etc/rc.conf
```

```
xrdp_enable="YES"
```

```
xrdp_sesman_enable="YES"
```



最終行に追加する

## 続き | 7-4. 他のパソコンからリモートデスクトップでFreeBSDに接続したい

### ③ 起動ファイルを設定する

```
% cp ~/.xinitrc ~/startwm.sh  
% chmod 700 ~/startwm.sh
```

### ④ ファイアウォールの設定

```
% sudo vi /etc/pf.conf  
:  
pass in on em0 proto tcp to port { 22, 445, 3389 } keep state
```

追記する (3389番がリモートデスクトップ)

### ⑤ 再起動する

```
% sudo shutdown -r now
```

再起動後有効になる

### ⑥ 他のパソコンからFreeBSDへリモートデスクトップで接続する。

## 続き | 7-4. 他のパソコンからリモートデスクトップでFreeBSDに接続したい

下記の操作は、リモートデスクトップ上の xfce4-terminal で作業を行なってください。

### ⑦ キー設定を行う ※1

※ 日本語キーボードの場合は、設定不要と思われる

```
% setxkbmap -print > ~/.xkb/keymap/rdpkbd
```

```
% vim ~/.xkb/keymap/rdpkbd
```

```
xkb_keymap {  
    xkb_keycodes { include "xfree86+aliases(qwerty)" };  
    xkb_types { include "complete" };  
    xkb_compat { include "complete" };  
    xkb_symbols { include "pc+us+myuser(userkeys)" };  
    xkb_geometry { include "pc(pc104)" };  
};
```

追加する

※1：当方、Mac の英語キーボードにて Windows Appより FreeBSD にログインしたところ、カーソルキーの挙動がおかしかったために、本設定を行った。よってこの編集をする際、カーソルを移動させるには「h」「j」「k」「l」キーを使ってください。

### ⑧ キー設定を有効にする

※ 日本語キーボードの場合は、設定不要と思われる

```
% vim ~/startwm.sh  
{  
xset m 5 3  
xkbcomp -I$HOME/.xkb ~/ .xkb/keymap/ rdpkbd $DISPLAY  
}  
↑  
修正する
```

⑨ リモートデスクトップを終了する。（ワークスペース→終了...）

⑩ もう一度、リモートデスクトップで接続し、キーが正しく入力できることを確認する。

## 続き | 7-4. 他のパソコンからリモートデスクトップでFreeBSDに接続したい

### 【参考メモ】リモートデスクトップ接続にて苦労した設定一覧

問題	対応
マウスをクリックしても、ウィンドウがフォーカスされない。 (致命的)	.fvwm2rc に IgnoreModifiers L25 を追加して対応済み。
fvwm3 の config を参考にして作成した閉じるボタン「x」の右上が1ドット欠ける。	.fvwm2rc の "閉じるボタン"の部分にて対応済み。
一部のパスが設定されない。	.cshrc のパス設定のコメントを外して、明示的に設定するようにした。
カーソルキーの動きがおかしい。	本マニュアルにて設定方法を記載済み。
便利な使い方9-2.にて設定した xload のアイコンをクリックしても、conkyコマンドが起動しない。	(未解決)
[CapsLock]キーで日本語入力ON/OFFの切り替えができなくなることがある。	xkbcomp -I\$HOME/.xkb ~/ .xkb/keymap/rdpkbd \$DISPLAY を手動で実行する。

## 7-5. FreeBSDから、Windowsにリモートデスクトップで接続したい

### ① Windows側の設定（下記は Windows10 Pro の例）

- スタートメニュー → 設定
- システム
- リモートデスクトップ
- リモート デスクトップ

リモート デスクトップを使用すると、リモート デスクトップ クライアントアプリ（Windows、Android、iOS、macOS で利用可能）を使用してリモート デバイスからこの PC に接続して制御できます。この PC で直接作業しているかのように、別のデバイスで作業できます。

リモート デスクトップを有効にする



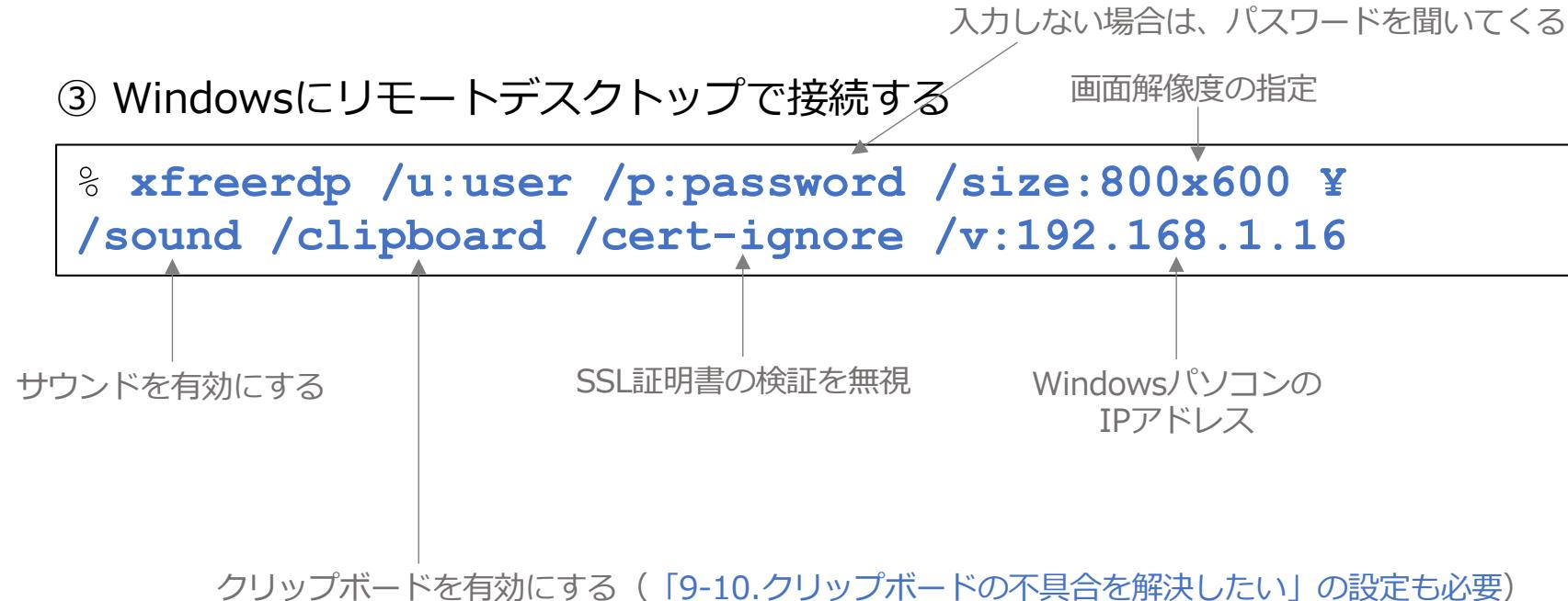
オンにする

## 続き | 7-5. FreeBSDから、Windowsにリモートデスクトップで接続したい

### FreeBSD側の設定

- ② xfreerdp（リモートデスクトップクライアント）をインストールする。

```
% sudo pkg install -y freerdp
```



## 続き | 7-5. FreeBSDから、Windowsにリモートデスクトップで接続したい

※注記（英語キーボードをお使いの方へ）

Windows側で autohotkey (Ver1系) を使用し、[CapsLock] を [半角/全角] に設定変更している場合、リモートデスクトップ接続時に [CapsLock] を押すと、IMEが瞬時にON→OFF してしまいます。これを防ぐには、autohotkey の設定を変更します。

※変更前 AutoHotkey Script.ahk

```
CapsLock::send, {vkF3sc029}
```

※変更後 AutoHotkey Script.ahk

← 通常、ユーザーのスタートアップフォルダー\*1へ入れておく

```
*CapsLock Up::send, {vkF3sc029}
```

\*1 C:\Users\ユーザー名\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup

補足：リモートデスクトップクライアントの Remminaアプリでは、このような現象は発生しない。

## 続き | 7-5. FreeBSDから、Windowsにリモートデスクトップで接続したい

※注記 (.fvwm2rcの設定について)

下記設定が有効になつてはいる、リモートデスクトップ内のウインドウ操作ができなくな  
るため、コメントアウトしてください。

```
# ALT+左クリックでウインドウを移動する(タイトルバーがないウインドウ対策)  
Mouse 1 W M Move
```



```
# ALT+左クリックでウインドウを移動する(タイトルバーがないウインドウ対策)  
#Mouse 1 W M Move
```

## 7-6. シフトJIS表示の機器類にtelnetでログインしたい

ここでは古いネットワークルーターなど、シフトJIS表示の機器類にTelnetでログインする方法を解説します。※mlterm用です。インストールは8-25.を参照。

### ① mltermの設定

```
% vim ~/.mlterm/aafont
```

```
ISO8859_1=IPAGothic ← 最終行に追加する  
JISX0208_1983=IPAGothic ←
```

### ② 接続

```
% mlterm -km=Shift_JIS ← 新しいウインドウが表示される
```

```
% telnet -8 192.168.1.1 ← 接続先を 192.168.1.1 の例で説明。
```

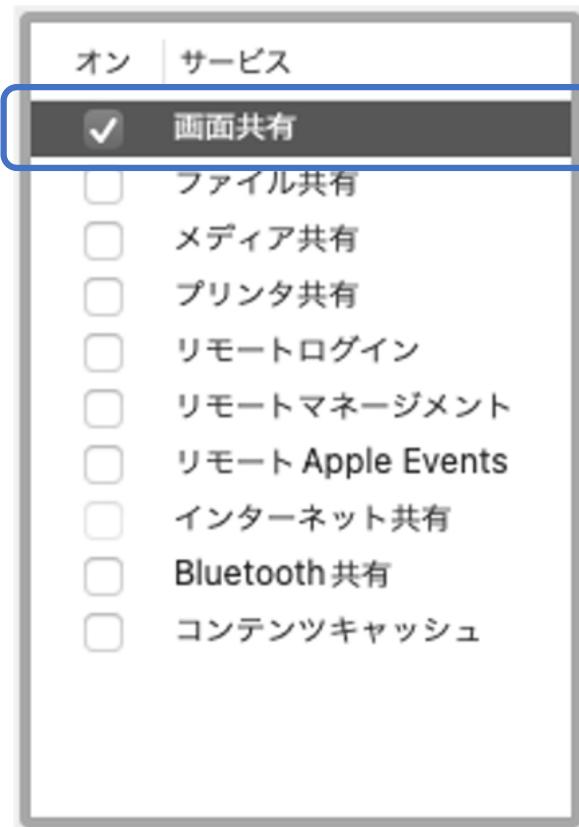
他にも、①を設定すれば、mlterm起動後に CTRL+マウス右ボタン で エンコーディング を SJIS にし、適用すれば表示可能。

## 7-7. FreeBSDから、MacにVNC接続したい

### ① Mac側の設定（下記は macOS Monterey 12 の例）

- アップルメニュー → システム環境設定
- 共有

macOS Ventura 13 では、  
システム設定  
macOS Ventura 13 では、  
一般 → 共有



オンにする

## 続き | 7-7. FreeBSDから、MacにVNC接続したい

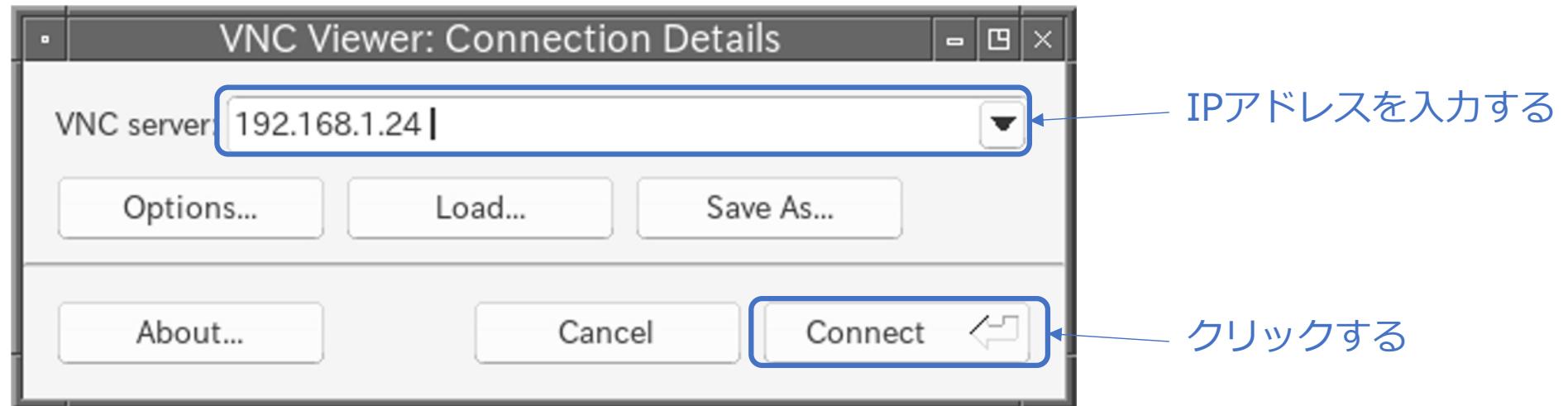
### FreeBSD側の設定

- ② TigerVNC (VNCクライアント) をインストールする。

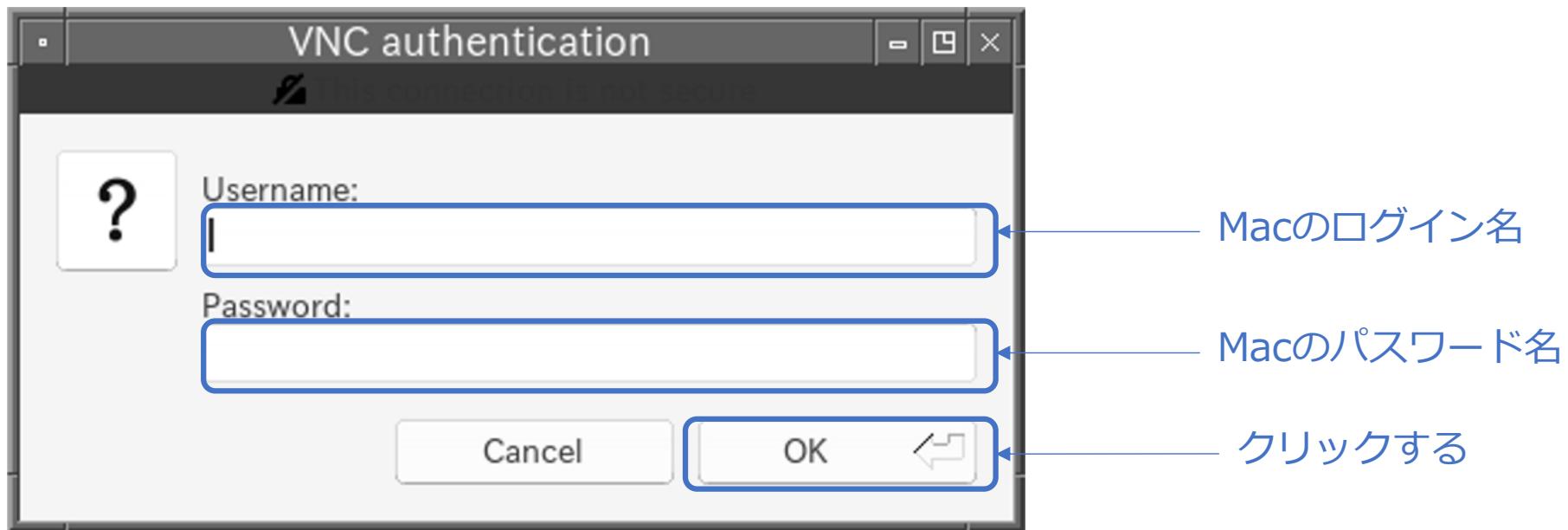
```
% sudo pkg install -y tigervnc-viewer
```

- ③ MacにVNC接続する

```
% vncviewer
```



## 続き | 7-7. FreeBSDから、MacにVNC接続したい



※このあと、画面上に非常に小さいウインドウ？（見た目はゴミ）や幅の無いウインドウ？が表示される場合があるが、隅の方をドラッグしてウインドウを拡大してください。

※ Mac側で英語キーボードを使い、Caps LockでIME ON/OFFにしている場合、TigerVNC経由で接続してもON/OFFしない。

## 7-8. テスト用FTPサーバーを立てたい

他のパソコンからFTP（SFTP）を使うなら、7-10.を参照。

### FTPサーバーの起動設定

```
% sudo vi /etc/rc.conf  
:  
ftpd_enable="YES"
```

### 再起動する

```
% sudo shutdown -r now
```

### FreeBSDからテスト接続する

```
% ftp localhost  
Name (localhost:pcuser): pcuser  
Password: *****
```

## 7-9. FTPにて、FreeBSDパソコンとサーバー間のファイルを同期したい

### lftpのインストール

```
% sudo pkg install -y lftp
```

同期する（ホームページなど、ローカルファイルを基準にリモートファイルを同期する例）

```
% lftp -u username ftp.example.com  
-e "mirror --reverse --delete ./localweb ./remoteweb; bye"
```

パスワード: \*\*\*\*\*

サーバー側のユーザー名

サーバーのアドレス

同期後、シェルに戻る

同期する対象のディレクトリ名（サーバー側）

同期する対象のディレクトリ名（FreeBSDパソコン側）

FreeBSDパソコン側に無いファイルをサーバーから削除

ミラーリングの方向をFreeBSDパソコン → サーバーとする

## 続き | 7-9. FTPにて、FreeBSDパソコンとサーバー間のファイルを同期したい

- ワイルドカードを用いてファイルをサーバーへアップロードしたい

アップロード (例)

```
% lftp -u username,password ftp.example.com
      -e "cd ./remoteweb; mput *.html; mput *.jpg; bye"
```

サーバー側のユーザー名・  
パスワード

サーバーのアドレス

アップロード後、シェルに戻る

拡張子が.jpgのファイルをアップロードする

拡張子が.htmlのファイルをアップロードする

アップロード対象のサーバー側ディレクトリ名

## 7-10. 家庭内FTPサーバーを立てたい（SFTP編）

このマニュアルではFreeBSDインストールの際に、sshdをインストール／起動していますので、そのままSFTPが使用できます。

他のパソコンより接続する（Windowsのコマンドプロンプト、Macのターミナルから）

```
> sftp pcuser@192.168.1.8 ← FreeBSDパソコンを192.168.1.8としています。
```

```
ED25519 key fingerprint is SHA256:*****  
This key is not known by any other names.
```

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

```
(pcuser@192.168.1.8) Password for pcuser@pc100001: *****
```

```
Connected to 192.168.1.8.
```

```
sftp> ← コマンドはFTPと同じです（説明は省略）
```

} 初回接続時に表示 \*1

\*1 7-2.において、ssh接続を公開鍵方式にし、パスワード認証を禁止した場合には、sftpでもその設定は有効となり、正しく設定されていればパスワードは聞かれません。

※ WindowsのフリーソフトWINSCPからも接続できます。（FFFTPは不可）

## 7-11. 家庭内Webサーバーを立てたい (nginx)

サーバー上の index.html を表示させるように設定します。

エンジンエックス  
nginxのインストール

```
% sudo pkg install -y nginx
```

起動設定

```
% sudo vi /etc/rc.conf
```

```
nginx_enable="YES" ← 最終行に追加する
```

公開するディレクトリの作成

```
% sudo mkdir -p /home/www
```

## 続き | 7-11. 家庭内Webサーバーを立てたい (nginx)

公開するディレクトリを pcuser から読み書きできるように、また nginx (www) から読み取りできるようにする。

```
% sudo pw groupmod www -m pcuser ← pcuserをwwwグループに所属させる  
% sudo chown pcuser:www /home/www  
% chmod 750 /home/www  
% sudo chmod g+s /home/www ← pcuserがこのディレクトリ以下でファイルを作成すると、自動的にグループwwwが付与される設定
```

テスト用の index.html ファイルを作成する

```
% vim /home/www/index.html  
  
<h1>Hello World!</h1> ← 新規に追加する
```

# 続き | 7-11. 家庭内Webサーバーを立てたい (nginx)

## nginx設定ファイルの編集

```
% sudo mv /usr/local/etc/nginx/nginx.conf /usr/local/etc/nginx/nginx.conf.org  
% sudo vi /usr/local/etc/nginx/nginx.conf  
  
worker_processes 1; ← 実際の通信処理を行うプロセスを1つとする  
  
events {}  
  
http {  
    server {  
        listen 80; ← 80番ポートを利用する  
        server_name _; ← どんなホスト名でも受け付ける *1  
  
        root /home/www; ← htmlファイルのディレクトリ  
        index index.html; ← デフォルトのファイル指定 *2  
    }  
}
```

新規に追加する

\*1 server\_name Webブラウザからアクセスされた時のURLの「ホスト名」の部分（例：localhost や 192.168.1.8）を通常は指定する。「\_」（アンダーバー）は、どんなホスト名でも受け付ける。

http://localhost  
http://192.168.1.8

\*2 Webブラウザからhtmlファイルの指定なしにアクセスされた時（例：http://192.168.1.8/）に、デフォルトで読み込むファイル名を指定する。

## 続き | 7-11. 家庭内Webサーバーを立てたい (nginx)

### ファイアウォールの設定

```
% sudo vi /etc/pf.conf
:
pass in on em0 proto tcp to port { 22, 80, 445, 3389 } keep state
```

追記する (80番がWebサーバー)

### 再起動

```
% sudo shutdown -r now
```

## 続き | 7-11. 家庭内Webサーバーを立てたい (nginx)

Webブラウザから確認する

**http://localhost**

他のパソコンのWebブラウザーから確認する

**http://192.168.1.8**



FreeBSDのパソコンを 192.168.1.8 としています

## 8-1. Firefoxで、ローカルファイルをブラウズしたい

```
% firefox .
```

※画像ファイルやPDFファイルなどが閲覧可能。

## 8-2. ハングル文字や簡体字・繁体字、絵文字を表示させたい

```
% sudo pkg install -y noto-sans-jp noto-emoji
```

## 8-3. Firefoxで、ダウンロードフォルダーを「~/Downloads」に変更したい

- Firefoxで、ダウンロードフォルダーを「~/Downloads」に変更したい

```
% mkdir ~/Downloads  
% firefox
```

1. 画面右上のアプリケーションメニュー「☰」→「設定」を選択する。
2. スクロールし、「ダウンロード 次のフォルダーに保存する(V)」の「参照...(O)」ボタンをクリックする。
3. pcuser の Downloads フォルダーを選択し、「開く(O)」ボタンをクリックする。

## 8-4. 付箋アプリ（Xpad）を使いたい

（※インストール済みです）

ワークスペース→プログラム→付箋... を選択する

- ・起動時にxpadを起動させるようにしたい

```
% vim ~/.xinitrc
{
xfce4-terminal  . . .
xpad -s &
}
fvwm3
```

この位置に追加する

## 続き | 8-4. 付箋アプリ (Xpad) を使いたい

- ・ウインドウ枠を表示させない。

右クリック→View→Window Decorationsのチェックを外す

- ・付箋の初期フォントや背景色などを設定したい

右クリック→Edit→設定(P)

- ・付箋ごとにフォントや背景色などの変更を行いたい

右クリック→Pad→プロパティ(P)

背景色参考

[黄#F9F89D] [青#B8F3FF] [緑 #B6FFA2] [赤 #F9C7C7] [紫 #BDC6FE] [灰 #EEEEEE] )

- ・付箋の大きさを変更したい

付箋右下部分をドラッグする

- ・付箋を移動したい

CTRLキーを押しながらドラッグする。または付箋の上部をドラッグする。

## 8-5. Firefoxを起動し、radikoでラジオNIKKEI第1を自動的に再生したい

※ 事前に一度だけFirefoxでradikoのページにアクセスし、適当な放送局を聴き、アンケートに答え、Firefoxを終了してください。

- GUI自動化ツールのインストール

```
% sudo pkg install -y xdotool
```

- シェルスクリプトの作成

```
% vim ~/rn1.tcsh

#!/bin/tcsh
firefox "https://radiko.jp/¥#!/live/RN1" &
sleep 10
set wid=`xdotool search --onlyvisible --name Firefox` 
xdotool windowactivate $wid
xdotool key Tab
sleep 1
xdotool key Tab
sleep 1
xdotool key Tab
sleep 1
xdotool key Return
```

- 実行権限の付与と実行

```
% chmod +x ~/rn1.tcsh
% ~/rn1.tcsh
```

※うまく動作しない場合は、Firefoxの画面を最大化するなどする。

## 続き | 8-5. Firefoxを起動し、radikoでラジオNIKKEI第1を自動的に再生したい

- 前ページの画面のまま、ラジオNIKKEI第2に切り替えたい

※ radiko の画面が表示されていることが前提。

- ・シェルスクリプトの作成

```
% vim ~/rn2.tcsh

#!/bin/tcsh
set wid=`xdotool search --onlyvisible --name Firefox`
xdotool windowactivate $wid
xdotool key Ctrl+l
sleep 1
xdotool type https://radiko.jp/¥#!live/RN2
sleep 1
xdotool key Return
sleep 5
xdotool key Tab
sleep 1
xdotool key Tab
sleep 1
xdotool key Tab
sleep 1
xdotool key Return
```

- ・実行権限の付与と実行

```
% chmod +x ~/rn2.tcsh
% ~/rn2.tcsh
```

※うまく動作しない場合は、Firefoxの画面を最大化するなどする。

## 8-6. Chromium（ウェブブラウザ）を使用したい

```
% sudo pkg install -y chromium webfonts
```

等幅フォントを正しく表示させるために必要 \*1

```
% chrome &
```

\*1 HTMLの<code>タグで思い通りのフォントが表示されていないので追加した。なお、Firefoxは webfonts をインストールしなくても思い通りの表示だった。

## 続き | 8-6. Chromium (ウェブブラウザ) を使用したい

- 座標、サイズを指定して Chromium を起動したい

```
% chrome --window-position=128,64 --window-size=800,600 &
```

y座標は適宜調整する

タイトルレバーやウインドウ枠の内側の座標やサイズを指定します。

- Chromiumを2画面表示で起動したい（画面解像度1366x768の例）

```
% chrome --window-position=5,14 --window-size=673,739 &
% chrome --new-window -window-position=688,14 &
  -window-size=673,739 -user-data-dir=/tmp/chrome1 &
```

同時に2つ以上のChromiumを起動するにはダミーのプロファイル指定が必須

\* Firefoxでは上記に該当する起動オプションが無く（サイズ指定はあるらしい）、またfvwm3でウィンドウの設定を指定してもうまくいかない。

## 続き | 8-6. Chromium（ウェブブラウザ）を使用したい

- Chromiumの初回起動前にブックマークを設定しておきたい

※システム管理者向け

Yahoo!やGoogleマップをあらかじめブックマークに登録しておく例。

```
⌚ mkdir -p ~/.config/chromium/Default
```

```
⌚ vim ~/.config/chromium/Default/Bookmarks
```

```
{
  "roots": {
    "bookmark_bar": {
      "children": [ {
        "id": "5",
        "name": "Yahoo! JAPAN",
        "type": "url",
        "url": "https://www.yahoo.co.jp/"
      }, {
        "id": "6",
        "name": "Google マップ",
        "type": "url",
        "url": "https://www.google.co.jp/maps/"
      } ],
      "id": "1",
      "name": "ブックマーク バー",
    }
  }
}
```

```
  "type": "folder"
},
"other": {
  "children": [ ],
  "id": "2",
  "name": "その他のブックマーク",
  "type": "folder"
},
"synced": {
  "children": [ ],
  "id": "3",
  "name": "モバイルのブックマーク",
  "type": "folder"
}
},
"version": 1
}
```

```
⌚ chrome &
```

## 続き | 8-6. Chromium（ウェブブラウザ）を使用したい

- Chromiumの初回起動前に初期設定をしておきたい

※システム管理者向け

設定例：

- ・ダウンロードフォルダーを ~/Downloads に設定する。
- ・ブックマークバーを常時表示する。
- ・デフォルトブラウザの確認をさせない。

```
% vim ~/.config/chromium/Default/Preferences

{
  "download": {
    "default_directory": "~/Downloads"
  },
  "bookmark_bar": {
    "show_on_all_tabs": true
  },
  "browser": {
    "check_default_browser": false
  }
}
```

効き目があるかどうかは不明（次ページの説明も参照）

```
% chrome &
```

## 続き | 8-6. Chromium（ウェブブラウザ）を使用したい

- デフォルトブラウザの確認をさせたくない

```
% chrome --no-default-browser-check &
```

起動オプションで指定したほうが効き目がある感じ

## 8-7. Macのユーザー辞書をインポートしたい

① Mac側の作業（概要） ※詳細な手順は一部省いて説明しています。

1. 画面右上の入力メニュー「A」→ユーザ辞書を編集... をクリックする。
2. 「ユーザ辞書」タブをクリックする（表示されない場合は次へすすむ）。
3. command+A キーを押す。
4. 選択された範囲を、デスクトップにドラッグ&ドロップする。
5. ユーザ辞書.plist が作成されるのを確認する。（中身はXML形式のファイル）
6. ユーザ辞書.plist をFreeBSDにコピーする。

ファイル共有でコピーすること。  
(scpコマンドでは、ファイル名のユーザ辞書.plistの  
「ザ」が「サ」扱いとなる)

## 続き | 8-7. Macのユーザー辞書をインポートしたい

FreeBSD側の作業 (ユーザ辞書.plistをAnthyのユーザー辞書ファイル形式に変換する)

### ② 変換スクリプトの作成

```
% vim conv_dic_anthy.py

import plistlib

# ユーザー辞書を読み込む
with open('ユーザ辞書.plist', 'rb') as f:
    plist_data = plistlib.load(f)

# よみがなと単語を抽出し、標準出力に表示する
for item in plist_data:
    reading = item.get('shortcut', '')
    word = item.get('phrase', '')

    # 整形して出力する
    print(f'{reading} #T39*1 {word}')
```

■フォーマット  
よみがな #T39\*1 単語

半角スペース      名詞の意

※本スクリプトは生成AIにて作成後、若干手直しした。

## 続き | 8-7. Macのユーザー辞書をインポートしたい

- ③ 変換スクリプトの実行（ユーザ辞書.plistが同じディレクトリにあることが前提）

```
% python3.11 conv_dic_anthy.py > anthydic.txt
```

- ④ 文字コード順\*1に並び替えて、Anthyのユーザー辞書に登録する

```
% cat ~/.anthy/private_words_default anthydic.txt | \
env LC_ALL=C sort > sortdic.txt
% mv sortdic.txt ~/.anthy/private_words_default
```

\*1 Anthyのユーザー辞書における読みがなは、「文字コード順」に並べる必要があるため、環境変数LC\_ALLをCに設定し、sortコマンドで並び替えを行っている。

（なお、Macのユーザー辞書は一見すると並び替えられているように見えるが、実際には文字コード順ではなく、辞書順と思われる。このまま登録すると、辞書登録したのにもかかわらず変換候補に現れないなどの現象が発生する。）

## 8-8. 日本語を含む PostScript ファイルを、PDF ファイルに変換したい

PostScript内の日本語フォント指定が Ryumin-Light および Gothic BBB-Medium の場合、IPA明朝、および IPAゴシックフォントに置き換えます。

```
% sudo pkg install -y ghostscript10
```

すでにインストール済みかもしれません。

```
% cd /usr/local/share/ghostscript
```

```
% ls
```

```
10.05.1 fonts
```

バージョン名のディレクトリへ cd する

```
% cd 10.05.1
```

```
% sudo vi Resource/Init/cidfmap
```

```
% Aliases
```

```
/Ryumin-Light /IPAMincho ;  
/Gothic BBB-Medium /IPAGothic ;
```

```
% IPA Fonts
```

```
/IPAMincho << /FileType /TrueType /CSI [(Japan1) 6] /Path (/usr/local/share/fonts/ipa/ipam.otf) >> ;  
/IPAGothic << /FileType /TrueType /CSI [(Japan1) 6] /Path (/usr/local/share/fonts/ipa/ipag.otf) >> ;
```

大文字・小文字正確に！

最終行に追加する

```
% ps2pdf filename.ps
```

PDFファイルに変換される

※ イロイロ試しましたが、IPAフォント以外では変換できないことが多かったです。

## 8-9. 画面スライドショーをしたい

```
% sudo pkg install -y feh
```

実行例

```
% feh - . -Z -F -D 5 -Y ~/Pictures/*.*
```

- フォルダーやファイル名など
- マウスカーソルを表示しない
- 5秒ごとに表示を切り替える
- 全画面表示
- 表示サイズに合わせて画像を拡大（ただし、縦横比は維持）
- 表示サイズに合わせて画像を縮小（ただし、縦横比は維持）

プログラムの終了は、[ESC]キー、または マウス右クリック→Exit を選択する。

※ スライドショーに必要と思われるオプション全てを記載した。

## 8-10. Firefoxの初期設定を、起動せずに行いたい

(ユーザー毎ではなく、パソコン全体の設定となる)

※システム管理者向け

```
% sudo mkdir /usr/local/lib/firefox/distribution  
% sudo vi /usr/local/lib/firefox/distribution/policies.json
```

```
{  
    "policies": {  
        "DownloadDirectory": "${home}/Downloads", ← ダウンロードフォルダーの設定  
        "DontCheckDefaultBrowser": true, ← デフォルトブラウザのチェックをさせない  
        "Homepage": {  
            "URL": "https://www.google.com/", ← ホームページの設定（URL、ブラウザ起動時に表示、  
            "StartPage": "homepage", ← 変更可能）  
            "Locked": false  
        },  
        "OverrideFirstRunPage": "", ← ようこそ！ページの無効化  
        "OverridePostUpdatePage": "", ← 新機能紹介ページの無効化  
        "DisplayBookmarksToolbar": true, ← ブックマークツールバーを表示する  
        "NoDefaultBookmarks": true, ← デフォルトのブックマークを表示しない  
        "Bookmarks": [  
            {  
                "Title": "Google",  
                "URL": "https://www.google.com/",  
                "Placement": "toolbar"  
            },  
            {  
                "Title": "YouTube",  
                "URL": "https://www.youtube.com/",  
                "Placement": "toolbar"  
            },  
            {  
                "Title": "Googleマップ",  
                "URL": "https://www.google.com/maps/",  
                "Placement": "toolbar"  
            }  
        ]  
    }  
}
```

## 続き | 8-10. Firefoxの初期設定を、起動せずに行いたい

```
{  
    "Title": "ChatGPT",  
    "URL": "https://chatgpt.com/",  
    "Placement": "toolbar"  
},  
{  
    "Title": "Google翻訳",  
    "URL": "https://translate.google.com/",  
    "Placement": "toolbar"  
},  
{  
    "Title": "47NEWS",  
    "URL": "https://www.47news.jp/",  
    "Placement": "toolbar"  
},  
{  
    "Title": "日本気象協会",  
    "URL": "https://tenki.jp/",  
    "Placement": "toolbar"  
},  
{  
    "Title": "radiko",  
    "URL": "https://radiko.jp/",  
    "Placement": "toolbar"  
}  
]  
}
```

↑ ブックマークの設定

### 課題 :

- ・初回起動時に日本語表示モードにする方法は見つかっていない。
- ・前回のセッションを復元しますか？の表示も防ぎたいが、policies.jsonでは設定できない。
- ・その他は、ChatGPTに質問してみてください。

## 8-11. GIMPを使いたい

ここでは、GIMPのバージョン2系をインストールします。

```
% sudo pkg install -y gimp  
% gimp
```

- ・ダークモードを解除したい

- ・編集(E) → 設定(P)
- ・テーマ → System を選択
- ・アイコンテーマ → Symbolic-Inverted を選択 → OK(O)

テーマをSystemにするとアイコンが分かりにくくなるための設定。

## 8-12. GIMPで画像編集したい

- ・縦横比を維持したままトリミングしたい



②チェック

③縦横比を選択

④縦横比を入力

「1:1」「9:16」「16:9」「4:3」「3:2」などでも可

⑤画像のトリミング範囲を選択

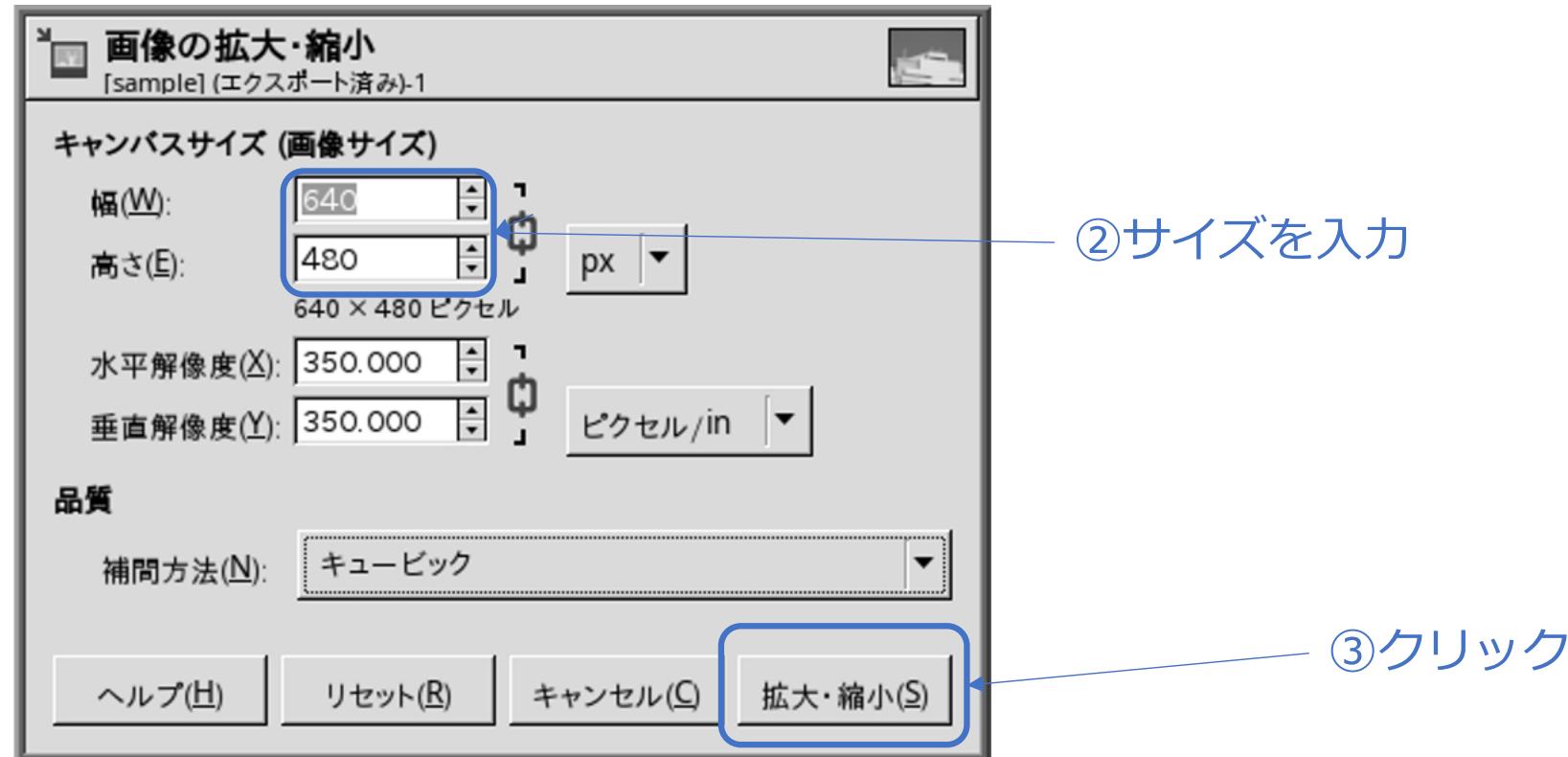
- ・サイズ変更は四隅部分をドラッグ
- ・範囲部分の移動は中央部分をドラッグ

⑥左クリックで確定

## 続き | 8-12. GIMPで画像編集したい

- ・画像を拡大・縮小したい（リサイズ・スケール変更）

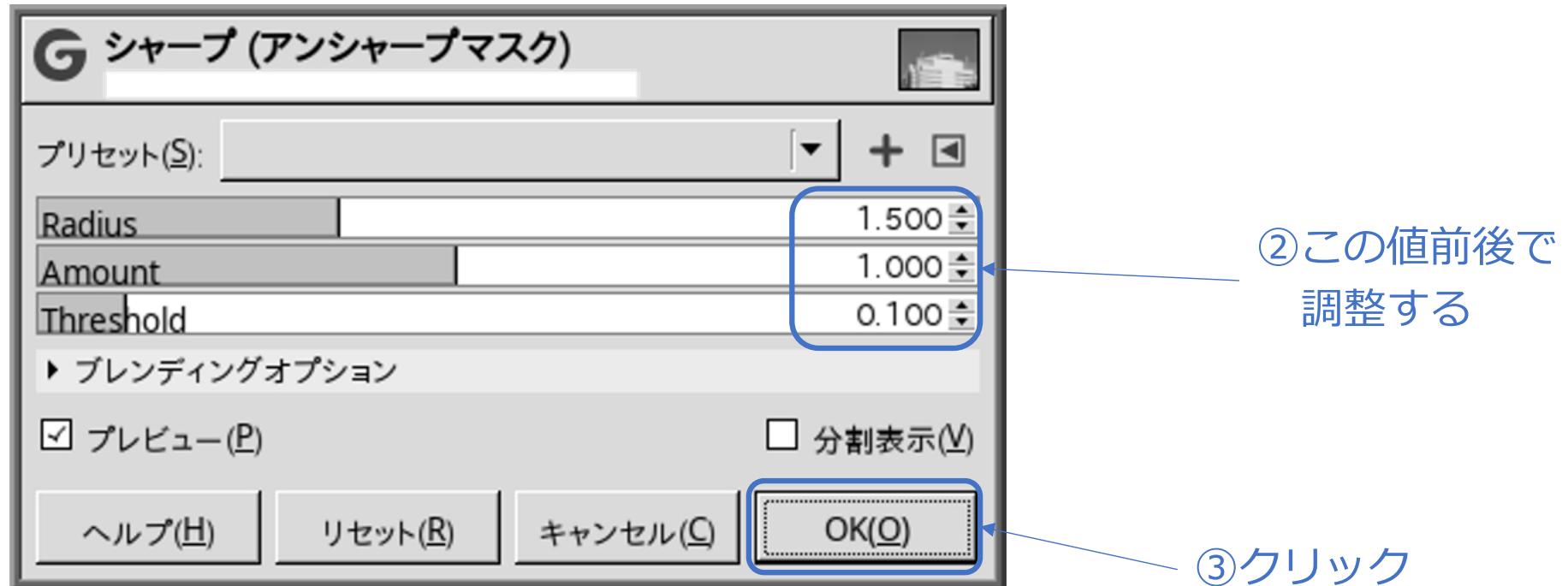
① 画像(I) → 画像の拡大・縮小(S)



## 続き | 8-12. GIMPで画像編集したい

- ・画像を少しクッキリさせたい

① フィルター(R) → 強調(H) → シャープ (アンシャープマスク) (U)



## 続き | 8-12. GIMPで画像編集したい

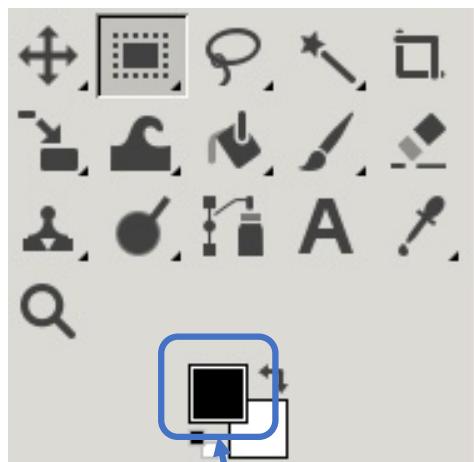
- ・編集した画像の保存

ファイル(F) → 名前を付けてエクスポート(X)...

## 続き | 8-12. GIMPで画像編集したい

- ・画像に余白を加えたい

余白の色を決める



①クリック



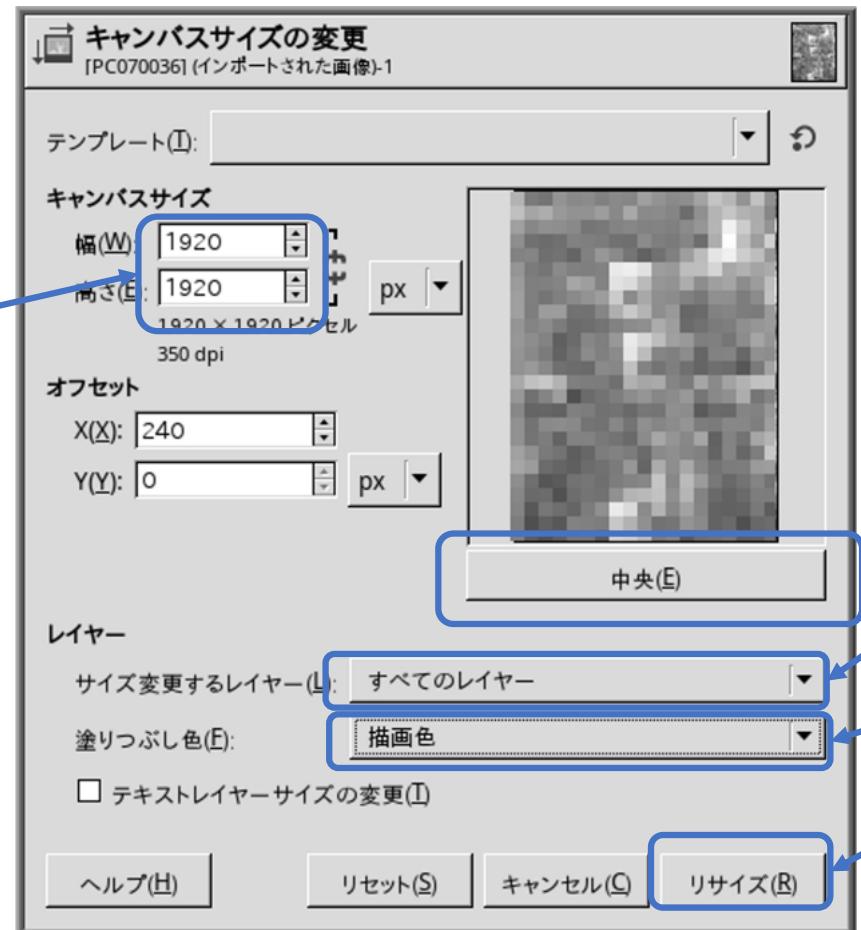
② ffffff と入力

③ OKをクリック

# 続き | 8-12. GIMPで画像編集したい

- ・画像に余白を加えたい（続き）

画像(I) → キャンバスサイズの変更(V)...



①余白を加えた後の画像サイズを入力

②中央(E)をクリック

③全てのレイヤーを選択

④描画色を選択

⑤クリック

## 続き | 8-12. GIMPで画像編集したい

- ・画像の一部にモザイクを入れたい



- ③マウスクリックで、モザイクしたい範囲を囲む
- ④フィルター(R)→ぼかし(B)→モザイク処理(P)... を選択する

# 続き | 8-12. GIMPで画像編集したい

- ・画像の一部にモザイクを入れたい（続き）



## 続き | 8-12. GIMPで画像編集したい

- ・画像の一部を修復・補修したい

例えば、ライブ・コンサートなどで、最前列の人の頭が写り込んでしまった場合、うまい具合に修復・補修する方法です。



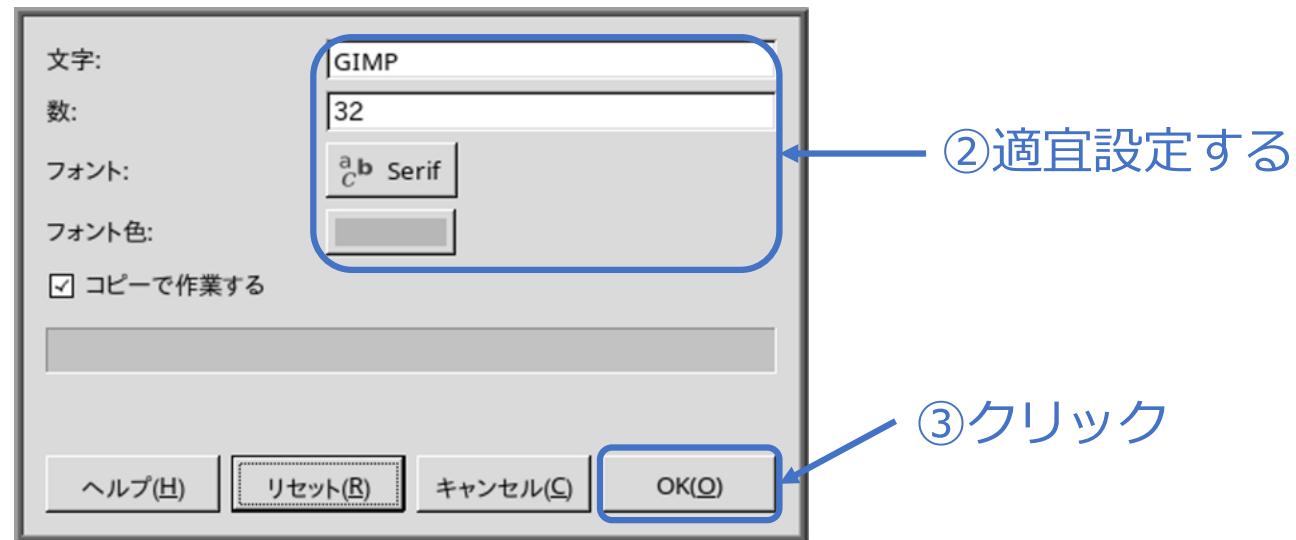
- ② 背景として使いたい場所を、CTRL+クリックする。 (○印が表示される)
- ③ 補修したい部分を、マウスの左ボタンを押しながら消しゴムで消すかのようにマウスを動かす。 (すると徐々に補正後の画像が浮かび上がってくる)

## 続き | 8-12. GIMPで画像編集したい

- ・画像をスライドフィルム風にしたい

リバーサルフィルムのスライドのようにする方法です。

- ① フィルター(R)→装飾(D)→スライド(S)... を選択する。



# 8-13. OpenSCADで通信鉄塔をモデリングしたい

※ CATIAやSOLIDWORKSなどのGUIベースのモデリングではなく、スクリプトを読み込むことによって描画を行う3次元CADソフトである。

```
% sudo pkg install -y openscad  
% openscad
```

注記：2026年1月時点で当方の環境では正常に動作しません

- Newボタンをクリックする
- 左側のEditor欄に下記のスクリプトを記述する

※ 各パラメータの詳細はChatGPTに質問してください。

```
// 初期設定  
trans = 0.75 ; // 透明度  
$fn = 50 ; // 円や円柱をなめらかに表示  
  
// 局舎を描画する  
color("Gray",trans) {  
    cube([9,9,4]);  
}  
  
// 鉄塔を描画する  
color("White",trans) {  
    translate([2.5,2.5,4]) {  
        cylinder(h=15,r=0.25) ;  
    }  
    translate([6.5,2.5,4]) {  
        cylinder(h=15,r=0.25) ;  
    }  
    translate([2.5,6.5,4]) {  
        cylinder(h=15,r=0.25) ;  
    }  
    translate([6.5,6.5,4]) {  
        cylinder(h=15,r=0.25) ;  
    }  
}
```

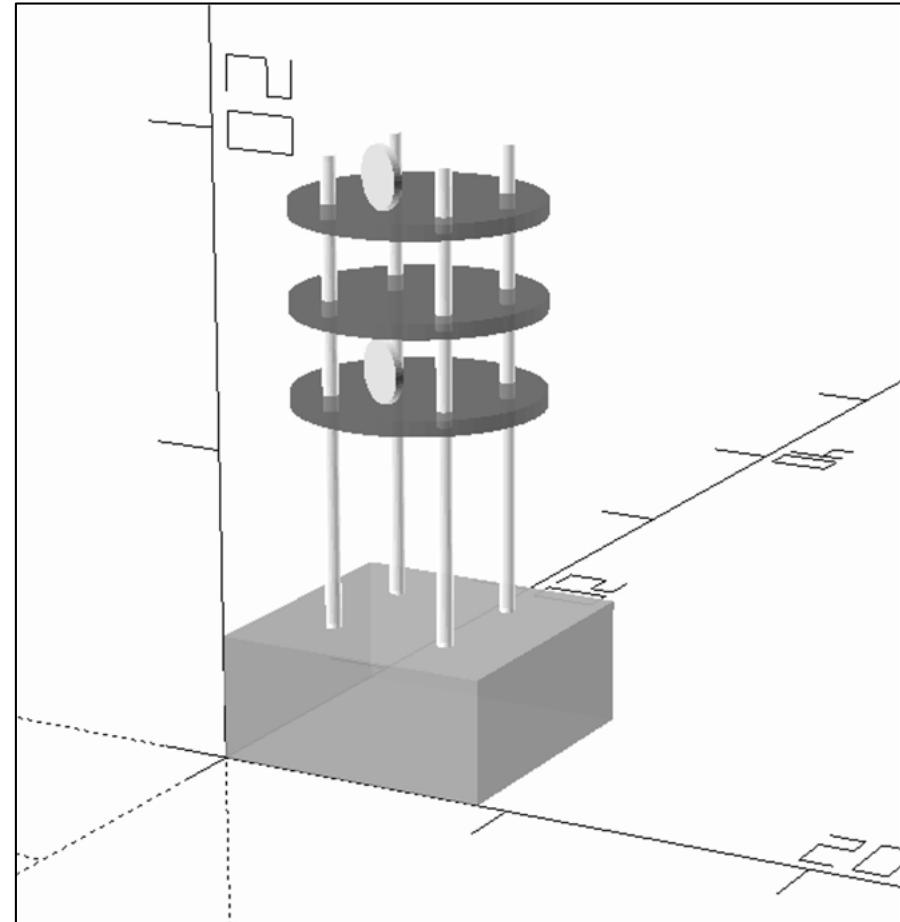


```
// プラットホームを描画する  
color("Red",trans) {  
    translate([4.5,4.5,11]) {  
        cylinder(h=0.5,r=4) ;  
    }  
    translate([4.5,4.5,14]) {  
        cylinder(h=0.5,r=4) ;  
    }  
    translate([4.5,4.5,17]) {  
        cylinder(h=0.5,r=4) ;  
    }  
}  
  
// パラボラアンテナを描画する  
color("White",100) {  
    translate([4.5,2.5,12.5]) {  
        rotate([120,90,0]) {  
            cylinder(h=0.25,r=1) ;  
        }  
    }  
    translate([4.5,2.5,18.5]) {  
        rotate([120,90,0]) {  
            cylinder(h=0.25,r=1) ;  
        }  
    }  
}
```

## 続き | 8-13. OpenSCADで通信鉄塔をモデリングしたい

- F5キー、または  を押す。
- 不必要なウインドウは、 で閉じる。
- 左ボタンを押したままドラッグ=回転
- 右ボタンを押したままドラッグ=移動
- マウスホイールをスクロールさせる=ズーム
- スクリプトを保存するのを忘れない（終了メニューは、保存せずにそのまま終了するため）

描画例：



## 8-14. サムネイル一覧から画像を選択して表示したい

```
% sudo pkg install -y nsxiv
```

Macの写真アプリに少し似ている・・・かも。ただし、大量に画像がある場合、スクロールバーが表示されないのは難点。

以下、「12-5. デジカメなどの画像データを、撮影日（年/月/日）別に整理したい」で整理した画像ファイルを、サムネイル表示させる例で紹介

```
% nsxiv -t -r ~/Pictures
```

指定ディレクトリ以下全ての画像ファイルが対象

起動時サムネイル表示

キー操作：

画面	キー／マウス	内容
サムネイル	Enter ダブルクリック	選択した画像を表示する
サムネイル	カーソル クリック	画像を選択する
サムネイル	右クリック	画像を複数選択する
画像表示	Enter 右クリック	サムネイル表示に戻る
画像表示	スペース n 画面右側をクリック	次の画像を表示する
画像表示	p 画面左側をクリック	前の画像を表示する

画面	キー／マウス	内容
画像表示	=	100%表示 (Dot by Dot)
画像表示	ホイール	ズーム
画像表示	カーソル CTRL+マウス移動	移動
画像表示	w	ウインドウ内全体表示
画像表示	< >	左回転 右回転
画像表示	f	フルスクリーン表示/解除
サムネ／画像	q	終了する

※ サムネイル表示では、デジカメ写真の縦横を認識するが、画像表示するときは認識していない模様（解決方法不明）。なお、サムネの大きさ、キーアサインの変更は、ソースコードを変更するのが一般的な模様（例外あり＝次ページにて解説）。⇒ 8-26. を参照のこと。

## 続き | 8-14. サムネイル一覧から画像を選択して表示したい

- 表示中の画像を外部コマンドに渡して加工したい

以下、キーアサインと起動プログラムを設定している。

```
% mkdir -p ~/.config/nsxiv/exec
```

```
% vim ~/.config/nsxiv/exec/key-handler
```

```
#!/bin/tcsh
switch ($1)                                $1にキーコードが渡される
    case "C-g":                            ←
        tr '\n' '\0' | xargs -0 tcsh -c 'gimp $*:q &' ←
        breaksw
endsw
```

ファイル名が標準出力で渡されるための処理  
(複数選択されている場合は、複数行でファイル名が渡される)

```
% chmod +x ~/.config/nsxiv/exec/key-handler
```

- 上記設定した上で、外部コマンドへ渡したい画像を表示させ、「CTRL-X」「CTRL-G」を続けて押すことにより、gimpが起動する（サムネイル画面でも可。複数選択でも可）。

# 続き | 8-14. サムネイル一覧から画像を選択して表示したい

- 表示中画像のExif情報を見たい

```
% sudo pkg install -y p5-Image-ExifTool  
% vim ~/.config/nsxiv/exec/image-info
```

```
#!/bin/tcsh  
  
# 一時ファイル名  
set tmp="/tmp/exif_output.txt"  
  
# ファイルのフルパスを取得  
set filepath = `realpath "$1"  
  
# exiftoolで出力を一時ファイルに保存  
exiftool -d "%Y%m%d %H%M%S" -DateTimeOriginal -ImageSize -  
Model -T "$1" > ${tmp}  
  
# 出力結果を行ごとに処理する  
set datetime = `cat ${tmp} | cut -f 1`  
set imagesize = `cat ${tmp} | cut -f 2`  
set model = `cat ${tmp} | cut -f 3`  
  
# 撮影日と時刻の処理  
set date_part = `echo $datetime | cut -d " " -f 1`  
set time_part = `echo $datetime | cut -d " " -f 2`  
  
if ("$date_part" == "--" || "$time_part" == "--") then  
    set capture_date = "撮影日:不明"  
else  
    set year = `echo "$date_part" | cut -c 1-4`  
    set month = `echo "$date_part" | cut -c 5-6`  
    set day = `echo "$date_part" | cut -c 7-8`  
    set weekday = `date -j -f "%Y%m%d"  
"${year}${month}${day}" "+%a"  
" ${capture_date}
```

```
set hour = `echo "$time_part" | cut -c 1-2`  
set minute = `echo "$time_part" | cut -c 3-4`  
set second = `echo "$time_part" | cut -c 5-6`  
  
set capture_date = "撮影日:${year}年${month}月${day}日  
(${weekday}) ${hour}時${minute}分${second}秒"  
endif  
  
# 画像の解像度を取得  
if ("$imagesize" == "-") then  
    set resolution = "サイズ:不明"  
else  
    set resolution = "サイズ:$imagesize"  
endif  
  
# デジカメの機種名を取得  
if ("$model" == "-") then  
    set camera_model = "機種名:不明"  
else  
    set camera_model = "機種名:$model"  
endif  
  
# 結果を標準出力に表示  
echo "$filepath"  
echo "$capture_date"  
echo "$resolution"  
echo "$camera_model"  
  
# 一時ファイルを削除  
rm ${tmp}
```

- 上記設定した上で、画像を表示すると、画面下に「パス名」「タイムスタンプ」「画像縦横サイズ」「デジカメの機種名」が表示される。

## 続き | 8-14. サムネイル一覧から画像を選択して表示したい

- 表示中画像のExif情報を見たい（続き）

```
₩ chmod +x ~/.config/nsxiv/exec/image-info
```

## 8-15. システム情報を表示したい (conky)

- ・アプリのインストール

```
% sudo pkg install -y conky
```

- ・設定ファイルをダウンロードする

```
% cd  
% fetch ¥  
https://raw.githubusercontent.com/moginwc/freebsd143/refs/heads/main/.conkyrc
```

## 続き | 8-15. システム情報を表示したい (conky)

- conkyを実行する

⇒ **conky**

画面右側にシステム情報が表示される。

※ネットワーク情報が表示されない場合は、`~/.conkyrc` の ネットワークインターフェース名が記載されている部分を修正してください（例：`em0 → re0` など）。

# 続き | 8-15. システム情報を表示したい (conky)

## 【補足】 .conkyrcの解説

```
-- conky基本設定
-----
conky.config = {
    double_buffer = true,           -- 画面のチラつきを防止する
    update_interval = 5.0,          ←
    own_window = true,             ←
    own_window_type = 'normal',    -- normal desktop
    own_window_transparent = true, -- 透過する
    alignment = 'middle_right',   -- 表示位置 ←
    maximum_width = 256,
    default_color = 'white',
    font = '7x13',                ← ビットマップフォント指定
    gap_x = 15,                   ←
    gap_y = 25,                   ←
};

-----  
-- conky画面設定
-----
conky.text = [[  

# システム表示(os名,リリース名,アーキテクチャ名,ホスト名,稼働時間)  

${color white}SYSTEM ${hr 2}  

OS: ${exec uname -o} ${kernel} ${exec uname -m}  

Hostname: $nodename  

Uptime: $uptime
```

### 更新頻度

ウインドウ内に状況を表示する。falseにするとデスクトップ上に表示される。

top_left	top_middle	top_right
middle_left	middle_middle	middle_right
bottom_left	bottom_right	bottom_middle

alignment設定がmiddle\_rightになっているので、ここでは表示位置を右端から15px内側に寄せる、という意味であるが、own\_window・own\_window\_typeの設定、およびウインドウマネージャによっても表示位置が変わるので、適宜変更する。思い通りにしたい場合は、own\_window\_type = overrideに設定する（ただしウインドウにならずにデスクトップ上に表示される）。

```
SYSTEM —  
OS: FreeBSD 14.3-RELEASE-p7 amd64  
Hostname: pc100001  
Uptime: 0h 0m 50s
```

→ 改行は、行間を空ける。  
(conky.text = [[ 以降は改行も意味がある)

# 続き | 8-15. システム情報を表示したい (conky)

## .conkyrcの解説（続き）

```
# CPU表示(平均負荷,CPU使用率,周波数,温度,プロセス)
${color white}CPU ${hr 2}
Load: ${loadavg} ${alignr}${cpu}%
Freq: ${freq_g}GHz ${alignr}${exec sysctl -a | grep temperature | awk '{print $2}'}
${cpugraph cpu1 40,256 3d78ba 3d78ba}
```

NAME	PID	CPU%	MEM%

負荷の高い上位5プロセスを表示。  
(CPU%は、そのプロセスが何%CPUを使っているかという意味であり、CPU全体リソースの中で何%という意味ではない)

高さ40、幅256ピクセルのCPU使用率グラフを描く。  
色指定はグラデーションの開始・終了色の意味である  
が、同じ色を指定することで単色となる。



NAME	PID	CPU%	MEM%
firefox	1819	24.32	12.60
firefox	1803	17.72	8.60
Xorg	1284	2.78	1.57
firefox	1823	2.64	2.52
pulseaudio	1839	0.39	0.16

# 続き | 8-15. システム情報を表示したい (conky)

## .conkyrcの解説（続き）

```
# メモリー表示(使用状況、スワップ使用状況)
```

```
 ${color white}MEMORY ${hr 2}
```

```
 RAM: $mem / $memmax ${alignr}$memperc%
```

```
 Swap: $swap / $swapmax ${alignr}$swapperc%
```

```
 ${color dedede}${membar 20,256}${color}
```

MEMORY

RAM: 2.81GiB / 7.43GiB 37%

Swap: 0B / 2.00GiB 0%



高さ20、幅256ピクセルのメモリ使用率バーを描く。

```
# ディスク表示(使用状況、ディスクアクセス状況)
```

```
 ${color white}DISK ${hr 2}
```

```
 Used: ${fs_used } / ${fs_size } ${alignr}${fs_used_perc }%
```

```
 ${color dedede}${fs_bar 20,256 }${color}
```

```
 Read: ${diskio_read}/s ${alignr}Write: ${diskio_write}/s
```

```
 ${diskiograph_read 40,128 53aa11 53aa11} ${alignr}${diskiograph_write 40,128 9f520a 9f520a}
```

左側に高さ40、幅128ピクセルのすべてのディスク読み取り速度グラフ（緑色）を描く。

右側に高さ40、幅128ピクセルのすべてのディスク書き込み速度グラフ（赤色）を描く。

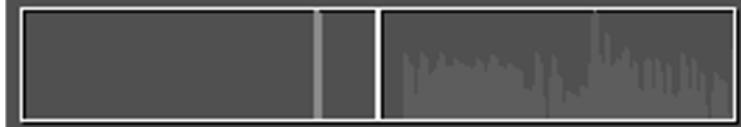
DISK

Used: 15.3GiB / 48.7GiB 31%



Read: 0B/s

Write: 180KiB/s



# 続き | 8-15. システム情報を表示したい (conky)

## .conkyrcの解説（続き）

```
# ネットワーク表示(IPアドレス, 使用状況)
```

```
 ${color white}NETWORK ${hr 2}
```

```
IP: ${addr em0}
```

```
Down: ${downspeed em0} /s ${alignr}Up: ${upspeed em0} /s
```

```
 ${downspeedgraph em0 40,128 53aa11 53aa11} ${alignr} ${upspeedgraph em0 40,128 9f520a 9f520a}
```

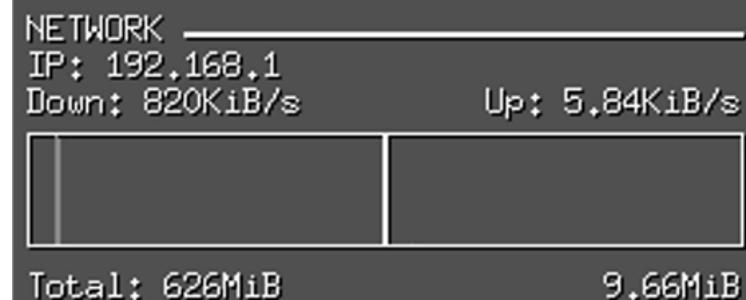
```
Total: ${totaldown em0} ${alignr} ${totalup em0}
```

```
];
```

em0 は、対象となるネットワークインターフェース  
を意味する。

左側に高さ40、幅128ピクセルのネットワーク受信  
速度グラフ（緑色）を描く。

右側に高さ40、幅128ピクセルのネットワーク送信  
速度グラフ（赤色）を描く。



## 8-16. マイク端子から録音をしたい

- ・デバイスの確認

```
% cat /dev/sndstat
```

Installed devices:

```
pcm0: <Realtek ALC269 (Analog 2.0+HP/2.0) > (play/rec) default
pcm1: <Intel Panther Point (HDMI/DP 8ch) > (play)
pcm2: <Intel Panther Point (HDMI/DP 8ch) > (play)
pcm3: <Intel Panther Point (HDMI/DP 8ch) > (play)
No devices installed from userspace.
```

録音可能な  
デバイス

## 続き | 8-16. マイク端子から録音をしたい

- ・ 録音ツールのインストール

```
% sudo pkg install -y sox
```

- ・ マイク端子にケーブルを接続する（コンボジャック（4極端子）の場合、機器によっては分岐・変換ケーブルなどが必要）。接続しない場合は、内蔵マイクが使用される。

- ・ 録音

```
% env AUDIODEV=0 rec -c 2 -r 44100 output.wav trim 0 10
```

-c 2 でステレオ録音

サンプリングレート指定

0秒から10秒間録音する

- ・ 再生

```
% env AUDIODEV=0 play output.wav
```

# 続き | 8-16. マイク端子から録音したい

## □ 録音日時付きで録音したい

アナログラジオの録音に使える、、、かも?

概略：音声をマイク端子から取り込み、日時付きの動画ファイルを作成します。

- FFmpegのインストール

```
% sudo pkg install -y ffmpeg
```

- 録音

```
% ffmpeg ¥  
-f oss -i /dev/dsp0 ¥ ← 入力デバイスの設定  
-f lavfi -i color=color=black:size=640x240 ¥ ← 真っ黒な画面を作成  
-t 00:00:30 ¥ ← 30秒間の録音  
[  
  -vf "drawtext=fontfile=/usr/local/share/fonts/ipa/ipag.otf:  
        text='%{localtime}': fontsize=32: fontcolor=white: x=16: y=16" ¥  
  -c:v libx264 -c:a aac -shortest output.mp4 ← エンコード設定
```

映像への  
日時描画  
指定

- 再生

```
% firefox output.mp4
```

## 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

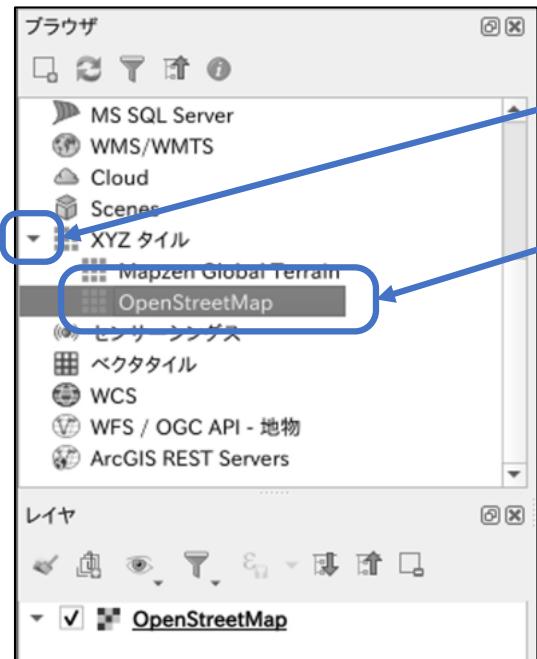
- ・インストール

```
% sudo pkg install -y qgis open-sans
```

- ・起動

```
% qgis
```

- ・初期設定（地図の設定）



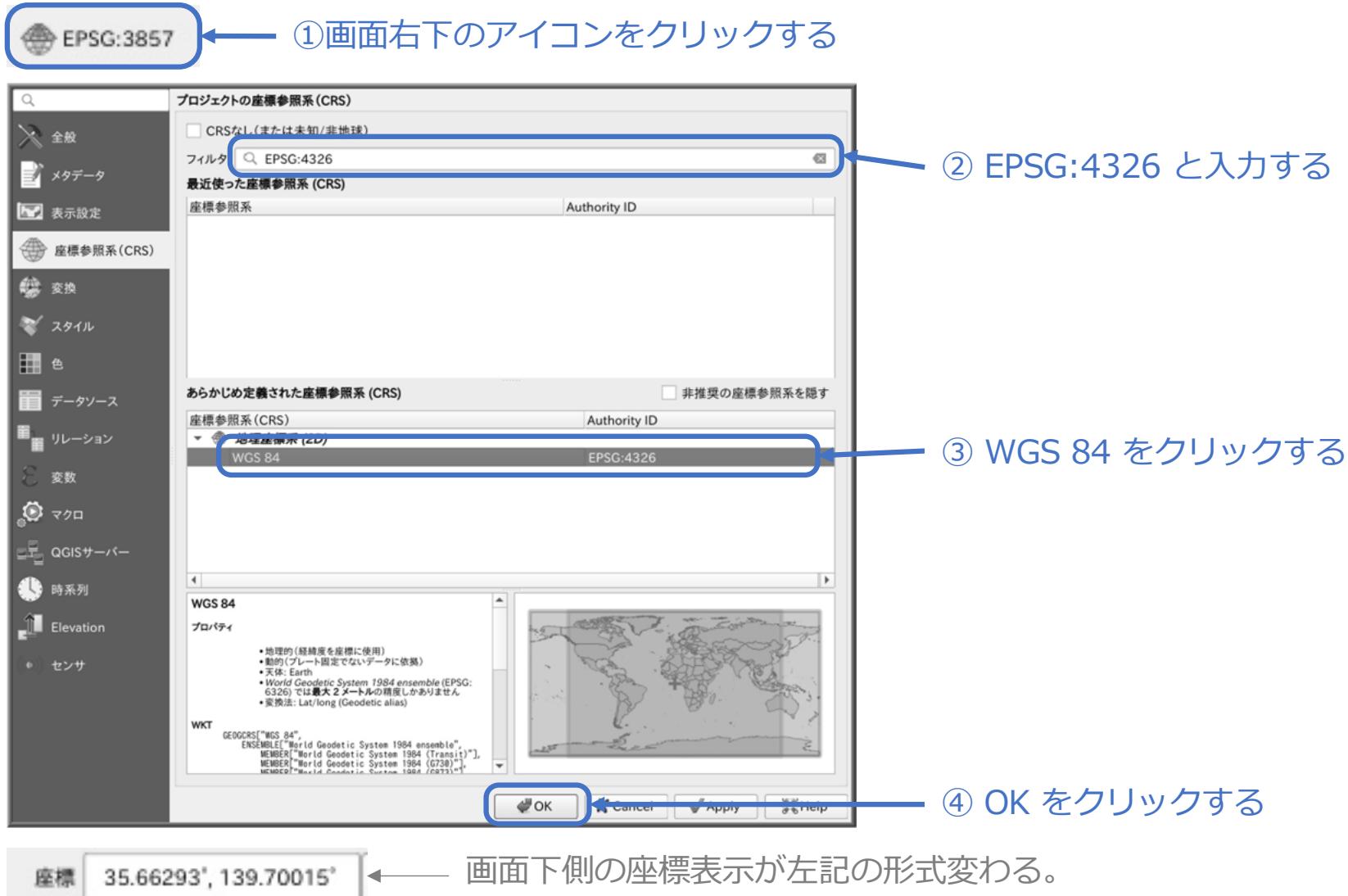
XYZ Tilesの左の▶をクリックする

OpenStreetMapを  
ダブルクリックする

画面右側に世界地図  
が表示される。

## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

### □ 座標表示を変更したい



## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

### □国土地理院の地図を使いたい

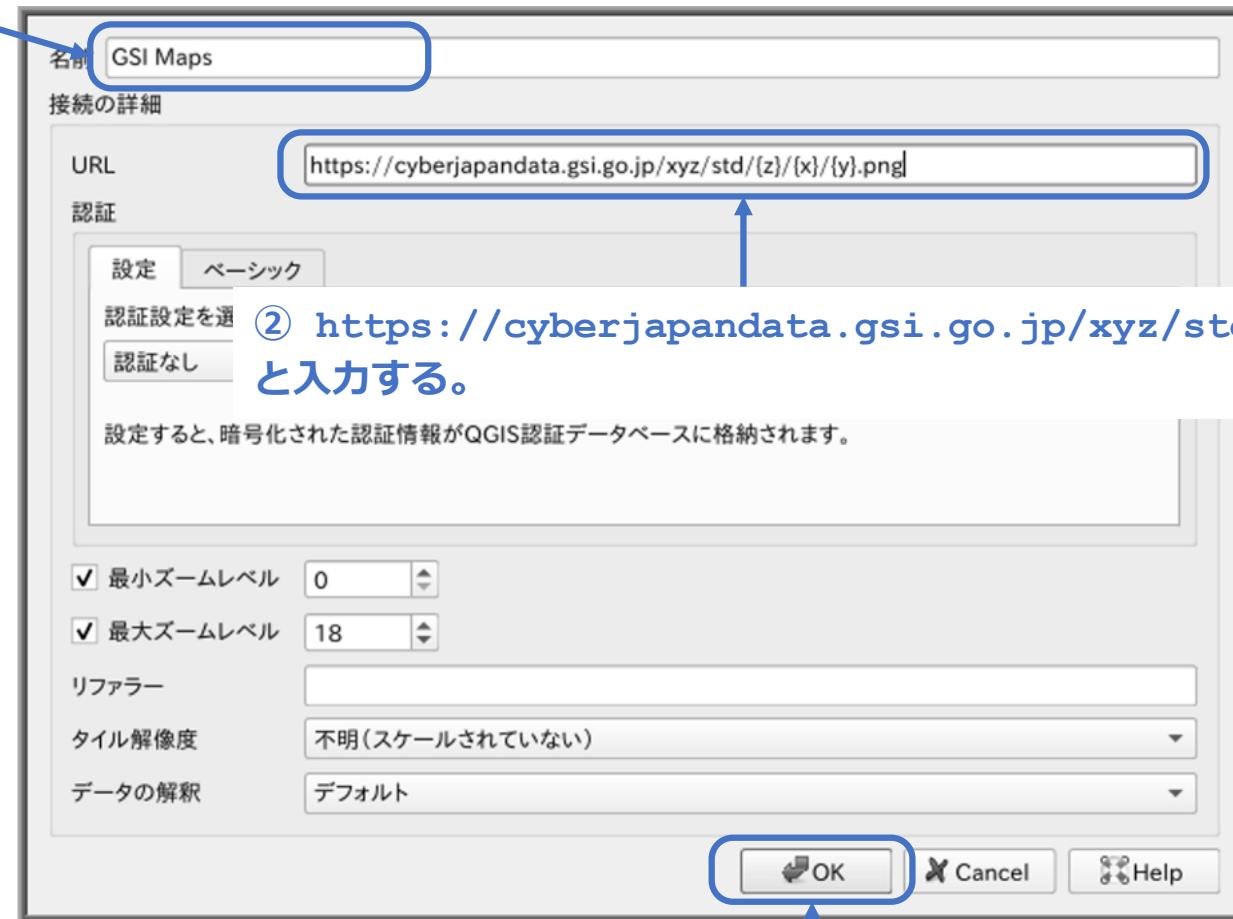
国土地理院の地図を使用するには、下記の設定を行ってください。



XYZ Tilesを右クリックし、  
新規接続...をクリックする

## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

### ① GSI Maps と入力する



② `https://cyberjapandata.gsi.go.jp/xyz/std/{z}/{x}/{y}.png` と入力する。

③ OK をクリックする

XYZ Tilesの中に「GSI Maps」が登録される。→ダブルクリックで地図の閲覧が可能。

## 続き | 8-17. QGIS（地理空間情報の閲覧、編集、分析）を使いたい

□ CSVファイルからポイント（座標）データを読み込み、表示させたい

ポイント（座標）データを作成する。

```
% vim point.csv
```

名前,緯度,経度

A点,35.6650016,139.6963361

B点,35.710063,139.8107

C点,36.069147,139.627710

緯度・経度については、

- ・10進法表記
- ・世界測地系

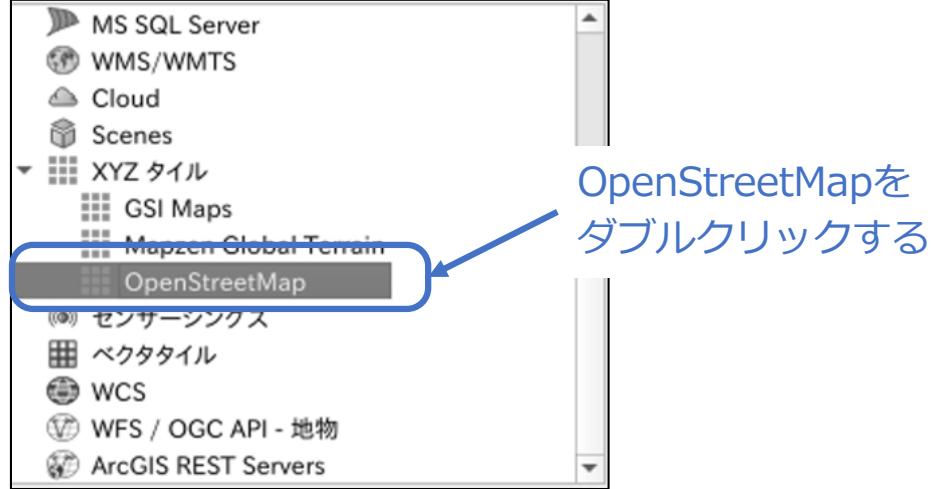
QGISを起動する

```
% qgis
```

プロジェクトテンプレートをダブルクリックする



## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

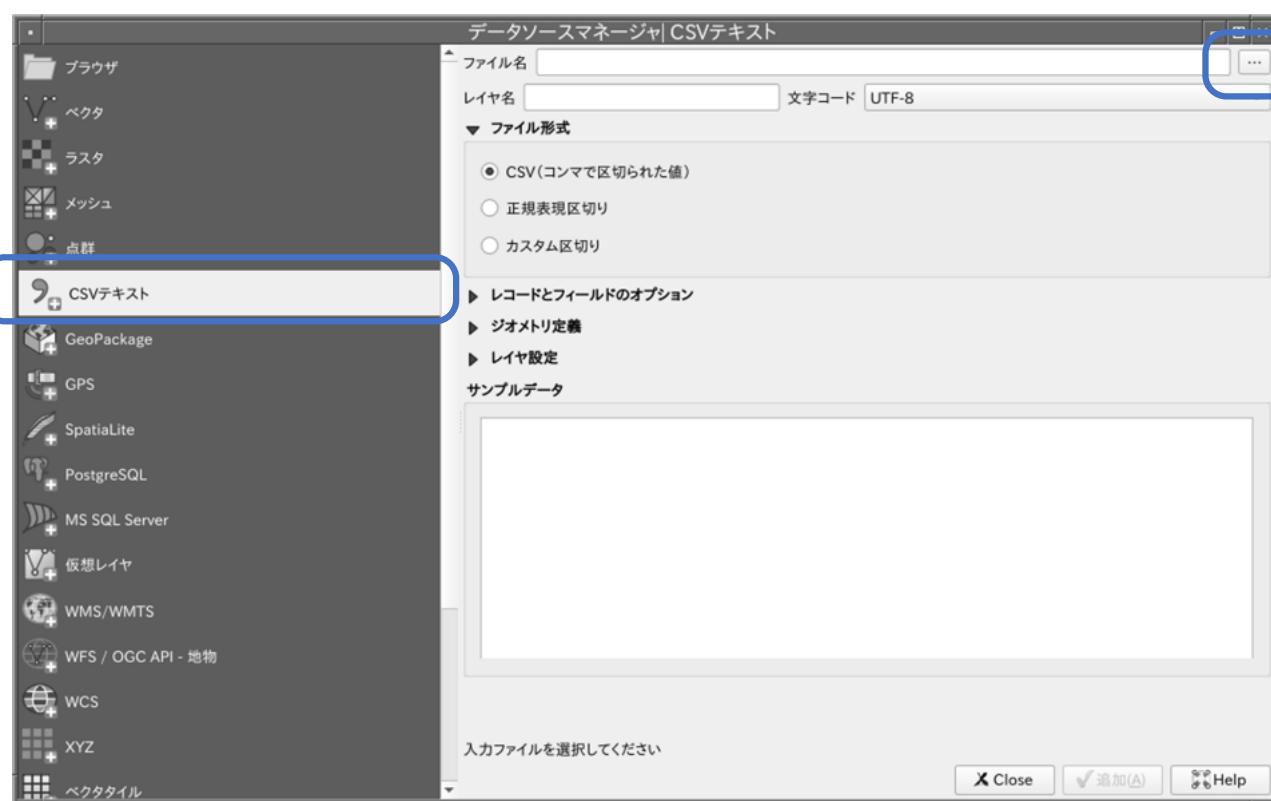


OpenStreetMapを  
ダブルクリックする

レイヤ(L)→レイヤを追加→CSVテキストレイヤを追加...を選択する

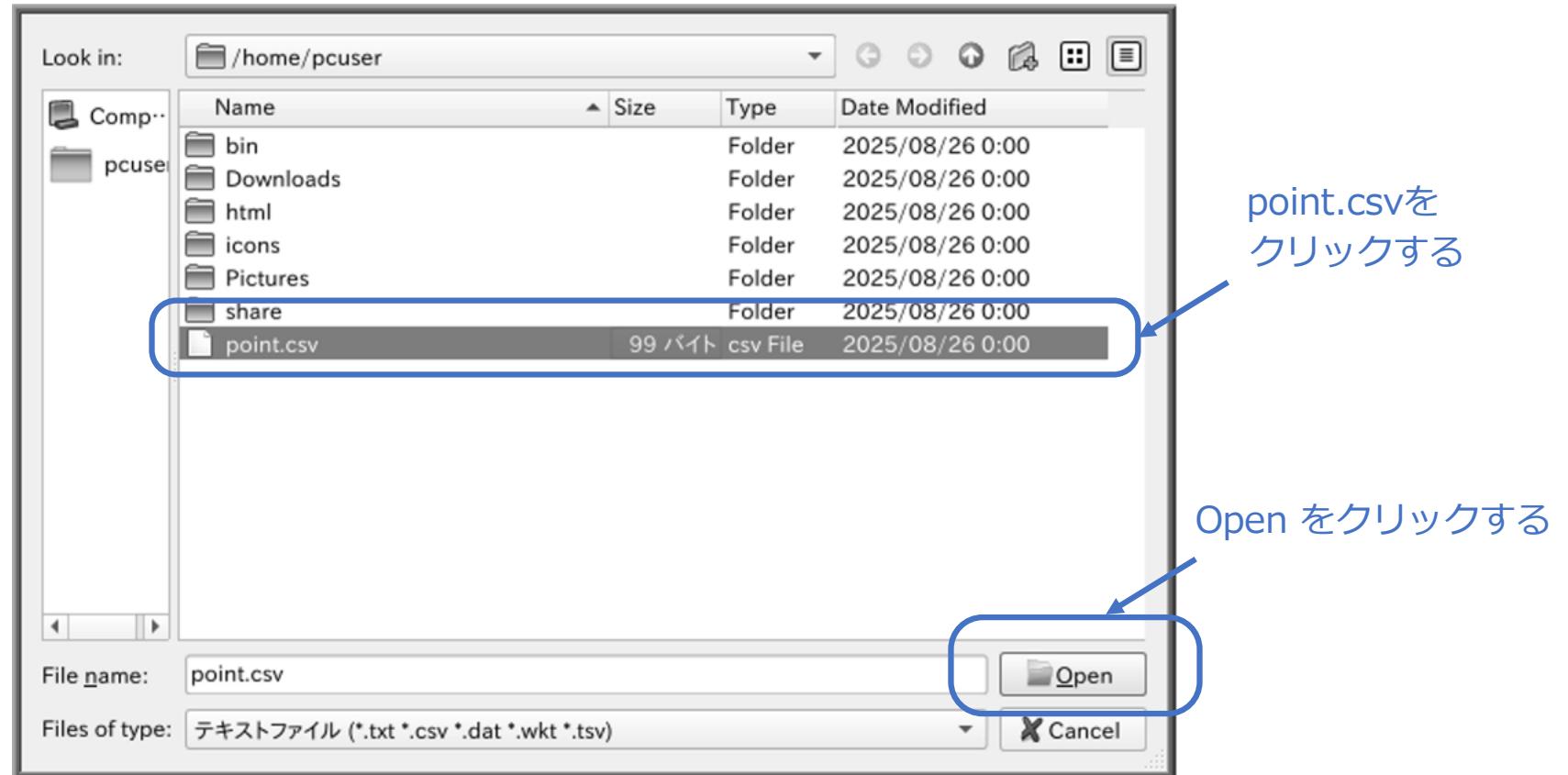
## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

CSVテキストを  
クリックする

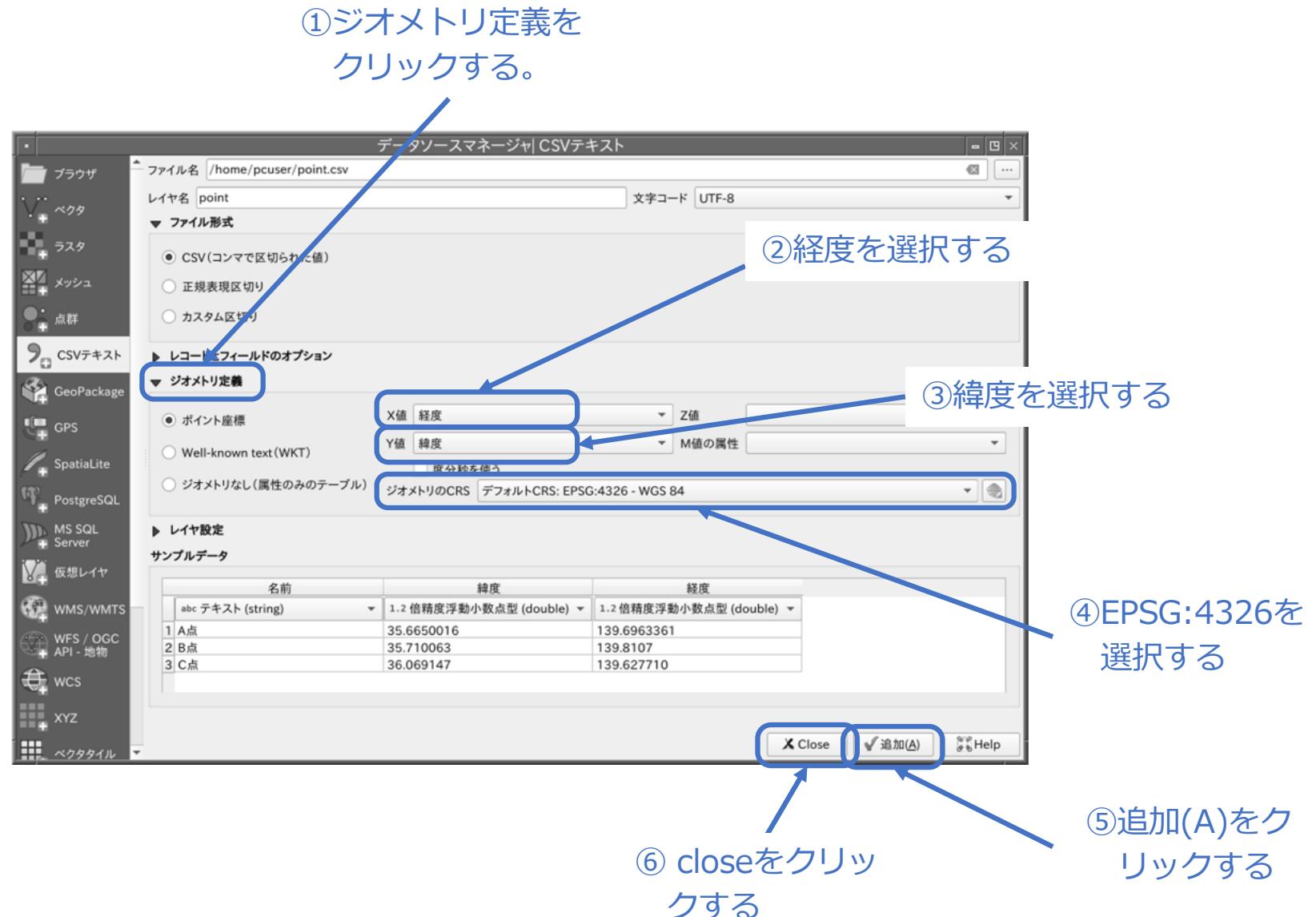


...をクリックする

## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

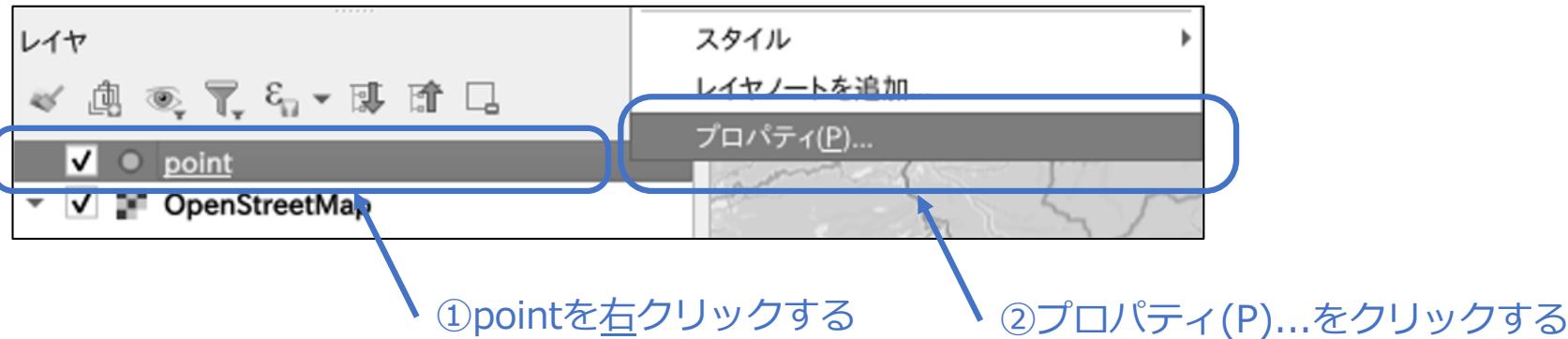


## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい



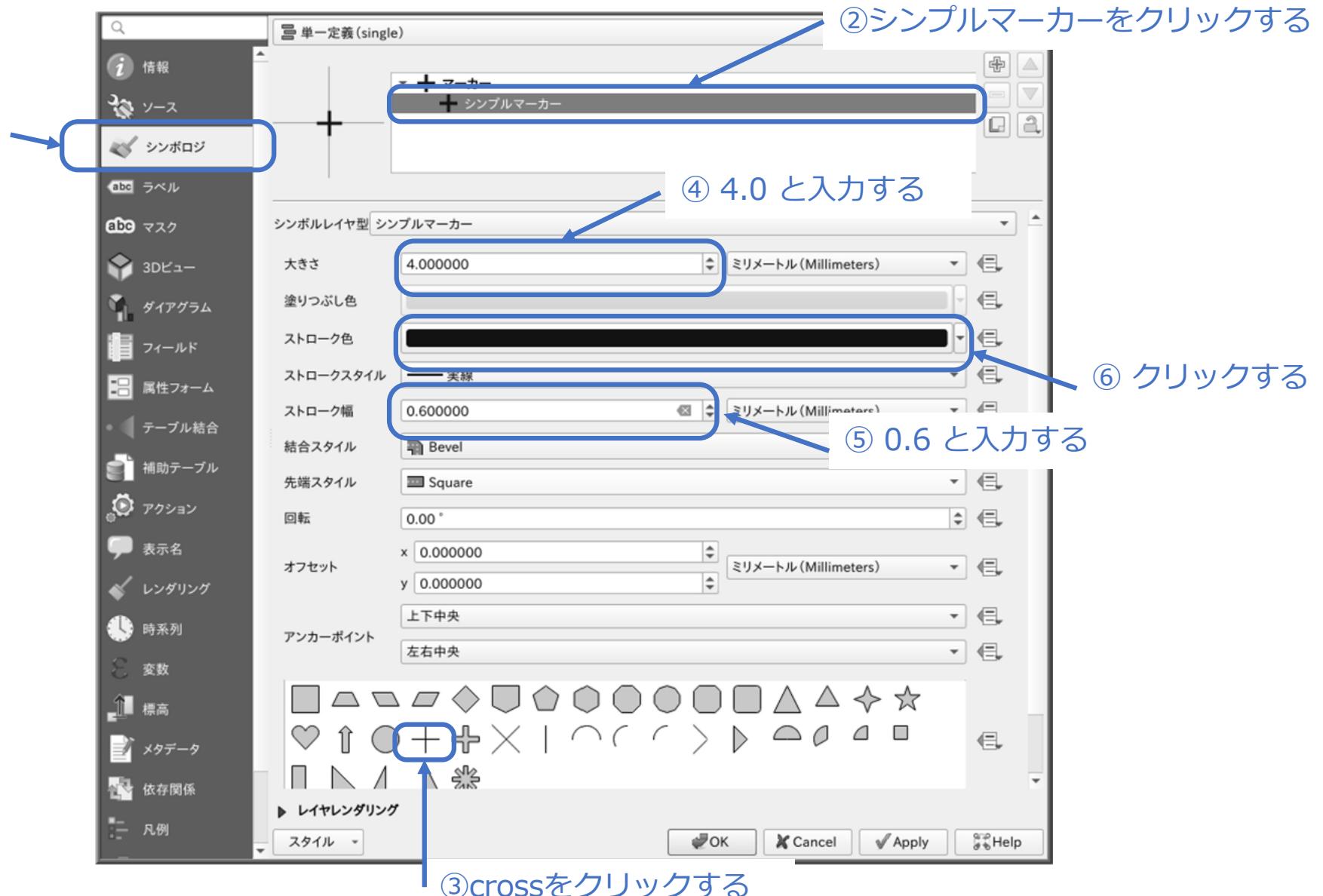
## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

※この先は見やすくするための設定です。



## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

①シンボロジ  
をクリックす  
る

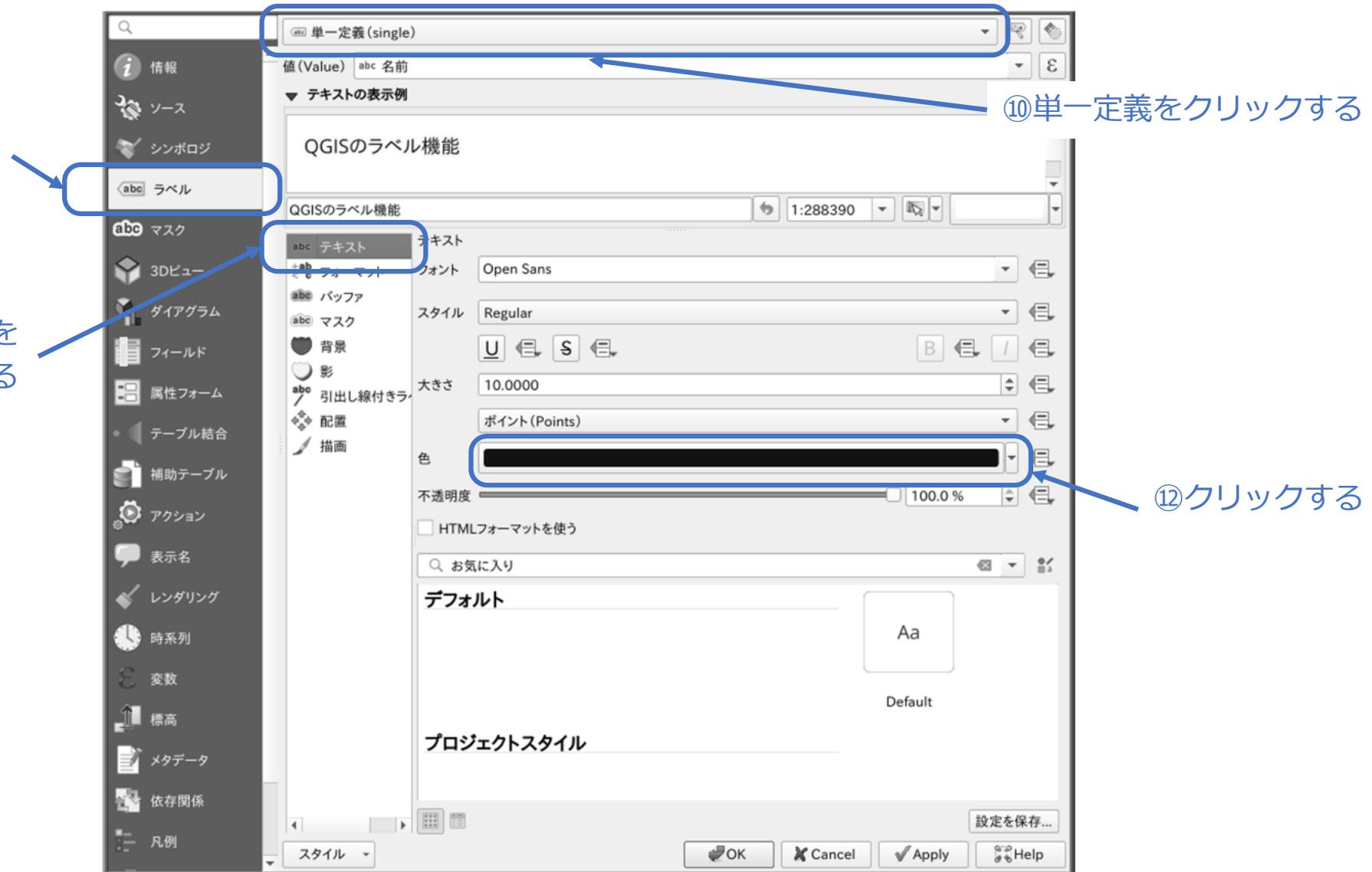


## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい



## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

⑨ラベルをクリックする

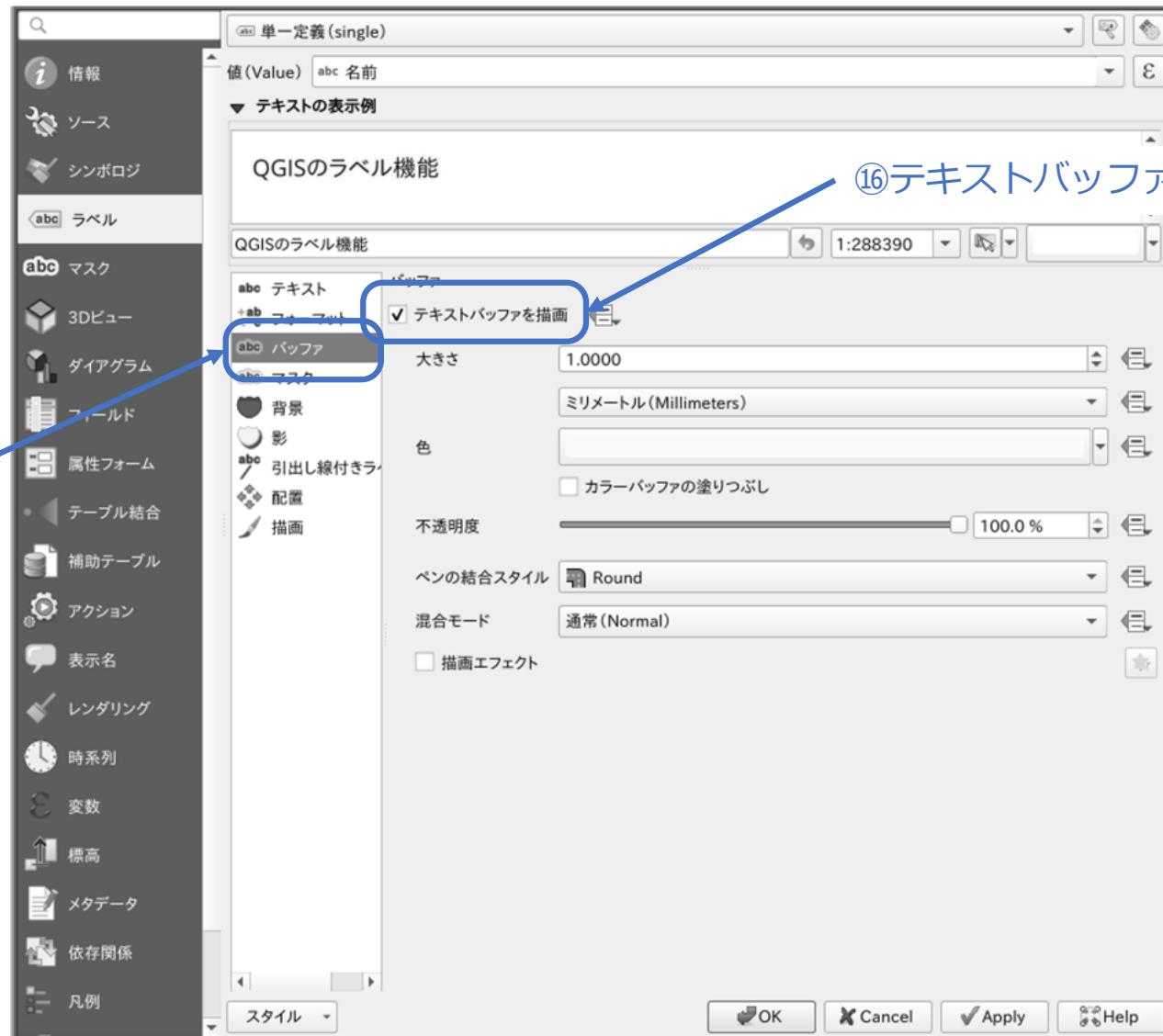


## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい



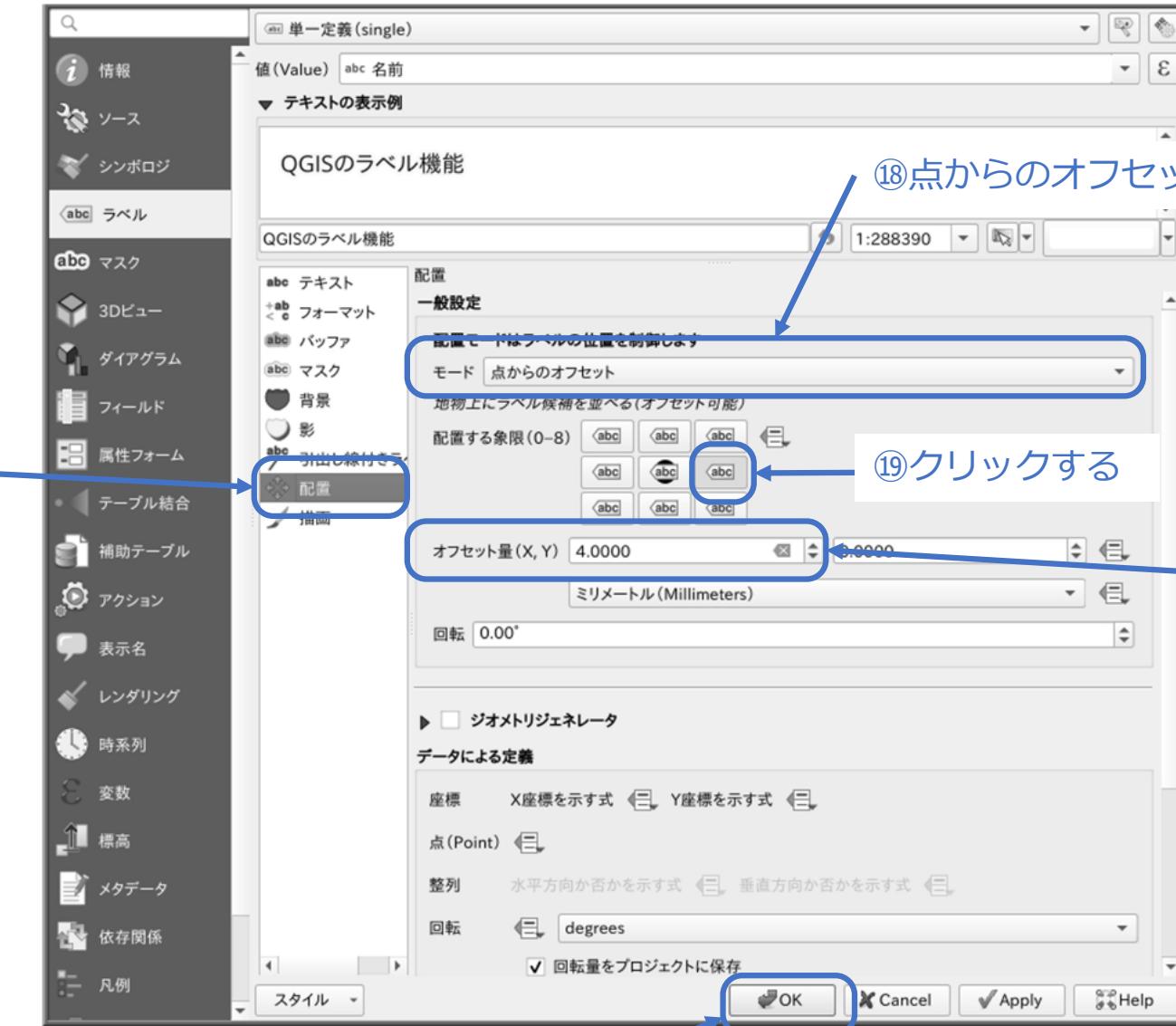
## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

⑯バッファをクリックする



## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

⑯配置をクリックする



⑰点からのオフセットを選択する

⑲クリックする

⑳4.0と入力する

㉑OKをクリックする

上記設定で、見やすく表示されると思います。

## 続き | 8-17. QGIS（地理空間情報の閲覧、編集、分析）を使いたい

□ CSVファイルからラインデータを読み込み、2点間の線を表示させたい

※前ページまでの続きで操作を説明します。

ラインデータを作成する。

```
⌘ vim line.csv
```

座標

```
LINESTRING(139.6963361 35.6650016, 139.8107 35.710063)  
LINESTRING(139.8107 35.710063, 139.627710 36.069147)
```

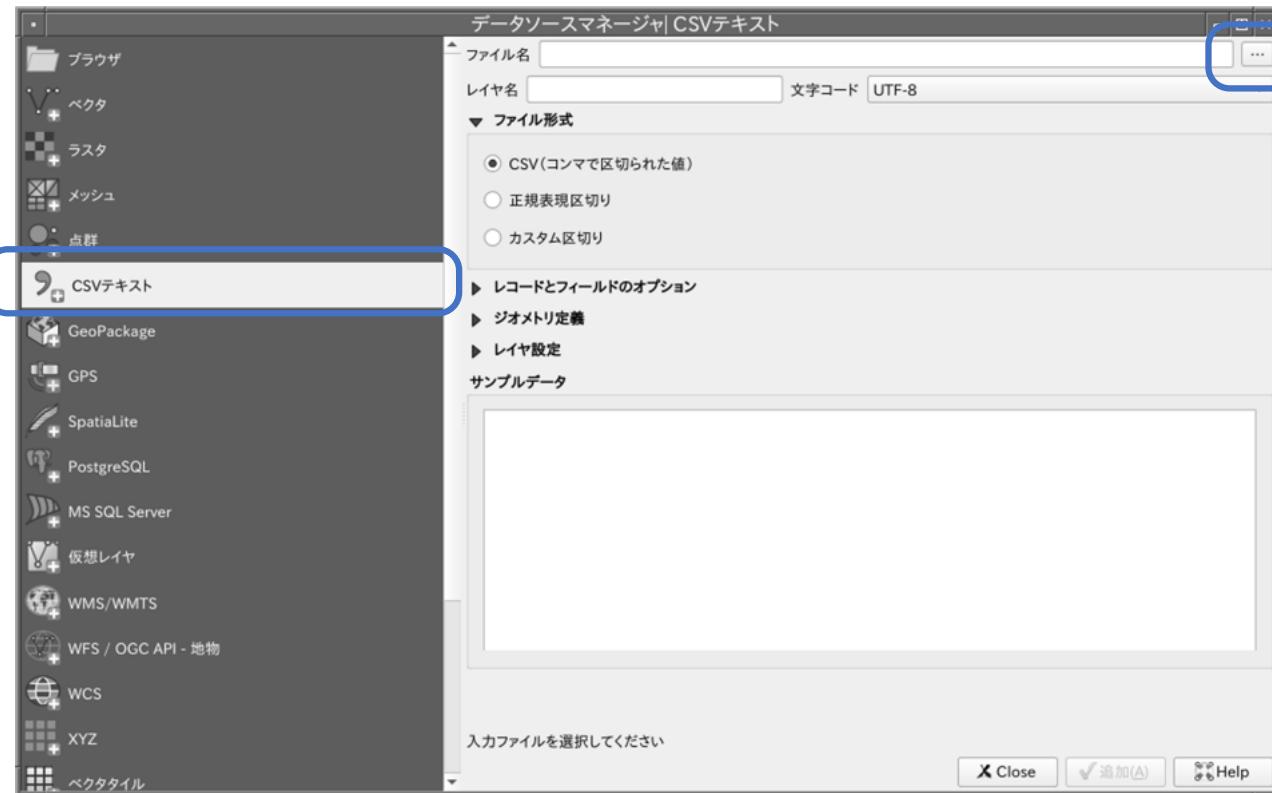
- ・経度1、緯度1、経度2、緯度2の順番
- ・経度と緯度の間はスペース
- ・経度1・緯度1と経度2・緯度2の間はカンマ

## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

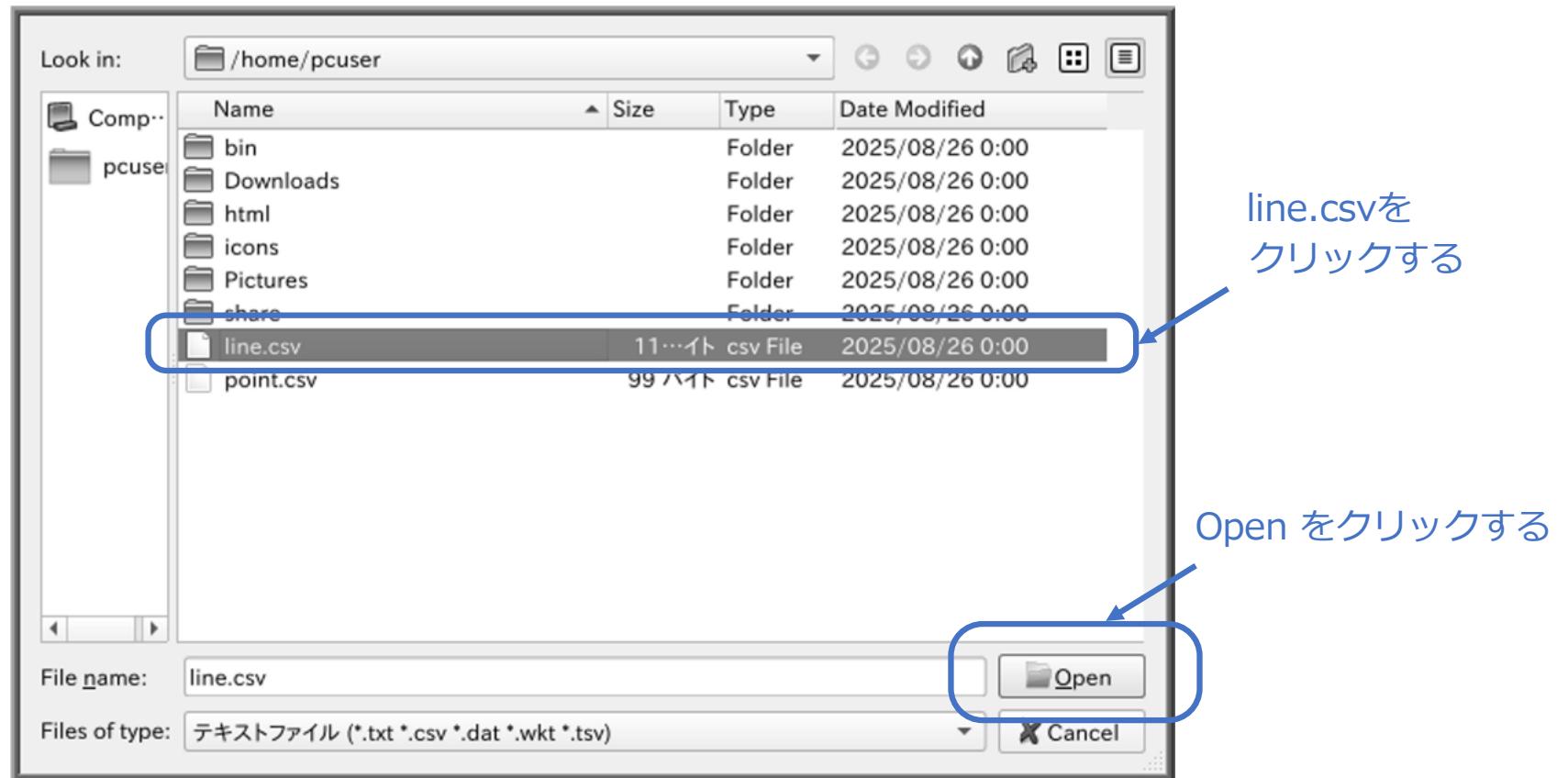
レイヤ(L)→レイヤを追加→CSVテキストレイヤを追加...を選択する

CSVテキストを  
クリックする

...をクリックする



## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい



## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

①ファイル形式を展開する

②カスタム区切りをクリックする

④ジオメトリ定義を展開する

⑤Well-known text(WKT)になっていることを確認する

③タブをチェックする

タブ区切りのファイルではないが、データ内にカンマが含まれており、誤認識を防ぐためにタブを選択している。

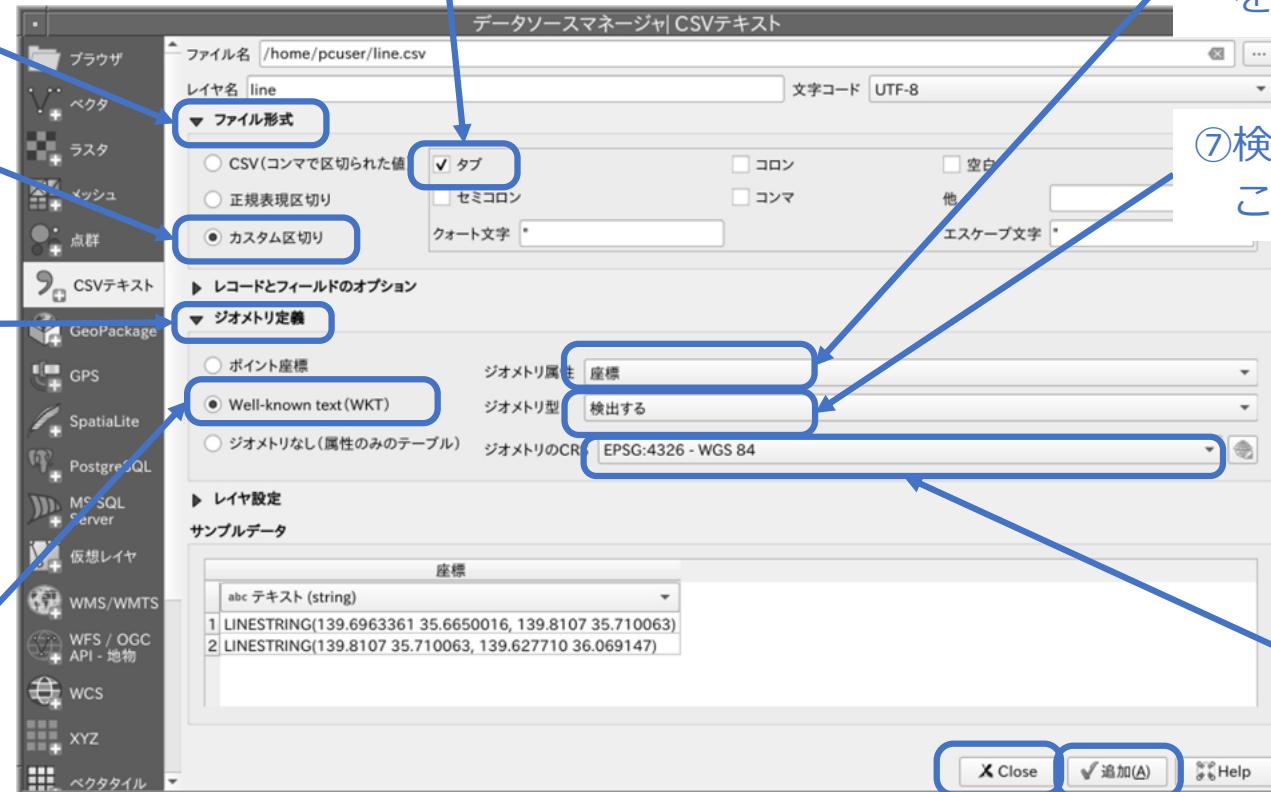
⑥座標になっていることを確認する

⑦検出するになっていることを確認する

⑧EPSG:4326になっていることを確認する

⑨追加(A)をクリックする

⑩closeをクリックする



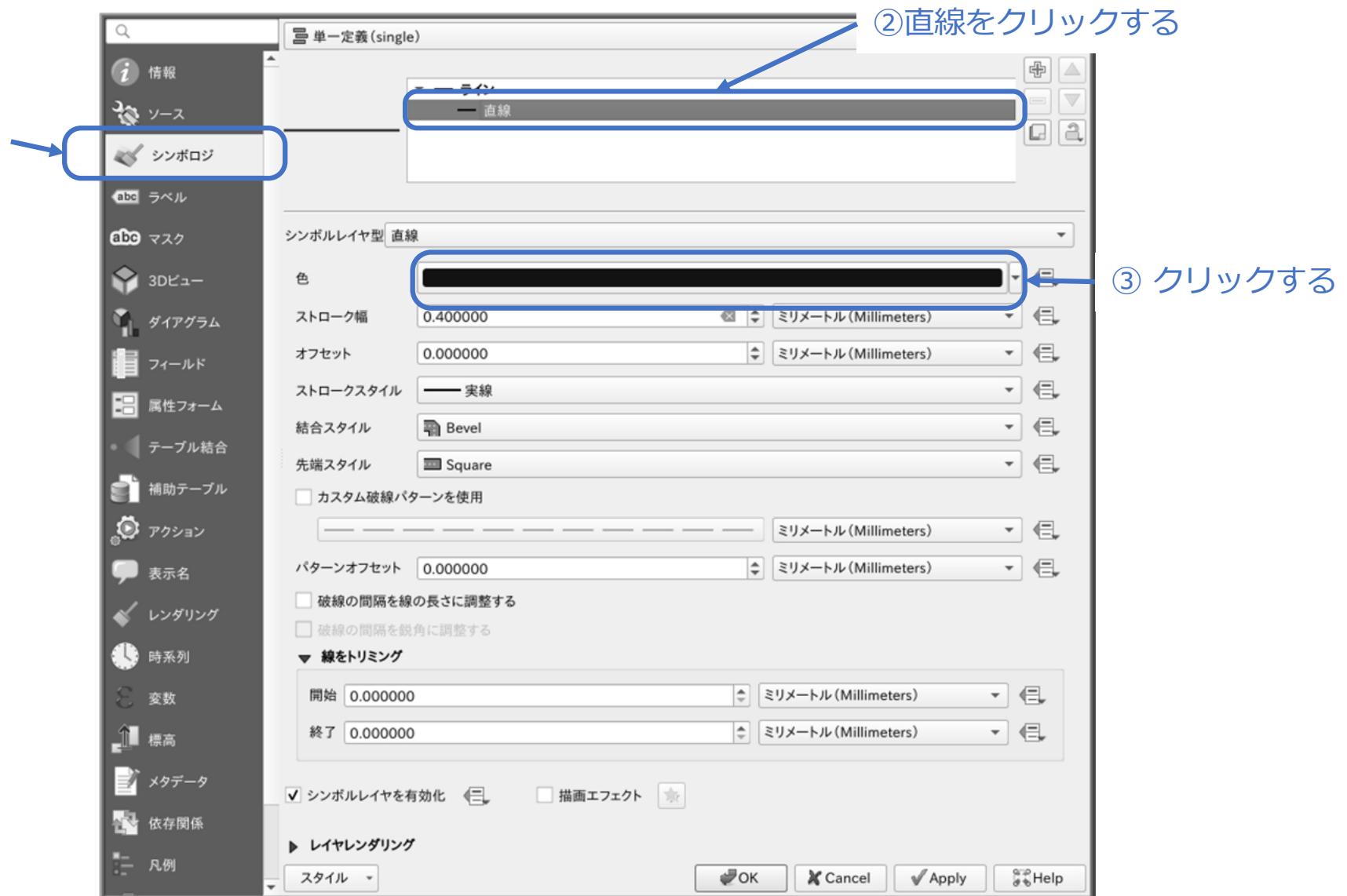
## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

※この先は見やすくするための設定です。



## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

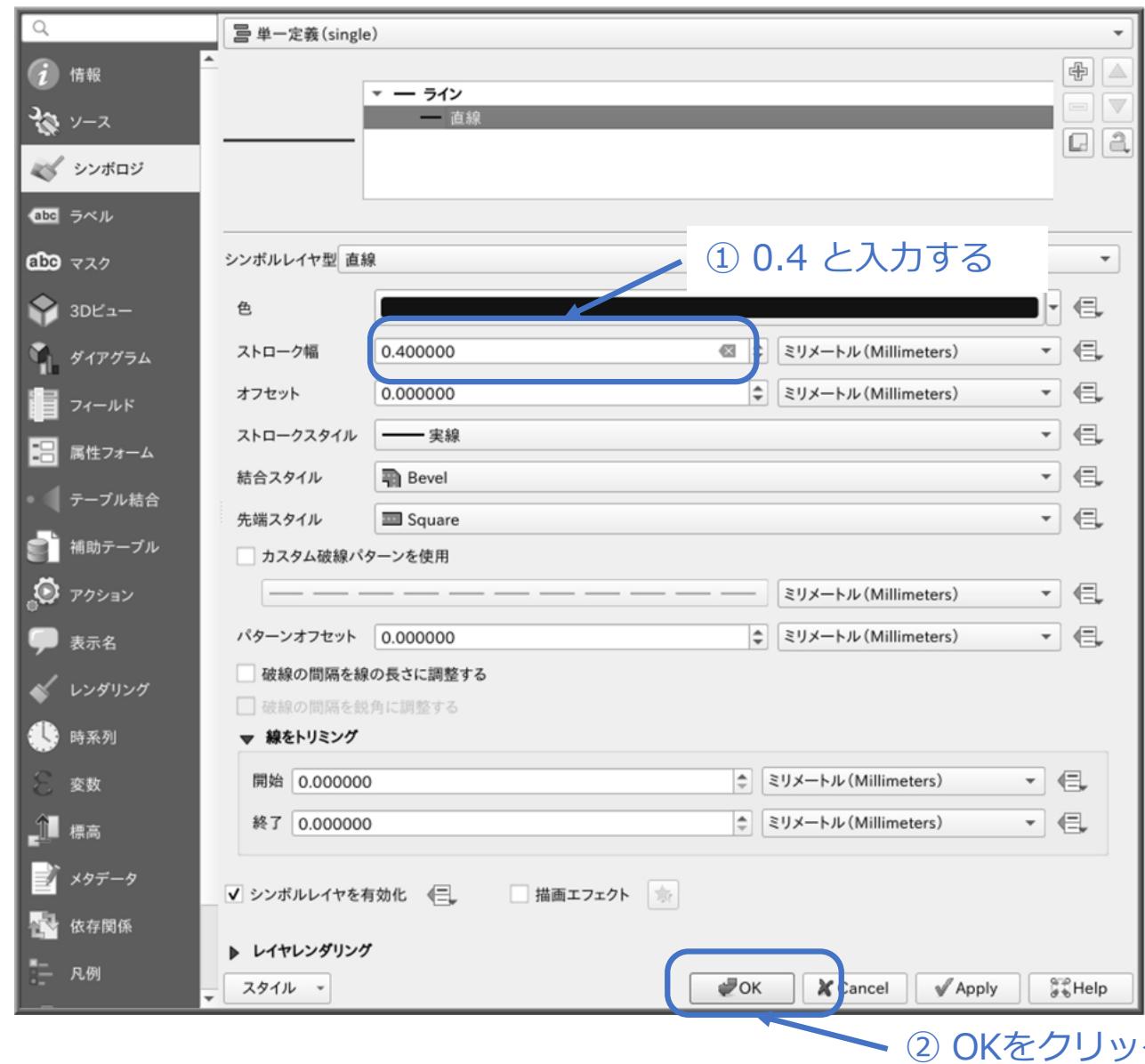
①シンボロジ  
をクリックす  
る



## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい



## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい



## 続き | 8-17. QGIS（地理空間情報の閲覧、編集、分析）を使いたい

補足：

- ・ ポイントデータ（point.csv）やラインデータ（line.csv）を直接編集（追加・修正・削除など）し、QGISのプロジェクトファイルを再度開くと、自動的に反映されます。

## 続き | 8-17. QGIS（地理空間情報の閲覧、編集、分析）を使いたい

□GPSロガーのデータ（NMEA0183形式）を読み込みたい

変換プログラムを作成する

○ vim conv\_nmea0183.py

```
import pynmea2
import csv
from datetime import datetime, time, timedelta

# 入力NMEAファイルと出力csvファイルの指定
input_file = "gps_data.nmea"
output_file = "gps_data.csv"

# JST (UTC+9) のオフセット
JST_OFFSET = timedelta(hours=9)

# CR+LFではなくLFを明示的に使用
with open(input_file, "r") as infile, open(output_file, "w") as outfile:
    writer = csv.writer(outfile, lineterminator="\n") # LFを指定
    writer.writerow(["Latitude", "Longitude", "Timestamp
(JST)"]) # ヘッダー行

    for line in infile:
        try:
            # NMEA文を解析
            msg = pynmea2.parse(line)

            # RMC文のみを処理
            if isinstance(msg, pynmea2.RMC):
                # タイムスタンプをJSTに変換
                if msg.datetime and msg.timestamp:
```

```
# msg.timestampをdatetime.timeに変換
timestamp_time = time(
    hour=msg.timestamp.hour,
    minute=msg.timestamp.minute,
    second=msg.timestamp.second,
)
# UTCでの日時を計算
utc_datetime = datetime.combine(msg.datetime,
    timestamp_time)

# JSTに変換
jst_datetime = utc_datetime + JST_OFFSET
# JSTタイムスタンプをフォーマット
timestamp = jst_datetime.strftime("%Y-%m-
%d %H:%M:%S")

else:
    timestamp = "N/A" # 日付や時刻がない場合

# 緯度、経度、JSTタイムスタンプを書き込み
writer.writerow([msg.latitude, msg.longitude,
    timestamp])

except pynmea2.ParseError:
    continue # 無効な行をスキップ
except AttributeError:
    # タイムスタンプが欠けている場合もスキップ
    continue
```

※本スクリプトは生成AIにて作成した。

## 続き | 8-17. QGIS（地理空間情報の閲覧、編集、分析）を使いたい

①ライブラリをインストールする

```
% python3.11 -m pip install pynmea2
```

②NMEA0183形式のファイルを、gps\_data.nmea という名前で準備する。

③変換プログラムを実行する

```
% python3.11 conv_nmea0183.py
```

gps\_data.csv が作成される。

④QGISの

レイヤ(L)

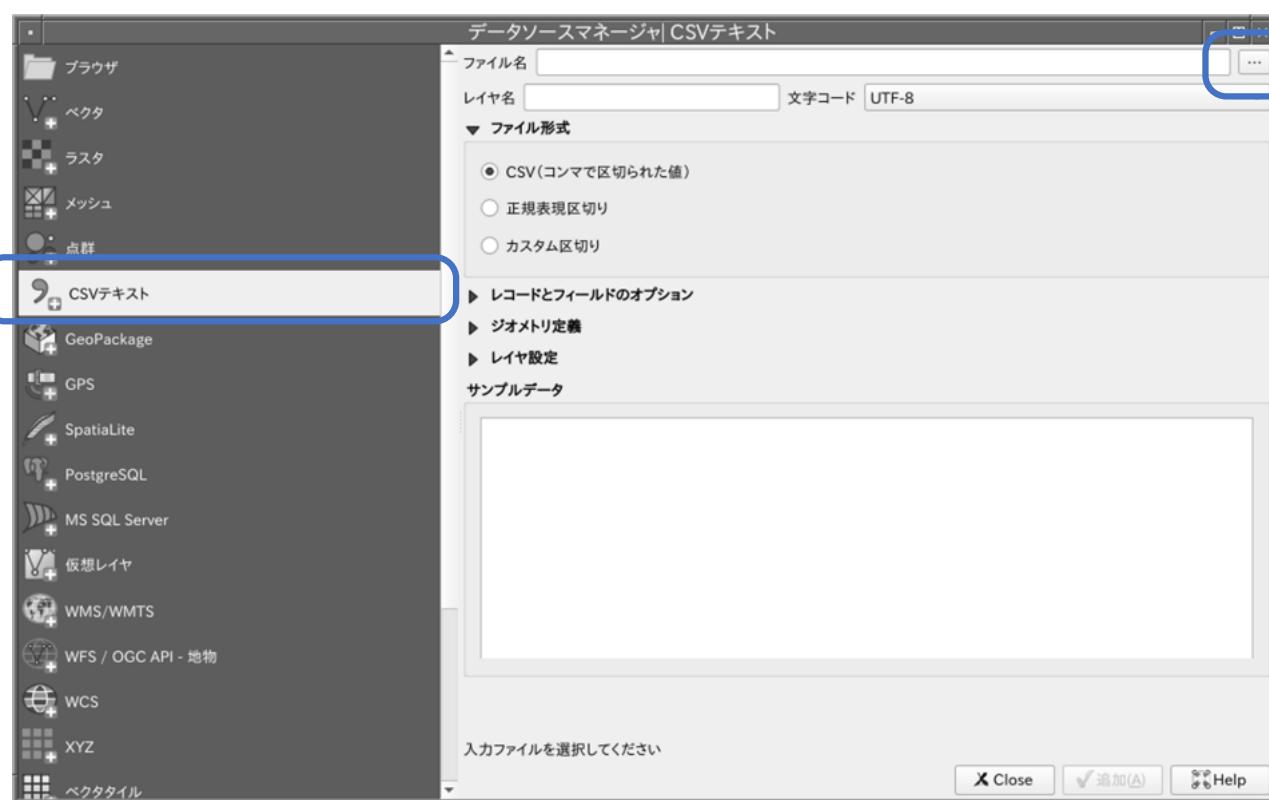
→レイヤを追加

→CSVテキストレイヤを追加...

を選択する

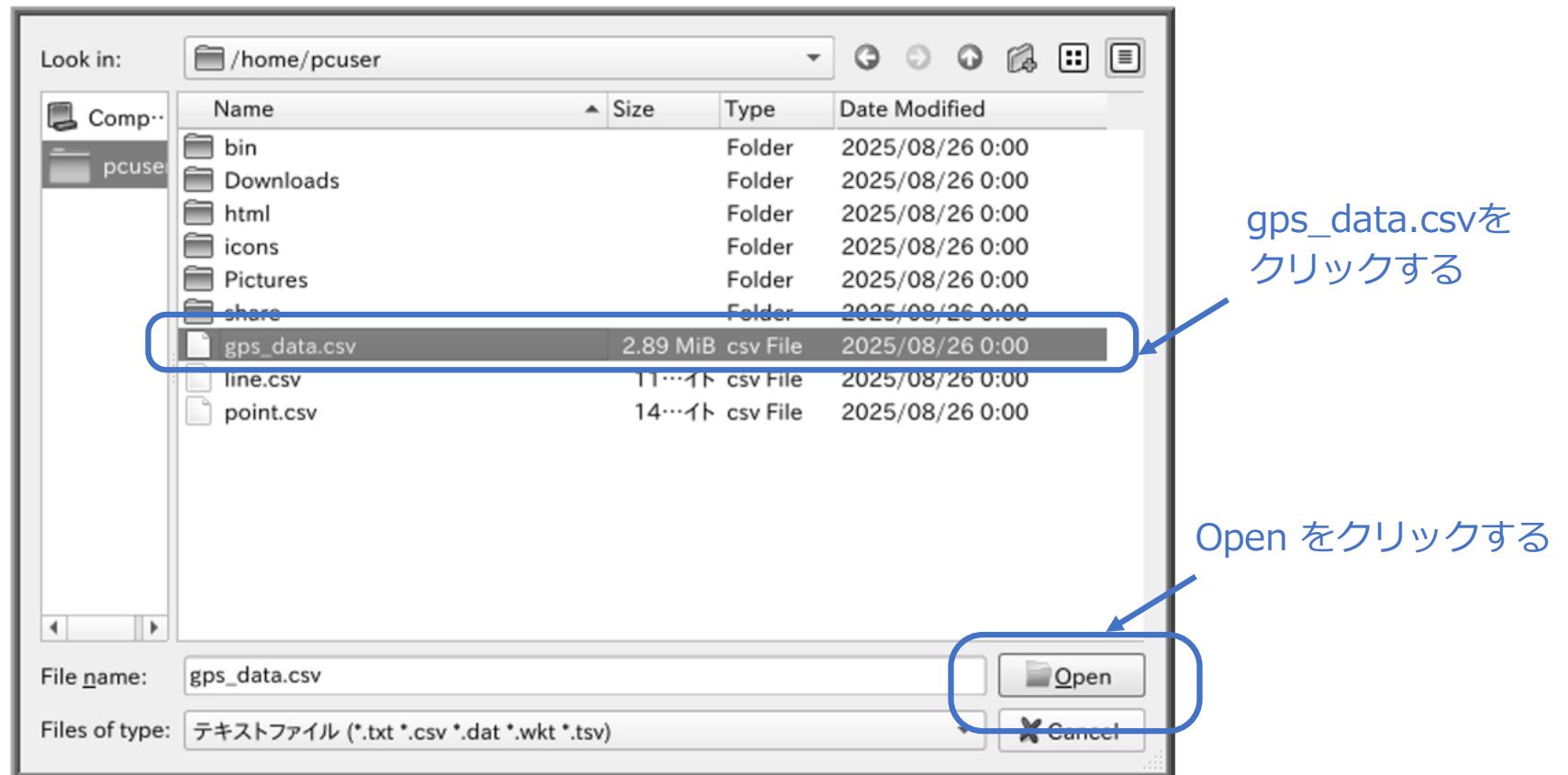
## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

CSVテキストを  
クリックする



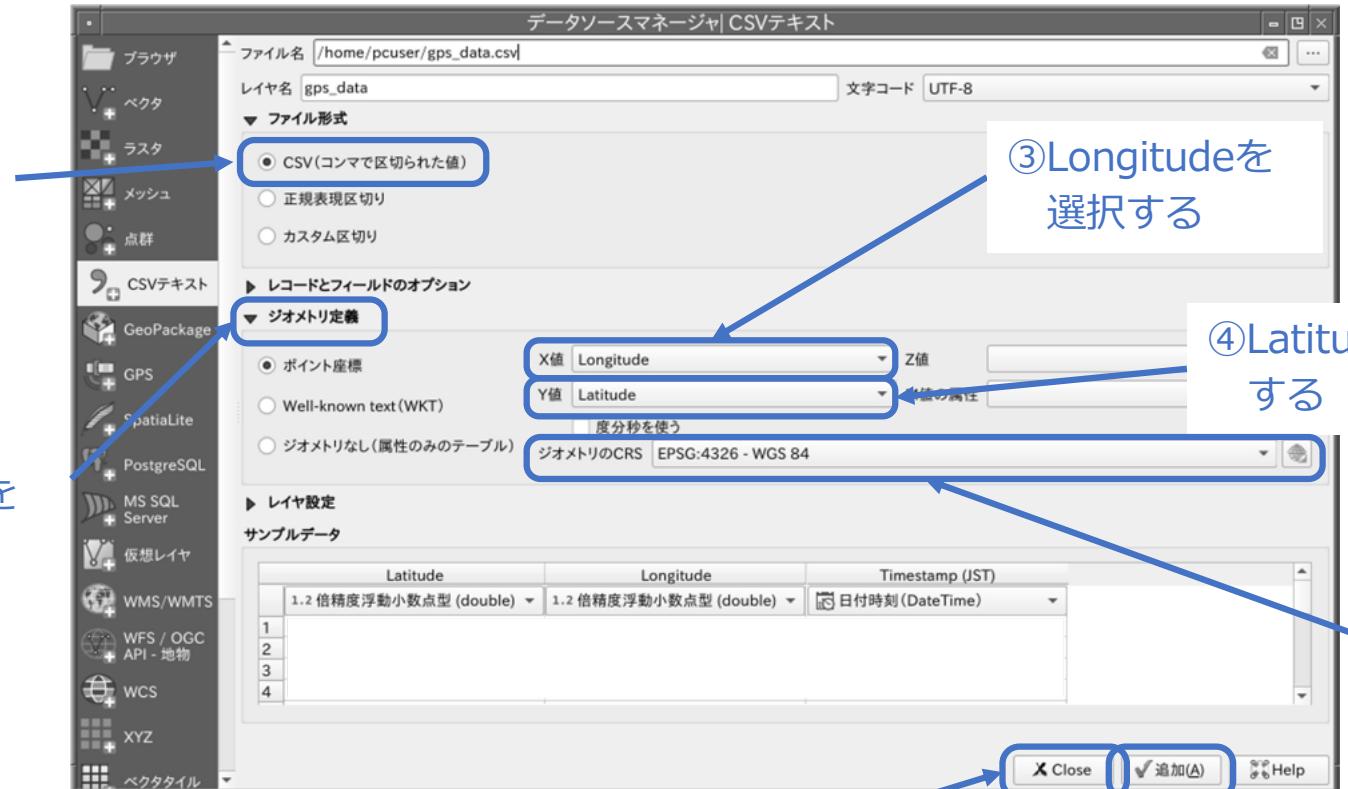
...をクリックする

## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい



## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

①CSVをチェックする。



②ジオメトリ定義を展開する。

③Longitudeを選択する

④Latitudeを選択する

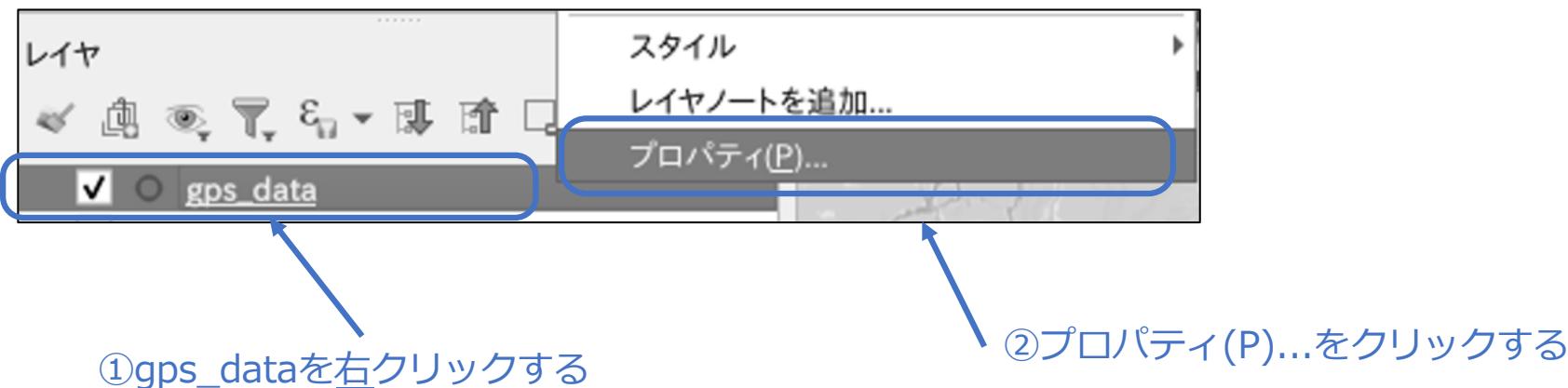
⑤EPSG:4326を選択する

⑥追加(A)をクリックする

⑦Closeをクリックする

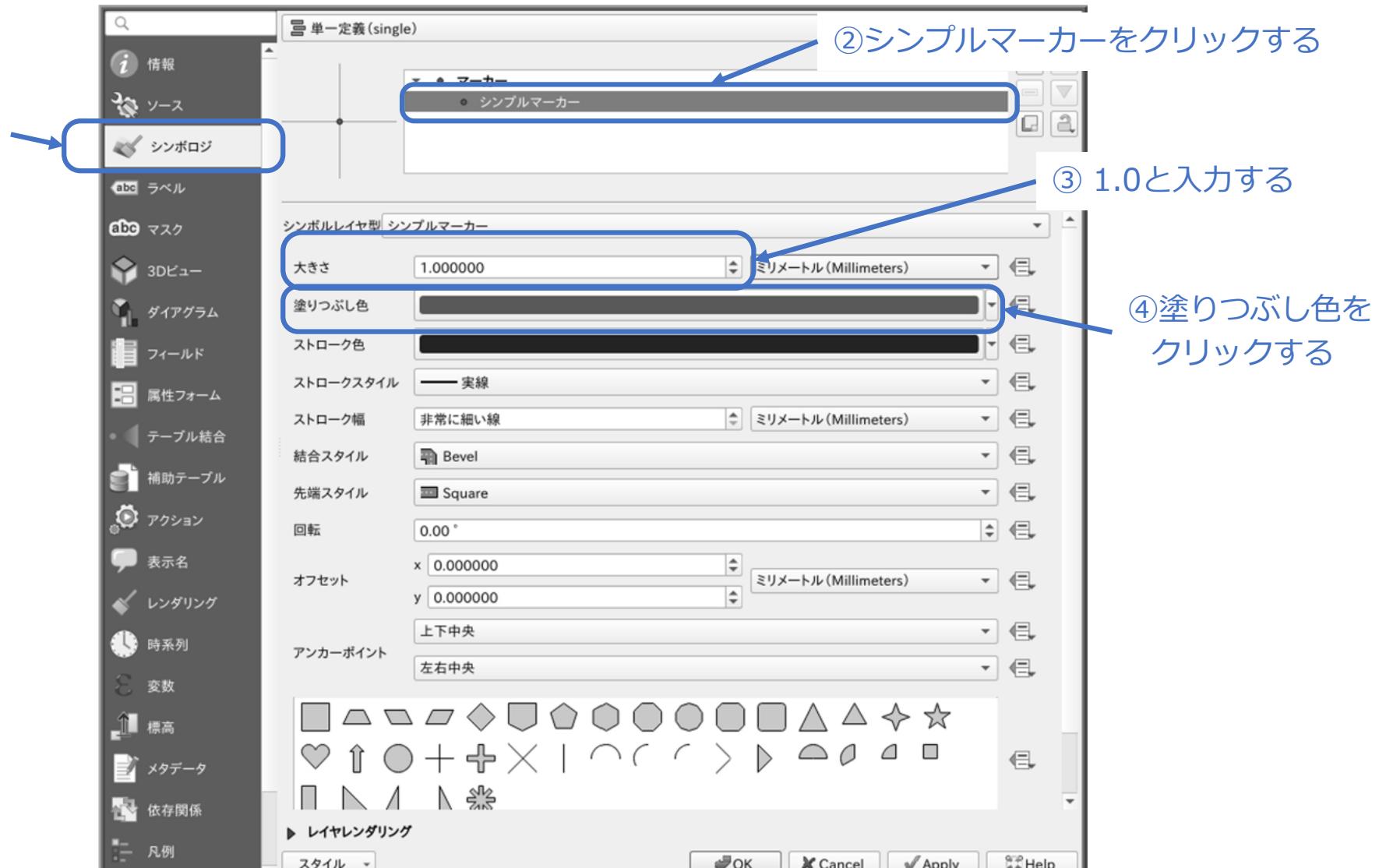
## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

※この先は見やすくするための設定です。

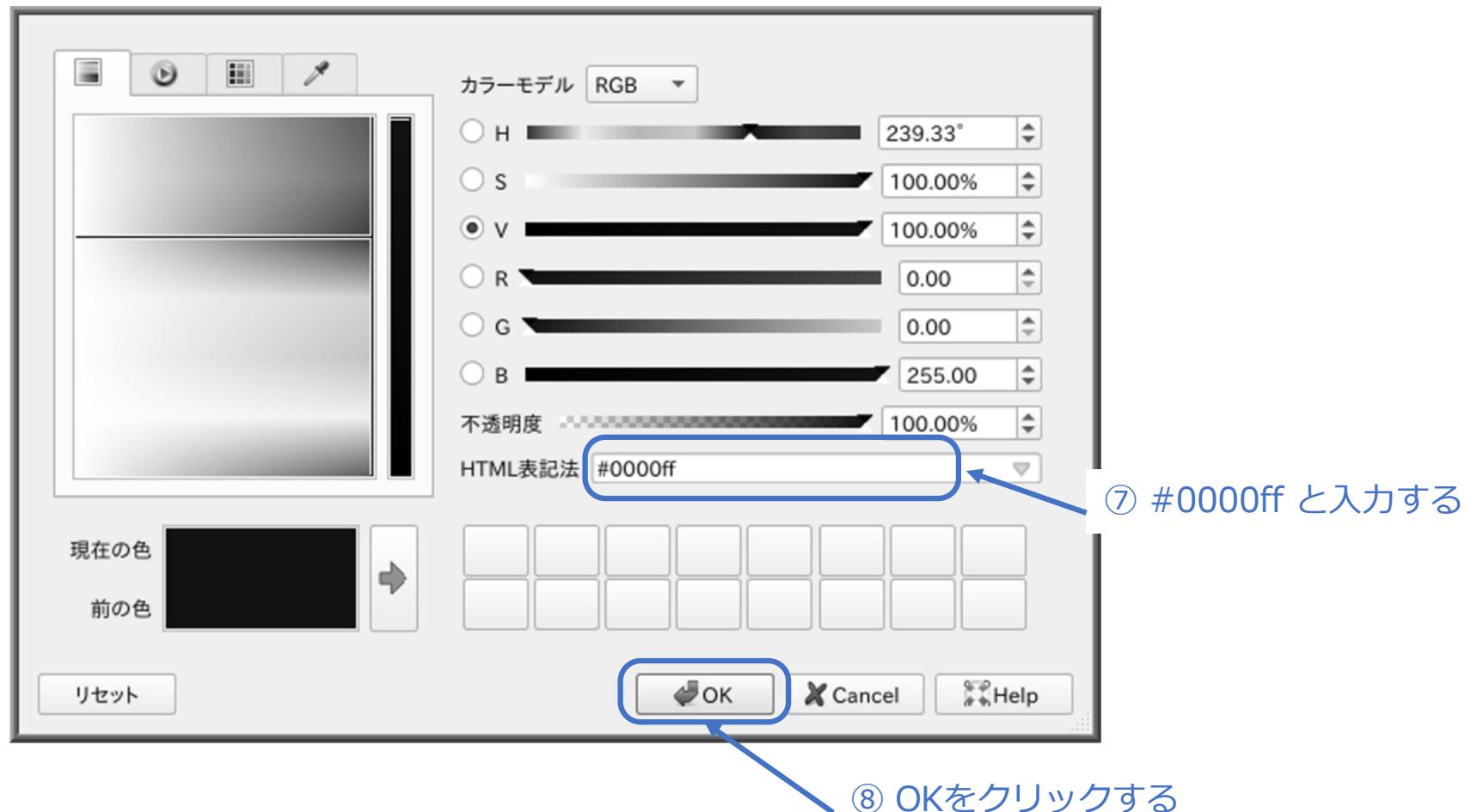


## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

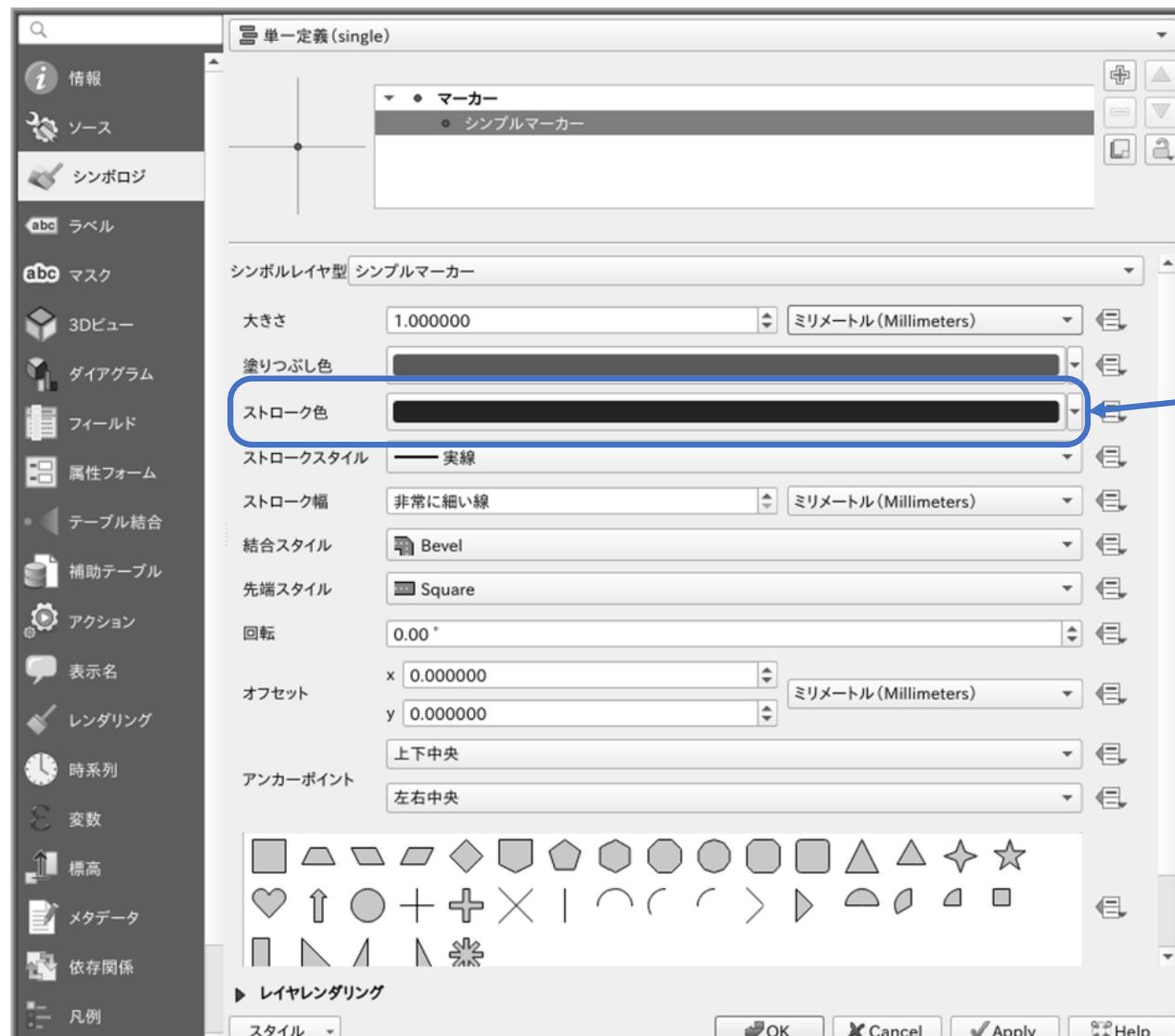
①シンボロジ  
をクリックす  
る



## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

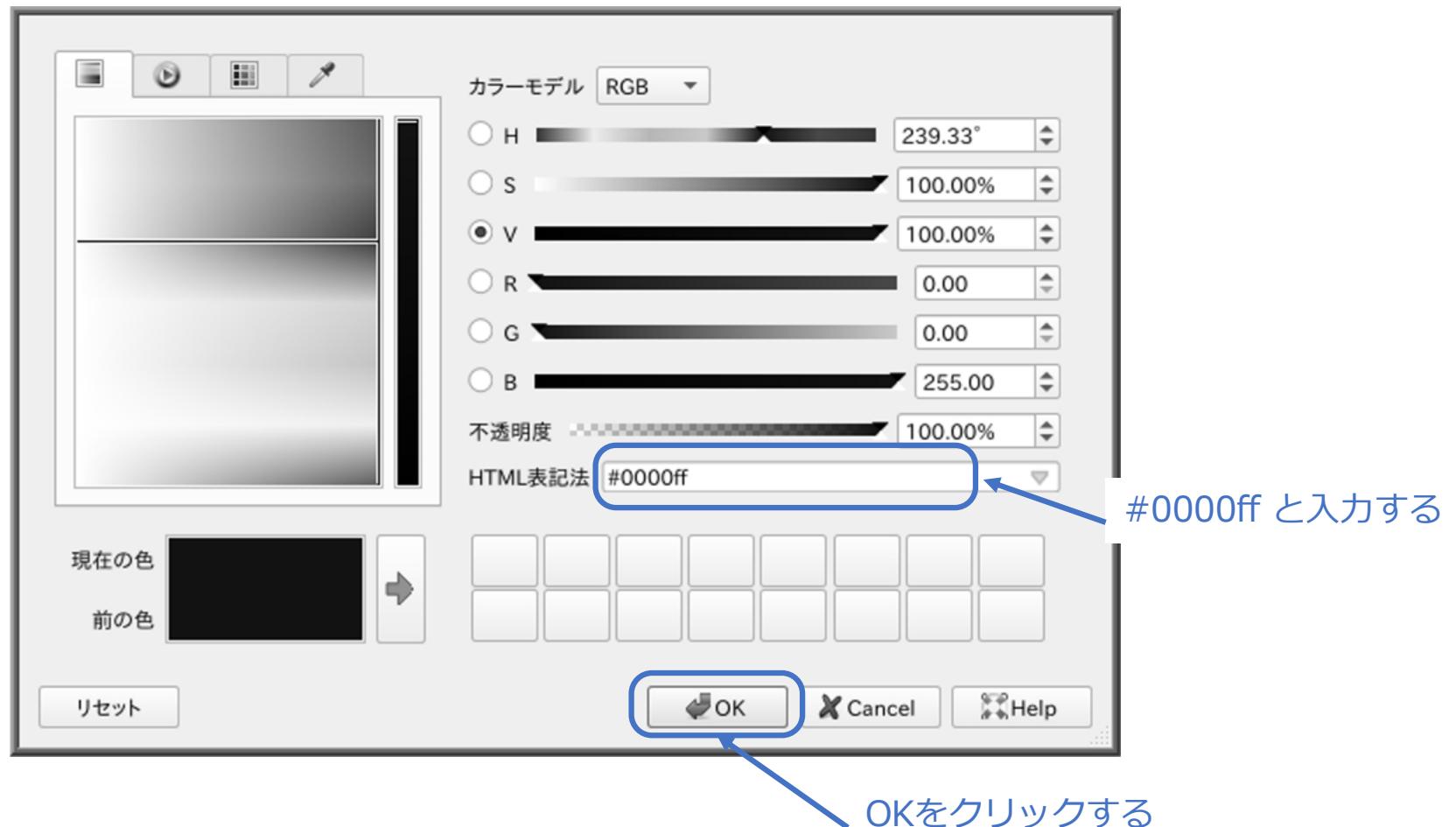


## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

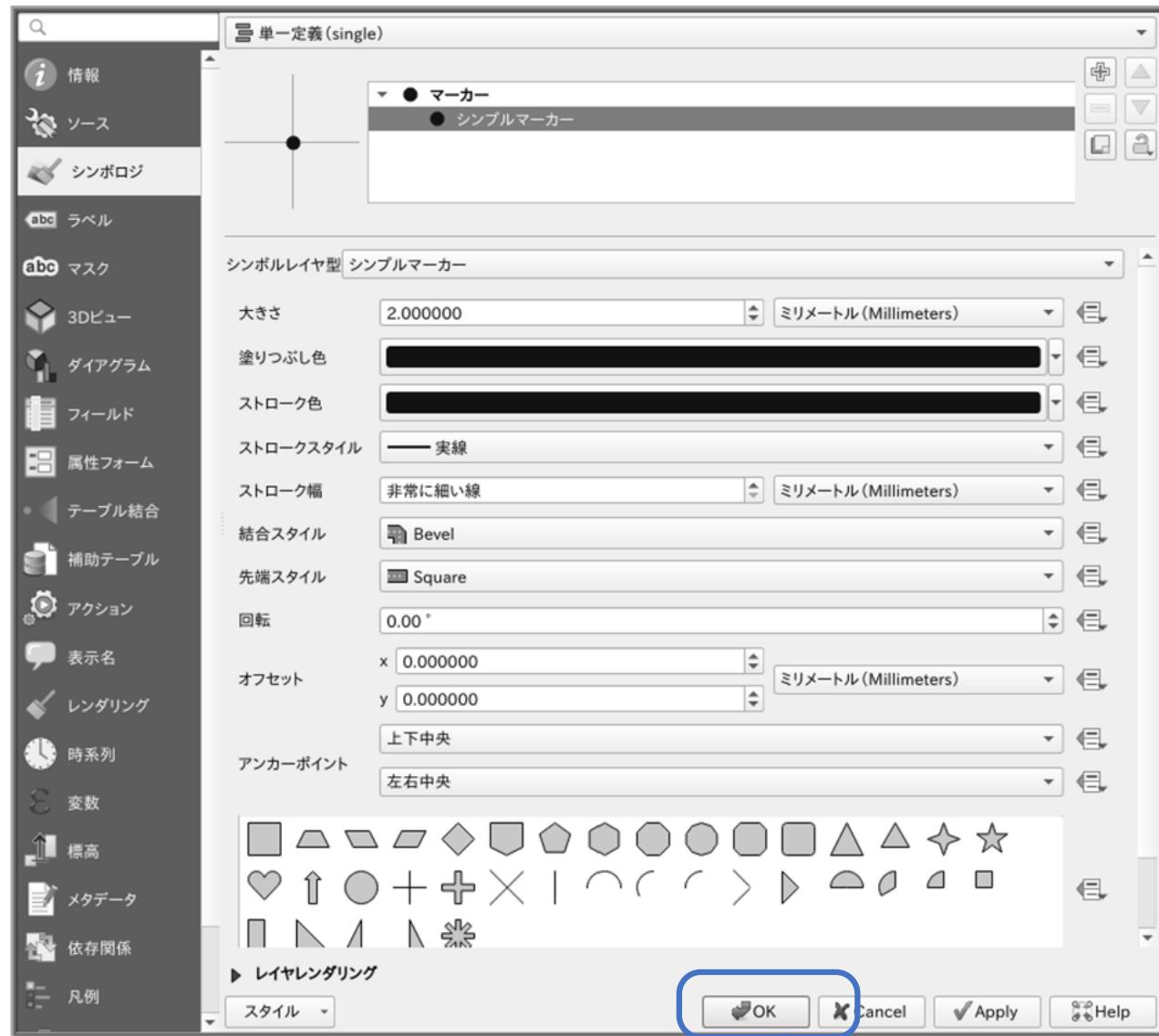


ストローク色を  
クリックする

## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい



## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

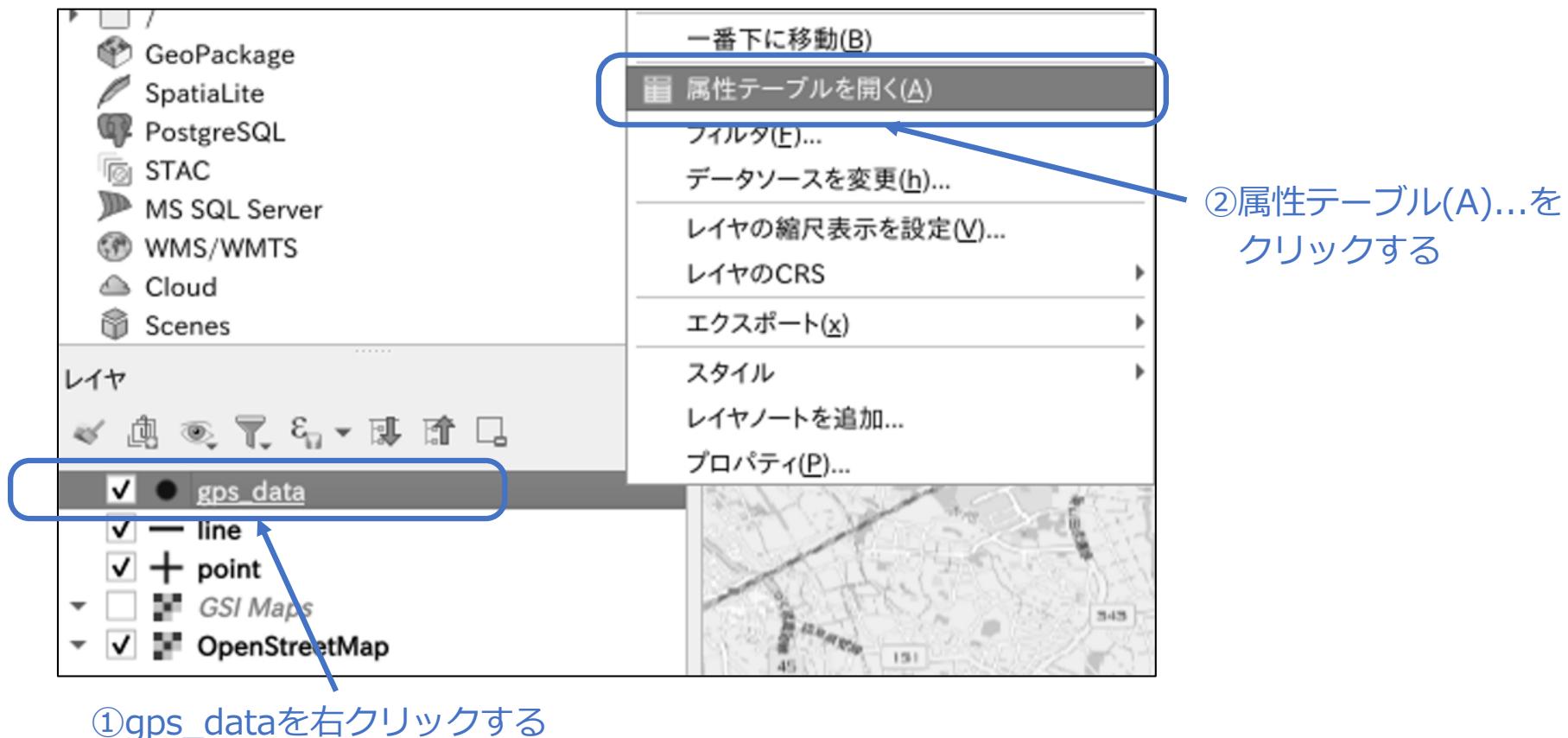


OKをクリックする

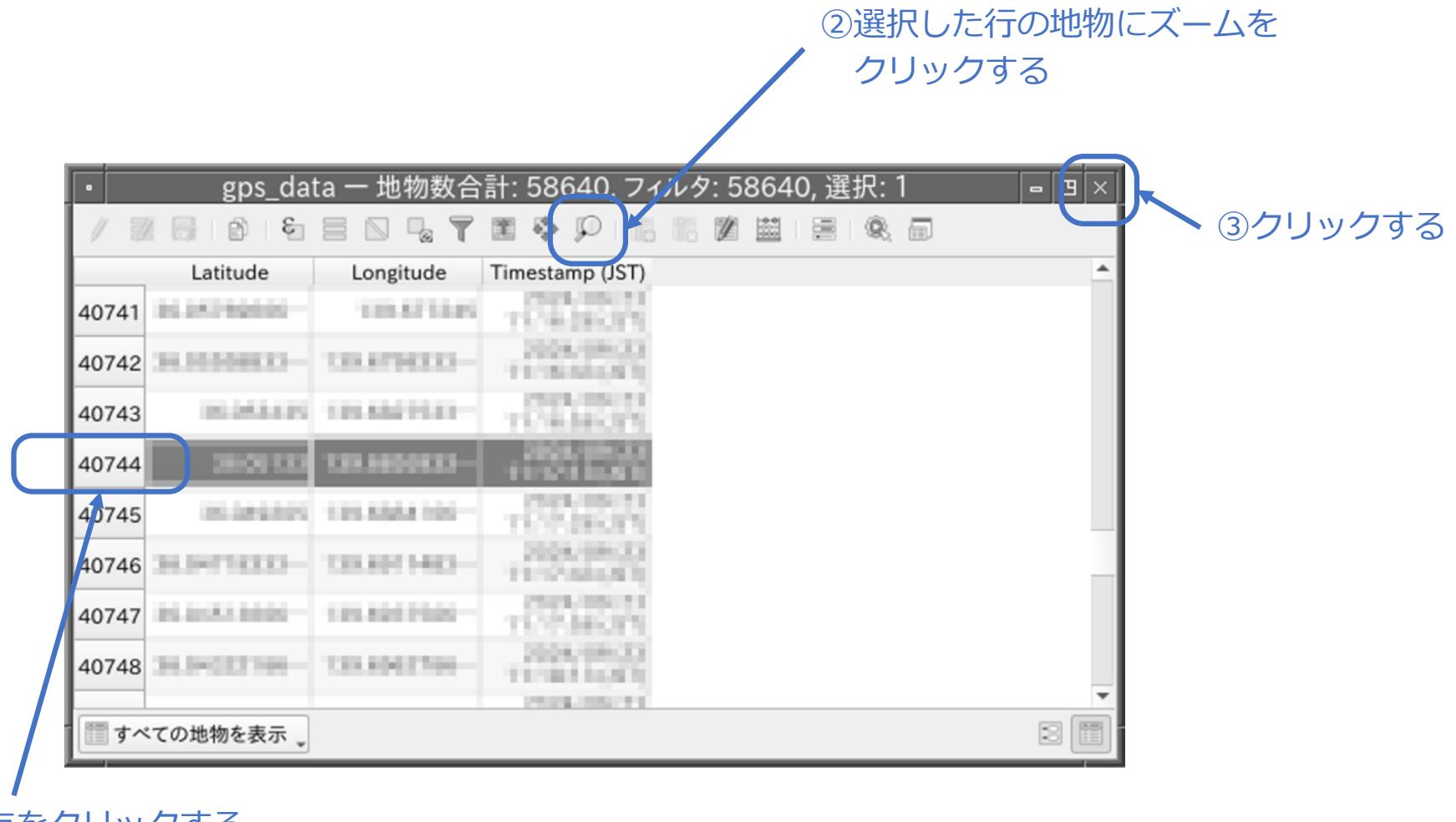
## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい

□GPSロガーの特定の地点を表示させたい

※前ページまでの続きで操作を説明します。



## 続き | 8-17. QGIS (地理空間情報の閲覧、編集、分析) を使いたい



①表示させたい地点をクリックする

## 8-18. 閲覧専用でメールを使いたい

普段はGmailなどのwebメールを使用するが、過去の（大量の）メールについては、ローカルで閲覧したい場合の説明をします。

- ・インストール

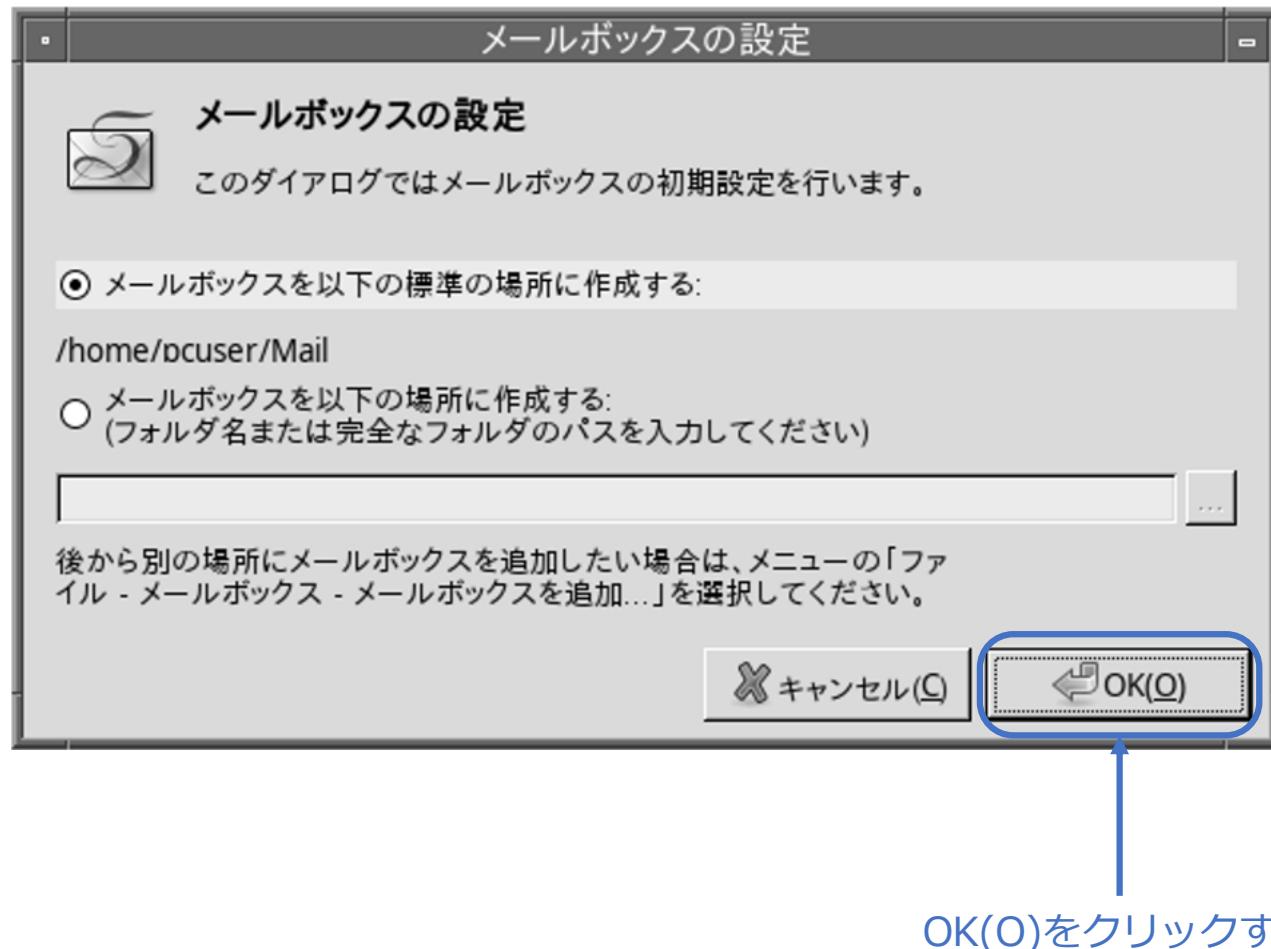
```
% sudo pkg install -y sylpheed
```

- ・起動

```
% sylpheed
```

## 続き | 8-18. 閲覧専用でメールを使いたい

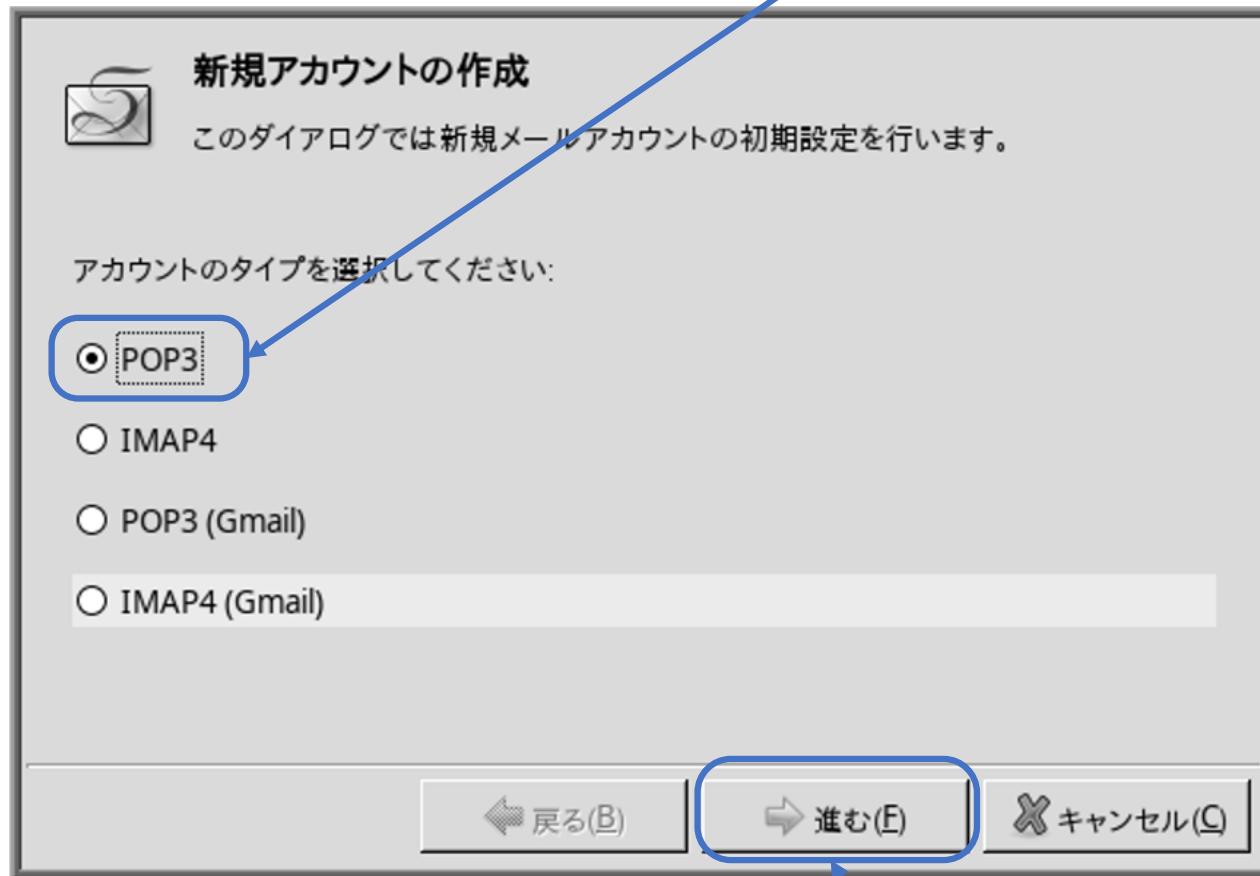
- ・ダミーアカウントの作成を行う



## 続き | 8-18. 閲覧専用でメールを使いたい

- ダミーアカウントの作成を行う（続き）

① POP3をチェックする

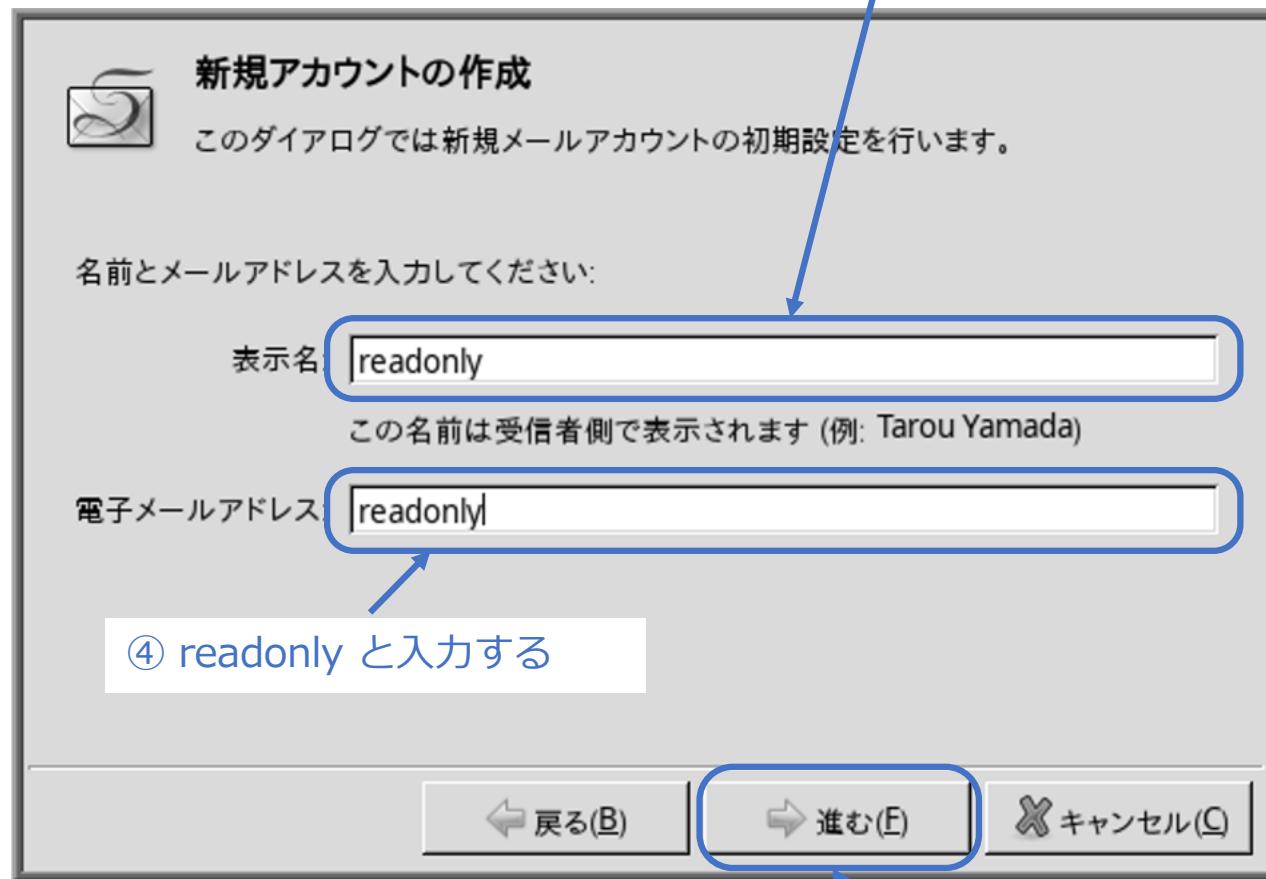


② 進む(F)をクリックする

## 続き | 8-18. 閲覧専用でメールを使いたい

- ・ダミーアカウントの作成を行う（続き）

③ readonly と入力する

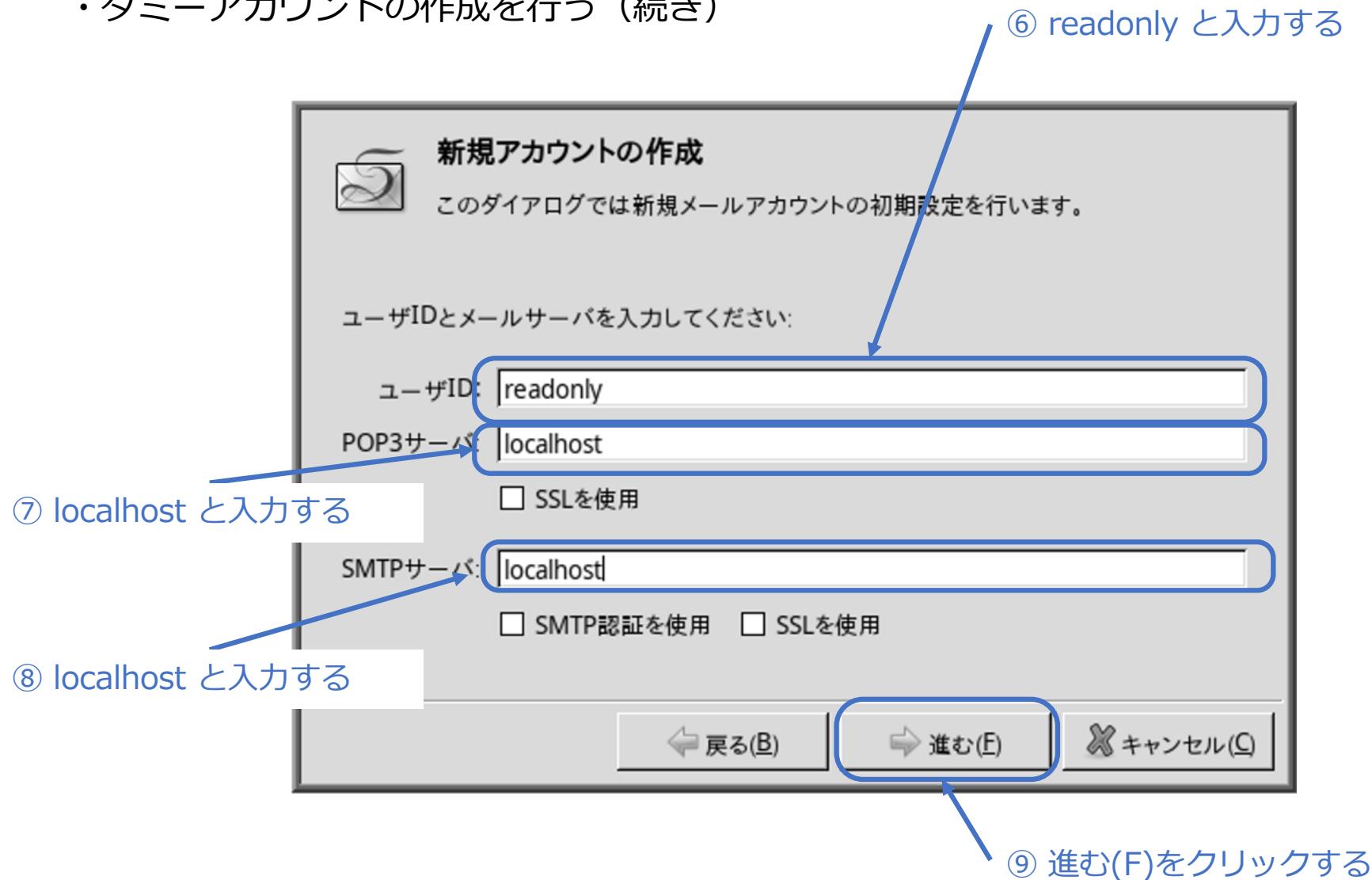


④ readonly と入力する

⑤ 進む(F)をクリックする

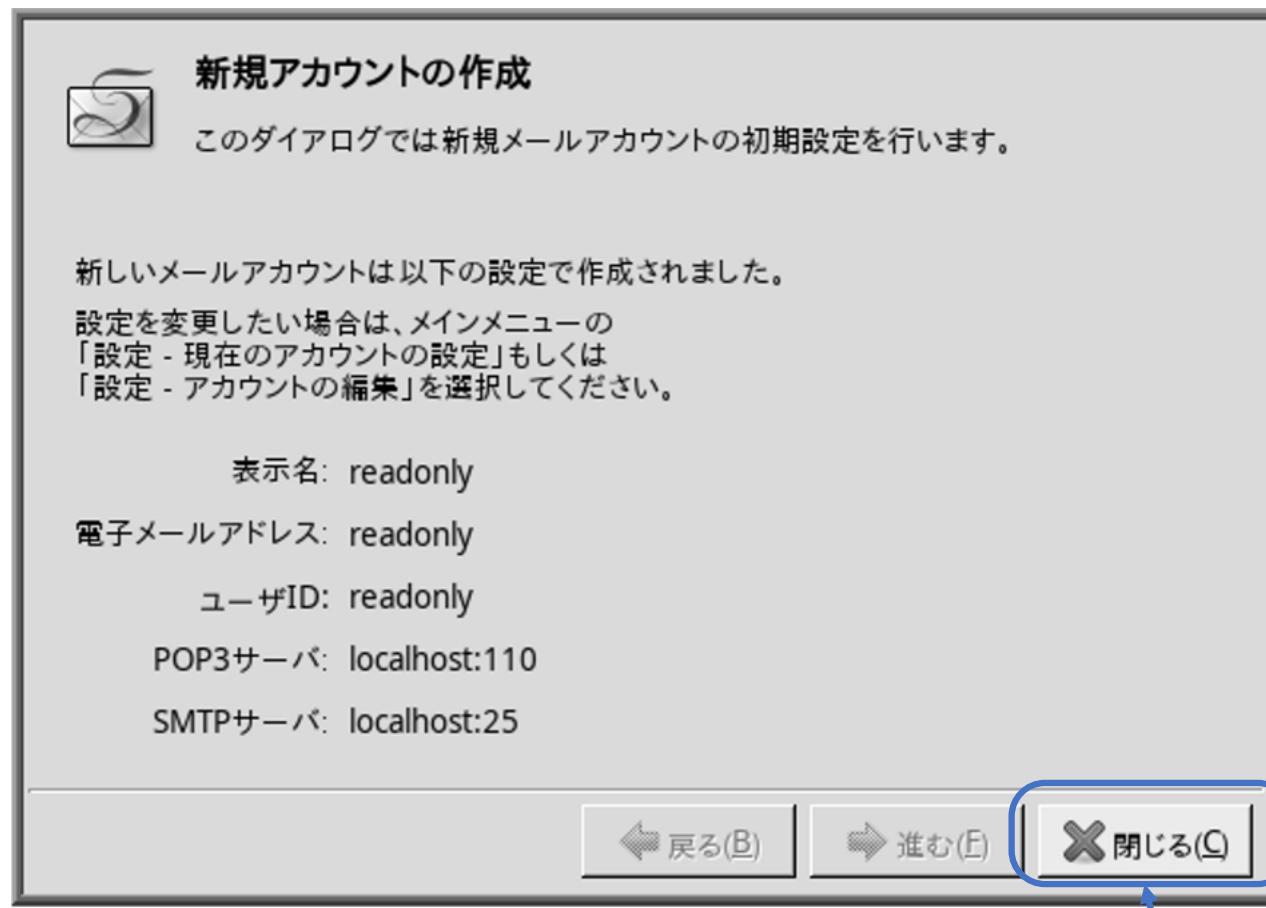
## 続き | 8-18. 閲覧専用でメールを使いたい

- ・ダミーアカウントの作成を行う（続き）



## 続き | 8-18. 閲覧専用でメールを使いたい

- ・ダミーアカウントの作成を行う（続き）



⑩ 閉じる(C)をクリックする

# 続き | 8-18. 閲覧専用でメールを使いたい

## □ Macのメールアプリからインポートしたい

以下、概要を説明します。

### Mac側作業

1. Macのメールアプリにて、任意のメールボックスを右クリック→メールボックスを書き出すを選択。書き出したいフォルダーを選択する。
2. メールボックス名.mboxというフォルダーが作成されるので、このフォルダーをFreeBSDのパソコンにコピーする。

### FreeBSD側作業

1. FreeBSD上のsylpheedを起動する。
2. ファイル(F)→フォルダ(F)→新規フォルダを作成(N)...を選択し、新規フォルダ名を入力する。
3. ファイル(F)→メールデータをインポート(I)...を選択し、①インポート元にインポートするメールのmboxファイルを選択、②インポート先フォルダーを選択して、OK(O)ボタンを押す。

※メールデータは ~/Mail に保存される。

※ sylpheedでは、UNIX mbox、eml（フォルダ）、Outlook Express（dbx）形式が読み取り可能。

※ mbox形式なのにうまくインポートできない場合（文字化けする、一件しか読み込まれないなど）は、文字コード：JISコード、改行コード：LF、への変換を行う（旧MacOS9時代のメールからエクスポートした場合に、このような事例あり）。

## 続き | 8-18. 閲覧専用でメールを使いたい

### □ フォントを変更したい

- ・メッセージ本文

設定(C) → 全般の設定(C)... → 表示タブ → テキストフォント

- ・上記以外 (メニューなど)

```
% vim ~/.sylpheed-2.0/gtkrc
```

```
gtk-font-name = "Noto Sans JP" ← 新規に追加する
```

Noto Sans JP 12 のようにフォントサイズを付与しても可

## 続き | 8-18. 閲覧専用でメールを使いたい

□ スレッド表示をやめたい

表示(V) → スレッド表示(R) を選択することで解除される。

□ メールを検索したい



クリックする

または、Shift + Ctrl + F を押す。

## 8-19. webカメラを利用したい

Zoomなどで使用するwebカメラの設定を行います。

- ・インストール

```
% sudo pkg install -y webcamd
```

- ・設定

```
% sudo vi /etc/rc.conf
```

:

webcamd\_enable="YES"



← 追加する

- ・ユーザーグループの追加設定

```
% sudo pw groupmod webcamd -m pcuser
```

# 続き | 8-19. webカメラを利用したい

- 再起動

```
% sudo shutdown -r now
```

- webカメラを接続する（パソコン内蔵webカメラの場合は不要）

- webカメラが認識されているかの確認

```
% ls -l /dev/video*
```

```
crw-rw---- 1 webcamd webcamd 0x9c Aug 26 08:26 /dev/video0
crw-rw---- 1 webcamd webcamd 0x9d Aug 26 08:26 /dev/video1
```

- Zoomのミーティングテスト（<https://zoom.us/ja/test/>）に接続し、テストを行う（「ブラウザから参加」リンクをクリックする）

## 8-20. 外付けカメラを使いたい

### ■ 外付けカメラを使いたい

USBに接続されたカメラから、写真を撮影したり映像を録画することができます。**基本的な設定は、「8-19. webカメラを使いたい」と同じ**です。次のページからは、具体的な活用例について説明します。

# 続き | 8-20. 外付けカメラを使いたい

- 外付けカメラの画像をキャプチャしたい（静止画1枚）

```
% ffmpeg -f video4linux2 -i /dev/video0 -vframes 1 snapshot.jpg
```

※ 機種によってはカメラ起動後の自動露出・ホワイトバランス調整・フォーカスに時間がかかるため、上記方法では綺麗にキャプチャできない場合があります。その対策については、以下となります。

- 外付けカメラの画像をキャプチャしたい（画質安定版、静止画1枚）

```
% ffmpeg -f video4linux2 -framerate 30 -i /dev/video0 -  
-ss 3 -vframes 1 -y snapshot.jpg
```

↑  
3秒後

- 外付けカメラの画像をキャプチャしたい（画質安定版、静止画、1秒ごと）

```
% ffmpeg -f video4linux2 -framerate 30 -i /dev/video0 -  
-vf "fps=1" snapshot_%04d.jpg
```

※ qキーで終了します。

## 続き | 8-20. 外付けカメラを使いたい

□外付けカメラの画像をキャプチャしたい  
(画質安定版、静止画、1秒ごと、日時付き)

IPAゴシックのフォントファイル指定

```
% ffmpeg -f video4linux2 -framerate 30 -i /dev/video0 ¥
-vf "fps=1, ¥
drawtext=fontfile=/usr/local/share/fonts/ipa/ipag.otf: ¥
text=' %{localtime} ': fontcolor=white: fontsize=24: box=1: ¥
boxcolor=black@0.5: x=w-tw-8: y=h-th-8" ¥
snapshot _%05d.jpg
```

※ qキーで終了します。

※ 作成された画像ファイルは firefox <ファイル名> や feh <ファイル名> で閲覧できます。

## 続き | 8-20. 外付けカメラを使いたい

- 外付けカメラの映像を録画したい（動画、音声なし、30秒間）

```
% ffmpeg -f video4linux2 -framerate 15 -i /dev/video0 -t 00:00:30 ¥  
output.mp4
```

↑  
1秒間に15コマ

↑  
30秒間

※ 容量試算：640x480ピクセル、30秒間、フレームレート15fps=約8MByte

- 外付けカメラ録画に日時を埋め込みたい（動画、音声なし、30秒間、日時付き）

```
% ffmpeg -f video4linux2 -framerate 15 -i /dev/video0 -t 00:00:30 ¥  
-vf "drawtext=fontfile=/usr/local/share/fonts/ipa/ipag.otf:  
text='%{localtime}': fontcolor=white: fontsize=24: box=1: ¥  
boxcolor=black@0.5: x=w-tw-8: y=h-th-8" ¥  
output.mp4
```

※ 作成された動画ファイルは firefox <ファイル名> で閲覧できます。

# 続き | 8-20. 外付けカメラを使いたい

## □ 動きを検出して映像を録画したい（動体検知）

- ・動体検知カメラソフトをインストールする

```
% sudo pkg install -y motion
```

- ・設定ファイルを作成する

```
% vim ~/motion.conf
```

```
video_device /dev/video0 ← カメラのデバイス名  
framerate 15 ← フレームレート  
picture_output off ← 動画を出力する  
movie_output on ← 動画の保存先  
target_dir /home/pcuser/Videos ← 日付・時刻を画像の右下に記録する  
text_right %Y-%m-%d %T ← 動き検知終了後、3秒間録画し続ける  
post_capture 30 ←
```

※ この他にもパラメーターは多数あります。詳細はChatGPTまで。

# 続き | 8-20. 外付けカメラを使いたい

- ・動画保存フォルダーを作成する

```
% mkdir ~/Videos
```

- ・ソフトを起動する

```
% motion
```

※ 終了は、CTRL+C。xx-yyyymmddhhmmss.mkvの形式の動画ファイルが作成される。

- ・録画した動画を閲覧する

```
% cd ~/Videos
```

```
% ffmpeg -i xx-20250826120000.mkv -c copy output.mp4
```

```
% firefox output.mp4
```

## 続き | 8-20. 外付けカメラを使いたい

### □ 外付けカメラの映像をそのまま見たい

- ・メディアプレイヤーアプリをインストールする

```
% sudo pkg install -y mpv
```

- ・アプリを起動する

```
% mpv av:///v412:/dev/video0 ¶  
--demuxer-lavf-o=video_size=640x480
```

入力フォーマット

カメラデバイスファイル

解像度指定（カメラデバイスによつて指定できる解像度は決まっている。  
確認方法は次ページを参照）

# 続き | 8-20. 外付けカメラを使いたい

補足：カメラデバイスが対応する解像度を確認する

```
% ffmpeg -f v4l2 -list_formats all -i /dev/video0
```

```
[video4linux2,v4l2 @ 0xb4921429280] Compressed:      mjpeg :          Motion-JPEG : 1920x1080 1280x960  
1280x720 1024x576 800x600 640x480 640x360  
[video4linux2,v4l2 @ 0xb4921429280] Raw       :      yuyv422 :          YUYV 4:2:2 : 640x480 640x360
```

Raw側が対応解像度と思われる

## 続き | 8-20. 外付けカメラを使いたい

### □ 外付けカメラの映像をYouTubeで配信したい

ここでは、OBSなどの配信ソフトを使用せず、FFmpegにより、映像のみ・音声なしで配信する例を紹介します（ライブカメラ風）。なお、YouTube側の操作説明は一部省略しています。

- ① YouTubeにログインする（画面いっぱいにウインドウを表示させてください）。



## 続き | 8-20. 外付けカメラを使いたい

「タイトル」の右側



④クリックする

⑤タイトルを入力する

⑥タイトルを入力する

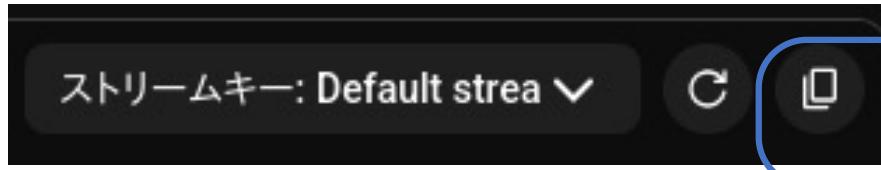
## 続き | 8-20. 外付けカメラを使いたい

スクロール  
させて



## 続き | 8-20. 外付けカメラを使いたい

- ・ストリームキーの確認



⑩クリックする

- ⑪配信する

貼り付ける

```
% ffmpeg %
-f v4l2 -framerate 30 -video_size 640x480 -i /dev/video0 %
-f lavfi -i anullsrc=channel_layout=stereo:sample_rate=44100 %
-shortest %
-vcodec libx264 -pix_fmt yuv420p %
-preset veryfast -g 60 -b:v 2500k %
-acodec aac -b:a 128k -ar 44100 %
-f flv rtmp://a.rtmp.youtube.com/live2/*****-*****-*****-*****
```

## 続き | 8-20. 外付けカメラを使いたい

- しばらくすると、画面が表示されるようになります。

- 補足

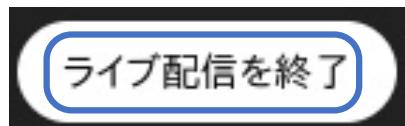
画面右上



← ライブ配信のURLを取得できます。

### ⑫終了する

画面右上



← クリックする。

- ffmpegを、Q で終了する。
- 配信された映像は、「YouTube Studio」→「コンテンツ」→「ライブ配信」に残っていますので、適宜削除するなりしてください。

# 続き | 8-20. 外付けカメラを使いたい

## 【補足】 ffmpegのオプション説明

種別	オプション	内容
映像入力	-f v4l2	映像入力の形式を Video4Linux2 に設定
	-framerate 30	映像のフレームレート (1秒あたり30フレーム)
	-video_size 640x480	解像度。USBカメラが対応している必要あり
	-i /dev/video0	映像入力デバイス
音声 (ダミー)	-f lavfi	入力形式を「FFmpeg内部フィルタ」に設定
	-i anullsrc=...	無音の音声を生成する仮想入力
	channel_layout=stereo	ステレオ (2ch) として出力
	sample_rate=44100	サンプリングレート (44.1kHz)
制御	-shortest	映像と音声の長さが異なるとき、短いほうに合わせて終了する
映像 エンコード	-vcodec libx264	映像を H.264 形式でエンコードする (YouTube推奨)
	-pix_fmt yuv420p	色フォーマット。YouTube互換の標準
	-preset veryfast	エンコード速度
配信設定	-g 60	YouTubeでは、2秒に1回はIフレーム*1が必要(30fpsなので60)
	-b:v 2500k	映像のビットレート (2.5Mbps)
音声エンコード (ダミー)	-acodec aac	音声をAAC形式でエンコードする (YouTube必須)
	-b:a 128k	音声ビットレート (128kbps)
	-ar 44100	音声サンプリングレート (44.1kHz)
出力設定	-f flv	出力形式を FLV に指定 (RTMP配信では通常はこれを指定する)
	rtmp://a.rtmp.youtube.com/live2/S TREAM_KEY	YouTube (RTMPサーバー) に映像+音声を送る

YouTubeでは、映像のみのストリームは受けつけないので、無音の音声をストリームとして送信している。

\*1 Iフレームとは、独立して表示できる「完全な画像」

## 続き | 8-20. 外付けカメラを使いたい

- 外付けカメラの映像をYouTubeで配信したい（日付時刻入り）
  - ・手順は、前ページまでの方法と同じです。コマンドを下記に変更してください。

```
% ffmpeg ¥
-f v4l2 -framerate 30 -video_size 640x480 -i /dev/video0 ¥
-f lavfi -i anullsrc=channel_layout=stereo:sample_rate=44100 ¥
-shortest ¥
-vcodec libx264 -pix_fmt yuv420p ¥
-preset veryfast -g 60 -b:v 2500k ¥
-acodec aac -b:a 128k -ar 44100 ¥
-vf "drawtext=fontfile=/usr/local/share/fonts/ipa/ipag.otf: ¥
      text='%{localtime}': fontsize=32: fontcolor=white: x=16: y=16" ¥
-f flv rtmp://a.rtmp.youtube.com/live2/*****-*****-*****-*****
```



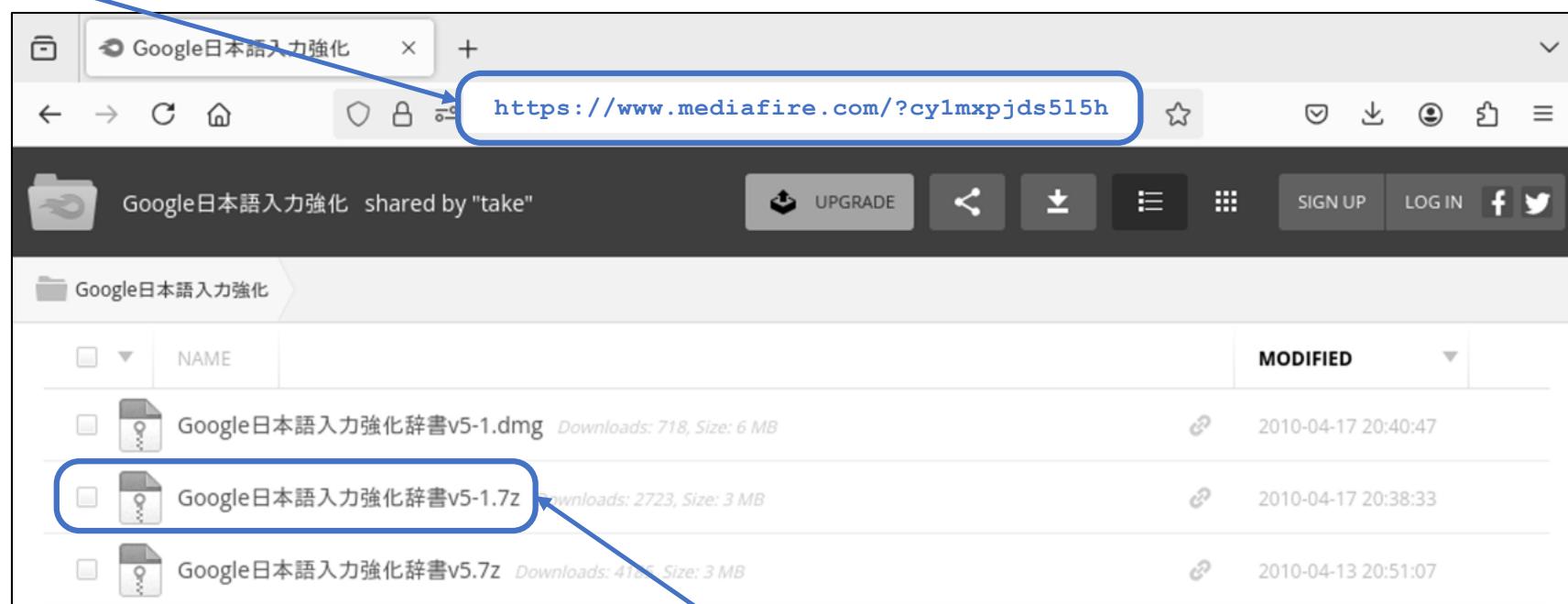
→ 画面左上に日時を挿入

## 8-21. Anthyの辞書を強化したい

ここでは、Google日本語入力強化辞書のファイルを使った例を紹介します\*1。

### □ 辞書ファイルをダウンロードする

- ① <https://www.mediafire.com/?cy1mwpjds515h> ヘアクセスする



- ② v5-1.7z をダウンロードする

\*1 ネット上に多数アップされている「Anthy 辞書 強化」の情報を参考にしています。

# 続き | 8-21. Anthyの辞書を強化したい

- 展開ツール 7-zip と、文字コード変換ツール nkf のインストールを行う

```
% sudo pkg install -y 7-zip ja-nkf
```

- ダウンロードした辞書を展開し、文字コードをUTF-8、改行コードをLFへ変換する

```
% cd ~/ダウンロード ← または ~/Downloads  
% 7z x Google日本語入力強化辞書v5-1.7z  
% cd Google日本語入力強化辞書v5-1  
% nkf -w -Lu Google日本語入力強化辞書.txt > google_dic.txt
```

今回はこの辞書ファイルのみを対象にしています。

## 続き | 8-21. Anthyの辞書を強化したい

- userdic - 日本語入力ユーザー辞書変換スクリプトをダウンロードする

```
% fetch http://startide.jp/comp/im/userdic/userdic
```

- ダウンロードしたスクリプトを動かすための ruby と関連モジュールをインストールする

```
% sudo pkg install -y ruby rubygem-rexml
```

- ダウンロードした辞書を Anthy 形式に変換する

```
% ruby userdic mozc anthy < google_dic.txt | \$  
env LC_ALL=C sort > anthy_dic.t
```

## 続き | 8-21. Anthyの辞書を強化したい

- ユーザー辞書用のディレクトリを作成する

```
% mkdir ~/.anthy/imported_words_default.d
```

- 変換した辞書を上記ディレクトリにコピーする

```
% cp anthy_dic.t ~/.anthy/imported_words_default.d/
```

- これで、変換効率が少し良くなる・・・かも？

(ただし、文節区切りがうまくいかないのはそのまま・・・です)

## 8-22. DOS/V環境を構築したい

### ■ DOS/V環境を構築したい

1990年代前半のIBM-PC/AT互換機で使われていた、DOS/V環境（IBM DOS/Jx.x/V、MS-DOS x.x/Vなど）をFreeBSD内に構築します。\*1

- DOS環境用ミュレータ DOSBox-Xをインストールする

```
% sudo pkg install -y dosbox-x
```

- DOS環境用のディレクトリを作成する

```
% mkdir ~/dos
```

\*1 個人的に思いつく利用方法としては、PC-VAN、NIFTY-Serveといったパソコン通信時代の掲示板ログ（当時はシフト SJIS）がLZH形式で圧縮されおり、それらを専用のビューアーで閲覧する、などが考えられます。今回、まずはDOS/Vを使う上での必要最小限と思われる設定の説明をします。

# 続き | 8-22. DOS/V環境を構築したい

## □ 設定ファイルを編集する

```
% mkdir ~/.config/dosbox-x  
% vim ~/.config/dosbox-x/dosbox-x.conf
```

```
[dosv]  
dosv=jp
```

```
[autoexec]
```

```
mount c ~/dos
```

```
c:
```

```
mode con: rate=30 delay=1
```

新規に追加する

DOSのautoexec.batに相当する行は、このdosbox-x.conf側に記述する

前ページで作成した ~/dos ディレクトリを、DOSからはC: ドライブに見せる設定

キーリピートの設定

※日本語キーボードの場合、keyb jp を設定しても肝心な、¥, \_, | などが入力不可。。。

## □ DOSBox-X を起動する

```
% dosbox-x
```

※起動ドライブやDOSのユーティリティプログラムの格納場所は、Z: ドライブとなります。

※DOS/V本体、\$FONTX、DISPV無しでOSが起動し、日本語表示が可能となります。日本語ファイル名も普通に表示され、FreeBSDからも文字化け無しに表示されます。

# 続き | 8-22. DOS/V環境を構築したい

## ■ LHAやFILMTNを使いたい

DOS環境上で、ファイル圧縮ソフト『LHA』や、ファイルメンテナンスソフト『FILMTN』を使用できるまでの手順を説明します。前ページに引き続き操作方法を説明します。

### □ ディレクトリを作成する (DOSBox-x内で操作) ※DOS時代風に命名

```
C:¥> md UTIL ← 実行ファイルの保存場所  
C:¥> md OLS ← ダウンロードしたソフトの保存場所 (OnLine Softwareの略)
```

### □ いったん DOSBox-xを終了する (DOSBox-x内で操作)

```
C:¥> EXIT
```

## 続き | 8-22. DOS/V環境を構築したい

- LHAとFILMTNをwebブラウザでダウンロードする

ファイル名	主なダウンロード先
lha255.exe	<a href="https://www.vector.co.jp/soft/dl/dos/util/se002413.html">https://www.vector.co.jp/soft/dl/dos/util/se002413.html</a>
fmv245.lzh	<a href="https://www.vector.co.jp/soft/dl/dos/util/se010373.html">https://www.vector.co.jp/soft/dl/dos/util/se010373.html</a>

- dos/olsディレクトリにコピーする

```
% cd ~/ダウンロード ← または cd ~/Downloads  
% cp lha255.exe ../dos/OLS/LHA255.EXE  
% cp fmv245.lzh ../dos/OLS/FMV245.LZH
```

※DOSBox-xから参照されるファイル名は全て大文字に統一した方がトラブルが少ないと思います。

## 続き | 8-22. DOS/V環境を構築したい

- パスを通すため、設定ファイルに追記する

```
% vim ~/.config/dosbox-x/dosbox-x.conf  
:  
set path=%path%;c:\util ← 最終行に追加する
```

- DOSBox-X を起動する

```
% dosbox-x
```

- LHAとFILMTNを展開する (DOSBox-x内で操作)

```
C:\> cd \OLS  
C:\OLS> LHA255.EXE C:\UTIL  
解凍を始めますか？！[Y/N] Y  
  
C:\OLS> LHA E FMV245.LZH C:\UTIL\
```

# 続き | 8-22. DOS/V環境を構築したい

## □ FILMTNの初期設定を行う (DOSBox-x内で操作)

```
C:¥OLS> cd ¥UTIL  
C:¥UTIL> FMCUSTV
```

- ・好みで設定する。
- ・[End]キーで終了する。

- 【個人的に設定必須の項目】
- ・タイトル画面スキップ [2] する
  - ・ファイル名表示方法 [2] 2列
  - ・Enterキーの扱い [2] 表示
  - ・Viewのエディタ起動キー [2] Shift+Enter
  - ・改行マークの表示 [2] する
  - ・環境ファイル出力パス C:¥UTIL

## □ FILMTNを起動する (DOSBox-x内で操作)

```
C:¥UTIL> FILMTN
```

※懐かしの画面が表示されます。

## □ DOSBox-xの終了 (DOSBox-x内で操作)

```
C:¥> exit
```

# 続き | 8-22. DOS/V環境を構築したい

## □ 注意事項

DOSBox-x起動中に、~/dosファイル内をFreeBSD上から操作すると、DOSBox-x上で反映されないことがあります。その場合は、DOSBox-x内から下記コマンドを実行してください。

```
> Z:  
Z:¥> mount -u c ← C: ドライブをアンマウントする  
Z:¥> mount c ~/dos  
Z:¥> C:  
C:¥> dir
```

## □ 設定ファイルについて

デフォルトの設定ファイルは `~/.config/dosbox-x/dosbox-x.conf` ですが、任意のディレクトリ、任意のファイル名を使用することも可能です。その場合、`dosbox-x -conf filename.conf` などと `-conf` オプションをつけて起動してください。

## □ 他

DOSBox-xは、なんとPC-9801のエミュレーションもできるとのことです。詳細はネット上の記事を参考にしてください。

## 8-23. ファイル管理ソフトThunarを使いたい

### ■ ファイル管理ソフトThunarを使いたい

ファイル管理ソフトThunar（スナー）のインストール、使いやすくするための設定方法、上級者向け設定方法などを紹介します（使い方そのものの説明は省略します）。

- Thunarと圧縮ファイル関連をインストールする

```
% sudo pkg install -y thunar thunar-archive-plugin xarchiver
```

このアーカイブプラグインをインストールすることにより、圧縮ファイルの右クリックで「ここに展開」「別の場所に展開」「アーカイブを作成」などのメニューが表示されるようになる。

- Thunarを起動する

```
% thunar
```

実際に呼び出されるプログラム

# 続き | 8-23. ファイル管理ソフトThunarを使いたい

## □ 使いやすくするための設定（表示回り）

- ・ファイル表示をリスト形式（詳細表示）にしたい

表示 (V) →リスト表示 (L) ※CTRL+2 でもリスト形式になる

- ・ファイル表示をリスト形式（詳細表示）で初期設定したい

編集 (E) →設定 (E) →表示タブ

→新規フォルダーの表示 (N) を「リスト表示」にする

→閉じる (C)

- ・リスト形式（詳細表示）表示で、今日・昨日の表示はいいが、○曜日と表示されるのは違和感があるので変更したい。

編集 (E) →設定 (E) →表示タブ

日付 形式 (F) を「今日 / カスタム」にし、「%Y-%m-%d(%a) %H:%M」に変更する

→閉じる (C)

今日、昨日はそのまま表示され、それ以外は  
2025-08-26(火) 12:00 の形式で表示される。

- ・左側の「場所」欄によくアクセスするフォルダーを追加したい

フォルダーをクリック（選択）し、

ブックマーク (B) →ブックマークを追加 (A)

（ドラッグ&ドロップで並び替え可能）

# 続き | 8-23. ファイル管理ソフトThunarを使いたい

## □ 使いやすくするための設定

- 右クリックした際の「ここでTerminalを開く」を正常に実行できるようにしたい  
mlterm用  
編集 (E) →カスタムアクションを設定 (G)  
→「Open Terminal Here」を選択→右側の歯車アイコンをクリック  
→コマンド (C) の欄に「`mlterm -working-directory %f`」と記入→OK (O)  
→閉じる (C)
- 圧縮ファイルのデフォルトの展開先を /tmp ではなく、/home/pcuser/work に変更したい  
(xarchiveアプリの範囲なので、起動・設定します。また、workディレクトリも作成します)  
`% mkdir ~/work`  
`% xarchiver`  
アクション (C) →設定 (P)  
→詳細→Preferred extraction directory:を 「/home/pcuser/work」 に設定する。→OK (O)

# 続き | 8-23. ファイル管理ソフトThunarを使いたい

## □ 使いやすくするための設定（ファイルタイプに応じた起動プログラムの指定）

- ・画像閲覧 (feh) 時、画面にフィットした感じで表示させつつ、同一フォルダーの画像も矢印キーまたはマウスのホイールで簡単に閲覧できるようにしたい

画像ファイルを右クリック→アプリで開く (N) →他のアプリで開く (N) をクリックする。

→画面下の「指定コマンドを使用する (C) :」に「`feh - . --start-at %f`」と記入→開く (O) をクリックする。

- ・上記方法で設定すると、メニュー名がコマンド名+オプションとなってしまい、分かりにくくなるのを修正したい

```
% cd ~/.local/share/applications/
```

```
% ls
```

```
userapp-feh - . -start-at %f-xxxxxx.desktop
```

```
% vim "userapp-feh - . -start-at %f-xxxxxx.desktop"
```

```
:
```

Name=**feh (画面フィット)**

Name=欄を変更する

# 続き | 8-23. ファイル管理ソフトThunarを使いたい

mlterm用です

- テキストファイルなどは、vimで閲覧したい。

テキストファイルを右クリック→他のアプリケーションで開く (A) をクリックする。

→画面下の「指定コマンドを使用する (C) :」に「`mlterm -e vim`」と記入→開く (O) をクリックする。

- 上記方法で設定すると、メニュー名がコマンド名+オプションとなってしまい、分かりにくくなるのを修正したい

```
% cd ~/.local/share/applications/
```

```
% ls
```

```
userapp-mlterm -e vim-xxxxxx.desktop
```

```
% vim "userapp-mlterm -e vim-xxxxxx.desktop"
```

```
:
```

Name=**vim** ← Name=欄を変更する

# 続き | 8-23. ファイル管理ソフトThunarを使いたい

## □ 使いやすくするための設定（複数ファイルの一括処理）

ここでは、一つまたは複数ファイルを選択し、画像のリサイズなどを行う例を紹介します。

### ・リサイズ用のシェルスクリプト作成

```
% mkdir ~/bin  
% vim ~/bin/conv_img_480s.tcsh  
  
#!/bin/tcsh  
foreach fn ($argv) ← 複数ファイルがスペース区切りでthunarから渡される  
    magick "${fn}" \$  
        -resize 480x480\$> \$ ← 画像の長辺側を480pxにする  
        -unsharp 1.5x1.0+0.1 \$ ← 画像を少しクッキリさせる  
        ~/work/`basename ${fn}`  
end
```

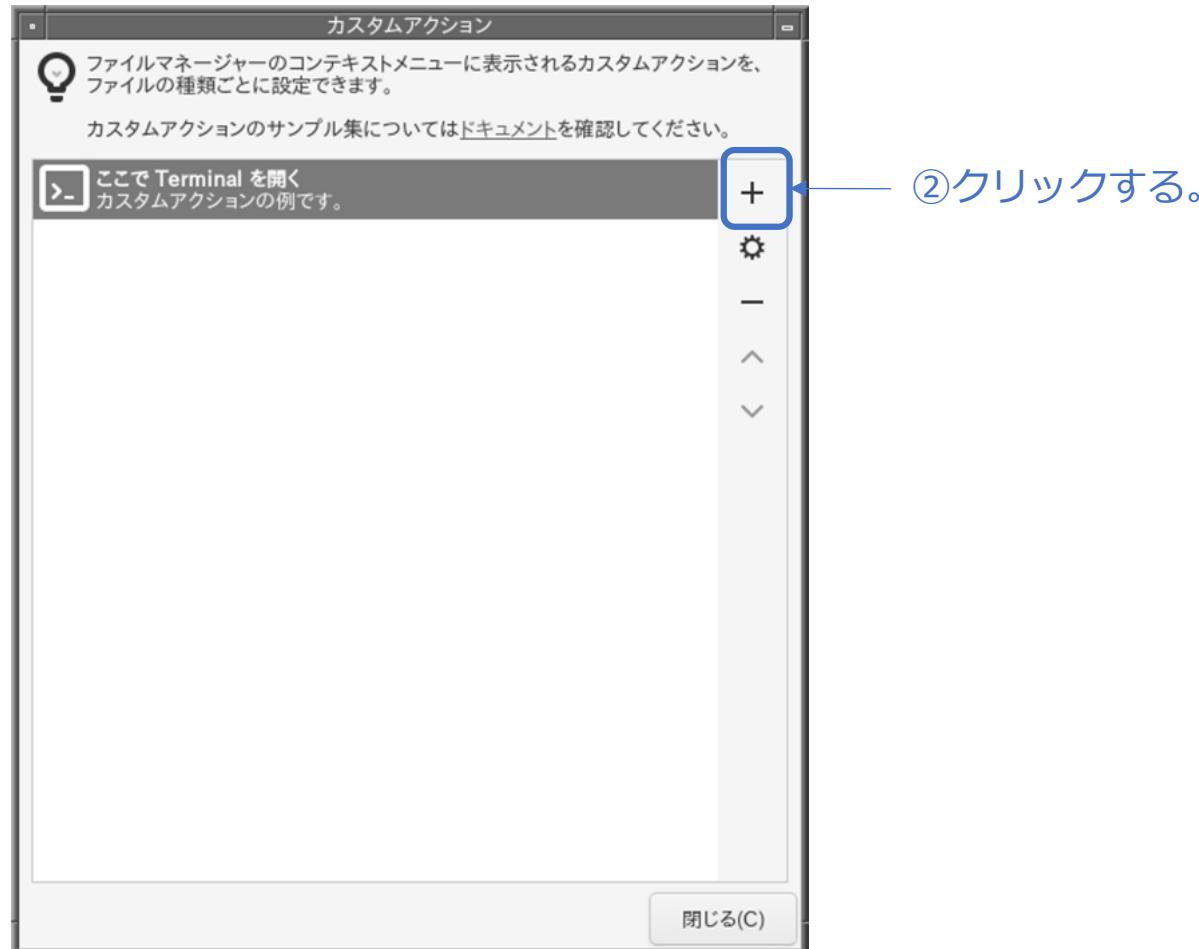
### ・実行権を付与する

```
% chmod +x ~/bin/conv_img_480s.tcsh
```

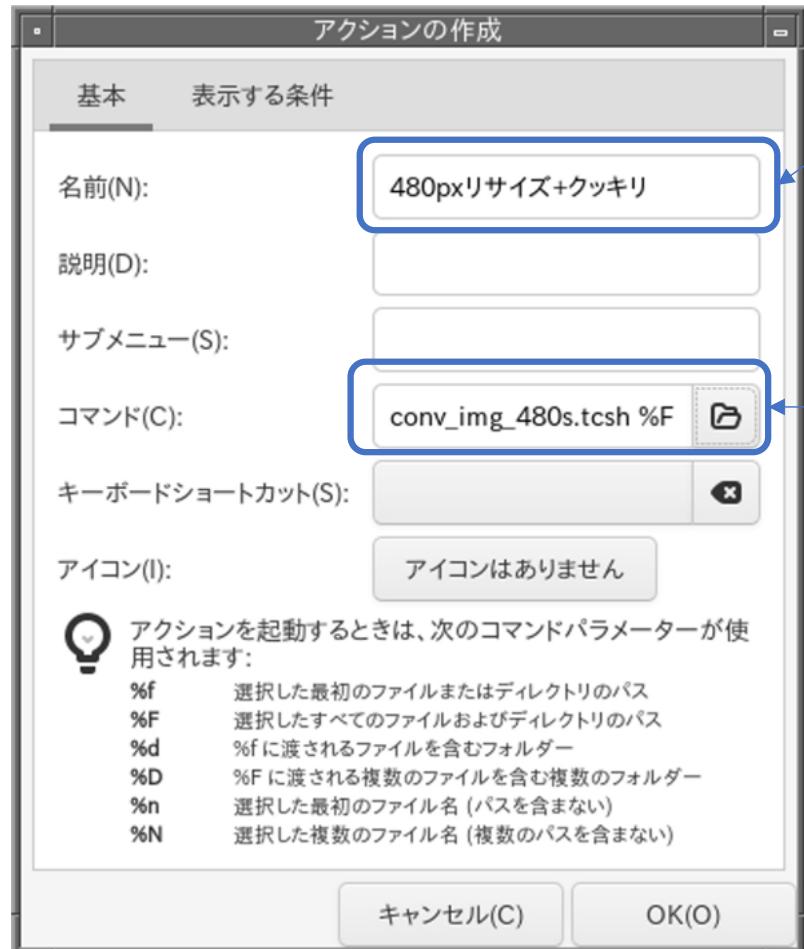
# 続き | 8-23. ファイル管理ソフトThunarを使いたい

- ・ Thunar側の設定

- ① メニューから、編集 (E) → カスタムアクションの設定 (G) ... を選択する。



## 続き | 8-23. ファイル管理ソフトThunarを使いたい

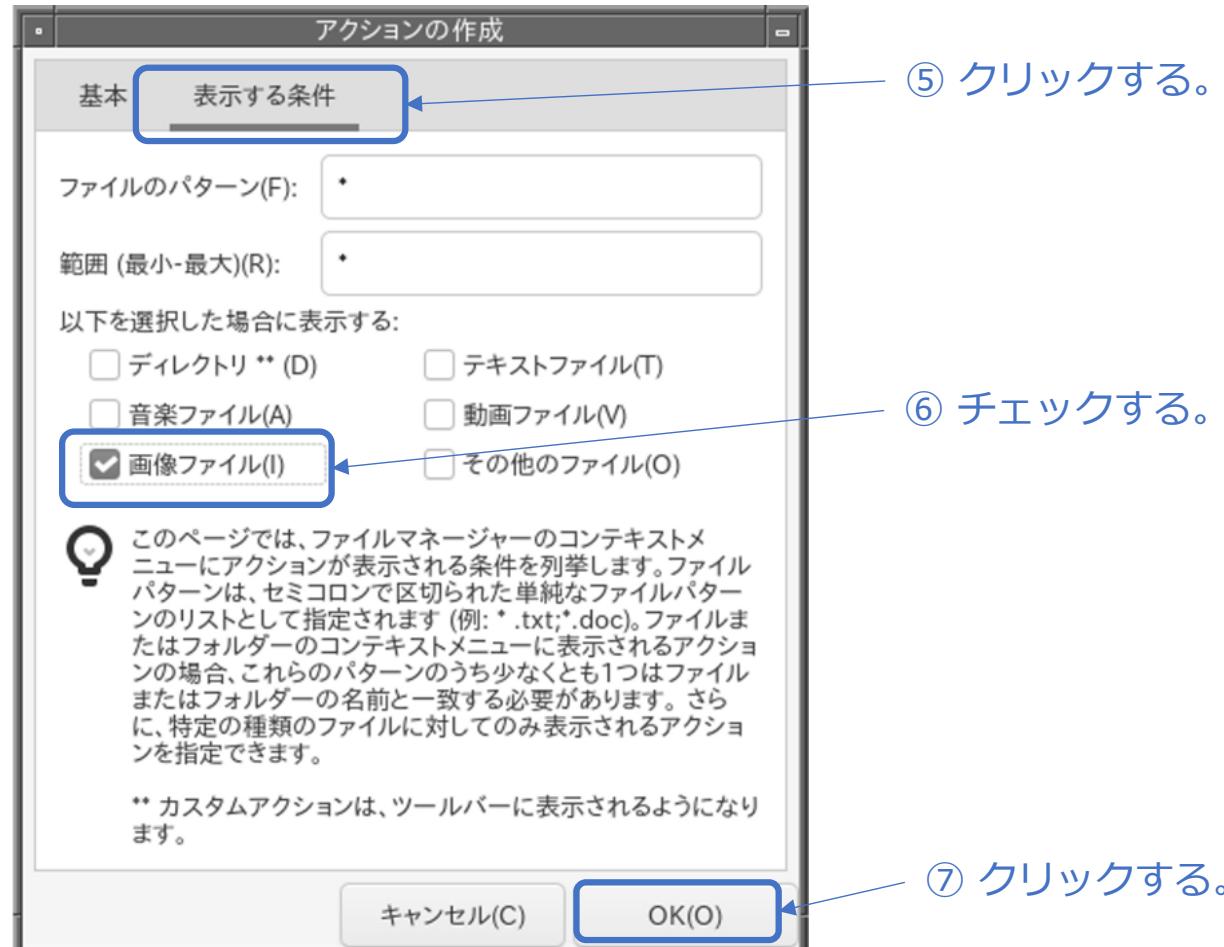


③ 480pxリサイズ+クッキリ と記入する。  
(この名称で、右クリック時に表示されます)

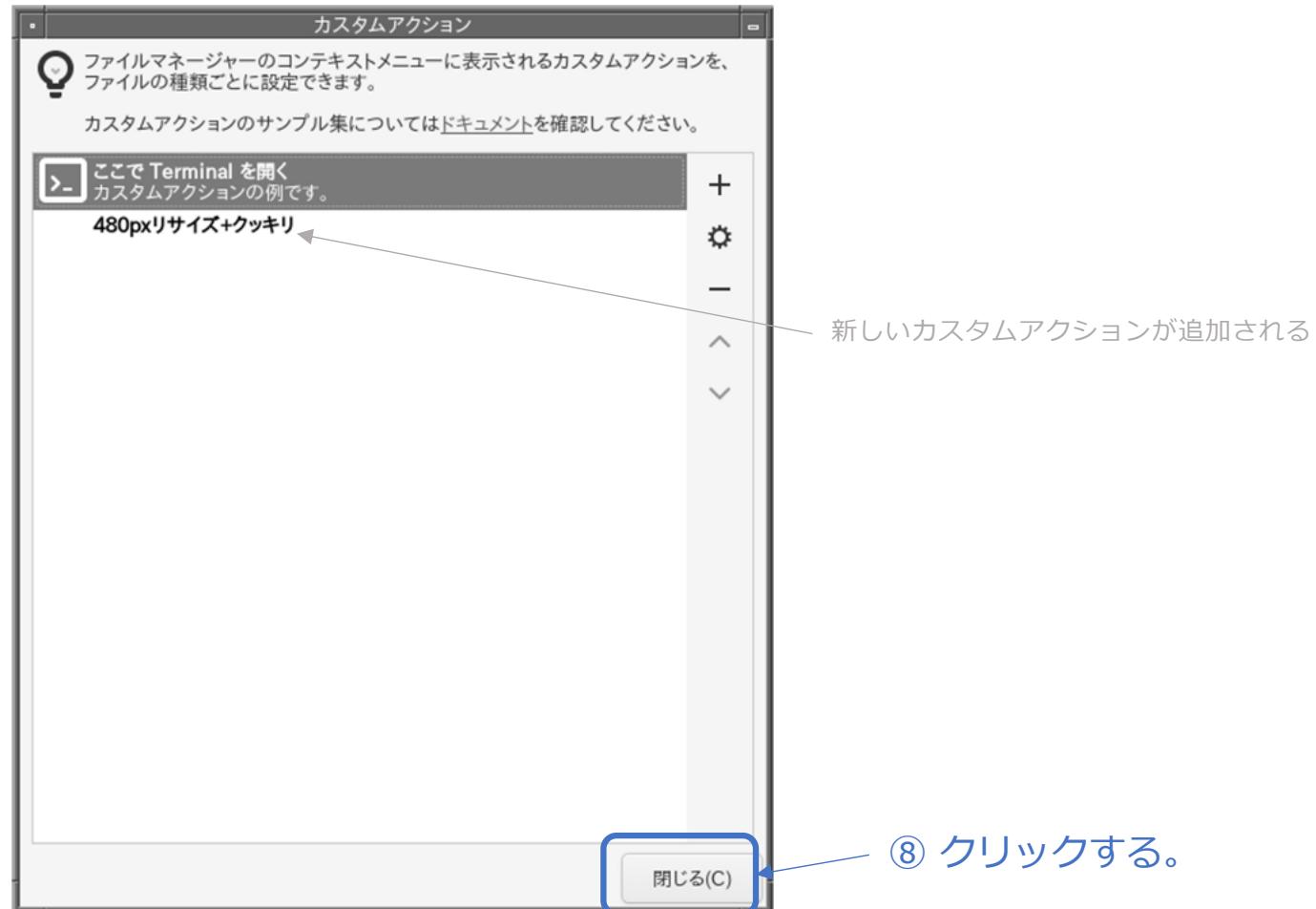
④ conv\_img\_480s.tcsh %F  
と入力する。

この %F で複数ファイルのパス+ファイル名がスペース区切りでシェルに引数として渡される。

# 続き | 8-23. ファイル管理ソフトThunarを使いたい



# 続き | 8-23. ファイル管理ソフトThunarを使いたい



# 続き | 8-23. ファイル管理ソフトThunarを使いたい

- 実際の使用方法



- ~/work ディレクトリに、画像変換後のファイルが保存されます。

# 続き | 8-23. ファイル管理ソフトThunarを使いたい

## □ 使いやすくするための設定（右クリックの「送る」の活用）

ここではファイルを選択時、右クリックした際に「送る」メニューの中に、選択したファイルのSHA256ハッシュを表示するメニューを追加する方法について説明します。

- ・sendtoディレクトリの作成

```
% mkdir -p ~/.local/share/Thunar/sendto
```

- ・送るメニューの作成

```
% vim ~/.local/share/Thunar/sendto/sha256.desktop
```

[Desktop Entry]

Type=Application

Name=SHA256ハッシュを表示する

Exec=xfce4-terminal -e 'tcsh -c "sha256sum %F; stty -icanon; dd bs=1 count=1"'

- ・右クリックに「SHA256ハッシュを表示する」が追加され、選択すると結果が表示されます（何かキーを押すと元に戻ります）。

# 続き | 8-23. ファイル管理ソフトThunarを使いたい

## □ 使いやすくするための設定（ファイルタイプ表示名の変更）

例えば、Thunarのリストビュー（詳細表示）において、「平文テキストドキュメント」または「平文文書」\*1という表示に違和感がある場合、これを修正することができます。以下、その手順を説明します。

- ・ディレクトリの作成

```
% mkdir -p ~/.local/share/mime/packages ← すでにあるかもしれません
```

- ・設定ファイルのコピーと編集

```
% cp /usr/local/share/mime/packages/freedesktop.org.xml ~/.local/share/mime/packages
% vim ~/.local/share/mime/packages/freedesktop.org.xml
:
<comment xml:lang="ja">テキストファイル</comment> ← 平文テキストドキュメントまたは平文文書と記載されている箇所を「テキストファイル」に変更する。
```

- ・有効化の設定

```
% update-mime-database ~/.local/share/mime
```

- ・これでリストビュー（詳細表示）には、「テキストファイル」と表示されます。

\*1 Plain Text を直訳したのだと考えられます。

# 続き | 8-23. ファイル管理ソフトThunarを使いたい

mlterm用です

- 使いやすくするための設定 (vimを右ダブルクリックして終了する)

数ページ前の「ファイルタイプに応じた起動プログラムの指定」にて「テキストファイルなどは、vimで閲覧したい」という設定を行いましたが、ちょっと閲覧するだけのつもりでも、終了するのにキーボードから「:」「q」「Enter」を入力するのが面倒になります。

ここでは Lotus Notes/Domino のように?、右ダブルクリックで戻る（終了する）方法を紹介します。なお、閲覧専用vim（読み取り専用）として新たに設定します。

- ・起動設定

```
% cd ~/.local/share/applications/  
% vim userapp-vim-readonly.desktop
```

```
[Desktop Entry]  
Type=Application  
Exec=mlterm -e vim -R %f ←  
Name=vim (Thunarから読み取り専用で起動)
```

新規に追加する

mltermのウインドウ内から  
vimを読み取り専用 (-R) で起動する設定

# 続き | 8-23. ファイル管理ソフトThunarを使いたい

- vimの設定

```
% vim ~/.vimrc
:
augroup ReadOnlyViewer
  autocmd!
  autocmd BufReadPost * if &readonly | set mouse=n | map <2-RightMouse> :q<CR> | endif
augroup END
```

追加する

もし、読み取り専用だった場合、  
①マウスを使用できるようにする  
②右ダブルクリックで、「:」「q」「<CR>」をキーボードに送る  
ように設定する。

- これでファイル選択時に右クリックで「vim(Thunarから読み取り専用で起動)」（もしくは、アプリケーションで開く→他のアプリケーションで開く(A)...→ vim(Thunarから読み取り専用で起動)）でvimが読み取り専用モードで開き、右ダブルクリックで終了します。

なお、マウスのホイールも使えるようになります。ちょっとした閲覧に便利になります。

# 続き | 8-23. ファイル管理ソフトThunarを使いたい

## 【補足】

もっと簡単に、`~/.vimrc`に

```
set mouse=n  
map <2-RightMouse> :q<CR>
```

と記述することで vim を右ダブルクリックで終了することも可能ですが、`set mouse=` を設定すると、いろいろ副作用がある\*1（使い勝手に影響する）ので、今回は前ページまで説明のように vim を読み取り専用で起動した時のみ、右ダブルクリックで終了するように設定しました。

\*1 ①文字列を選択するときにSHIFTキーを押す必要があること、②複数のウインドウ間を移動する際、ウインドウをクリックした場所にカーソルが移動してしまう、など。

## 8-24. フォントファイルの中身を閲覧したい

フォントファイルがどのように構成されているのかを閲覧することができます。

- fontforgeアプリをインストールする

```
% sudo pkg install -y fontforge
```

- IPAゴシックファイルを閲覧する

```
% fontforg /usr/local/share/fonts/ipa/ipag.otf
```

## 続き | 8-24. フォントファイルの中身を閲覧したい

### □ フォント表示の改善

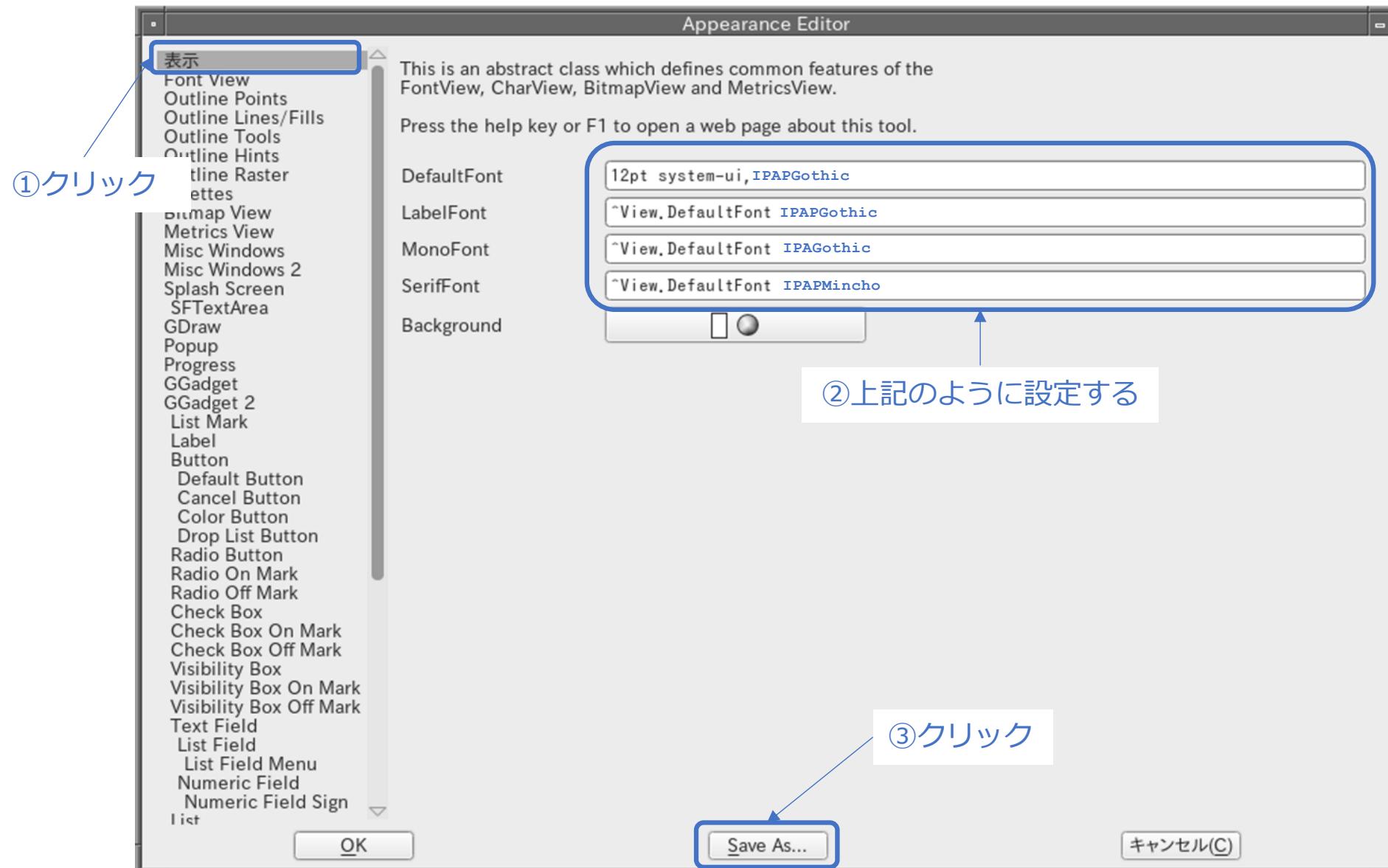
フォントファイルを扱うアプリなのに、メニュー部分など、フォント表示が妙におかしい（英文字と漢字のバランスなど）を改善する設定を説明します。

### □ 起動する（適当にフォントファイルを指定する）

```
% fontforge /usr/local/share/fonts/ipa/ipag.otf
```

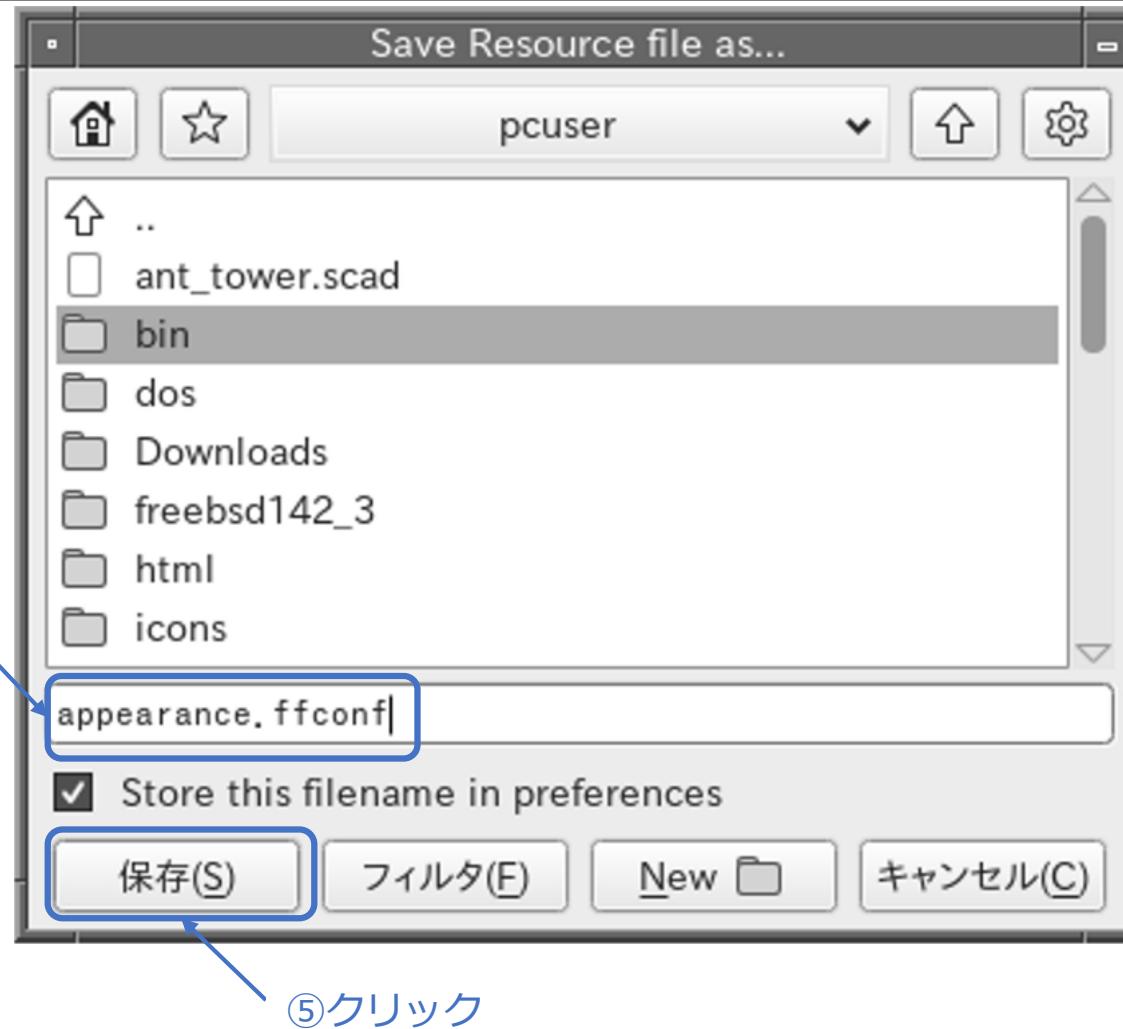
ファイル (F) →Appearance Editor... をクリックする。

# 続き | 8-24. フォントファイルの中身を閲覧したい



## 続き | 8-24. フォントファイルの中身を閲覧したい

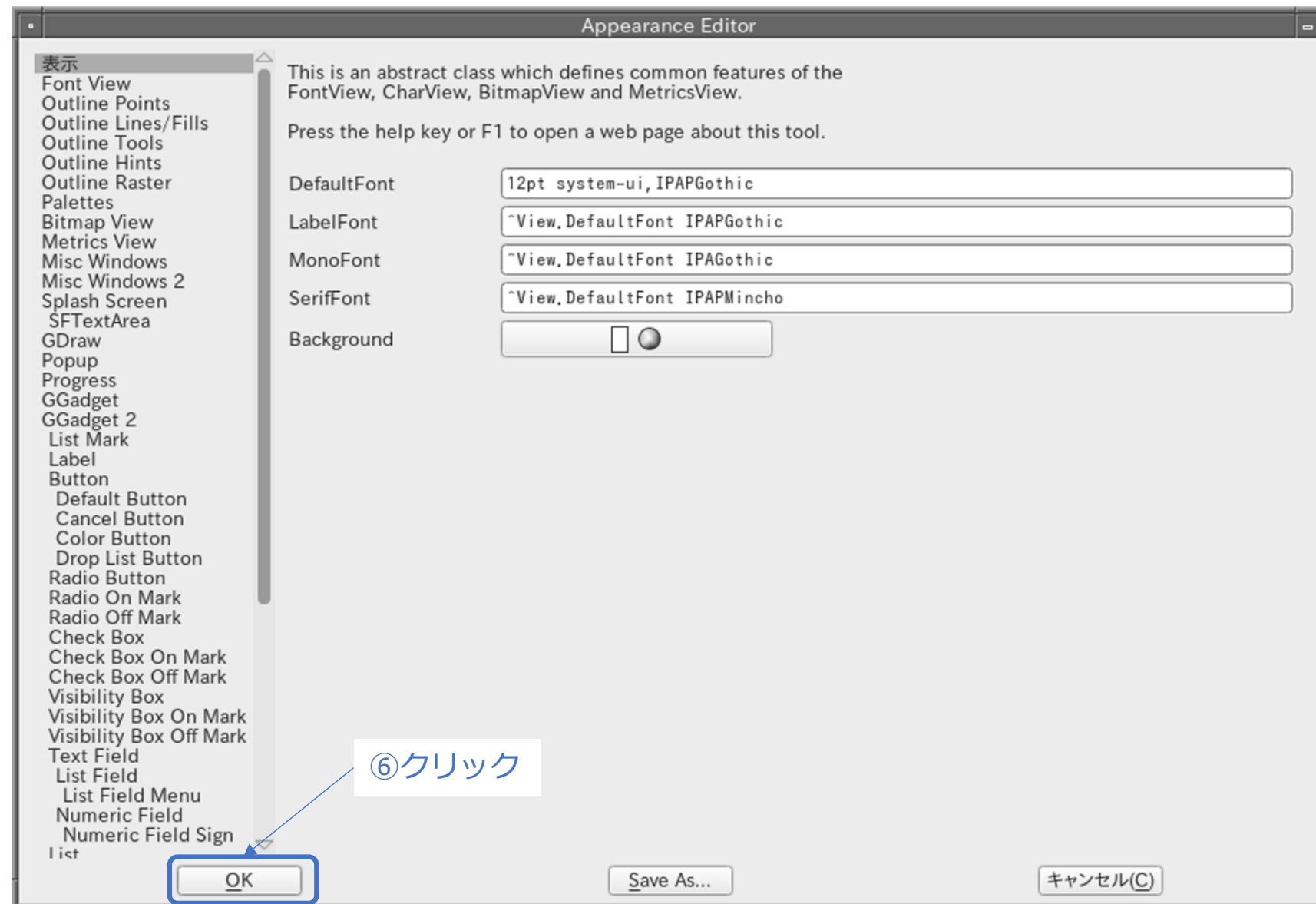
④appearance.ffconf  
と入力する。



⑤クリック

- ・次回以降は、この appearance.ffconf ファイルを参照して起動します。  
(ファイル (F) →環境設定 (E) →一般→リソースファイルに定義されます)

# 続き | 8-24. appendix. 便利な使い方



## 8-25. mltermを使いたい（再コンパイル）

今まで通り mlterm を中心に使いたい場合は、下記順序でpkg経由でのインストールと、再コンパイルを行ってください。

※2025年春頃より、ウィンドウ外にマウスがある状態で文字入力を行うと、ハングアップする現象が確認されています。X Window Systemのximまわりのライブラリの変更 (libx11) が要因とされていますが、本件はその暫定対策となります (uim専用)。

- mltermのインストール（いったん依存関係も含めてインストール）

```
% sudo pkg install -y mlterm
```

- gitのインストール

```
% sudo pkg install -y git
```

- portsツリーの準備（再コンパイルに必要）

```
% sudo git clone https://git.FreeBSD.org/ports.git /usr/ports
% cd /usr/ports
% sudo git checkout 2025Q2
```

← FreeBSD 14.3がリリースされた頃の ports を指定

# 続き | 8-25. mltermを使いたい

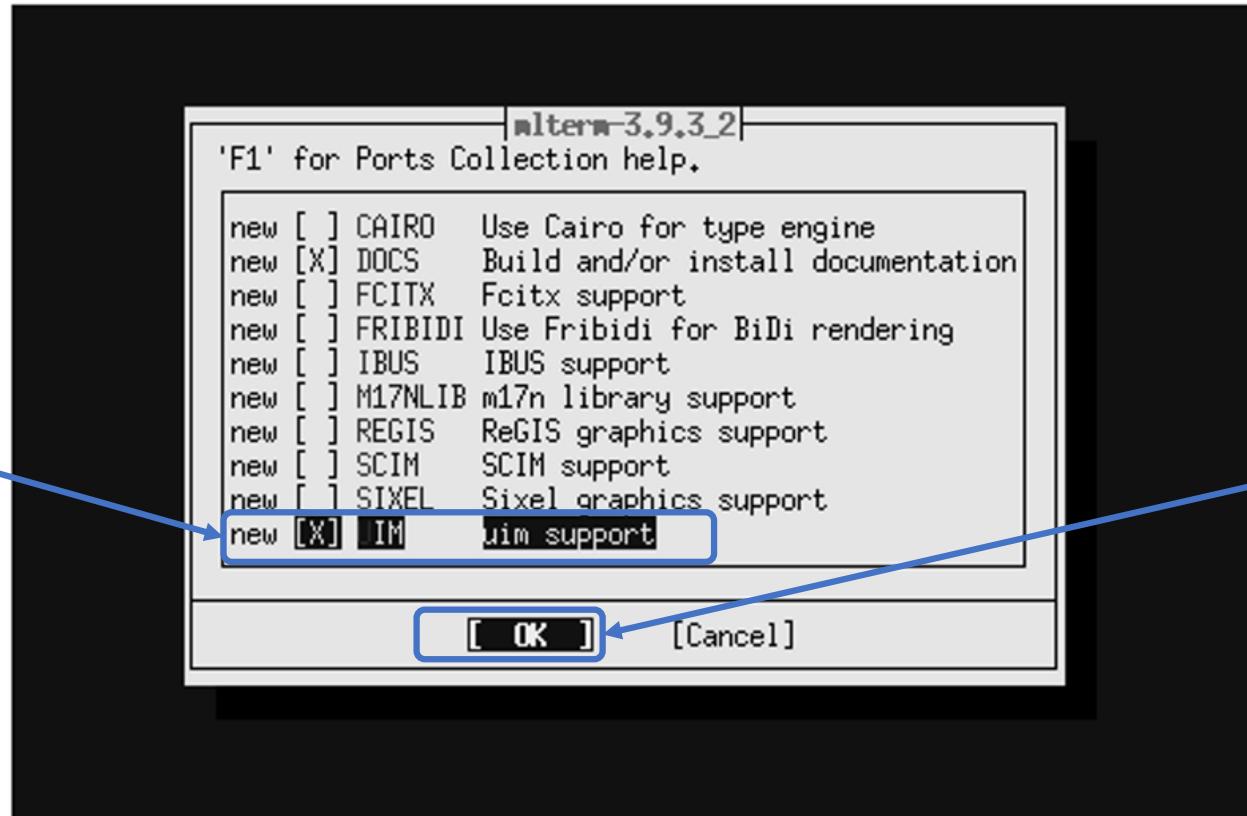
## □ mltermのコンパイル設定

⌚ **xterm** ← xfce4-terminalで曖昧幅を全角にすると罫線が正しく表示されないので、いったんxtermを起動します。

下記のコマンドは、xterm上で入力してください。

⌚ **cd /usr/ports/x11/mlterm**  
⌚ **sudo make config**

↓で uim support を  
選択し、スペースキー  
を押す。



## 続き | 8-25. mltermを使いたい

### □ mltermのコンパイル

xfce4-terminalに戻って、下記を入力してください。

<CTRL+C>

```
% cd /usr/ports/x11/mlterm  
% sudo make
```

### □ pkgでインストールされたmltermの削除と、コンパイルしたmltermのインストール

```
% sudo make deinstall  
% sudo make reinstall
```

### □ mlterm設定フォルダーを作成する

```
% mkdir ~/.mlterm
```

# 続き | 8-25. mltermを使いたい

- `~/.mlterm/main` に以下の内容を新規に追加する

```
% vi ~/.mlterm/main

input_method = uim
fontsize = 16
scrollbar_mode = right
scrollbar_view_name = athena
use_anti_alias = true
line_space = 2
bel_mode = none
bg_color = #fff7e9
blink_cursor = true
word_separators = "↑="
logsize = 8192      半角スペースあり
use_combining = false
static_backscroll_mode = true
box_drawing_font = decsp
```

この設定が重要！

- 入力メソッドに直接uimを使う
- フォントサイズを16にする
- スクロールバーを右側に表示する
- スクロールバーの形状を指定する
- アンチエイリアスフォントを使用する
- 行間を2に設定する
- ベル音を消す
- 背景色を設定する
- カーソルを点滅させる
- ダブルクリック時の単語区切りを設定する
- バックログの行数を設定する
- 結合文字列の処理をしない
- スクロールバック中は自動的にスクロールしない
- 罫線を正しく表示する

# 続き | 8-25. mltermを使いたい

- `~/.mlterm/aafont` に以下の内容を新規に追加する

```
% vi ~/.mlterm/aafont
```

```
ISO10646_UCS4_1 = IPAGothic
```

IPAゴシックを利用する

- `~/.mlterm/key` に以下の内容を新規に追加する

```
% vi ~/.mlterm/key
```

```
Control+Shift+V=INSERT_SELECTION
```

Ctrl+Shift+Vでペーストする

```
Control+Shift+N=OPEN_SCREEN
```

Ctrl+Shift+Nで新規にウインドウを開く

- mltermの起動

```
% mlterm
```

参考 : <https://github.com/arakienn/mlterm/blob/3.9.4/doc/ja/README.ja>

# 続き | 8-25. mltermを使いたい

## ■ 便利に使う方法

□ 横方向にウインドウを分割したい

- SHIFT + F1

□ 縦方向にウインドウを分割したい

- SHIFT + F2

8-29. 端末エミュレータを画面分割して起動したい (tmux) もご覧ください。

参考 : <https://github.com/arakienn/mlterm/blob/3.9.4/doc/ja/README.ja>

※当方の環境では、この操作を何度か行なった時点で mlterm が固まることがあります。

## 8-26. 軽量画像ビュアnsxivをカスタマイズして使いたい

現在 pkg でインストールできる nsxiv はバージョンが古い（28）ので、画像表示するときにデジカメ写真の縦横（exifの回転情報）を認識せず、正しく表示されません。

①exifの回転情報を認識させ正しく表示させる、②キーアサインを追加・変更する、③ステータスバーの文字サイズを大きくすること目的として、ソースコードをダウンロードし、該当部分に変更を加えてコンパイル、インストールする手順を紹介します。なお、pkgでnsxivがインストールされている前提で説明します。

FreeBSD標準の BSD make ではコンパイルできないのでインストールする

- GNU make と git をインストールする

```
% sudo pkg install -y gmake git
```

- nsxiv の最新版をダウンロードする ← 2025/9/20時点での最新版は nsxiv 33。

```
% mkdir ~/work
% cd ~/work
% git clone https://codeberg.org/nsxiv/nsxiv.git
```

## 続き | 8-26. 軽量画像ビュアnsxivをカスタマイズして使いたい

### □ ソースコードの修正①

```
% cd ./nsxiv
% vim config.mk
:
# Uncomment on OpenBSD
HAVE_INOTIFY = 0
lib_fonts_bsd_0 =
lib_fonts_bsd_1 = -lfreetype -L/usr/X11R6/lib/freetype2
inc_fonts_bsd_0 =
inc_fonts_bsd_1 = -I/usr/X11R6/include/freetype2
CPPFLAGS = -I/usr/X11R6/include -I/usr/local/include $(inc_fonts_bsd_$(HAVE_LIBFONTS))
LDLIBS = -L/usr/X11R6/lib -L/usr/local/lib $(lib_fonts_bsd_$(HAVE_LIBFONTS))
```

→ 行頭の # を削除する

# 続き | 8-26. 軽量画像ビュアnsxivをカスタマイズして使いたい

## □ ソースコードの修正② (フォント・キーアサインの変更)

```
% vim config.def.h
:
static const char *BAR_FONT[] = { "Nsxiv.bar.font",
                                  "IPAGothic:size=10.5" }; ← 変更する
:
/* keyboard mappings for image and thumbnail mode: */

:
/*{ 0,           XK_space,          i_navigate,      +1 }, */ ← コメントアウトする
:

/*
 * 画像表示モード *
 * 0, XK_Right, i_navigate, +1 }, /* →キーで次画像 */
 * 0, XK_Down,  i_navigate, +1 }, /* ↓キーで次画像 */
 * 0, XK_Left,  i_navigate, -1 }, /* ←キーで前画像 */
 * 0, XK_Up,    i_navigate, -1 }, /* ↑キーで前画像 */
 * 0, XK_space, g_switch_mode, None }, /* Spaceでサムネイル画面へ */
 * 0, XK_Escape, g_switch_mode, None }, /* ESCでサムネイル画面へ */
/* サムネイル表示モード */
 * 0, XK_Prior, g_scroll_screen, DIR_UP }, /* PgUp */
 * 0, XK_Next,  g_scroll_screen, DIR_DOWN }, /* PgDn */
};

};
```

追加する (189行目付近)  
macOSの写真アプリみたいな  
キーアサインにしました

# 続き | 8-26. 軽量画像ビュアnsxivをカスタマイズして使いたい

- コンパイルする

```
% gmake CC=cc
```

- インストールする

```
% sudo gmake install
```

※ /usr/local/binに nsxiv がインストールされ、他、マニュアル等も最新版に変更されます。

## 8-27. 画面録画したい

デスクトップの操作など、操作説明動画を録画する方法を解説します。

### □ 画面を録画する

```
% ffmpeg -f x11grab -r 30 -s 1366x768 -i :0.0 ¥  
-c:v libx264 -preset ultrafast -crf 25 ¥  
-pix_fmt yuv420p -movflags +faststart output.mp4
```

※終了は q キー

画面サイズ

### □ 秒数指定して画面を録画する

```
% ffmpeg -t 30 -f x11grab -r 30 -s 1366x768 -i :0.0 ¥  
-c:v libx264 -preset ultrafast -crf 25 ¥  
-pix_fmt yuv420p -movflags +faststart output.mp4
```

秒数

# 続き | 8-27. 画面録画したい

## □ オプションの説明

種別	オプション	内容
映像入力	-f x11grab	映像入力の形式をX11の画面キャプチャモードに設定
	-r 30	映像のフレームレート（1秒あたり30フレーム）
	-s 1366x768	キャプチャ領域のサイズ
	-i :0.0	入力ディスプレイ指定。:0.0は通常のXディスプレイ。
エンコード	-c:v libx264	ビデオコーデックにH.264 (x264) を使用
	-preset ultrafast	エンコード速度設定。ultrafastは圧縮率を犠牲にして軽く録画する。
	-crf 25	画質。0～51の範囲で、低いほど高画質（＝ファイル大）。一般的には 23 が標準、25 は軽め。
出力設定	-pix_fmt yuv420p	ピクセルフォーマット指定
	-movflags +faststart	MP4ヘッダをファイル先頭へ移動。

} 範囲指定する場合は、  
-s 800x600 -i :0.0+128,64  
などと指定する。

## 8-28. 端末を閉じても、作業を続けられるようにしたい (tmux)

下記のような用途で便利な、ターミナル・マルチプレクサ tmux の導入手順と簡単な使い方を説明します。

- 大容量のファイルコピーや長時間実行するコマンドを安全に続けたい
- SSH接続中に、端末を閉じたり接続が切れたりしても作業を継続したい
- ウィンドウマネージャーや端末を再起動しても作業環境を保ちたい

ティーマックス

OSの再起動の場合は作業環境を保てません

### インストール

```
% sudo pkg install -y tmux
```

## 続き | 8-28. 端末を閉じても、作業を続けられるようにしたい (tmux)

### 【使用例】

①使用方法（端末エミュレータ内より）

```
% tmux
```

②画面下に縁のステータスバーが表示される。

③例えば、長時間かかるバックアップやコンパイルなどを実行する

```
% rsync ...  
% make ...
```

④上記コマンドの実行中に、端末エミュレータ内を×で閉じる

## 続き | 8-28. 端末を閉じても、作業を続けられるようにしたい (tmux)

### 【使用例（続き）】

⑤端末エミュレータを再度起動する

⑥下記のコマンドを実行する

% **tmux a** ← attachの略

⑦画面が復活する（処理も継続している）

⑧tmuxの終了

% **exit**

## 続き | 8-28. 端末を閉じても、作業を続けられるようにしたい (tmux)

### 【カスタマイズ】

- ・ステータスバーのカスタマイズ

```
% vim ~/.tmux.conf

set-option -g status-right '"#H" #(date +"%Y-%m-%d(%a) %H:%M")' ← 日付書式を変更
set -g status-style "bg=gray,fg=black" ← ステータスバーの色を変更
```

```
% tmux
```

ステータスバーの背景色がグレーになり、画面右下には黒色で「"pc100001" 2025-08-26(火) 12:00」などと表示されます。

## 8-29. 端末エミュレータを画面分割して起動したい (tmux)

ここでは、画面分割（左1+右2（上下）レイアウト）して端末エミュレータを起動する例を紹介します。8-28.の設定が終わっていることが前提とします。また、ここでは8-25.でコンパイルした mlterm を使います。 (xfce4-terminalではウインドウ枠が正しく表示されないなどの不具合が発生します)

### ・ tmuxの設定

```
% vim ~/.tmux.conf
```

```
set -g mouse on
set-option -g pane-border-style fg=gray
set-option -g pane-active-border-style fg=gray
set -g word-separators " ="
```

分割した画面をマウスで移動できるようにする

最終行に追加する

分割した枠の色

mouse on にすると、マウスのダブルクリック動作が tmux 独自になるので、ここでは mlterm と同じになるように設定する。

### ・ mltermの設定（すでに8-25.にて設定済みです）

```
% vim ~/.mlterm/main
```

```
box_drawing_font = decsp
```

最終行に追加する

## 続き | 8-29. 端末エミュレータを画面分割して起動したい

- ・端末エミュレータを大きな画面サイズで起動

```
% mlterm --geometry=161x34 --title=端末エミュレータ
```

- ・tmuxを起動

```
% tmux new-session $; split-window -h $; $  
select-pane -t 1 $; split-window -v $; $  
attach $; select-pane -t 0
```

- ・使い方

内容	方法
ウインドウ間の移動	マウスをクリック、他にも「CTRL+B」に続けて「→」「↓」「←」「↑」
終了	exit、logout、「CTRL+D」
全てを終了	「CTRL+B」に続けて「&」、kill-windows tcsh? (y/n) に「y」

## 8-30. xfce4-terminalの色を標準16色にしたい

vimの設定にて、syntax onにし、ソースコード等を編集する際、xfce4-terminalでは、mltermと微妙に配色が異なるので、これをmltermと同じANSI 16色パレットに変更する方法について紹介します。

- ・設定ファイルを直接変更

```
% vim ~/.config/xfce4/xfconf/xfce-perchannel-xml/xfce4-terminal.xml  
:  
  
<property name="color-palette" type="string"  
value="rgb(0,0,0);rgb(205,0,0);rgb(0,205,0);rgb(205,205,0);rgb(0,0,238);rgb(  
205,0,205);rgb(0,205,205);rgb(229,229,229);rgb(127,127,127);rgb(255,0,0);rgb  
(0,255,0);rgb(255,255,0);rgb(92,92,255);rgb(255,0,255);rgb(0,255,255);rgb(25  
5,255,255)"/>  
</channel>
```

</channel>の上の行に、改行せずに  
追記する

## 8-31. (メモ書き) mltermとxfce4-terminalの主な違い

オプション名が似ているため、動作仕様も同じと思いがちですが、実際には両者には大きな違いがあります。ここでは、その相違点を整理してまとめておきます。

内容	mlterm	xfce4-terminal
曖昧幅の扱い	全角／半角切り替え可能。デフォルトで全角。	全角／半角切り替え可能。デフォルトは半角。
結合文字分離	できる。	できない。
-eオプション	それ以降を全部コマンドとして扱う	1引数のみ (ダブルクオーテーションで囲むなどの対策が必要)
半角罫線	box_drawing_font = decsp が指定されていれば曖昧幅が全角指定でも罫線は半角で描画される。	曖昧幅を全角にすると表示異常となるケースが多い。 (tmuxの複数画面がうまくいかない)
右クリック	何もしない (つまり、WMや他のソフトでカスタマイズ可能)	設定可 (無しにはできない?)

## 8-32. gnuplotでいい感じの棒グラフを作成したい

ニュープロット  
または  
グニュープロット

2次元や3次元のグラフを作成するためのアプリ「gnuplot」にて、（見た目が）いい感じの棒グラフ\*1を作成する方法を紹介します。なお、gnuplotの基本的な使い方は、ChatGPTやネット情報などを参照してください。

- gnuplotのインストール

```
% sudo pkg install -y gnuplot
```

- 画像閲覧ソフト feh のインストール

```
% sudo pkg install -y feh
```

\*1 EXCEL や gnuplot で描画したそのままのグラフではなく、新聞に掲載されるようなグラフです。

# 続き | 8-32. gnuplotでいい感じの棒グラフを作成したい

## ・設定ファイルの作成

```
% vim ~/bargraph.gp
```

```
#-----  
# 1. データ読み込み & Y軸最大値の取得（背景ストライプ計算のため）  
#-----  
set datafile separator ","      # 区切り文字の指定  
  
set term dumb                  # 値のみ取得のため、いったん端末上に表示  
set yrang [0:*]                # Y軸を0～自動最大  
plot 'bargraph.csv' using 1:2  # 1列目=ラベル, 2列目=値  
YMAX = int(GPVAL_Y_MAX)        # 最大Y値を取得して整数化  
  
#-----  
# 2. PNG出力設定  
#-----  
set terminal pngcairo size 640,480 font "IPAPGothic,10.5"  
set output "bargraph.png"  
  
#-----  
# 3. 背景ストライプ作成（偶数行だけ灰色にする）  
#-----  
do for [i=1:YMAX-1:2] {  
    set object i+1 rect $  
    from graph 0, first i to graph 1, first (i+1)  
    fc rgb "#f8f8f8" fillstyle solid 1.0 noborder behind  
}  
}
```

YMAXの値が欲しいがために、この一連の処理を行っている

pngcairo . . . png画像+高画質+日本語対応  
size . . . . 画像サイズ  
font . . . . フォントとサイズ設定

次ページへ

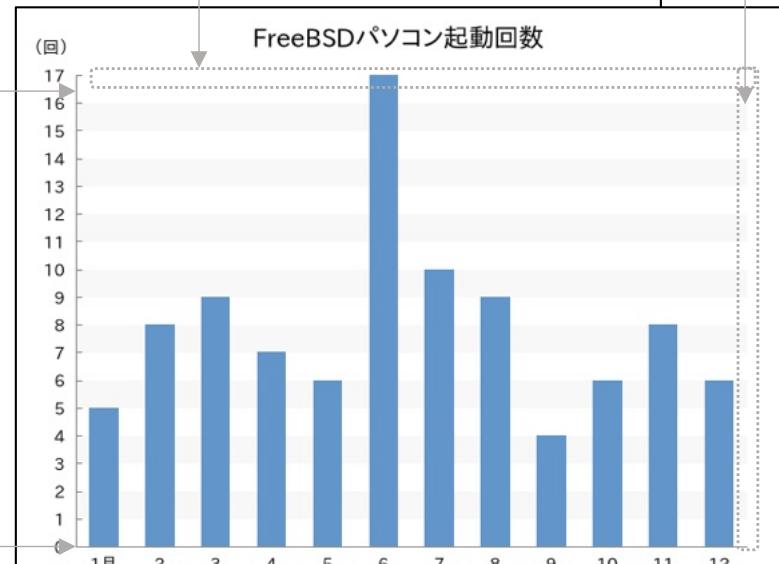
## 続き | 8-32. gnuplotでいい感じの棒グラフを作成したい

前ページより

```
#  
# 4. グラフ全体の設定（軸 / 目盛 / 枠線）  
#-----  
set ytics 1  
set xtics nomirror  
set ytics nomirror  
set tics textcolor rgb "#000000"  
  
# 外枠（左と下のみ） 線の色 線の太さ  
set border 3 lc rgb "#888888" lw 1  
10進数←2進数  
右上左下 0 0 1 1
```

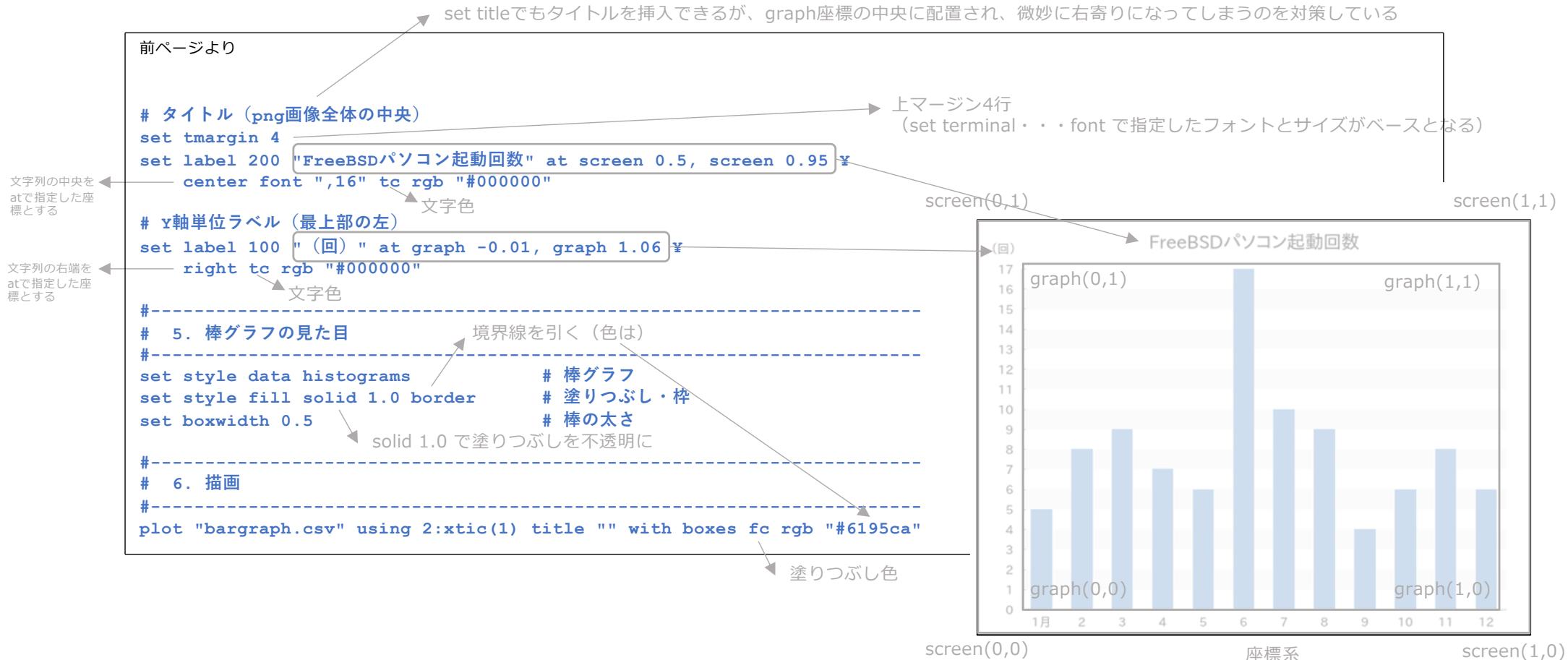
目盛なし

目盛なし



次ページへ

## 続き | 8-32. gnuplotでいい感じの棒グラフを作成したい



# 続き | 8-32. gnuplotでいい感じの棒グラフを作成したい

- データの作成

```
% vim ~/bargraph.csv
```

```
1月,5  
2,8  
3,9  
4,7  
5,6  
6,17  
7,10  
8,9  
9,4  
10,6  
11,8  
12,6
```

- グラフの作成

```
% gnuplot ~/bargraph.gp > /dev/null
```

- グラフの表示

```
% feh ~/bargraph.png
```

終了は Q キー

## 8-33. PlantUMLにて組織図を作成したい

プラントユーエムエル

PlantUMLは、テキストで記述した定義から各種図（構成図、組織図など）を自動生成するツールです。今回は、実務で利用可能（？）な組織図を作成するための手法を解説します（後述の「補足：うまくいかないこと、できること」もご覧ください）。

- PlantUML と graphviz のインストール

```
% sudo pkg install -y plantuml graphviz
```

- 画像閲覧ソフト feh のインストール

```
% sudo pkg install -y feh
```

本内容は <https://qiita.com/daisukeArk/items/a1e00d67471bd0ac4d52> を参考にしました。

# 続き | 8-33. PlantUMLにて組織図を作成したい

## ・定義ファイルの作成

```
% vim ~/organization.puml
```

```
' ※ 本図は架空の組織図です  
' ※ 拠点は package として表現しています  
  
@startuml organization  
  
object 会長 #ffffff {  
    #高橋CEO  
}  
object 社長 #ffffff {  
    #佐藤COO  
}  
  
package 本社 {  
    object 総務部 {  
        #田中 (部長、本社統括)  
        +中村  
        +近藤 (工場常駐)  
    }  
    object 管理部 {  
        #山本 (部長)  
        +中村 (総務部兼務)  
    }  
    object 営業部 {  
        #加藤 (部長)  
        +鈴木 (担当部長)  
    }  
    object 第1営業課 {  
        #加藤 (課長、営業部長兼務)  
        +山田  
        +小林  
    }  
}
```

```
object 第2営業課 {  
    #佐々木 (課長)  
    +山口  
}  
  
package 工場 {  
    object 生産管理部 {  
        #井上 (部長、工場統括)  
        +木村  
    }  
    object 開発部 {  
        #林 (部長)  
        +斎藤 (主幹)  
        +清水  
        +山崎  
        +阿部  
    }  
    object 品質保証部 {  
        #森 (部長)  
        +池田  
    }  
    object 製造部 {  
        #橋本 (部長)  
    }  
    object 製造1課 {  
        #山下 (課長)  
        +石川 (係長)  
        +中島  
        +伊藤  
        +松本  
    }  
}
```

```
object 製造2課 {  
    #岡田 (課長)  
    +長谷川  
    +佐藤 大  
}
```

```
会長 -- 社長  
社長 -- 総務部  
社長 -- 管理部  
社長 -- 営業部  
社長 -- 生産管理部  
社長 -- 開発部  
社長 -- 品質保証部  
社長 -- 製造部  
営業部 -- 第1営業課  
営業部 -- 第2営業課  
製造部 -- 製造1課  
製造部 -- 製造2課
```

```
総務部 - [hidden]r- 管理部  
営業部 - [hidden]r- 生産管理部  
開発部 - [hidden]r- 品質保証部
```

```
@enduml
```

組織の階層を表す

並び順が崩れないように設定

# 続き | 8-33. PlantUMLにて組織図を作成したい

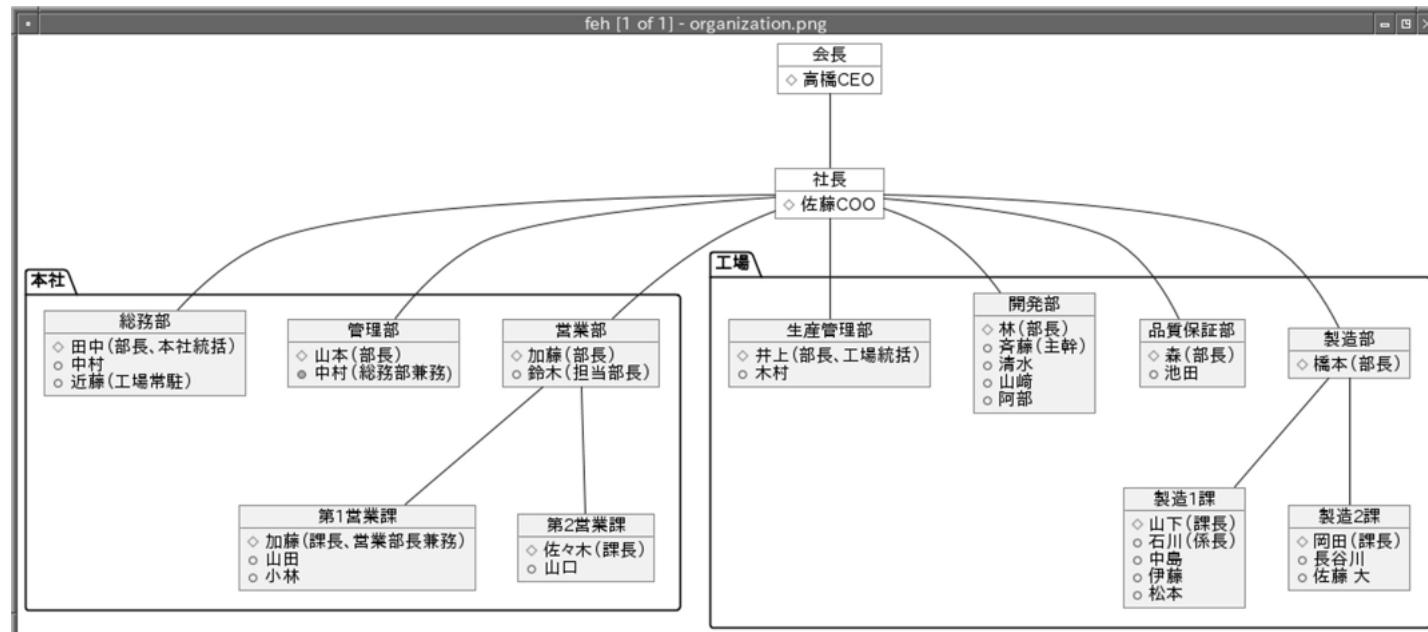
- 組織図の作成

```
% plantuml ~/organization.puml
```

- 組織図の閲覧

```
% feh ~/organization.png
```

終了は Q キー



- ・補足（うまくいかないこと、できること）

1. 左から右へ展開する組織図
2. 各部や各課の整列（上下が微妙にズれるなど）
3. 直線でなく、直交線にするには「skinparam linetype ortho」のパラメーターを加えれば可能だが、思い通りの配線にはならない。
4. 部を通り越して課のみを作成したい場合、ダミーの部を設けるなど、工夫次第では作画できるが、設定が複雑になってしまふのと、思ったようなレイアウトにはならない（配線が途切れてしまうなど）。

## 8-34. PDFにて、ページの入れ替え、追加、削除などを行いたい

PDFファイルにて、プレビュー画面を表示しながら、ページの入れ替え、他のPDFファイルの一部のページの追加、ページの削除などを行う場合には、PDF Arrangerが便利です。  
(macOSのプレビューみたいな感じです)

- PDF Arrangerのインストール

```
% sudo pkg install -y py311-pdfarranger
```

- 起動

```
% pdfarranger
```

- 使用方法は、省略します。

## 8-35. PDFにて、複数ファイルの結合や、偶数/奇数ページ毎の処理を行いたい

PDFファイルにて、複数ファイルの結合や、特定の範囲や偶数・奇数ページでの抽出・削除・回転、複数のファイルを交互にミックスする場合には、PDF Mix Toolが便利です。

- PDF Mix Toolのインストール

```
% sudo pkg install -y pdfmixtool-qt5
```

- 起動

```
% pdfmixtool
```

- 使用方法は、省略します。
- PDF作成の際、ファイル名には .pdf の拡張子を入力してください。

## 続き | 8-35. PDFにて、複数ファイルの結合や、偶数/奇数ページ毎の処理を行いたい

PDF Mix Tool 使用時、画面上の一部のボタンアイコンが表示されないことがあります。その対策方法を下記に説明します。

- ・アイコンテーマのインストール

```
% sudo pkg install -y kf6-breeze-icons
```

- ・Qt5アプリの見た目・挙動を一括設定するコントロールパネルのインストール

```
% sudo pkg install -y qt5ct
```

- ・環境変数を設定する

```
% vim ~/.xinitrc
:
export QT_QPA_PLATFORMTHEME=qt5ct ← この位置に追加する
xset r rate 250 40
:
```

続き | 8-35. PDFにて、複数ファイルの結合や、偶数/奇数ページ毎の処理を行いたい

ウィンドウシステムをいったん終了する

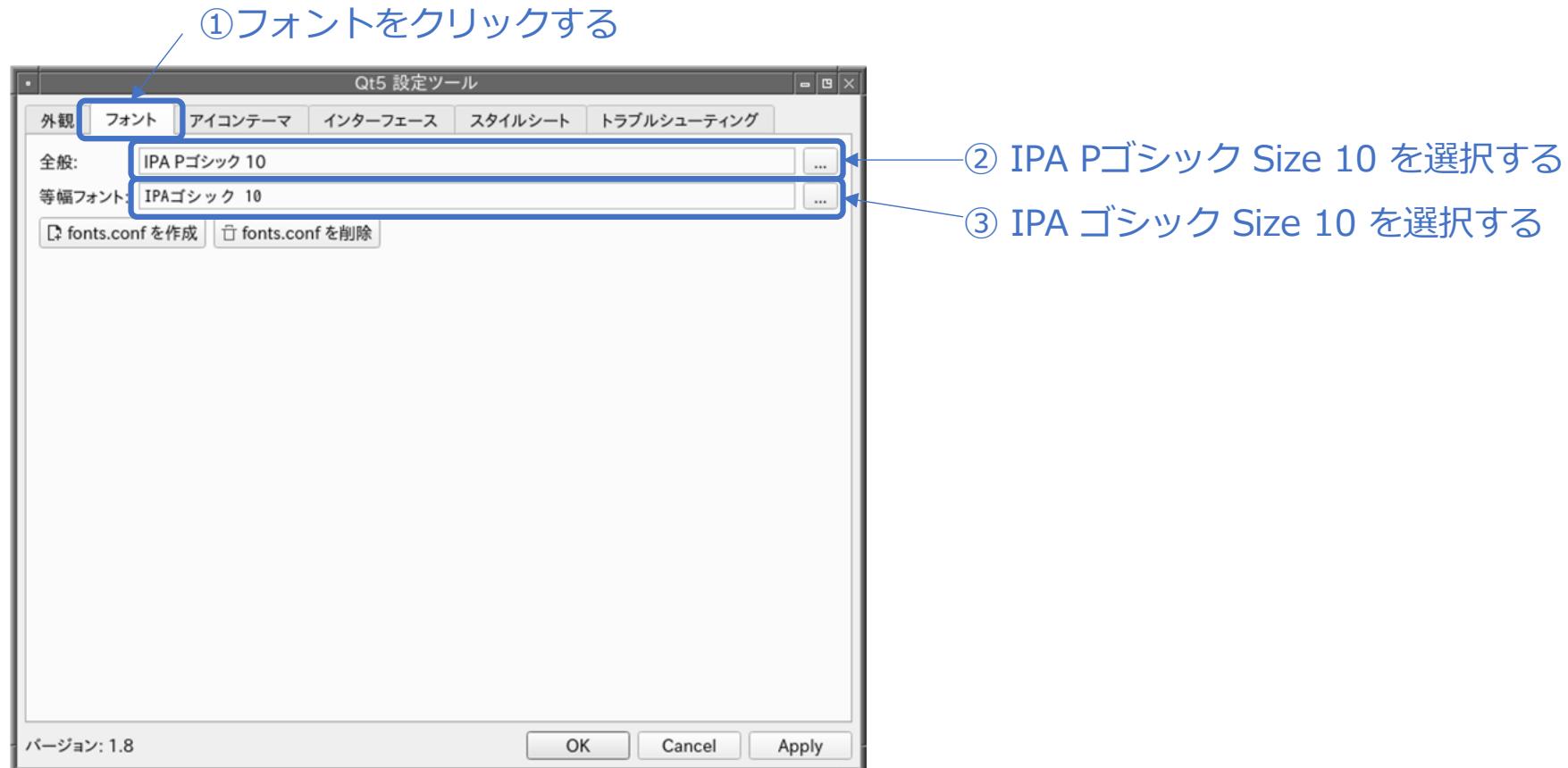


ウィンドウシステムを起動する

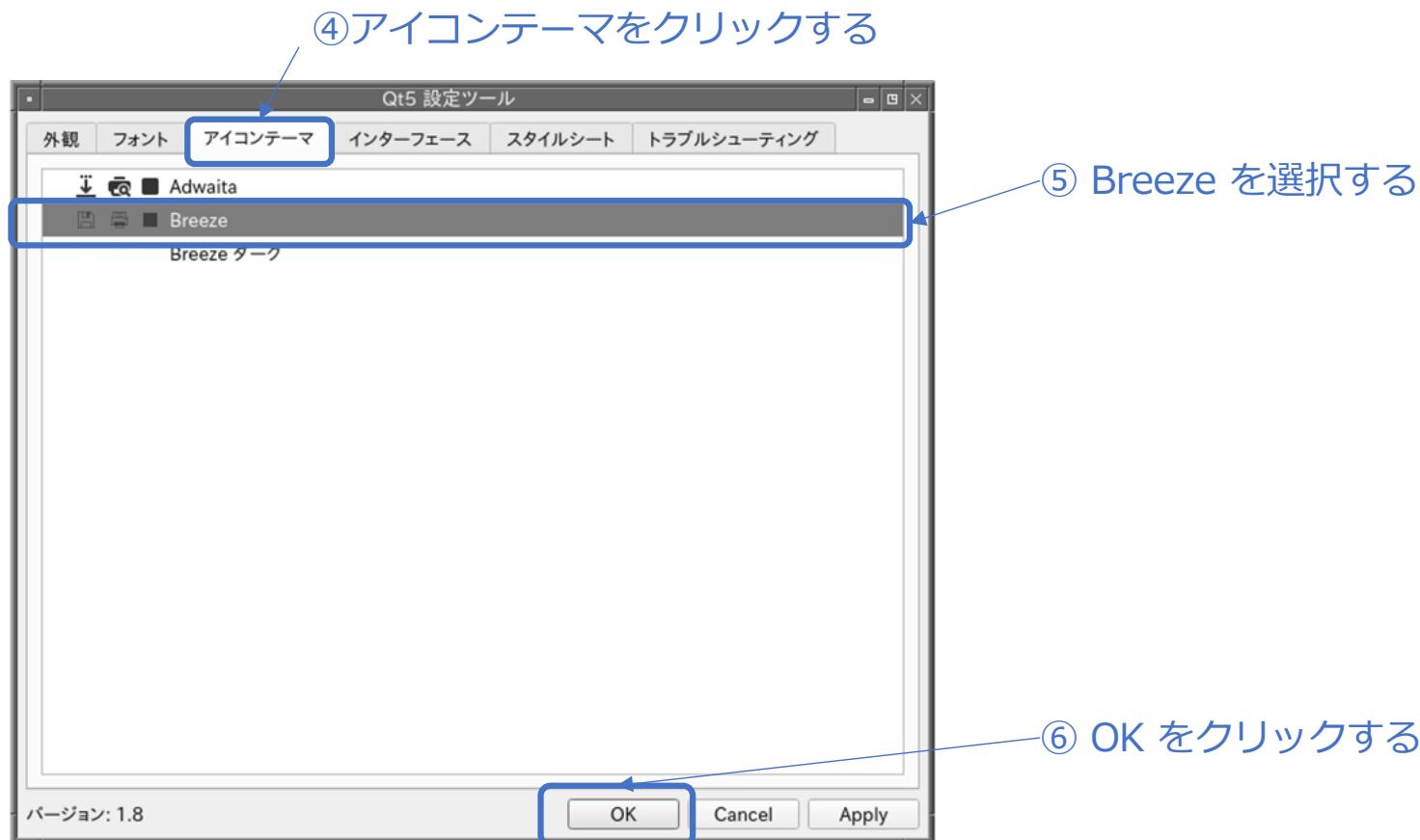
```
% startx
```

- Qt5コントロールパネルを起動する

qt5ct



## 続き | 8-35. PDFにて、複数ファイルの結合や、偶数/奇数ページ毎の処理を行いたい



- 再度 PDF Mix tool を起動する

```
% pdfmixtool
```

## 8-36. コマンドラインにて、PDFファイルの結合・分割処理などを行いたい

8-35.にて、PDF Mix Toolを用いたPDFファイルの結合、分割などの説明をしましたが、これらの処理を、コマンドライン上で実行できるようにするツールが qpdf です。

- qpdf のインストール

```
% sudo pkg install -y qpdf
```

- PDFファイルの単純結合 (file1.pdfとfile2.pdfを1つのPDFファイルにする)

```
% qpdf --empty --pages file1.pdf file2.pdf -- output.pdf
```

このオプションについてはChatGPTに質問してください

- PDFファイルの抽出 (file.pdfから1～5ページと7ページを抽出する)

```
% qpdf --empty --pages file.pdf 1-5,7 -- output.pdf
```

ページ範囲を指定する

■ 複数のPDFファイルからの抽出・結合

(file1.pdfの1~5と7ページ、file2.pdfの8,9ページを抽出し、1つのPDFファイルにする)

```
% qpdf --empty --pages file1.pdf 1-5,7 file2.pdf 8,9 -- output.pdf
```

ページ範囲を指定する

ページ範囲	内容
1	1ページ
1,2,3	1,2,3ページ
1-3	1から3ページ
1,2,5-11	1,2ページと、5から11ページ
3-z	3から最後のページ
z	最後のページ
3,1,2	3,1,2ページの順番
even	偶数ページ
odd	奇数ページ

■ 奇数ページを右に90度回転、偶数ページを左に90度回転させる

```
% qpdf file.pdf ¥  
--rotate=+90:1-z:odd ¥ ← --rotateの場合、範囲指定は 1-z:odd などとする  
--rotate=-90:1-z:even ¥ ← --rotateの場合、範囲指定は 1-z:even などとする  
-- output.pdf
```

※想定される利用用途 → 両面スキャンの設定を間違えた場合など。

■ 閲覧用のパスワードを付与する

```
% qpdf --encrypt 1234 9999 256 -- file.pdf output.pdf
```

↑  
↑  
↑ 暗号強度  
管理者用のパスワード例（閲覧用のパスワードを外す際に必要）  
↓ 閲覧用のパスワード例

■ 印刷禁止、編集不可、テキスト／画像抽出不可のPDFにする

```
% qpdf --encrypt "" 9999 256 ¥  
--print=none ¥ ← 印刷禁止  
--modify=none ¥ ← 編集禁止  
--extract=n ¥ ← 抽出禁止  
-- file.pdf output.pdf
```

↑ 閲覧用のパスワードを付与しない場合は "" を指定する

- 印刷禁止、編集不可、テキスト／画像抽出不可のPDFにし、さらに管理者パスワードをランダムに設定する

```
% set OWNERPW=`openssl rand -base64 48 | tr -dc 'A-Za-zA-Z0-9' | head -c 32`  
% qpdf --encrypt "" "$OWNERPW" 256 ¥  
--print=none ¥ ← 印刷禁止  
--modify=none ¥ ← 編集禁止  
--extract=n ¥ ← 抽出禁止  
-- file.pdf output.pdf
```

※想定される利用用途 → PDFファイルを自動的に生成する仕組みができており、PDFファイルそのものは管理する必要がない（管理したくない）場合や配布用のPDFなど

## 9-1. デスクトップに、プログラムを起動するランチャーを表示させたい

画面中央下にFirefox、Chrome、xfce4-terminalのランチャーを配置する例

最終行付近のコメント##を削除する

vim ~/.fvwm2rc

EwmhBaseStruts 0 0 0 48

ウインドウを最大化しても、  
ランチャーと重ならないよう  
にする（左右上下）

AddToFunc InitFunction "I" Module FvwmButtons  
AddToFunc RestartFunction "I" Module FvwmButtons

FvwmButtons自身を  
ALT+TABで表示される  
リストに表示しない

FvwmButtons自身に  
フォーカスさせない

Style "FvwmButtons" NoTitle, BorderWidth 0, HandleWidth 0, WindowListSkip, NeverFocus,  
FixedPosition

配置を固定する

\*FvwmButtons: Back #cecece

ランチャーのサイズと座標（適宜変更する\*1）

\*FvwmButtons: ButtonGeometry 48x48+611-0

ランチャーの行数

\*FvwmButtons: Rows 1

ランチャー間の幅

\*FvwmButtons: Frame 1

ホバー時の色

\*FvwmButtons: ActiveColorset 12

\*FvwmButtons: (Icon "firefox.png", Action (Mouse 1) Func-LaunchOrRaise\_firefox, Action (Mouse 3) Menu firefox\_menu)

\*FvwmButtons: (Icon "chrome.png", Action (Mouse 1) Func-LaunchOrRaise\_chromium, Action (Mouse 3) Menu chromium\_menu)

\*FvwmButtons: (Icon "xterm-sol.png", Action (Mouse 1) Func-LaunchOrRaise\_xfce4terminal, Action (Mouse 3) Menu xfce4terminal\_menu)

左クリック時、プログラムが起動していない場合は起動、起動している場合はフォーカスするように.fvwm2rcの先頭付近でファンクションを定義している

右クリックでメニューを表示し、選択できるようにした

\*1 ランチャー座標 = (画面の横解像度 - ランチャーニュ  $\times$  48)  $\div$  2

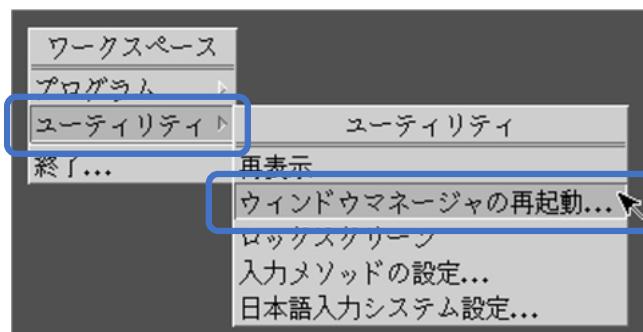
## 続き | 9-1. デスクトップに、アプリを起動するランチャーを表示させたい

Firefoxのアイコンをランチャー表示用にコピーする

```
% cp ¥  
/usr/local/lib/firefox/browser/chrome/icons/default/default32.png ¥  
~/icons/firefox.png
```

Chromiumのアイコンをランチャー表示用に変換・コピーする

```
% magick ¥  
/usr/local/share/icons/hicolor/64x64/apps/chrome.png ¥  
-resize 32x32 ~/icons/chrome.png
```



ウィンドウマネージャの  
再起動をクリックする

## 9-2. ランチャーに、システム負荷やバッテリー状態を表示させたい

最終行付近の「コメント##を外す

```
% vim ~/.fvwm2rc
```

```
*FvwmButtonsGeometry 48x48+563-1 ← ランチャーの座標を適宜変更する  
(画面の横解像度 - ランチャー数 × 48) ÷ 2
```

```
*FvwmButtons: (Swallow "xload" 'Exec xload -update 3 -nolabel -bg §§#cecece', Action(Mouse 1) Func-LaunchOrRaise_conky)  
*FvwmButtons: (Swallow "xbatt" 'Exec xbatt -bg §§#cecece', Padding 0 2)
```

xload（システムの負荷グラフを表示）をインストールする

```
% sudo pkg install -y xload
```

xbatt（バッテリーの状況を表示する）をインストールする

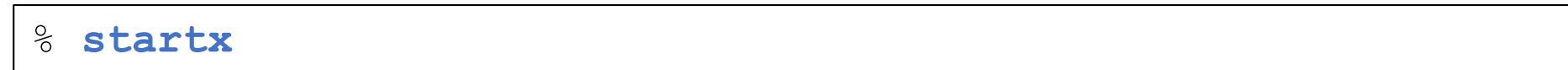
```
% sudo pkg install -y xbatt
```

## 続き | 9-2. ランチャーに、システム負荷やバッテリー状態を表示させたい

ウィンドウシステムをいったん終了する



ウィンドウシステムを起動する



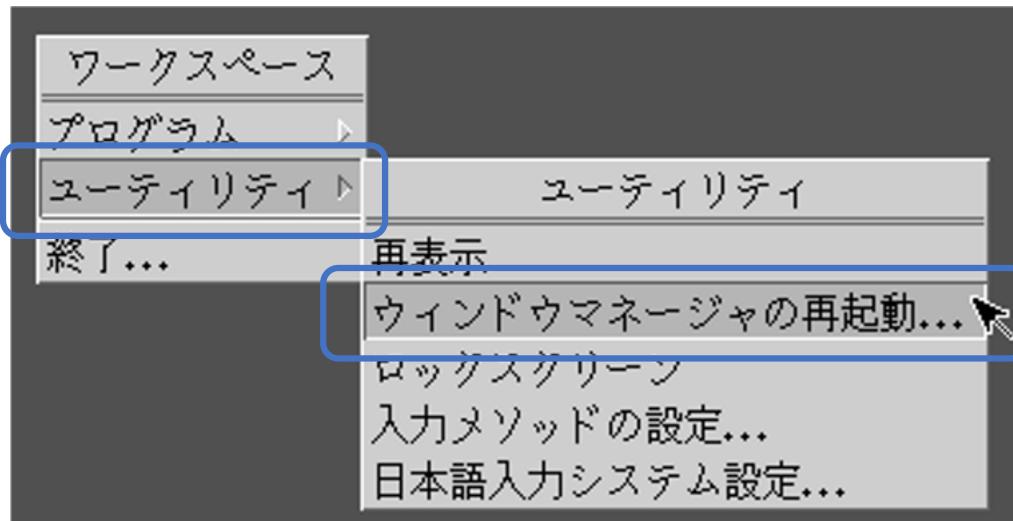
### 9-3. ファンクションキー押下で、アプリが起動するようにしたい

```
% vim ~/.fvwm2rc
```

```
Key F12 A 4 Exec scrot -b -u -f 'screenshot%Y%m%d%H%M%S.png'
```

最終行に追加する

※ここでは、Windowsキー+F12キーでアクティブなウィンドウ画面を  
画面キャプチャーするプログラムを起動している。



A · · · (Any)

C · · · Ctrl

S · · · Shift

M · · · Alt

N · · · (Nothing)

4 · · · WindowsKey

※指定しても、動作しない場合がある。

ウィンドウマネージャの  
再起動をクリックする

## 続き | 9-3. ファンクションキー押下で、アプリが起動するようにしたい

(補足) 前ページの画面キャプチャソフト scrot のよく使用されるオプション

オプション	内容
なし	全画面を画面キャプチャーする
-u	アクティブウィンドウをキャプチャーする
-d 秒数	秒数後に画面キャプチャーする
-b	タイトルバーやウィンドウ枠も含めてキャプチャーする
-s	デスクトップをクリックする→全画面キャプチャ ウィンドウをクリックする→そのウィンドウをキャプチャーする 範囲を選択する→その部分をキャプチャーする
-p	マウスカーソルを含めてキャプチャーする
-f 形式	ファイル名の形式を指定する。 %Y%m%d%H%M%Sで、年月日時分秒の形式となる。 例：scrot -u -b -f '%Y%m%d%H%M%S.png'

## 9-4. キーボード上のボリュームキーにて音量調節をしたい

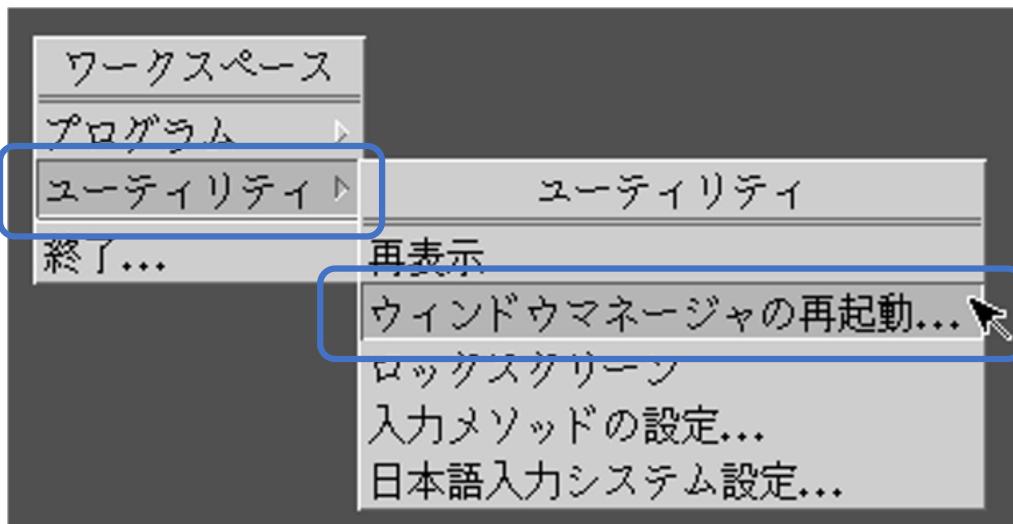
ノートパソコンなど、キーボード上に音量キーが存在する場合

コメント# #を外す

```
% vim ~/.fvwm2rc
```

```
Key XF86AudioRaiseVolume A A Exec mixer vol.volume=+0.05
Key XF86AudioLowerVolume A A Exec mixer vol.volume=-0.05
Key XF86AudioMute          A A Exec mixer vol.mute=^
```

※パソコンによっては、動作しない場合あり



ウィンドウマネージャの  
再起動をクリックする

上記方法では、画面上に何も表示されません。9-19.「音量調整時に、画面上に音量・ミュート状態を表示したい」もご覧ください。

## 9-5. Windowsのように、ALT-F4キーで、アプリを終了させたい

```
⌘ vim ~/.fvwm2rc
```

```
key F4 A M Close
```

すでに設定済み

※アクティブウィンドウがある場合にALT-F4キーを押すと、アクティブウィンドウのアプリが終了する。

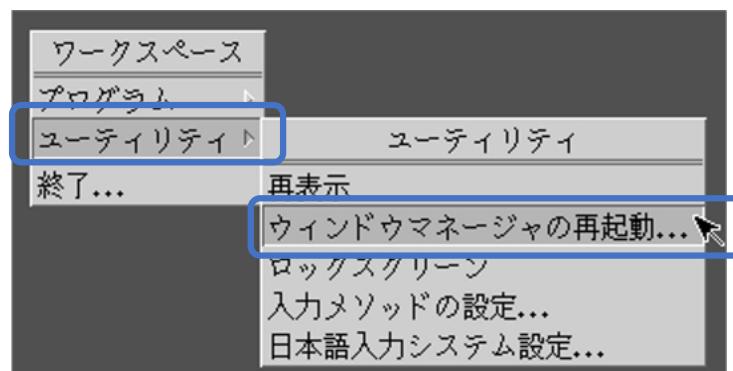
アクティブウィンドウが無い場合には、マウスがドクロマークに変化するので、終了したいアプリの上でクリックすると、終了する。

## 9-6. タイトルバーのダブルクリックで、ウィンドウシェードになるようにしたい

タイトルバーの部分のみをデスクトップ上に残します。

```
% vim ~/.fvwm2rc

{
# タイトルバーをクリック・ドラッグ・ダブルクリックした時の動作
DestroyFunc Func-TitleMouse
AddToFunc    Func-TitleMouse I Raise
+
+                               M Move
+                               D WindowShade ← 変更する
}
```

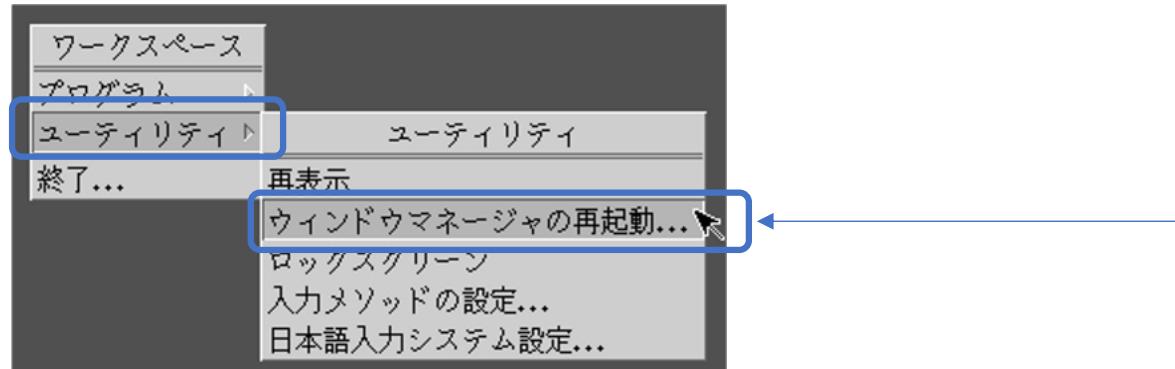


ウィンドウマネージャの再起動  
をクリックする

## 9-7. タイトルバーをダブルクリックした際は、縦方向のみ最大化したい

```
% vim ~/.fvwm2rc

{
# タイトルバーをクリック・ドラッグ・ダブルクリックした時の動作
DestroyFunc Func-TitleMouse
AddToFunc    Func-TitleMouse I Raise
+
+                               M Move
+                               D Maximize 0 100 ← 変更する
}
```



ウィンドウマネージャの再起動  
をクリックする

## 続き | 9-7. タイトルバーをダブルクリックした際は、縦方向のみ最大化したい

■前述した方法とは別に、①タイトルバーをCTRLキーを押しながらダブルクリックした場合、②最大化ボタンをCTRLキーを押しながらクリックした場合に、縦方向のみ最大化する方法を紹介します。

```
% vim ~/.fvwm2rc
{
# タイトルバーをクリック・ドラッグ・ダブルクリックした時の動作
DestroyFunc Func-TitleMouse
AddToFunc   Func-TitleMouse I Raise
+
+           M Move
+           D Maximize ← 元に戻す

# CTRLキーを押しながら
# タイトルバーをクリック・ドラッグ・ダブルクリックした時の動作
DestroyFunc Func-TitleMouseCtrl
AddToFunc   Func-TitleMouseCtrl I Raise
+
+           M Move
+           D Maximize 0 100
```

次ページへ

追加する

## 続き | 9-7. タイトルバーをダブルクリックした際は、縦方向のみ最大化したい

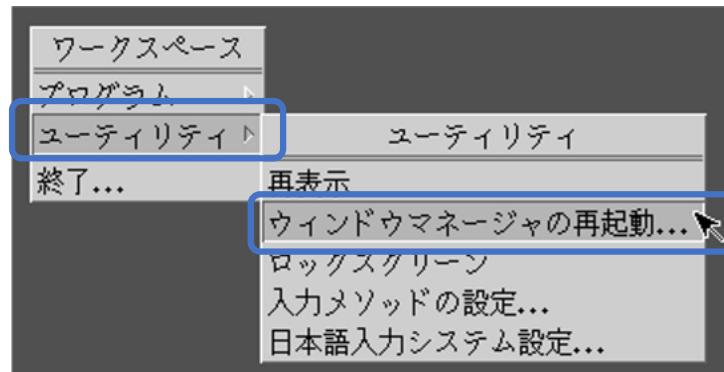
前ページより

```
{  
# タイトルバーの各ボタンをマウスで左クリックした時の動作  
Mouse 1 1 A Menu Window-Ops Close  
Mouse 1 2 A Close  
Mouse 1 4 A Maximize  
Mouse 1 4 C Maximize 0 100 ← 追加する  
Mouse 1 6 A Iconify  
}  
# タイトルバーをマウス操作した時の動作  
Mouse 1 T A Func-TitleMouse  
Mouse 1 T C Func-TitleMouseCtrl ← 追加する
```

次ページへ

## 続き | 9-7. タイトルバーをダブルクリックした際は、縦方向のみ最大化したい

前ページより



ウィンドウマネージャの再起動  
をクリックする

## 9-8. 画面表示スケールを変更したい

### ■ 画面表示スケール 125% の例

```
% xrandr  
Screen 0: minimum 320 x 200, current 1366 x 768, maximum 16384 x 16384  
LVDS-1 connected primary 1366x768+0+0 (normal left inverted right x axis y axis) 277mm x 156mm  
    1366x768      60.02*+
```

```
% xrandr --output LVDS-1 --scale 0.75x0.75
```

※どの程度実用的に使えるのかは不明です。

## 9-9. フォントを追加したい

- フォントを追加したい (pkg install以外の方法)

```
% mkdir ~/.fonts
```

```
~/fonts内にフォントファイルをコピーする
```

```
% fc-cache -fv ← システム対し、インストールされているフォントを再認識させる
```

```
% fc-list
```

追加したいフォント名が表示されることを確認する

## 9-10. クリップボードの不具合を解決したい

コピーしたのにペーストできない、などの不具合を解決する方法。

```
% sudo pkg install -y autocutsel
```

```
% vim ~/.xinitrc
```

```
{
```

```
autocutsel -selection PRIMARY &
autocutsel -selection CLIPBOARD &
```

```
exec fvwm3
```

} 追加する

X Window System上の2つのクリップボードを同期させる設定

## 続き | 9-10. クリップボードの不具合を解決したい

ウィンドウシステムをいったん終了する



ウィンドウシステムを起動する



※X Window Systemのクリップボードのメカニズム詳細は、「X11 CLIPBOARD PRIMARY」で検索。

## 9-11. 一部のフォントを無効にしたい

- 一部のフォントを無効にしたい (下記の例は梅明朝は使うが、梅明朝S3は使わない例)

```
% mkdir -p ~/.config/fontconfig
```

```
% vim ~/.config/fontconfig/fonts.conf
```

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<fontconfig>
  <selectfont>
    <rejectfont>
      <pattern>
        <patelt name="family">
          <string>梅明朝S3</string>
        </patelt>
      </pattern>
    </rejectfont>
  </selectfont>
</fontconfig>
```

繰り返すことで、複数の  
フォントを指定可能

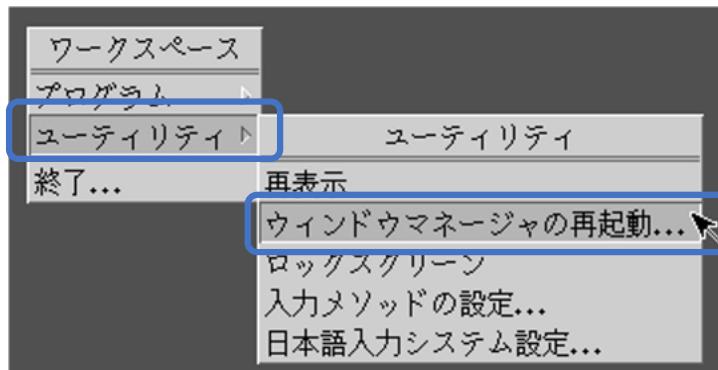
```
% fc-cache -fv
```

## 9-12. CTRL+ALT+DELで画面ロックしたい

### ■ CTRL+ALT+DELで画面ロックしたい

コメント##を外す

```
% vim ~/.fvwm2rc
{
    Key Delete A CM Exec xlock -mode blank -mousemotion -timeout 10
}
```



ウィンドウマネージャの再起動  
をクリックする

## 9-13. GTK系アプリのデフォルトフォントを変更したい

※3.初期設定（フォント設定）にて設定済みです。

### □GTK2系

```
% vim ~/.gtkrc-2.0  
  
gtk-font-name="Noto Sans JP" ← 新規に追加する
```

※当方の環境では、アプリ起動時に Default font does not have a positive size と表示されるが、  
フォントは変更される。

### □GTK3系

```
% mkdir -p ~/.config/gtk-3.0  
% vim ~/.config/gtk-3.0/settings.ini  
  
[Settings]  
gtk-font-name=Noto Sans JP 9 } 新規に追加する
```

Noto Sans JP のようにサイズ指定無しも可

## 9-14. フォントのアンチエイリアスをグレースケール方式にしたい

※3.初期設定（フォント設定）にて設定済みです。

■フォントのアンチエイリアスをグレースケール方式にしたい（GTK2系）

```
% vim ~/.Xresources
```

```
xft.rgba: none
```

新規に追加する

```
% vim ~/.xinitrc
```

```
{
```

```
xkbcomp -I$HOME/.xkb ~/xkb/keymap/mykbd $DISPLAY
```

```
xrdb -merge ~/.Xresources
```

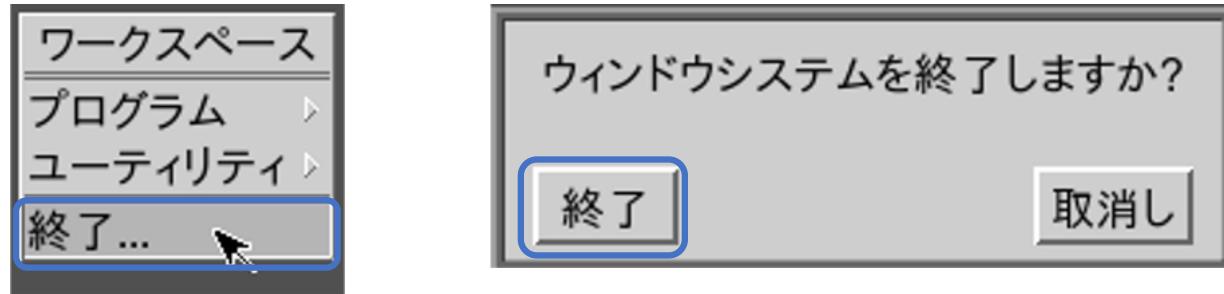
追加する

```
}
```

(次ページに続く)

## 続き | 9-14. フォントのアンチエイリアスをグレースケール方式にしたい

ウィンドウシステムをいったん終了する



ウィンドウシステムを再起動する

```
% startx
```

※ 9-17. IPA系フォントのアンチエイリアスを改善したい（GTK2系）も参照のこと。

## 9-15. デュアルモニター化（2画面表示）したい

### ■デュアルモニター化（2画面表示）したい

ここでは、ノートパソコンに外部のモニターを接続し、デュアルモニター化するように設定します。

- ・モニターを接続する。
- ・FreeBSDを起動する。
- ・ログインする。
- ・startxする。
- ・すでに、デュアルモニター化の状態となっている。

(次ページに続く)

## 続き | 9-15. デュアルモニター化（2画面表示）したい

□画面配置を変更したい

デュアルモニター時、X Window System が認識する解像度

```
% xrandr
```

Screen 0: minimum 320 x 200, current 2646 x 768, maximum 16384 x 16384  
LVDS-1 connected primary 1366x768+0+0 (normal left inverted right x axis y axis) 277mm x 156mm  
1366x768 60.02\*+  
HDMI-1 connected 1280x768+1366+0 (normal left inverted right x axis y axis) 330mm x 198mm  
1280x768 59.87\*+

外部モニター

ノートパソコンのモニター

```
% xrandr --output HDMI-1 --auto --left-of LVDS-1
```

ノートパソコンのモニターの左側に外部モニターを配置する例

配置	オプション	補足
左	--left-of	
右	--right-of	デフォルト
上	--above-of	
下	--below-of	

(次ページに続く)

## 続き | 9-15. デュアルモニター化（2画面表示）したい

- fvwmbuttons、xclock、uim-toolbar、xfce4-terminalの表示位置を変更する

```
% vim ~/.fvwm2rc
```

\* FvwmButtons: ButtonGeometry 48x48+**1803-0** ← 座標を適宜変更する

```
% vim ~/.xinitrc
```

uim-toolbar-gtk3 +**2495-1** ← 座標を適宜変更する

xclock ~ -geometry +**2437+1** ~ ← 座標を適宜変更する

xterm -geometry 109x5+**1280+1** ~ ← 座標を適宜変更する

xfce4-terminal --geometry=80x24+**1280+97** ~ ← 座標を適宜変更する

※上記は、ノートパソコンの1366x768のモニターに、1280x768の外部モニターを左側に設置した場合の例である。（次ページの図も参照）

(次ページに続く)

## 続き | 9-15. デュアルモニター化（2画面表示）したい

### □設定の恒久化

```
% vim ~/.xinitrc  
}  
xrandr --output HDMI-1 --auto --left-of LVDS-1 ← 追加する  
exec fvwm3
```



## 9-16. Firefoxから画像をドラッグ&ドロップしたい

### ■Firefoxから画像をドラッグ&ドロップでダウンロードしたい

「8-23.ファイル管理ソフト Thunar を使いたい」で Thunar をインストールすると、Chromium から画像をドラッグ&ドロップでダウンロードすることが可能ですが、FireFoxでは「サポートしていない操作です」と表示されます。これを回避する手順を紹介します。（8-23.でThunarがインストールされていることが前提です）

```
% dbus-launch thunar
```

- Firefox から、web ブラウザ上の画像を thunar のウィンドウにドラッグ&ドロップする。\*1

\*1 ウェブサイトの作りによっては、ダウンロードできない画像もあります（Google画像検索など）。  
その場合は、右クリックでの画像を保存...を推奨。

## 9-17. IPA系フォントのアンチエイリアスを改善したい

※3. 初期設定（フォント設定）にて設定済みです。

### ■IPA系フォントのアンチエイリアスを改善したい（GTK2,3系）

IPA系フォント（IPAゴシック、IPA Pゴシックなど）のアンチエイリアスに、違和感を感じた場合に対策する方法です。下記の説明では、「9-14. フォントのアンチエイリアスをグレースケール方式にしたい（GTK2系）」の設定も行なっていることを前提としています。

```
% vim ~/.Xresources
```

```
Xft.hintstyle: hintslight ← 追加する
```

ウィンドウシステムをいったん終了する



ウィンドウシステムを再起動する

```
% startx
```

## 9-18. ウィンドウのクラス名を取得したい

### ■ ウィンドウのクラス名を取得したい

.fvwm2rcの先頭部分に記載されたランチャーのアプリ名 = クラス名を取得する方法について解説します。

```
٪ xprop | grep WM_CLASS
```

- ・情報を得たいウィンドウ（アプリ）をクリックする。

```
WM_CLASS(STRING) = "Navigator", "firefox-esr"
```

これがクラス名

## 9-19. 音量調整時に、画面上に音量・ミュート状態を表示したい

9-4. キーボード上のボリュームキーにて音量調節をしたい、の改良版です。

※macOS風に表示します。

### □スクリプトのダウンロード

```
% mkdir ~/bin ← すでに存在するかもしれません  
% cd ~/bin  
% fetch https://raw.githubusercontent.com/moginwc/freebsd143/refs/heads/main/bin/volume_osd_client.tcsh  
% fetch https://raw.githubusercontent.com/moginwc/freebsd143/refs/heads/main/bin/volume_osd_daemon.py
```

上記スクリプトは生成AIで作成し、当方が若干修正しました。

次々ページに補足説明あり。

### □実行権の付与

```
% chmod +x ~/bin/volume_osd_client.tcsh
```

### □volume\_osd\_daemon.pyの修正（表示位置の修正）

```
% vim ~/bin/volume_osd_daemon.py  
  
self.root.geometry("200x200+583-136") # 必要に応じて位置/サイズを変更
```

(画面横の解像度 - 200) / 2

画面中央になるように修正する。

## 続き | 9-19. 音量調整時に、画面上に音量・ミュート状態を表示したい

- Arialフォントのインストール（スピーカーの絵文字を利用するため）

```
% sudo pkg install -y webfonts
```

※インストール済みかもしれません

- .fvwm2rcの修正（ノートパソコンなど、キーボード上に音量キーが存在する場合）

```
% cd ~  
% vim ~/.fvwm2rc
```

```
# ボリュームUP/DOWN/MUTEボタンの動作  
#Key XF86AudioRaiseVolume A A Exec mixer vol.volume=+0.05  
#Key XF86AudioLowerVolume A A Exec mixer vol.volume=-0.05  
#Key XF86AudioMute A A Exec mixer vol.mute=^  
  
# ボリュームUP/DOWN/MUTEボタンの動作(OSD版)  
Key XF86AudioRaiseVolume A A Exec tcsh -c "mixer vol.volume=+0.05; volume_osd_client.tcsh"  
Key XF86AudioLowerVolume A A Exec tcsh -c "mixer vol.volume=-0.05; volume_osd_client.tcsh"  
Key XF86AudioMute A A Exec tcsh -c "mixer vol.mute=^; volume_osd_client.tcsh mute"
```

コメント#を外す

コメントアウ  
トする

## 続き | 9-19. 音量調整時に、画面上に音量・ミュート状態を表示したい

### □.fvwm2rcの修正（デスクトップパソコンなど、101や106キーボードの場合）

```
⌚ cd ~  
⌚ vim ~/fvwm2rc
```

```
# 音量操作(CTRL+ALT+↑で音量大、CTRL+ALT+↓で音量小、CTRL+ALT+Mでミュート/解除)  
#Key Up A CM Exec mixer vol.volume=+0.05  
#Key Down A CM Exec mixer vol.volume=-0.05  
#Key M A CM Exec mixer vol.mute=^  
  
# ボリュームUP/DOWN/MUTEボタンの動作(OSD版)  
Key Up A CM Exec tcsh -c "mixer vol.volume=+0.05; volume_osd_client.tcsh"  
Key Down A CM Exec tcsh -c "mixer vol.volume=-0.05; volume_osd_client.tcsh"  
Key M A CM Exec tcsh -c "mixer vol.mute=^; volume_osd_client.tcsh mute"
```

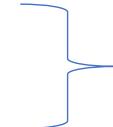
コメント#を外す

コメントアウ  
トする

## 続き | 9-19. 音量調整時に、画面上に音量・ミュート状態を表示したい

### □ .xinitrc の修正

```
% vim ~/.xinitrc
:
autocutsel -selection CLIPBOARD &
rm -f /tmp/volume_osd_value
python3.11 $HOME/bin/volume_osd_daemon.py &
exec fvwm3
```



この位置に  
追加する

## 続き | 9-19. 音量調整時に、画面上に音量・ミュート状態を表示したい

ウィンドウシステムをいったん終了する



ウィンドウシステムを再起動する

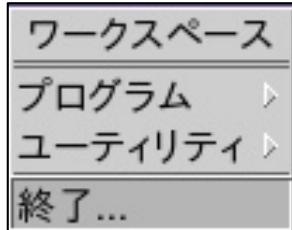
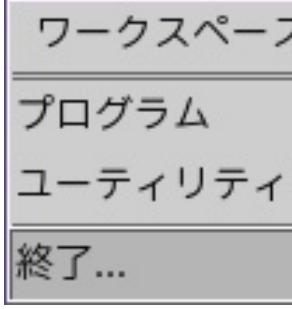
```
% startx
```

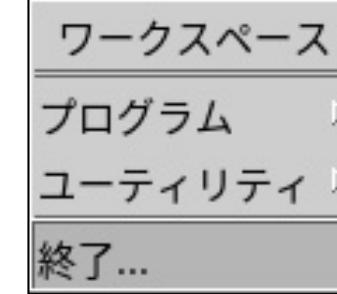
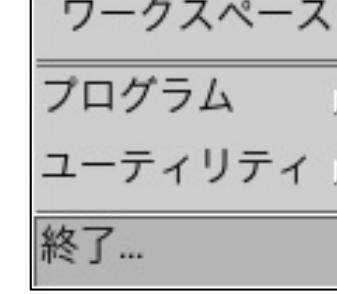
キーボード上の音量ボタン・ミュートボタンを押すと、  
画面上に音量（%）やMUTE状態が表示される。

□補足

volume\_osd\_client.tcsh は、現在の音量の状況を /tmp/volume\_osd\_value に書き込むスクリプトです。volume\_osd\_daemon.py は、/tmp/volume\_osd\_value を監視し、変化があった場合に画面に音量を表示する常駐スクリプトです。なお、両スクリプトとともに、音量設定そのものは行いません。2つのプログラムに分けた理由は、音量キーを連打した場合の対応のためです。

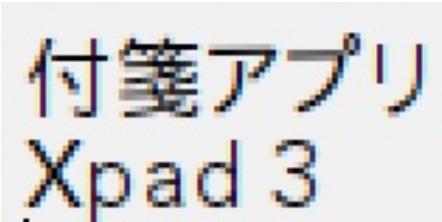
## 9-20. (メモ書き) ルートメニューのデフォルトフォント検証

フォント名	FVWMルートメニュー
<u>IPA Pゴシック</u> サイズ : 10.5	 <p>バランス良い。*1 このフォントを採用した。</p> <p>*1 小書き文字（ここでは「ィ」）の幅が違う。 よく見ると「リ」の幅も違う。</p>
<u>IBM Plex Sans JP</u> サイズ : 10.5	 <p>文字幅、行間がやや広め。</p>

フォント名	FVWMルートメニュー
<u>Noto Sans JP</u> サイズ : 10.5	 <p>文字幅、行間がやや広め。 文字がやや下寄り。</p>
<u>ヒラギノ角ゴ Pro</u> サイズ : 10.5 <small>Mac OS X 10.6頃のフォント</small>	 <p>文字幅、行間がやや広め。 文字が上寄り。</p>

## 9-21. (メモ書き) フォントのアンチエイリアスについて

9-14. フォントのアンチエイリアスをグレースケール方式にしたい (GTK2系)、および 9-17. IPA系フォントのアンチエイリアスを改善したい (GTK2,3系) の詳細をメモしておきます。

.Xresources フォント	なし	Xft.rgba: none	Xft.rgba: none Xft.hintstyle: hintslight
Xpad 3 (GTK2系)  IPA Pゴシック サイズ : 12	 輪郭に青や赤の色が混じる (サブピクセル方式)	 輪郭はグレーになる (グレースケール方式)	 フォントが若干なめらかになる
uim-toolbar-gtk3 (GTK3系)  IPA Pゴシック	 輪郭にはグレーになる (グレースケール方式)	 フォントが若干なめらかになる	

※上記アプリ以外でも、GTK2,3系アプリのメニューバーなどに .Xresources の設定は影響します。

## 続き | 9-21. (メモ書き) フォントのアンチエイリアスについて

GTK3系アプリでは、下記のファイルを作成することでも、アンチエイリアスの方法が指定可能です（.Xresourcesの設定よりも優先される？）。

```
% mkdir -p ~/.config/fontconfig
% vim ~/.config/fontconfig/fonts.conf

<?xml version='1.0'?>
<!DOCTYPE fontconfig SYSTEM 'fonts.dtd'>
<fontconfig>

    <!-- グレースケールレンダリング -->
    <match target="font">
        <edit name="rgba" mode="assign">
            <const>none</const>
        </edit>
    </match>

    <!-- ヒンティングを有効に -->
    <match target="font">
        <edit name="hinting" mode="assign">
            <bool>true</bool>
        </edit>
        <edit name="hintstyle" mode="assign">
            <const>hintslight</const>
        </edit>
    </match>

</fontconfig>
```

## 9-22. (メモ書き) .fvwm2rcファイルの解説

### ■ タイトルバーをクリック・ドラッグ・ダブルクリックした時の動作

#### # 1: ファンクション

##### # タイトルバーをクリック・ドラッグ・ダブルクリックした時の動作

```
DestroyFunc Func-TitleMouse  
AddToFunc Func-TitleMouse I Raise  
+ M Move  
+ D Maximize
```

:

#### # 5: マウスとキー設定

##### # タイトルバーでマウス操作した時の動作

```
Mouse 1 T A Func-TitleMouse
```

→ Func-TitleMouse を実行する

記号	状況	内容
I	クリック直後	ウインドウを前面にする Raise
M	ドラッグした場合	ウインドウを動かす Move
D	ダブルクリックした場合	ウインドウを最大化する Maximize

→ どの修飾キーが押されていても・押されていなくても

→ タイトルバーの上で

→ マウスの左ボタンをクリック

- 1 · · · 左ボタン
- 2 · · · 中ボタン
- 3 · · · 右ボタン

A · · · (Any)  
C · · · Ctrl  
S · · · Shift  
M · · · Alt  
N · · · (Nothing)  
4 · · · WindowsKey

複数記載も可 (例 : CS)

## 続き | 9-22. (メモ書き) .fvwm2rcファイルの解説

### ■ ランチャー(chromium) 起動またはフォーカスする

```
# ランチャー(chromium) 起動またはフォーカスする
DestroyFunc Func-LaunchOrRaise_chromium
AddToFunc Func-LaunchOrRaise_chromium
+ I All (Chromium-browser) Raise
+ I Next (Chromium-browser) Focus
+ I None (Chromium-browser) Exec chrome --no-default-browser-check
```

上から  
順番に  
即実行

ウインドウクラス名については9-18.を参照

コマンド	意味	上記スクリプトの実行内容
All	条件に合う全てのウインドウに対して、コマンドを実行する。	ウインドウクラス名「Chromium-browser」があれば、Raise（前面に表示）する。
Next	条件に合う「最初に見つけた」ウインドウに対して、コマンドを実行する。	ウインドウクラス名「Chromium-browser」があれば、最初に見つけたウインドウを Focus（フォーカス）する。
None	条件に合うウインドウがなければ、コマンドを実行する。	ウインドウクラス名「Chromium-browser」がなければ、chromium を起動する。

【注意点】ウインドウクラス名は、ウインドウが画面上に描画された段階から取得できるので、アプリが実行されてからウインドウが表示されるまでにタイムラグがあるアプリの場合、上記の例では2重起動する可能性がある。

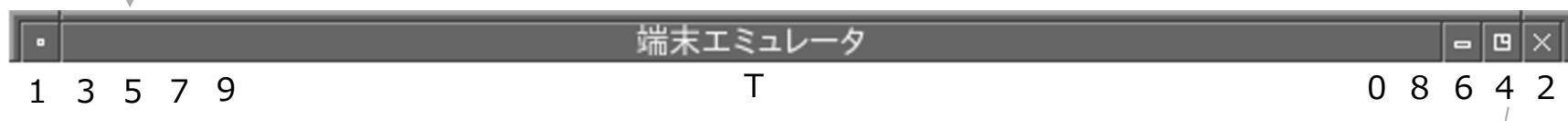
# 続き | 9-22. (メモ書き) .fvwm2rcファイルの解説

## ■ タイトルバーボタン関連

### # 5: マウスとキー設定

# タイトルバーの各ボタンをマウスで左クリックした時の動作

```
Mouse 1 1 A Menu Window-Ops Close  
Mouse 1 2 A Close  
Mouse 1 4 A Maximize  
Mouse 1 4 C Maximize 0 100  
Mouse 1 6 A Iconify
```



### # 6: ウィンドウの装飾

#### # 最大化ボタン

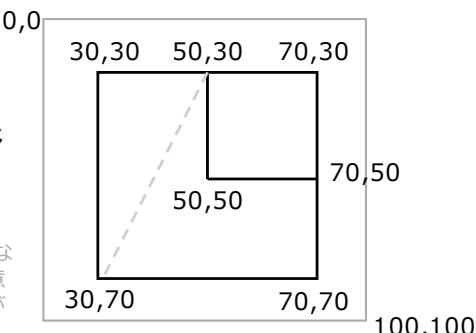
ButtonStyle 4 AllActive      Vector 8 30x70@3 30x30@3 70x30@3 70x70@3 30x70@3 50x30@4 50x50@3 70x50@3

8点分の座標に対し、その順番で線を引く、の意味 (①→②、②→③・・・)

#### 座標イメージ

(100x100固定)

100x100のイメージを小さなピクセルで表示するため、意図した表示にならないことがある（1ドットずれるなど）。



左記図の点線の部分  
(描画しない部分)

# 10-1. FreeBSD上でWindowsのアプリを使いたい

## ■ FreeBSD上でWindowsのアプリを使いたい (Wine)

※※※ 正常に動作しないアプリ多数あり ※※※

【おことわり】 安定した動作を期待するのであれば、Windowsパソコンを用意して、リモートデスクトップ経由でWindowsアプリを使用したほうが良さそうです。

ここでは、

- ・ Wineのインストール・初期設定
- ・ インストーラー形式のWindowsアプリのインストールと起動
- ・ zip形式のWindowsアプリのインストールと起動

を解説します。

Wineのインストール①

```
⌘ sudo pkg install -y wine wine-gecko wine-mono winetricks
```

# 続き | 10-1. FreeBSD上でWindowsのアプリを使いたい

## Wineのインストール②

```
% wine  
/home/pcuser/.i386-wine-pkg//usr/local/bin/wine doesn't exist!  
  
Try installing 32-bit Wine with  
    /usr/local/share/wine/pkg32.sh install wine-devel mesa-dri  
  
If using Poudriere, please make sure your repo is setup to use  
FreeBSD:14:i386  
and create symlinks for  
    FreeBSD:14:amd64 and  
    FreeBSD:14:i386  
to the relevant output directories. See pkg.conf(5) for more info.  
  
% /usr/local/share/wine/pkg32.sh install wine mesa-dri
```

Proceed with this action? [y/N] : **y**

## 初期設定①

```
% winecfg
```

何もせず OK ボタンで終了する。

# 続き | 10-1. FreeBSD上でWindowsのアプリを使いたい

## 初期設定②（フォントのインストール）

```
% winetricks corefonts
```

コマンドラインからインストールしたいフォントを指定すると、しつこいエラーメッセージは表示されない。

※ webfonts をFreeBSD側にインストールしていれば、上記の corefonts のインストールは不要と思われる（全く同じフォントをインストールする模様）が、winetricksではWindows内のレジストリも変更しているようなので、winetricksの利用を推奨。なおフォントは、~/.wine/drive\_c/windows/Fonts にインストールされる。

```
% sudo pkg install -y ja-font-ume
```

※ MS (P,UI) ゴシック・MS (P) 明朝に構造が似ているフォントをインストールしている。ただしumeフォントはビットマップフォントを内包していないので、アプリによっては20ポイント以下の文字はギザギザ表示されます。

# 続き | 10-1. FreeBSD上でWindowsのアプリを使いたい

## 初期設定③（代替フォントの設定）

```
% fetch https://raw.githubusercontent.com/moginwc/freebsd143/refs/heads/main/wine-japanese.reg.txt
```

※ 代替フォントの設定ファイルをダウンロードしている \*1。

Windows Registry Editor Version 5.00

' 文字コードをUTF-16リトルエンディアン、改行をCR+LFに変換してから  
使用してください

```
[HKEY_CURRENT_USER\Software\Wine\Fonts\Replacements]
"Meiryo UI"="Noto Sans JP"
"Meiryo"="Noto Sans JP"
"MS Gothic"="Ume Gothic"
"MS Mincho"="Ume Mincho"
"MS PGothic"="Ume P Gothic"
"MS PMincho"="Ume P Mincho"
"MS Sans Serif"="Ume P Gothic"
"MS Shell Dlg"="Ume UI Gothic"
"MS UI Gothic"="Ume UI Gothic"
"MS Pゴシック"="Ume P Gothic"
"MS P明朝"="Ume P Mincho"
"MS ゴシック"="Ume Gothic"
"MS 明朝"="Ume Mincho"
```

```
"Tahoma"="Noto Sans JP"
"Yu Gothic UI Light"="Noto Sans JP Light"
"Yu Gothic UI Semibold"="Noto Sans JP Medium"
"Yu Gothic UI Semilight"="Noto Sans JP DemiLight"
"Yu Gothic UI"="Noto Sans JP"
"Yu Gothic Light"="Noto Sans JP DemiLight"
"Yu Gothic Medium"="Noto Sans JP Medium"
"Yu Gothic"="Noto Sans JP"
"Yu Mincho Demibold"="Noto Serif JP SemiBold"
"Yu Mincho Light"="Noto Serif JP DemiLight"
"Yu Mincho"="Noto Serif JP"
"メイリオ"="Noto Sans JP"
"游ゴシック"="Noto Sans JP"
"游明朝"="Noto Serif JP"

' 追加
"SimSun"="Ume P Gothic"
"Segoe UI"="Ume P Gothic"
```

\*1 : nogajunさん@GitHub Gistのwine-japanese.reg 「LinuxのWine環境で日本語周りをいい感じに設定するレジストリデータ」を参考に、当方が若干手直しをしました。

# 続き | 10-1. FreeBSD上でWindowsのアプリを使いたい

## 初期設定③（代替フォントの設定）（続き）

```
% sudo pkg install -y ja-nkf ← インストール済みかもしれません  
% nkf -W8 -w16L -Lw wine-japanese.reg.txt > wine-japanese.reg
```

ファイルのエンコードをUTF-16リトルエンディアン、  
改行コードをCR+LFに変換している。

```
% regedit /s wine-japanese.reg
```

レジストリエディタを使用し、代替フォントを設定している。

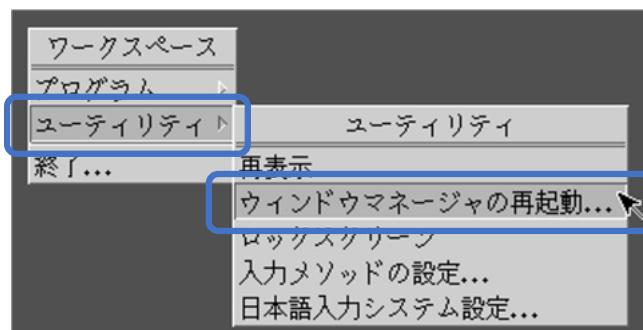
# 続き | 10-1. FreeBSD上でWindowsのアプリを使いたい

## 初期設定④ (ダイアログボックスのコントロール)

```
% vim ~/.fvwm2rc
:
Style * DecorateTransient
Style "* .exe" !DecorateTransient
Style * StackTransientParent
Style * FPRleaseFocusTransient
:
```

※Wine 10.0 と fvwm3 の組み合わせでは、ダイアログボックスを移動するとボタンが押せなくなるなど、操作不能になる不具合が発生します。そのため、ダイアログボックス表示時は、ウィンドウを移動できないよう、タイトルバーを非表示にする設定にしています。

①先頭の#wine#を削除する



②ウィンドウマネージャ  
の再起動をクリックする

# 続き | 10-1. FreeBSD上でWindowsのアプリを使いたい

- インストーラー形式のWindowsアプリのインストールと起動  
(※ここでは Windows版テキストエディタ Mery (64bit) の例)

- ・ ウェブブラウザにて、[Mery Ver 2.6.7 \(64ビット版\) インストーラー \(MerySetup-x64-2.6.7.exe\)](#) をダウンロードする

```
% cd ~/ダウンロード ← または cd ~/Downloads  
% wine ./MerySetup-x64-2.6.7.exe
```

インストーラーの指示に従う

```
% wine ~/.wine/drive_c/Program Files/Mery/Mery.exe
```

→ スペースをエスケープする

※ 32bitアプリは下記のように、カッコもエスケープする。

```
% wine ~/.wine/drive_c/Program Files (x86)/Mery/Mery.exe
```

# 続き | 10-1. FreeBSD上でWindowsのアプリを使いたい

- zip形式のWindowsアプリのインストールと起動  
(※ここでは Windows版テキストエディタ Mery (64bit) の例)

```
% mkdir ~/wine_bin
```

zip形式のアプリを保存するためのディレクトリを作成しておく。  
(必須ではなく、わかりやすいように作成しています)

- ・ウェブブラウザにて、Mery Ver 2.6.7 (64ビット版) ZIP (Mery-x64-2.6.7.zip) をダウンロードする

```
% cd ~/ダウンロード ← または cd ~/Downloads
% unzip -O cp932 Mery-x64-2.6.7.zip
% mv Mery ~/wine_bin
% wine ~/wine_bin/Mery/Mery.exe
```

大文字のO (オー)

## 10-2. Wine + uim-anthy, mozcでのアプリ動作状況・課題一覧

### ■ Wine (Ver10.0) + uim-anthy, mozc でのアプリ動作状況・課題一覧 \*1

アプリ	内容	対策
全般	ダイアログボックスなど、ウインドウ表示後に元の画面にフォーカスが戻らない。	.fvwm2rcに Style * Lenience を追加する。 .fvwm2rcの #ダイアログやポップアップウインドウに 対し、装飾・重なり順・フォーカス動作を制御する、の 部分の記述が必要
全般	テキスト入力、テキストボックスなどにおいて、日本語入力が不安定になる。	不明。
全般	<u>ダイアログボックスなどのウインドウが表示されると、日本語入力モードが強制的にOFFになってしまう。</u>	もう一度日本語入力モードをONにして入力する。根本対策は不明。なお、ダイアログボックスを閉じると、モードが復活する。
全般	<u>日本語入力中、変換候補がウインドウ左上に表示されてしまう。</u>	不明（切実な問題）。なおAnthyの場合、uim-pref-gtk3 → Anthy → 候補ウインドウのチェックを外す、で緩和可能。
全般	フォントが乱れる。フォントがおかしい。	フォントの設定をする。
全般	フォントを変更すると、漢字等が表示されなくなってしまう。	フォント指定時、Script:欄の「Japanese」を選択するのを忘れない。

\*1 一部、FreeBSD14.1、FreeBSD14.2、Wine9.0での検証結果も含みます（以降のページ含めて）。

※ 網掛けのアプリは、利用要注意、または実質利用不可。

## 続き | 10-2. Wine + uim-anthy,mozcでのアプリ動作状況・課題一覧

### □ Wine (Ver10.0) + uim-anthy,mozc でのアプリ動作状況・課題一覧 (続き)

アプリ	内容	対策
全般	uim-anthy/mozcで入力できる「～」は、 FULLWIDTH TILDEではなく、 WAVE DASHである。	
全般	uim-anthy/mozcで「-」を全角変換したものは、 FULLWIDTH HYPHEN-MINUSではなく、 MINUS SIGN（半角）である。	
全般	Windows (NTFS) では同一視されていたディレクトリ名・ファイル名の大文字小文字の挙動。	wineの仕様確認や各ソフトで検証が必要。
全般	Windowsで多用されているフォルダーナー名・ファイル名に空白を含む場合の対処方法。	cd "Program Files" とダブルクオーテーションで囲むか、 cd Program¥ Files などと、半角スペースをエスケープする。
全般	シンボリックリンクの取り扱い。	wineの仕様確認や各ソフトで検証が必要。
全般	ドットファイルの取り扱い。	wineの仕様確認や各ソフトで検証が必要。
全般	Windows (NTFS) では扱えないファイル名の取り扱い。	wineの仕様確認や各ソフトで検証が必要。

## 続き | 10-2. Wine + uim-anthy,mozcでのアプリ動作状況・課題一覧

### □ Wine (Ver10.0) + uim-anthy,mozc でのアプリ動作状況・課題一覧 (続き)

アプリ	内容	対策
全般	ファイル選択ダイアログで、特定のフォルダーのみアクセスできない。	そのフォルダーの中に、拡張属性を持ったファイルが存在し、それが影響している可能性大（特に、sambaで共有しているフォルダーで発生しやすい）。 % lsextattr user * にて、DOSATTRIBと表示されるファイルを削除するか、別のフォルダーにコピーして削除する。
テキストエディター 全般	日本語入力において、入力中の文字、および変換中の文字やカーソル位置がわかりにくい。	背景色を設定することで若干見やすくなる。 ただし、変換中の文字区切りのわかりにくさは改善しない（秀丸は改善可能）。
Visual Basic 6ランタイムを使用するアプリ 全般	起動しない。  タイトルバーなどが表示されない。  最大表示、アプリ終了のボタンが表示されない	<ul style="list-style-type: none"><li>「10-4. VB6のランタイムを必要とするソフトを起動させたい」を参照。</li><li>.fvwm2rcの #ダイアログやポップアップウィンドウに対し、装飾・重なり順・フォーカス動作を制御する、の部分が有効になっていることを確認する。</li><li>上記の内容を確認する。</li></ul>

## 続き | 10-2. Wine + uim-anthy,mozcでのアプリ動作状況・課題一覧

### □ Wine (Ver10.0) + uim-anthy,mozc でのアプリ動作状況・課題一覧 (続き)

アプリ	内容	対策
秀丸 9.50 64bit版	日本語入力において、入力中の文字、および変換中の文字やカーソル位置がわかりにくい。	その他(O)→ファイルタイプ別の設定(C)...→デザイン→IME変換中の色（※チェックしない）→入力モード(M)...→トゥルーインライン入力をチェックする。これにより入力中の文字にアンダーラインが付与される。また、変換中の文字区切りにカーソルキーが表示されるので多少わかりやすくなる（ただし違和感はある）。
秀丸	ファイルの読み書き時にエラーとなる。	winecfgにて、Applications・Hidemaru.exeのWindows Versionを「WindowsXP」にする。
秀丸 9.50 64bit版	IPAゴシックのサイズ設定が反映されない（12ポイント設定になる）。	不明。他のエディターでは問題なし。⇒10-14. Wineで使えるいい感じの等幅フォントを生成したいも参照。
秀丸 9.50 64bit版	ヘルプ表示時、左側のインデックスが再表示されない、検索できない、など。	不明。
秀丸 9.46 64bit版	winecfgのGraphicsタブにて、allow window manager to decorate the controlsのチェックを外した設定の際、アイコンを最小化した後に復帰させると、ウインドウタイトル色が非アクティブ時の色となってしまう。	実害なし。ウインドウ関係の操作をすると、復帰する。

## 続き | 10-2. Wine + uim-anthy,mozcでのアプリ動作状況・課題一覧

### □ Wine (Ver10.0) + uim-anthy,mozc でのアプリ動作状況・課題一覧 (続き)

アプリ	内容	対策
秀丸 9.50 64bit版	アウトライン解析の枠表示時、見出し表示部分のツールチップ表示がされた場合、速やかにクリックできない。(ホイールも効かない)	不明。枠の上の▼→詳細(S)...→ツールチップ表示のタイムアウト(O)を1秒にすることで緩和可能。
秀丸 9.50 64bit版	タブモードにすると、画面まわりの異常となる(マウスが表示されない、画面が黒くなるなど)。	不明。
Mery 2.6.7 64bit版	(簡単なチェック済み)	(テキストエディタ全般の課題をご覧ください)

## 続き | 10-2. Wine + uim-anthy,mozcでのアプリ動作状況・課題一覧

### □ Wine (Ver10.0) + uim-anthy,mozc でのアプリ動作状況・課題一覧 (続き)

アプリ	内容	対策
TeraPad 1.2.9 32bit版	(簡単なチェック済み)	(テキストエディタ全般の課題をご覧ください)
サクラエディタ 2.4.2.6048 32bit	終了系のメニューが効かないことがある。 コンソール側にメッセージが表示されることがある。	名前をつけて保存してから終了する。根本対策は不明。 不明。
秀丸フライヤー Classic 2.00 64bit版	ファイル名が右詰表示。ファイルサイズやファイル 変更時刻が表示されない。一部のアイコンが表示さ れない。	不明。
カシミール3D 9.4.2 32bit版	(簡単なチェック済み)	(なし)
Google Earth Pro 7.3.6.10441 64bit版	・インストーラーの画面が不安定。 <u>(簡単なチェック済み)</u>	・無視 <u>(なし)</u>
Radio Mobile 11.6.6 32bit版	・起動プログラムが不明 ・MSVBVM60.DLLが無いので起動しない ・起動直後から全般的に画面周りに異常がある	・rmweng.exe ・10-4. VB6のランタイムを必要とするソフトを起動さ せたい、を参考に設定。

## 続き | 10-2. Wine + uim-anthy,mozcでのアプリ動作状況・課題一覧

### □ Wine (Ver10.0) + uim-anthy,mozc でのアプリ動作状況・課題一覧 (続き)

アプリ	内容	対策
Binary Editor BZ 1.9.8.5 64bit版	フォントによっては、左側のバイナリーディスプレイと右側のキャラクターのカーソル位置が一致しなくなる場合がある。	一致するようなフォントを選択する。⇒10-14. Wineで使えるいい感じの等幅フォントを生成したいも参照。
WPS Office 2 Standard Edition	インストールできるものの、動作不安定。	不明。
Acrobat Reader 2025.001.21111 日本語 64bit版	「セットアップは中断されました」となる。	不明。
Adobe Reader 11.0.08 日本語 32bit版	「セットアップは中断されました」となる。	% winetricks atmlib msftedit mspatcha riched30 wsh57 → インストール → AdobeReader起動 → 2番目にチェック → OK ただし、検索窓などで日本語入力はできない。
TeraTerm 5.5.2 x64	(簡単なチェック済み)	(なし)
DiskMirroringTool Unicode 0.381 64bit版	シンボリックリンクは無視される。 (簡単なチェック済み)	他、ドットファイル、Windowsでは使えないファイル名などの挙動を確認する必要あり。 (なし)

## 続き | 10-2. Wine + uim-anthy,mozcでのアプリ動作状況・課題一覧

### □ Wine (Ver10.0) + uim-anthy,mozc でのアプリ動作状況・課題一覧 (続き)

アプリ	内容	対策
WinMerge 2.16.42.1 64bit版	インストール最後の「情報」ボックス内の文字が化けてしまう。 画面のちらつきが発生することがある。 (その他、簡単なチェック済み)	% winetricks msftedit  不明。 (なし)
WinMerge 2.16.52.2 64bit版	画面のちらつきが発生することがある。 メニューバーの表示がおかしい。 (その他、簡単なチェック済み)	不明。  (なし)
ねずみ将棋 1.02	ファイル名に日本語が含まれるので unzip -O cp932 NezumiShogi102.zip で展開する。 wine ねずみ将棋.exe で起動する。 (簡単なチェック済み)	(なし)
東方星蓮船 体験版 0.02a 32bit版	(簡単なチェック済み)	(なし)

## 続き | 10-2. Wine + uim-anthy,mozcでのアプリ動作状況・課題一覧

### □ Wine (Ver10.0) + uim-anthy,mozc でのアプリ動作状況・課題一覧 (続き)

アプリ	内容	対策
TCARD 7.97 32bit版	(簡単なチェック済み)	(なし)
付箋紙21FE v3.05 32bit版	「ようこそ、付箋紙21FEへ」の画面が表示されたまま、暴走する。	不明。
Kaku v1.3.2 32bit版	<ul style="list-style-type: none"> <li>常駐時のシステムトレイの表示に課題あり。</li> <li>指定できないフォントがある。</li> </ul> (簡単なチェック済み)	<ul style="list-style-type: none"> <li>stalonetrayなどを利用する。</li> <li>不明。</li> </ul> (なし)
FFFTP v5.8 64bit版	起動後に画面枠は表示されるものの、ハングアップする。	不明。
FFFTP v1.97b 32bit版	設定等のダイアログボックス画面で、右側や下側の設定項目が欠け、設定できない。 (簡単なチェック済み)	不明。 (なし)
JTrim V1.5.3c 32bit版	<ul style="list-style-type: none"> <li>MSVBVM60.DLLが無いので起動しない</li> <li>起動直後から全般的に画面周りに異常がある</li> </ul> (簡単なチェック済み)	<ul style="list-style-type: none"> <li>% winetricks vb6run を実行</li> <li>10-4. VB6のランタイムを必要とするソフトを起動させたい、を参考に設定。</li> </ul> (なし)

## 続き | 10-2. Wine + uim-anthy,mozcでのアプリ動作状況・課題一覧

### □ Wine (Ver10.0) + uim-anthy,mozc でのアプリ動作状況・課題一覧 (続き)

アプリ	内容	対策
AviUtil 1.10 32bit版	・フォント選択画面で、同じフォントが多数表示される。 (簡単なチェック済み)	不明 (なし)
AviUtil 2 beta29	・起動しない	不明
分度器で測りましょ Version 1.2.0 32bit版	透過できない (簡単なチェック済み)	不明 (なし)

## 続き | 10-2. Wine + uim-anthy, mozcでのアプリ動作状況・課題一覧

### 補足：Windowsアプリで日本語を安定表示させるためのフォントについて

国産のWindowsアプリでは、画面まわりにMSゴシック系（PやUIを含む）のフォントが使われることが多く、Wine上で安定して動かすためには、MSゴシックに似たフォント（字体というより、文字幅・行間・全角と半角のバランスなどが近いもの）を使用することが重要なポイントだと考えます。

しかし、意外と一筋縄ではいかず、イロイロな症状が発生するため、ここにその状況をまとめておきます。

フォント名	等幅	P	UI	ビットマップ	字形	販売・サポート状況	課題
HGゴシックB (リコー) *1	あり	あり	なし	あり 20dot以下	JIS 2004	購入可能。	UIフォントなし。
LXゴシック (リコー)	あり	あり	なし	あり 20dot以下	JIS 90	なし。中古品のみ。	UIフォントなし。入手困難（過去のVine LinuxやTurbo Linuxの商用版の一部にバンドルされていた）。
梅ゴシック	あり	あり	あり	なし	JIS 2004	不明。	サイズが18以下のフォントは、ギザギザフォントになってしまう。
IPA モナー ゴシック	あり	あり	あり	あり 8~12	JIS 90	不明。	文字幅がわずかに異なる。 UIフォントをメニューバーに設定すると、 フォントがかかれてしまう。

\*1 当方、未検証です。

## 続き | 10-2. Wine + uim-anthy, mozcでのアプリ動作状況・課題一覧

### 補足：Windowsアプリでの常用等幅フォントについて

Wine上のWindowsアプリで使用する等幅フォントについても、イロイロな症状が発生するため、ここにその状況をまとめておきます（フリーのものが対象）。なお、フォントサイズの検証範囲は、9～24までです。

アプリ名	梅ゴシック	IPAゴシック	IPAモナゴシック	BIZ UDゴシック
秀丸エディタ	サイズ9～18までは、ギザギザフォント。△	サイズ設定が効かないことがある。×	サイズ9～12までは、ビットマップフォント。○	サイズ12～24は、文字幅異常となる（16のみ大丈夫）。×
Mery	サイズ9～19までは、ギザギザフォント。△	サイズ9～12までは、ギザギザフォント。それ以降は縦方向のアンチエイリアスがかかりにくい。○△	サイズ9～12までは、ビットマップフォント。○	○
TeraPad	サイズ9～18までは、ギザギザフォント。△	サイズ9～12までは、ギザギザフォント。それ以降は縦方向のアンチエイリアスがかかりにくい。○△	サイズ9～12までは、ビットマップフォント。○	文字幅異常となる。×

次ページへ

## 続き | 10-2. Wine + uim-anthy, mozcでのアプリ動作状況・課題一覧

補足：Windowsアプリでの常用等幅フォントについて（続き）

アプリ名	梅ゴシック	IPAゴシック	IPAモナゴシック	BIZ UDゴシック
Bz	サイズ9～18までは、ギザギザフォント。△	サイズ9～12までは、ギザギザフォント。○	左側のバイナリーと右側の文字ウインドウでカーソル位置がズレる。×	左側のバイナリーと右側の文字ウインドウでカーソル位置がズレる。×
TCARD	サイズ9～18までは、ギザギザフォント。△	サイズ9～12までは、ギザギザフォント。○	サイズ9～12までは、ビットマップフォント。○	○
WinMerge	サイズ9～18までは、ギザギザフォント。△	サイズ9～12までは、ギザギザフォント。○	サイズ9～12までは、ビットマップフォント。○	○

結論：ギザギザフォントを許容さえすれば、梅ゴシックが一番安定している。ただし、アプリごとにも良し悪しがあるので、お好みで使用する。

### 【追記】

上記等幅フォントの諸問題の解決のため、『10-14. Wineで使えるいい感じの等幅フォントを生成したい』をご覧ください。

※急いで検証したため、間違っている可能性がありますが、ご了承ください。

## 10-3. 秀丸エディターを使用したい

### ■ 秀丸エディター\*1を使用したい 以下は、Ver9.50の例で説明します。

一部、細かい操作方法は省略しています。

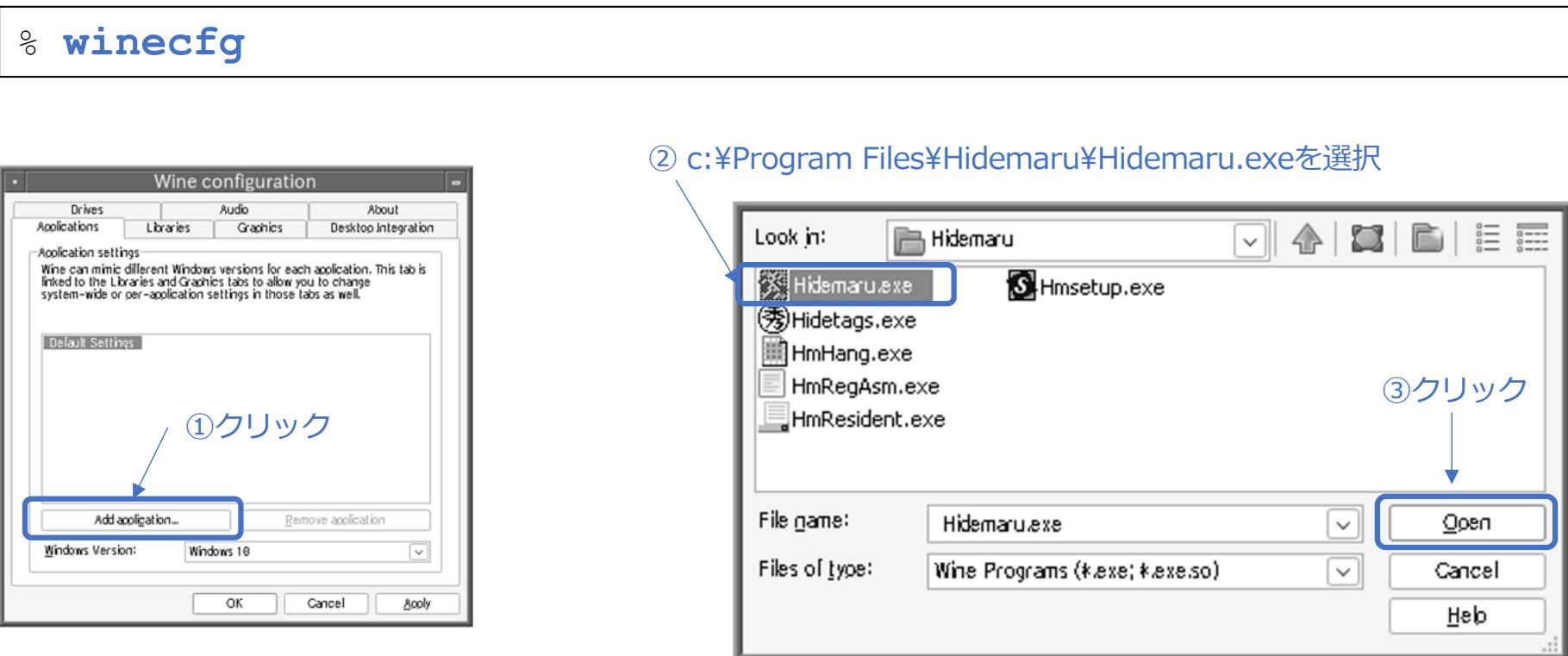
#### ・サイト企画のサイトより、秀丸エディタ64bit版をダウンロードする

```
% cd ~/ダウンロード ← または cd ~/Downloads  
% wine hm950_x64_signed.exe  
画面の指示に従ってインストールする
```

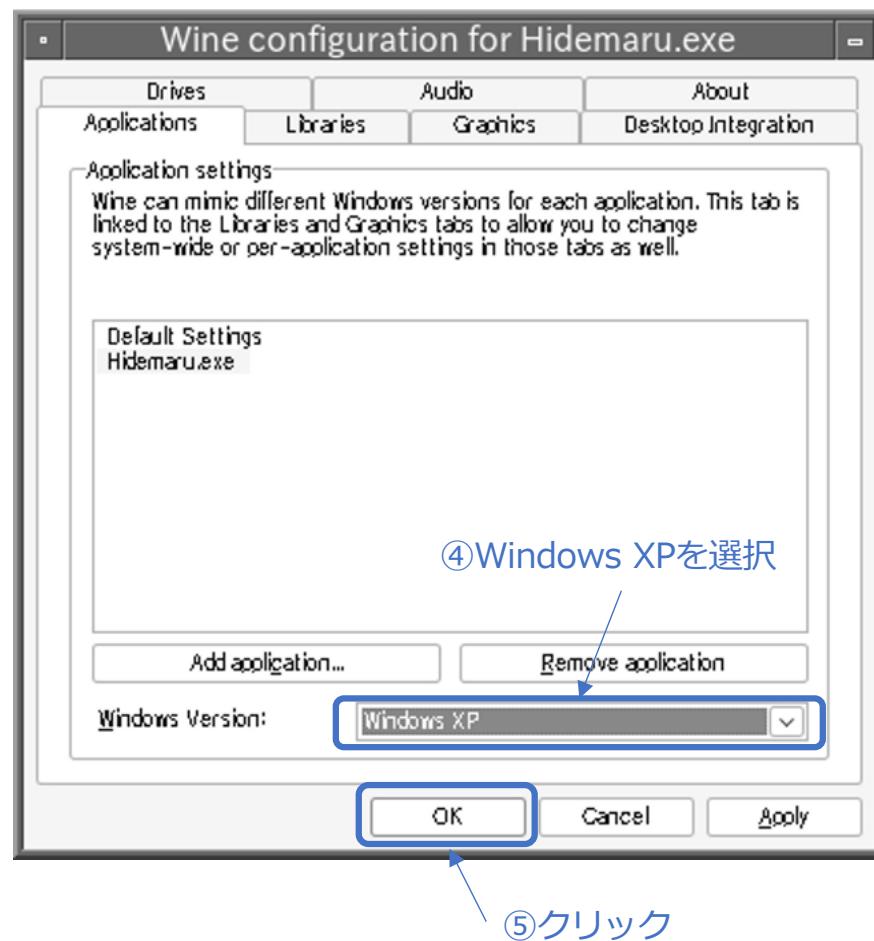
\*1 秀丸エディタを一定期間以上使用する場合は、ライセンス購入（有料）が必要です。  
起動時に「送金について」が表示されるようになると、その後の日本語入力ができなくなります。

## 続き | 10-3. 秀丸エディターを使用したい

以下は、ファイル読み書き時に動作異常を防止する設定である



## 続き | 10-3. 秀丸エディターを使用したい



## 続き | 10-3. 秀丸エディターを使用したい

秀丸エディターの起動と最低限の初期設定

```
% wine ~/.wine/drive_c/Program Files/Hidemaru/Hidemaru.exe
```

- ・その他(O) → 動作環境(E)... → 常駐機能 → 秀丸エディタの常駐(T) のチェックを外す
- ・その他(O) → ファイルタイプ別の設定(S)... → デザイン → IME変換中の色  
→ 入力モード(M)...ボタン → トゥルーインライン入力 にチェックする。 ← チェックはせずにクリックする。
- ・その他(O) → ファイルタイプ別の設定(S)... → 体裁 → ウィンドウ幅に合わせる(U)に、チェックする。
- ・その他(O) → ファイルタイプ別の設定(S)... → フォント → (適宜設定する)

以下、設定を推奨：

- ・その他(O) → 動作環境(E)... → 上級者向け設定(A)にチェック → ファイルエンコード1 → 新規作成やASCIIのとき：変更(D)... → エンコードの種類(C) : Unicode(UTF-8)、改行コード(R) : 改行=LF、BOMの有無：なし、にする。
- ・その他(O) → 動作環境(E)... → ウィンドウ → ステータスバー(R)の詳細(Z)... → カーソル位置の文字コードにチェック

## 続き | 10-3. 秀丸エディターを使用したい

秀丸の起動をカンタンにする設定

```
% vim ~/.cshrc
:
alias rm      'rm -i'

alias hidemaru 'wine ~/.wine/drive_c/Program Files/Hidemaru/Hidemaru.exe'
```

追加する

```
% source ~/.cshrc
% hidemaru
```

10-10. 「ランチャーにWindowsアプリを追加したい」を設定する

## 続き | 10-3. 秀丸エディターを使用したい

補足：Winecfgの設定項目について

少しわかりにくい winecfg の設定範囲をまとめました。

タブ名	設定範囲	内容
Applications	個別	アプリケーションごとに設定をするためのタブ。 Default Settingsは全アプリ共通。
Libraries	個別	Applicationsタブで指定したアプリごとに、ライブラリの設定を行う。
Graphics	個別	Applicationsタブで指定したアプリごとに、グラフィックの設定を行う。
Desktop Integration	共通	システム共通で、デスクトップの配色やフォルダーの設定を行う。
Drives	共通	システム共通で、ドライブの設定を行う。
Audio	共通	システム共通で、オーディオの設定を行う。

## 10-4. VB6のランタイムを必要とするソフトを起動させたい

- Windowsのフリーの画像レタッチソフト「JTrim」の例で紹介します。

<https://www.woodybells.com/jtrim.html> から JTrim Version 1.53cをダウンロードする

```
% cd ~/ダウンロード ← または ~/Downloads  
% mkdir -p ~/wine_bin/JTrim  
% unzip -O CP932 jt153c.zip -d ~/wine_bin/JTrim
```

- ・ランタイムのインストール

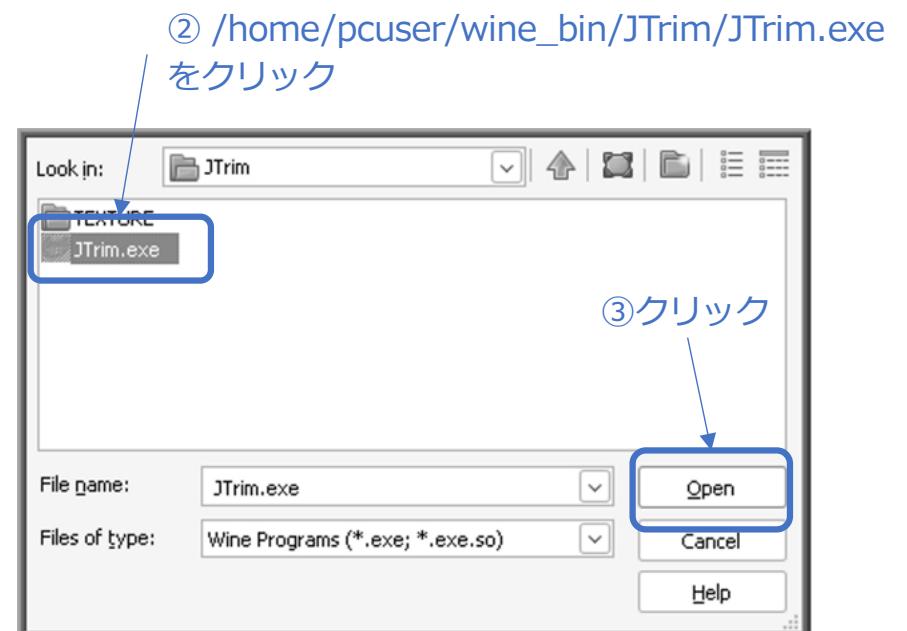
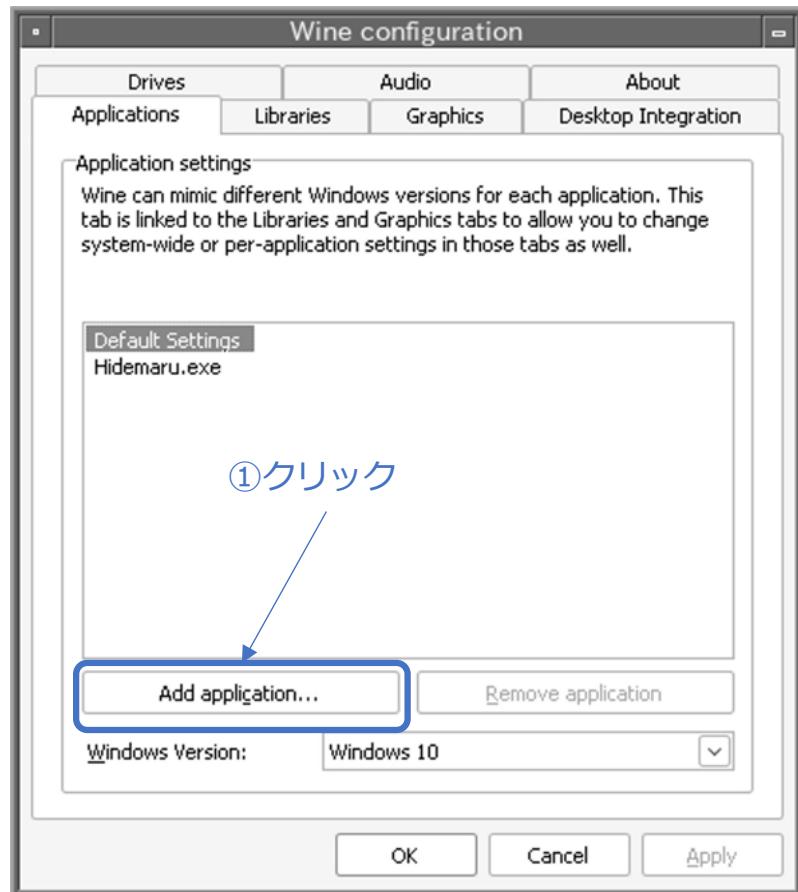
```
% winetricks vb6run
```

# 続き | 10-4. VB6のランタイムを必要とするソフトを起動させたい

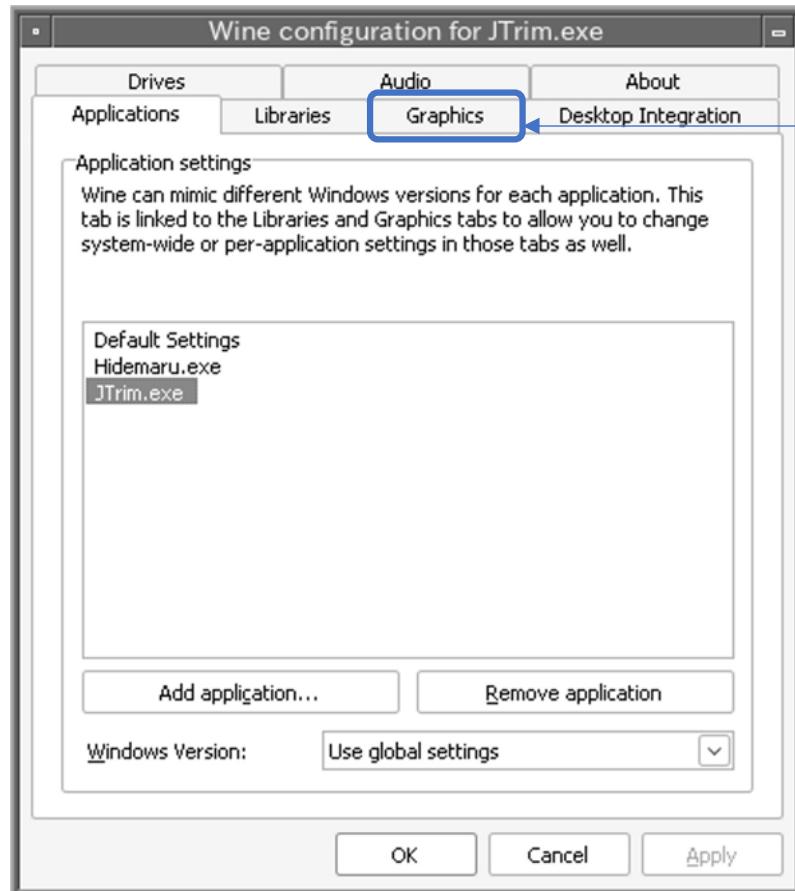
## ・ ウィンドウ周りの設定

※この設定は、Wine10.0とfwm3の組み合わせにおいて、VB6ランタイムを使用するWindowsアプリのウィンドウ周りがおかしくなる、などの対策上必要となります。

% winecfg

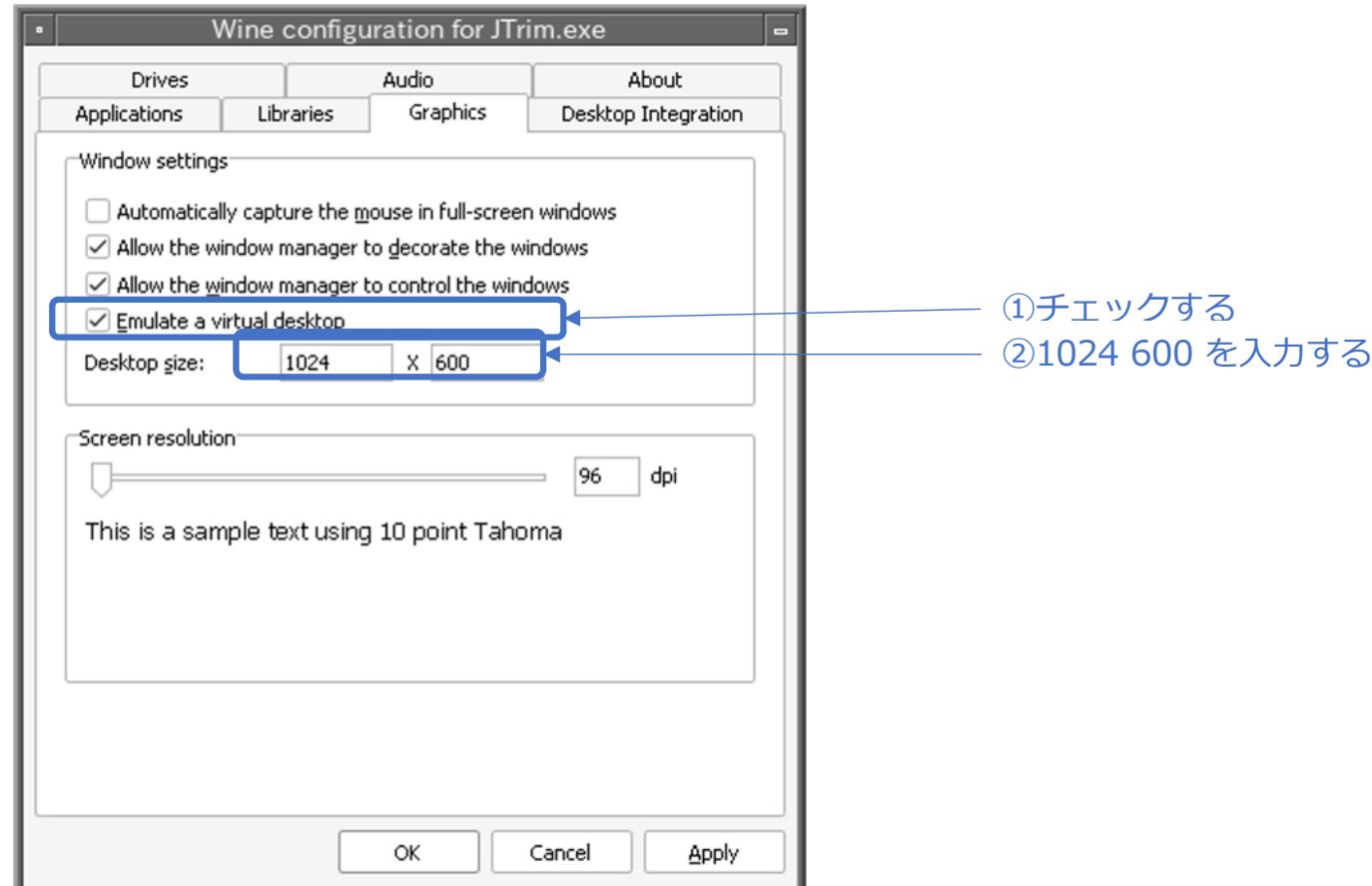


## 続き | 10-4. VB6のランタイムを必要とするソフトを起動させたい



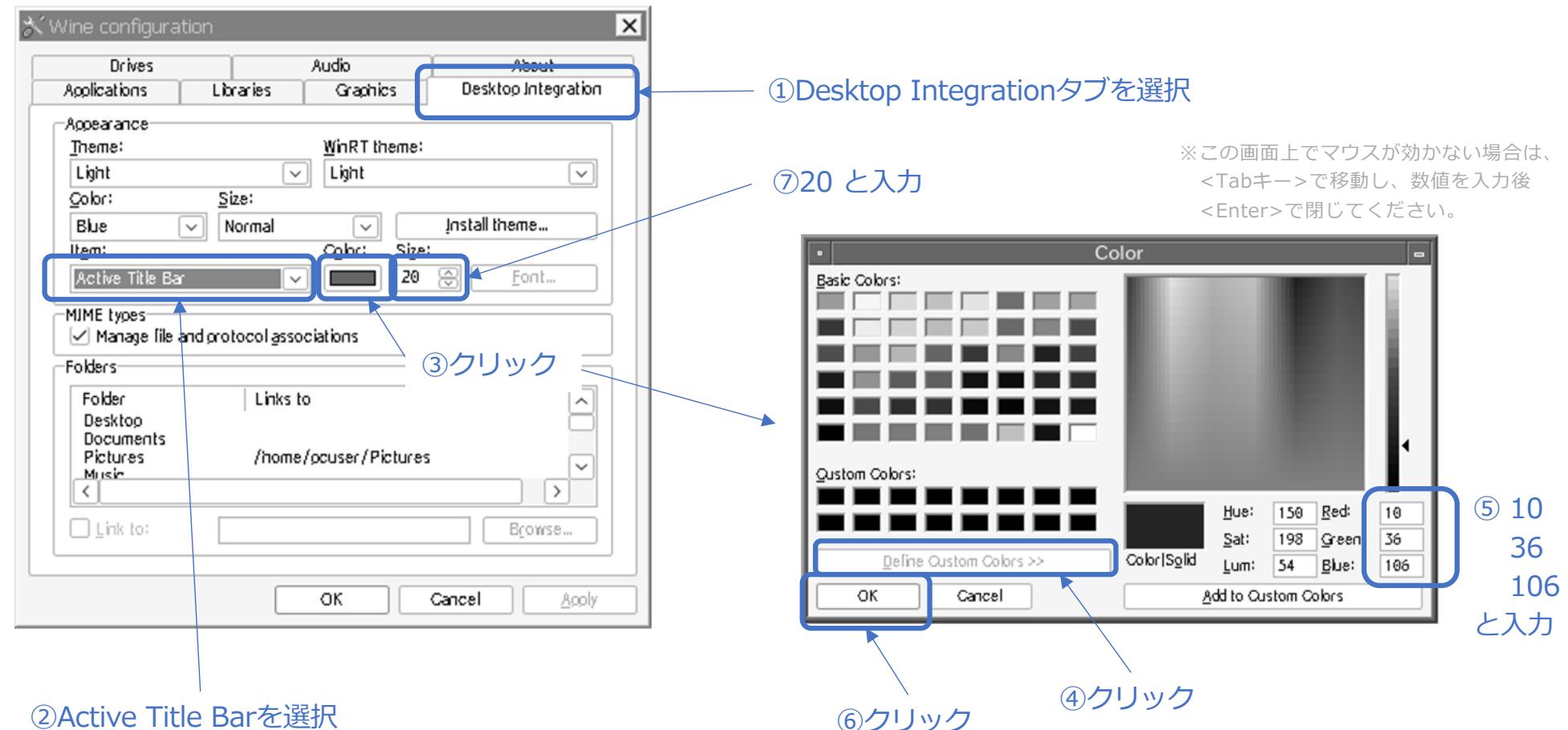
Graphicsタブをクリック

## 続き | 10-4. VB6のランタイムを必要とするソフトを起動させたい



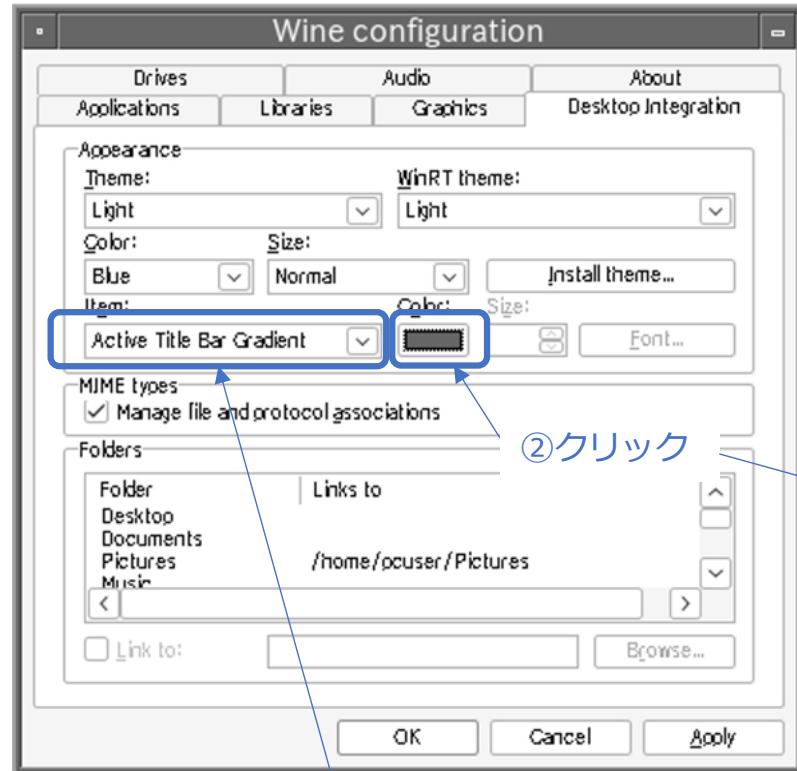
ここまで設定で基本的にはOKですが、タイトルバー周りがやや見にくいで次ページ以降で設定します。（次ページ以降は、一度設定すればOKです）。

# 続き | 10-4. VB6のランタイムを必要とするソフトを起動させたい

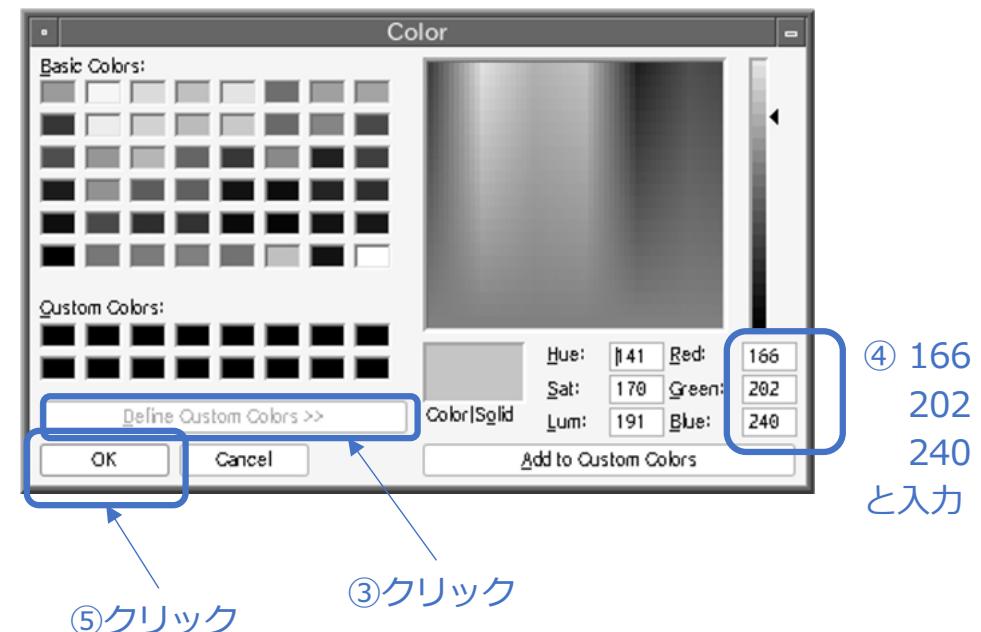


このページではタイトルバーの配色や右上のボタンの大きさを設定しています。

# 続き | 10-4. VB6のランタイムを必要とするソフトを起動させたい



① Active Title Bar Gradientを選択



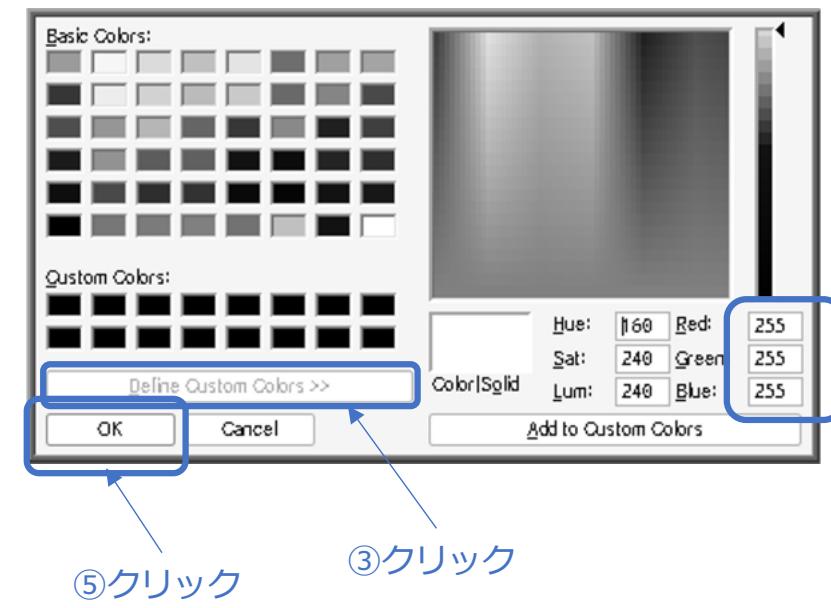
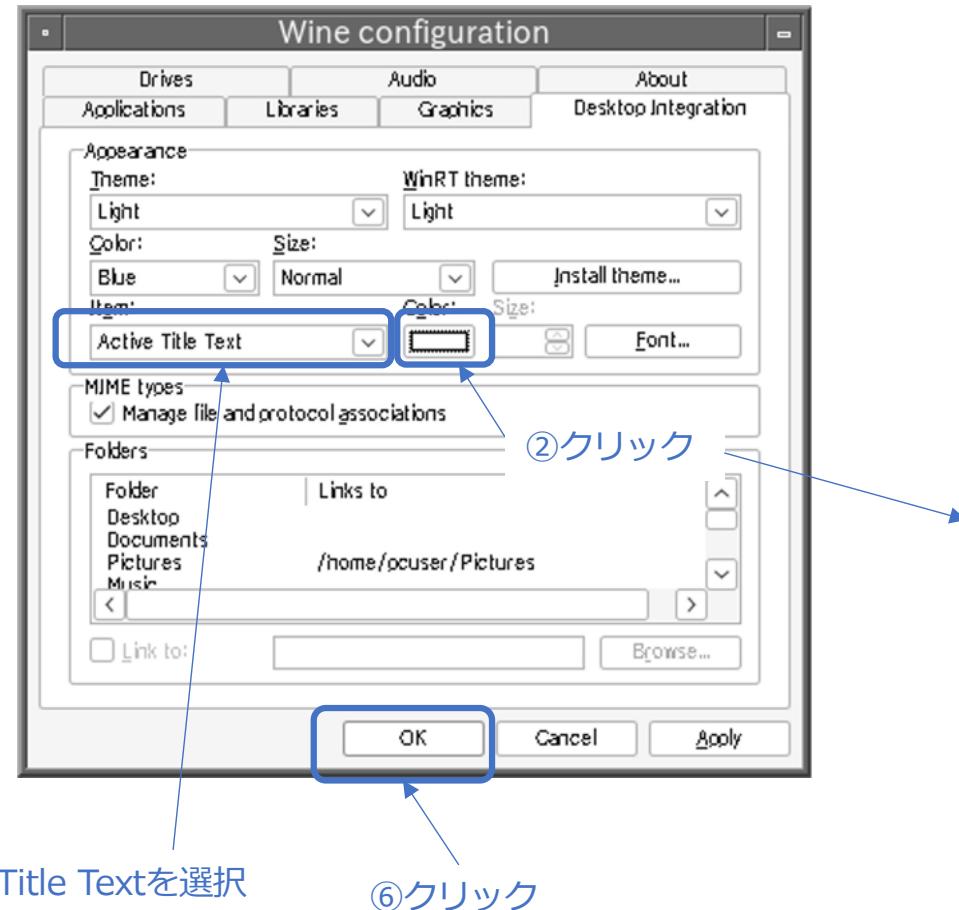
④ 166  
202  
240  
と入力

⑤ クリック

※この画面上でマウスが効かない場合は、  
<Tabキー>で移動し、数値を入力後  
<Enter>で閉じてください。

このページではタイトルバーの配色（グラデーション側）を設定しています。

# 続き | 10-4. VB6のランタイムを必要とするソフトを起動させたい



※この画面上でマウスが効かない場合は、  
<Tabキー>で移動し、数値を入力後  
<Enter>で閉じてください。

このページではタイトルバーの文字色を設定しています。

# 続き | 10-4. VB6のランタイムを必要とするソフトを起動させたい

- JTTrimの起動

```
⇒ cd ~/wine_bin/JTrim  
⇒ wine JTTrim.exe
```



※全画面表示になってしまふ場合は、505ページの Desktop Size を適当な値に変更してみてください。

## 10-5. Wineを完全消去したい

- Wineを完全消去したい (※インストーラー形式でインストールしたWindowsアプリも消去されます)

```
% sudo pkg delete -y wine wine-gecko wine-mono winetricks  
% sudo pkg autoremove  
% rm      ~/.config/menus/applications-merged/wine-*  
% rm -rf  ~/.i386-wine-pkg  
% rm      ~/.local/share/mime/application/x-wine*  
% rm      ~/.local/share/mime/packages/x-wine*  
% rm -rf  ~/.local/share/applications/wine  
% rm      ~/.local/share/applications/wine-*  
% rm      ~/.local/share/desktop-directories/wine-*  
% rm -rf  ~/.cache/winetricks  
% rm -rf  ~/.cache/wine  
% rm -rf  ~/.wine
```

さらにファイルが残っていないかどうかをチェックし、手作業で削除する。

```
% find ~/ -name *wine* -print
```

## 10-6. Wineで起動したWindowsアプリが暴走した場合の対処方法

```
% wineserver -k
```

## 10-7. 拡張子が .lzh のファイルを展開したい

インストール

```
% sudo pkg install -y lha
```

使い方

```
% lha x filename.lzh
```

# 10-8. 文字化けしているファイルやディレクトリを削除したい

- 文字化けしているファイルやディレクトリを削除したい（下記の例はディレクトリ）

確認方法

```
% ls -il  
2964981 drwxr-xr-x 2 pcuser wheel 512 10月 16 08:35 ???x???ó??????
```

削除方法

※数値を絶対に間違えないように！

```
% find . -inum 2964981 -exec rm -rf "{}" \$;
```

## 10-9. 日本語ファイル名を含むWindowsのzipファイルを展開したい

### 展開方法

```
% unzip -O cp932 filename.zip
```

大文字のO（オー）

## 10-10. ランチャーにWindowsアプリを追加したい

- ランチャー (FvwmButtons) にWindowsアプリを追加したい  
(下記の例は、秀丸エディターを追加する例)

① 秀丸エディター 64bit版 をインストールする (手順は10-3.を参照)

② .exeファイルからアイコンを抽出するユーティリティーのインストール

```
% sudo pkg install -y icoutils
```

③ .exeファイルからアイコンを抽出する

```
% cd ~/.wine/drive_c/Program Files/Hidemaru
% wrestool -x --output=/tmp -t14 Hidemaru.exe
```

ハイフン2つ

④ アイコンを確認する

```
% cd /tmp
% firefox *.ico
```

# 続き | 10-10. ランチャーにWindowsアプリを追加したい

- ⑤ .icoファイルから、.pngファイルに変換する Hidemaru.exe\_14\_102\_1041.ico が良さそうなので、これを.pngに変換する。

```
% magick Hidemaru.exe_14_102_1041.ico hidemaru.png
```

- ⑥ iconsフォルダーにコピーする hideamru-0~2.pngの3つのファイルが作成されるが、32x32ドットのファイルをアイコンファイルとして採用する。

```
% cp hidemaru-0.png ~/icons/hidemaru.png
```

- ⑦ .fvwm2rcに追加する

```
% vim ~/.fvwm2rc
```

\*FvwmButtonsButtonGeometry 48x48+539-0

\*FvwmButtons: (Icon "hidemaru.png", Action (Mouse 1) Func-LaunchOrRaise\_hidemaru, Action (Mouse 3) Menu hidemaru\_menu)

先頭の#wine#を削除する

変更する (画面の横解像度 - ランチャー数 × 48) ÷ 2

右クリックでメニュー表示

プログラムが起動していないければ起動、起動していれば  
フォーカスするように.fvwm2rcの先頭付近でファンク  
ションを定義している

- ⑧ ウィンドウマネージャの再起動

```
デスクトップ右クリック → ユーティリティ → ウィンドウマネージャの再起動...を選択
```

## 10-11. 拡張子が.cabのファイルを展開したい

- ・ユーティリティーのインストール

```
% sudo pkg install -y cabextract
```

- ・中身を確認する

```
% cabextract -l filename.cab
```

- ・特定のファイルを /tmp ディレクトリに展開する

```
% cabextract -d /tmp -F sample.txt filename.cab
```

## 10-12. Wine起動時のメッセージを抑制したい

- Wine起動時のメッセージを抑制したい

環境変数の設定

```
% vim ~/.xinitrc
:
export WINEDEBUG=-all ← この位置に追加する
xset r rate 250 40
:
```

ハイフン  
↓

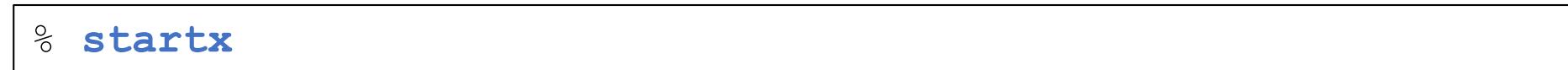
この位置に追加する

## 続き | 10-12. Wine起動時のメッセージを抑制したい

ウィンドウシステムをいったん終了する



ウィンドウシステムを起動する



## 10-13. Windowsアプリから、FreeBSD上のプログラムを起動したい

例：

```
cmd /c start /unix /home/pcuser/bin/program.tcsh
```

## 10-14. Wineで使えるいい感じの等幅フォントを生成したい

等幅フォントの諸問題を解決するために、IPAゴシックを基として新たにフォントを作成する方法を説明します。

- フォント関連ライブラリのインストール

```
% sudo pkg install -y fontforge py311-fonttools
```

- フォント作成手順 (スクリプトをダウンロードして実行する)

```
% mkdir ~/bin ← すでに存在しているかもしれません
% cd ~/bin

% fetch ¥
https://raw.githubusercontent.com/moginwc/freebsd143/refs/heads/main/bin/create_font.tcsh
% fetch ¥
https://raw.githubusercontent.com/moginwc/freebsd143/refs/heads/main/bin/create_font.py
% fetch ¥
https://raw.githubusercontent.com/moginwc/freebsd143/refs/heads/main/bin/create_font_unicode.txt
% fetch ¥
https://raw.githubusercontent.com/moginwc/freebsd143/refs/heads/main/bin/create_font_005c.glyph
% fetch ¥
https://raw.githubusercontent.com/moginwc/freebsd143/refs/heads/main/bin/create_font_005f.glyph

% tcsh ./create_font.tcsh
```

これで、『My932ゴシック (My932Gothic)』フォントが使えるようになります (WineでもFreeBSDでもどちらでも使えます) あまり詳細に検証していないので、不具合があるかもしれません。

# 続き | 10-14. Wineで使えるいい感じの等幅フォントを生成したい

## □ フォント生成の解説

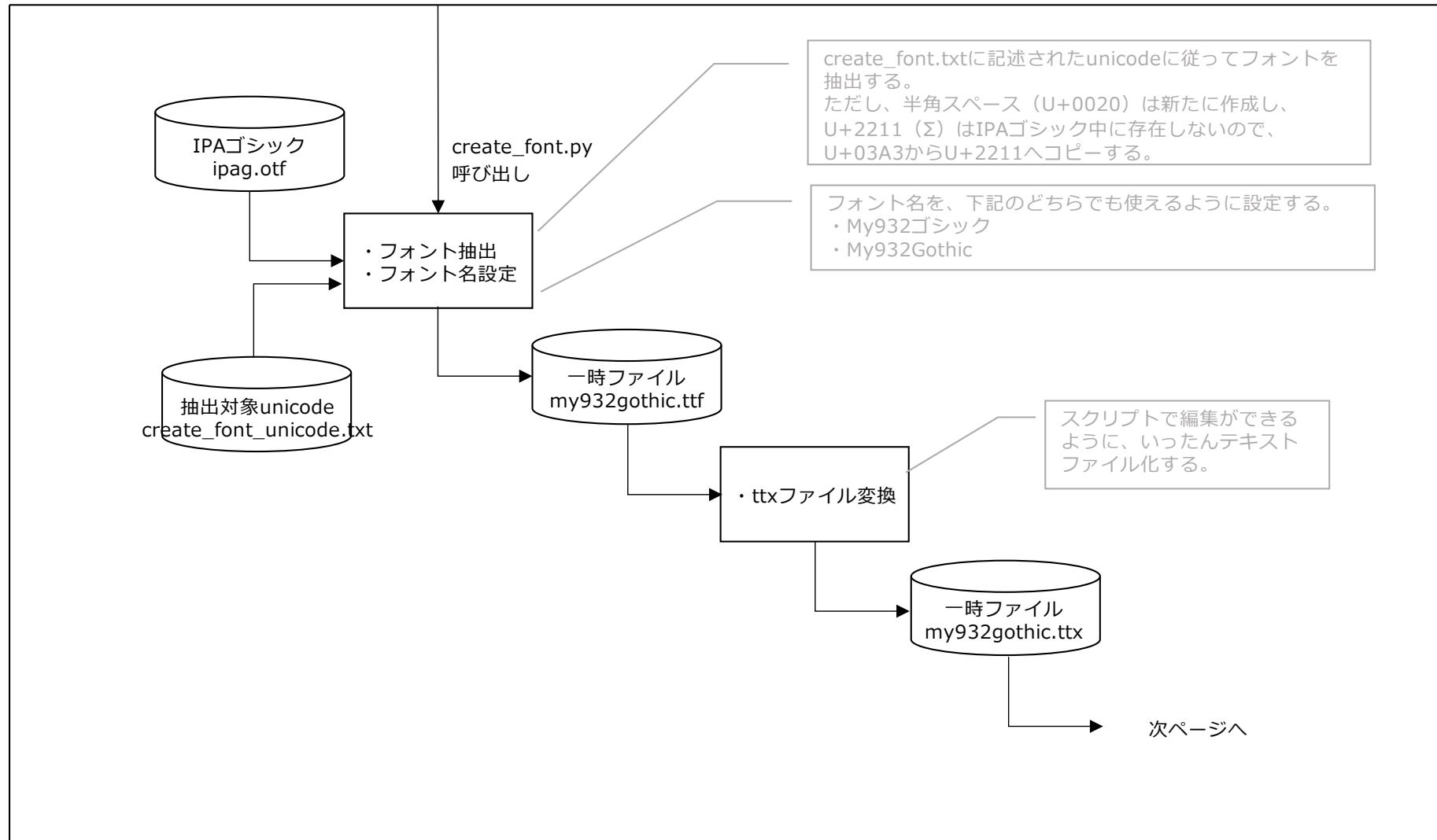
仕様：

項目	内容
目的	Wine上で動作する秀丸エディタ、WinMerge、Bz（以上、Windowsアプリ）において、等幅フォントが問題なく、キレイに表示・使用できること。
基となるフォント	IPAゴシック Ver.003.03 (/usr/local/share/fonts/ipa/ipag.otf)
対象文字	<ul style="list-style-type: none"><li>CP932 1バイト部に対応したunicode 158文字</li><li>半角英数字記号 0x20-0x7e 95文字</li><li>半角カナ 0xA1-0xDF 63文字</li><li>CP932 2バイト部に対応したunicode 7326文字</li><li>CP932とSJISでunicodeコードポイントが異なる 7(6)文字</li></ul> <p>①フォントを軽量化したい、②滅多に使わない文字を除外したい、などの理由。</p>
フォント名	My932ゴシック My932Gothic MとGは大文字
フォントファイル名・形式	my932gothic.ttf 全て小文字
フォント形状	文字幅、行間、半角と全角の文字幅割合は、MSゴシック互換
アンチエイリアス	すべてのサイズで有効であること。
入れ替えフォント	アンダースコア0x5fについては、半角ハイフン並みに厚みを持たる。 バックスラッシュ0x5cについては、スラッシュを反転させる。 (上記はIPAゴシックを参考にした)

# 続き | 10-14. Wineで使えるいい感じの等幅フォントを生成したい

変換プログラム入出力仕様：

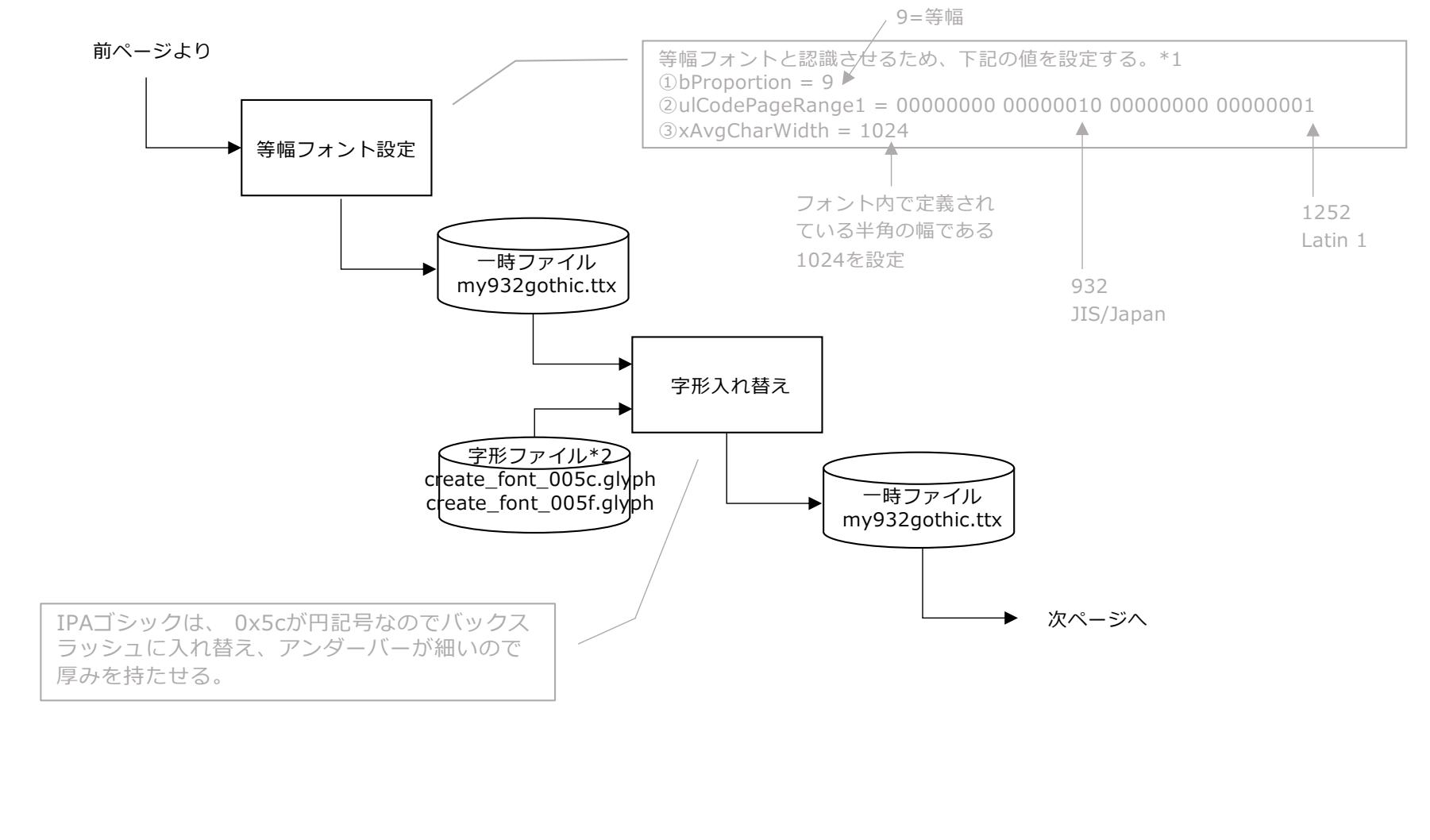
create\_font.tcsh



# 続き | 10-14. Wineで使えるいい感じの等幅フォントを生成したい

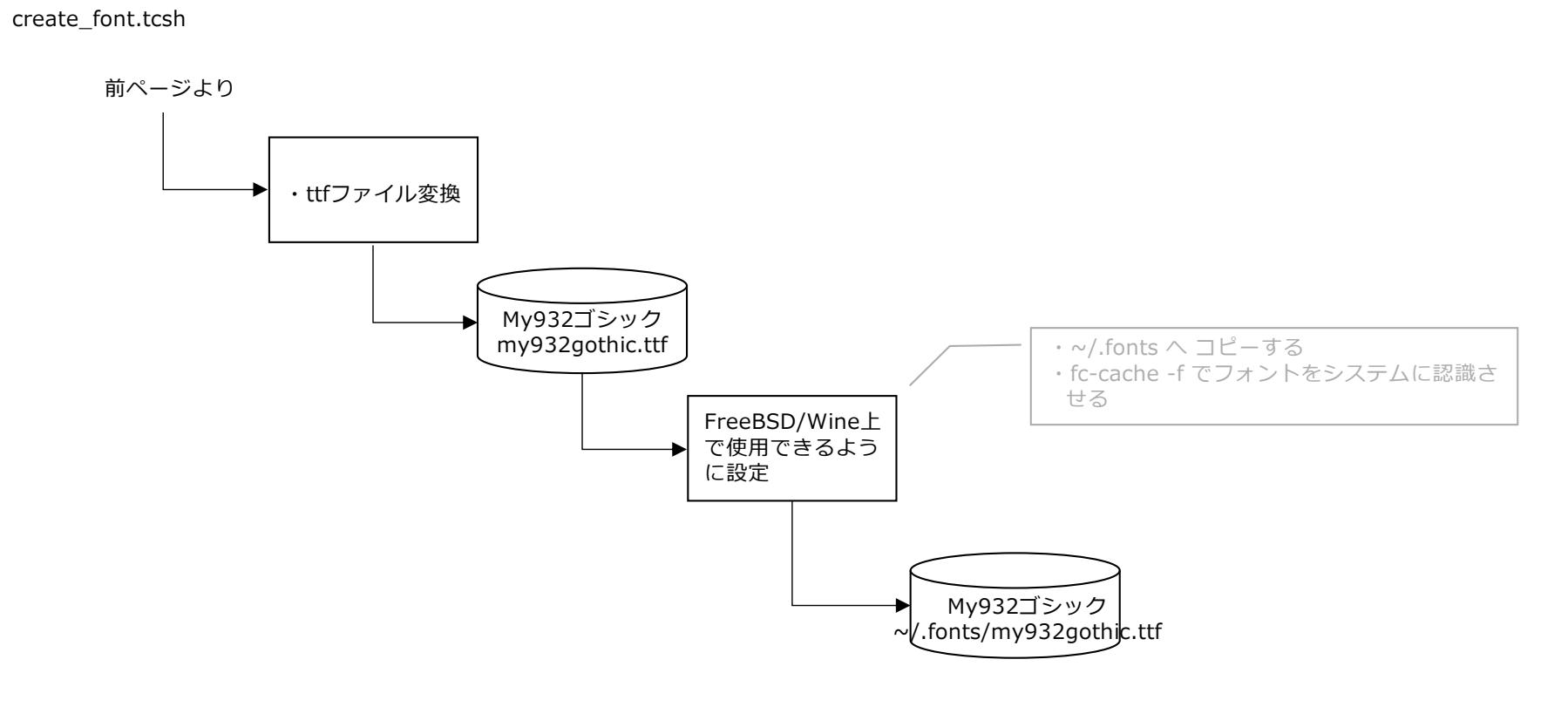
変換プログラム入出力仕様（続き）：

create\_font.tcsh



# 続き | 10-14. Wineで使えるいい感じの等幅フォントを生成したい

変換プログラム入出力仕様（続き）：



# 続き | 10-14. Wineで使えるいい感じの等幅フォントを生成したい

変換プログラム入出力仕様（補足）：

【補足】

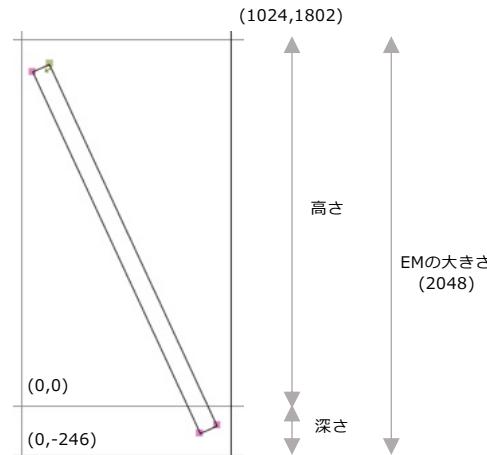
本スクリプトでは、単にフォントをエクスポート→再作成したのみで、文字幅・行間調整などは一切行なっていませんが、結果的にMSゴシックと似た構造（文字幅・行間のこと。書体ではない）となりました。

前々ページの \*1

- ①の設定を行わないと、秀丸エディタではプロポーショナルフォント扱いになる（表示は正常）。
- ②の設定を行わないと、Bzでフォントそのものが選択できない。
- ③の設定を行わないと、Bzでバイナリ表示部分と文字表示部分のカーソル位置がズレてしまう。

他にも等幅設定はたくさんある模様（ネット情報参照）だが、ここでは目的に沿って、必要最小限の設定とした。

前々ページの \*2 【参考】半角バックスラッシュの字形（0x5c）と座標の関係



## 10-15. Thunarの右クリックで、Binary Editor Bzを起動したい

ここでは Thunar (8-23.で紹介のファイル管理ソフト)において、ファイルを右クリックした際に「送る」メニューの中に、Windows用バイナリーエディターの Binary Editor Bz を追加する方法について説明します。

- ① ウェブブラウザにて、Bz1987Portable.zip (ポータブル版) をダウンロードする  
( <https://gitlab.com/devill.tamachan/binaryeditorbz/-/releases/1.9.8.7> )

- ② zipファイルの展開と、フォルダーの移動

```
% cd ~/ダウンロード ← または cd ~/Downloads
% unzip Bz1987Portable.zip
% mv Bz1987Portable Bz
% mkdir ~/wine_bin ← すでに10-1.で作成済みかと思います
% mv Bz ~/wine_bin
```

## 続き | 10-15. Thunarの右クリックで、Binary Editor Bzを起動したい

### ③ sendtoディレクトリの作成

```
% mkdir -p ~/.local/share/Thunar/sendto ← すでに存在するかもしれません。
```

### ④ 送るメニューの作成

```
% vim ~/.local/share/Thunar/sendto/bz.desktop
```

[Desktop Entry]

Type=Application

Name=Binary Editor Bzで開く

Exec=wine /home/pcuser/wine\_bin/Bz/Bz64.exe %f

新規に追加する

← Bzの起動コマンドを記述する

- 右クリックに「Binary Editor Bz で開く」が追加され、バイナリーエディタが開けるようになります。

# 11-1. mozcのインストールと設定

## ■ mozcのインストールと設定

※uim-anthyの設定が行われていることを前提としています。

※下記の不具合があります (uim-mozc 2.23.2815.102.01\_7)

1. mozc\_toolが起動しないので、設定や辞書登録などができない。

```
% mozc_tool
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
E0000 00:00:1729998065.028032 100834 descriptor_database.cc:633] File already exists in database:
ipc/ipc.proto
F0000 00:00:1729998065.029359 100834 descriptor.cc:2236] Check failed: GeneratedDatabase()->Add(encoded_file_descriptor, size)
*** Check failure stack trace: ***
アボート(coreを出力しました)
```

⇒ 次ページ以降、一時的に回避する方法を掲載しています。

2. Qt系アプリ\*1でアプリ終了時に「セグメントエラー」が発生する。

⇒ 対応方法不明。

\*1 LibreOffice や OpenSCADなど。Qt系のアプリと、uim-mozcの相性が良くない感じです。

# 続き | 11-1. mozcのインストールと設定

mozcのインストールと初期設定

## ① インストール

```
% sudo pkg install -y ja-uim-mozc
```

## ② ~/.xinitrc に以下の内容を追加する

```
% vim ~/.xinitrc
```

```
export LC_ALL=ja_JP.UTF-8
```

```
export GTK_IM_MODULE=uim
```

```
export QT_IM_MODULE=uim
```

```
export XMODIFIERS=@im=uim
```

```
export XIM=uim
```

```
/usr/local/bin/mozc start ← この位置に
```

```
uim-xim &
```

```
uim-toolbar-gtk3 -1-1 &
```

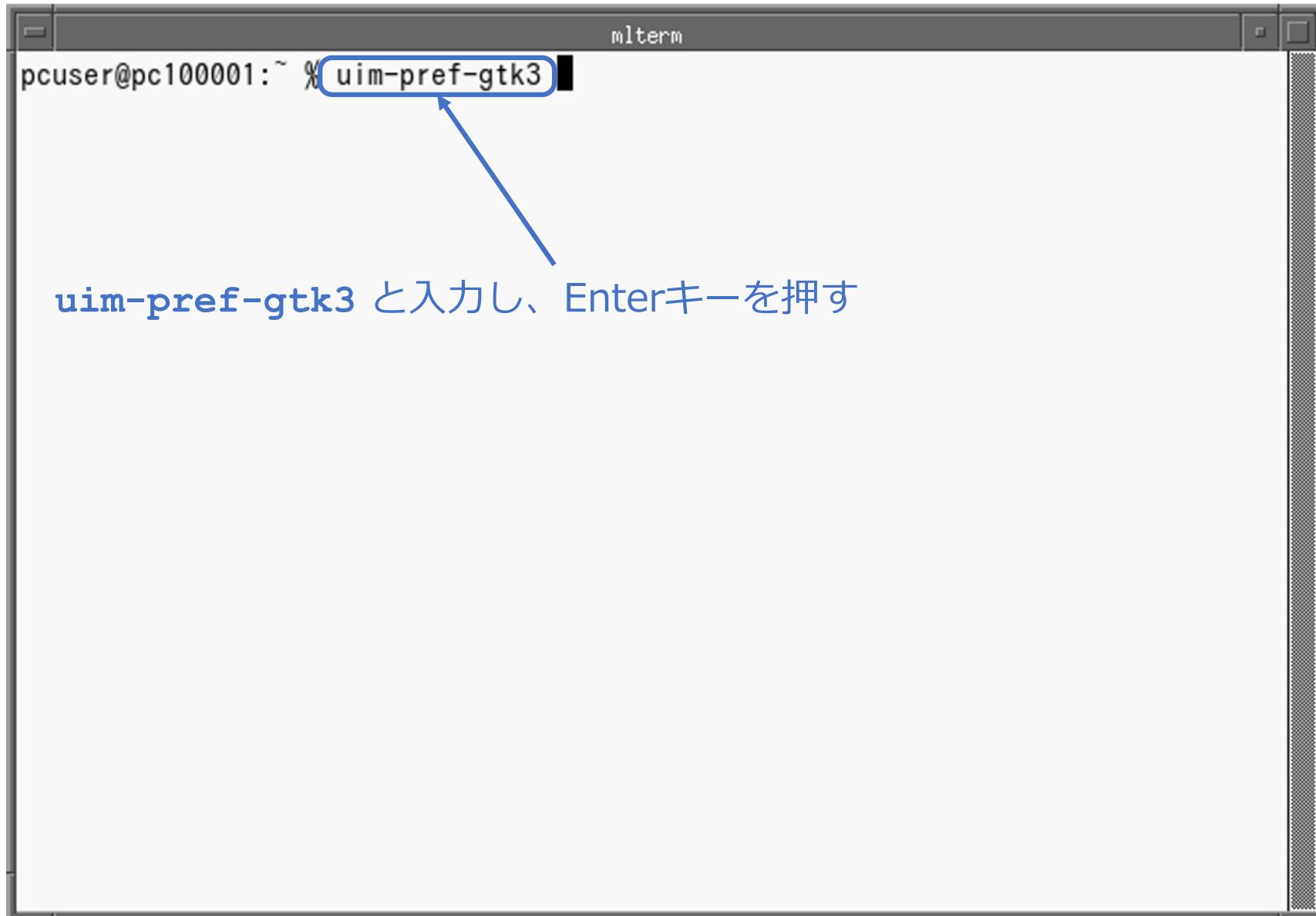
```
xset r rate 250 40
```

この位置に  
追加する

その他は  
uim-anthy  
と同じ設定

## 続き | 11-1. mozcのインストールと設定

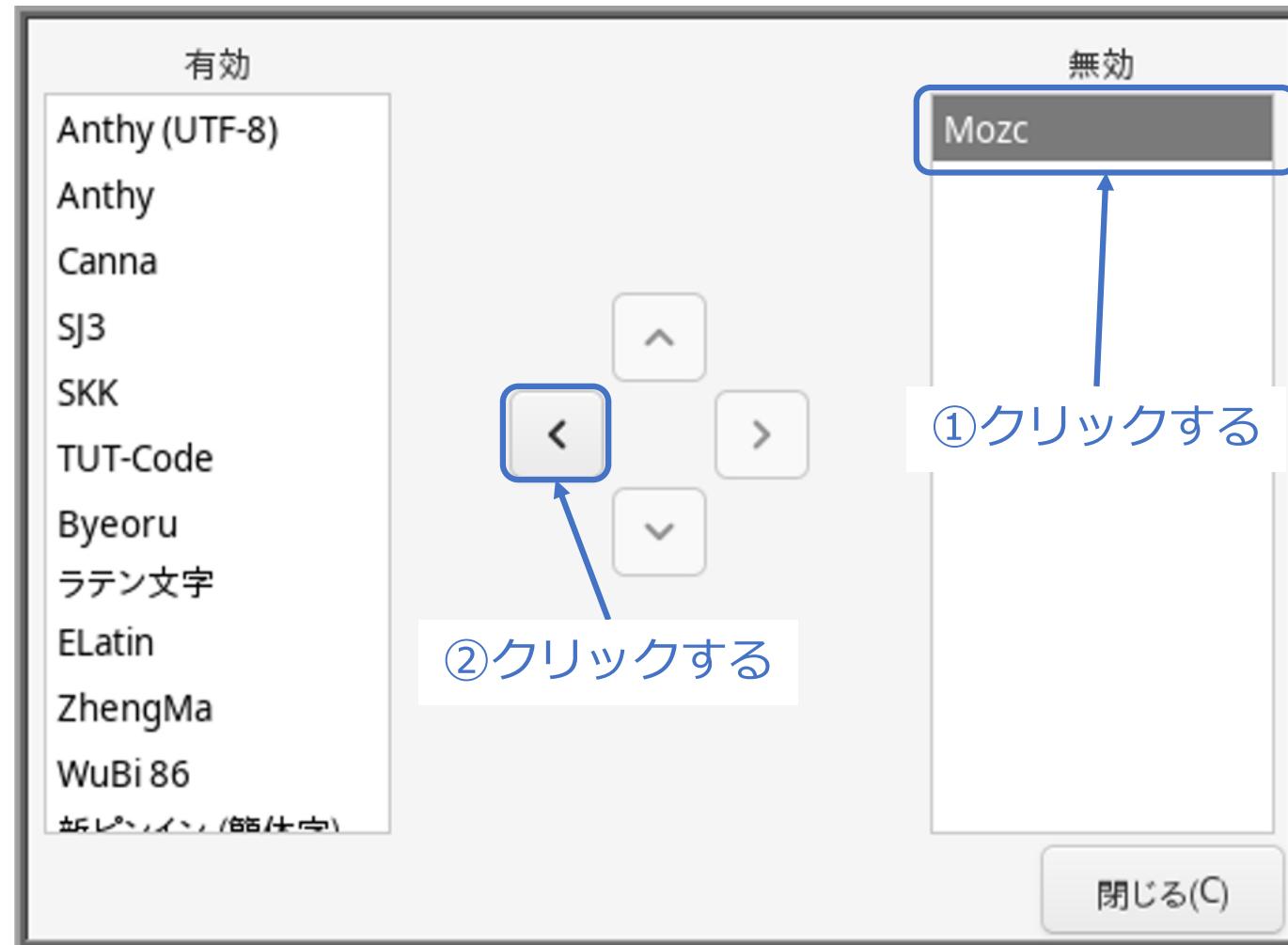
### ③ uimの初期設定



# 続き | 11-1. mozcのインストールと設定



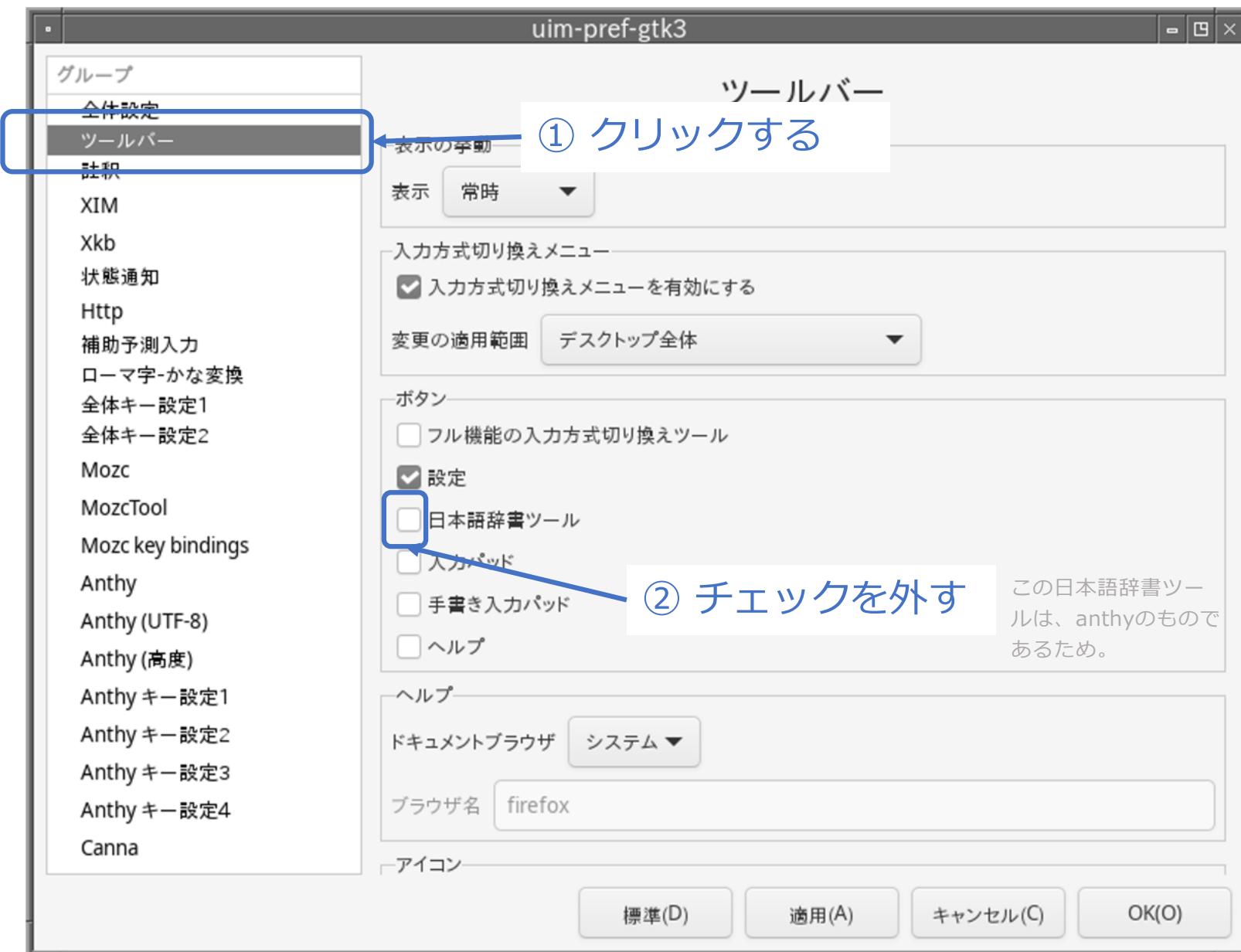
## 続き | 11-1. mozcのインストールと設定



## 続き | 11-1. mozcのインストールと設定



# 続き | 11-1. mozcのインストールと設定



# 続き | 11-1. mozcのインストールと設定



## 続き | 11-1. mozcのインストールと設定

ウィンドウシステムをいったん終了する

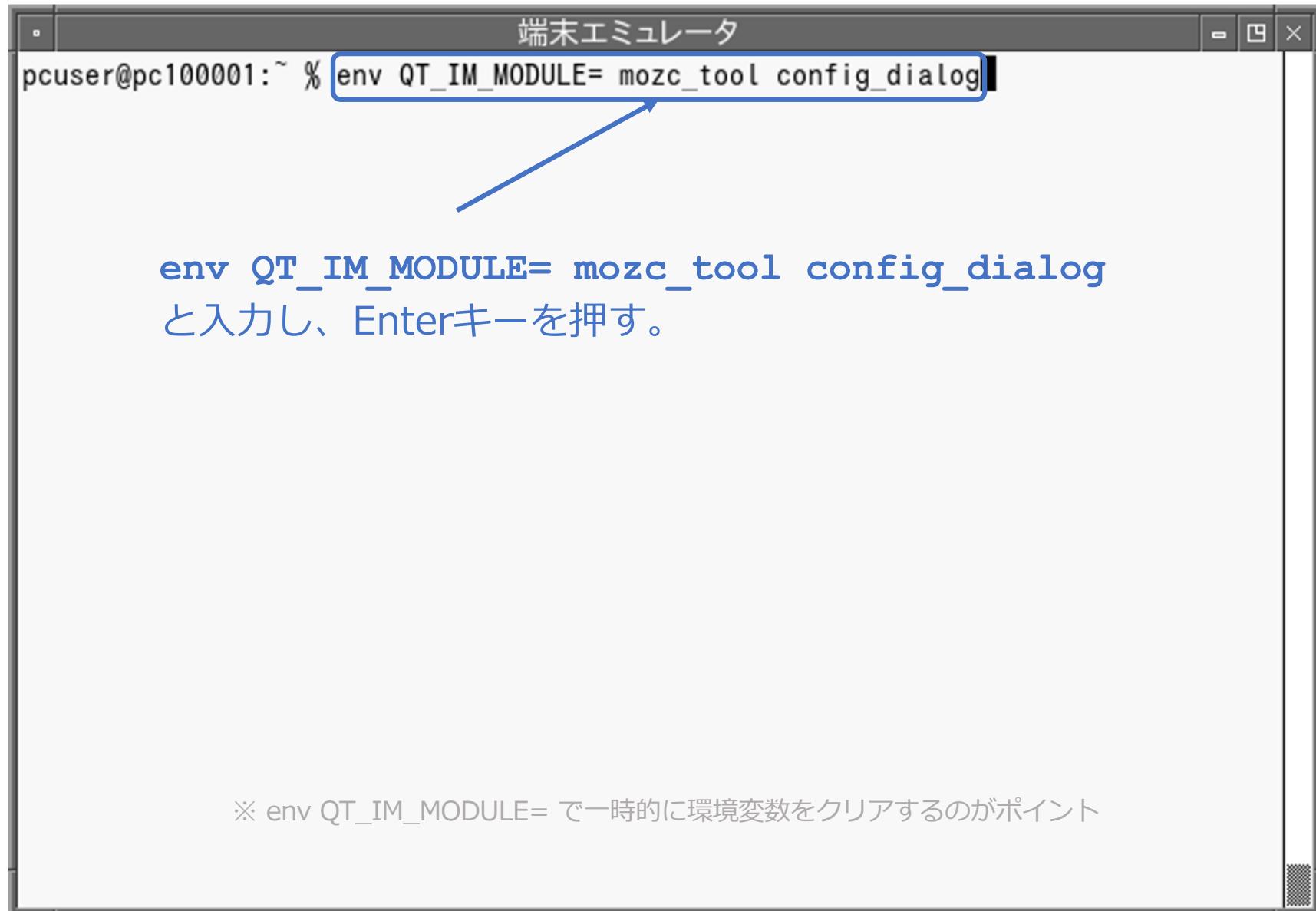


ウィンドウシステムを起動する

```
% startx
```

## 続き | 11-1. mozcのインストールと設定

### ④ mozcの初期設定

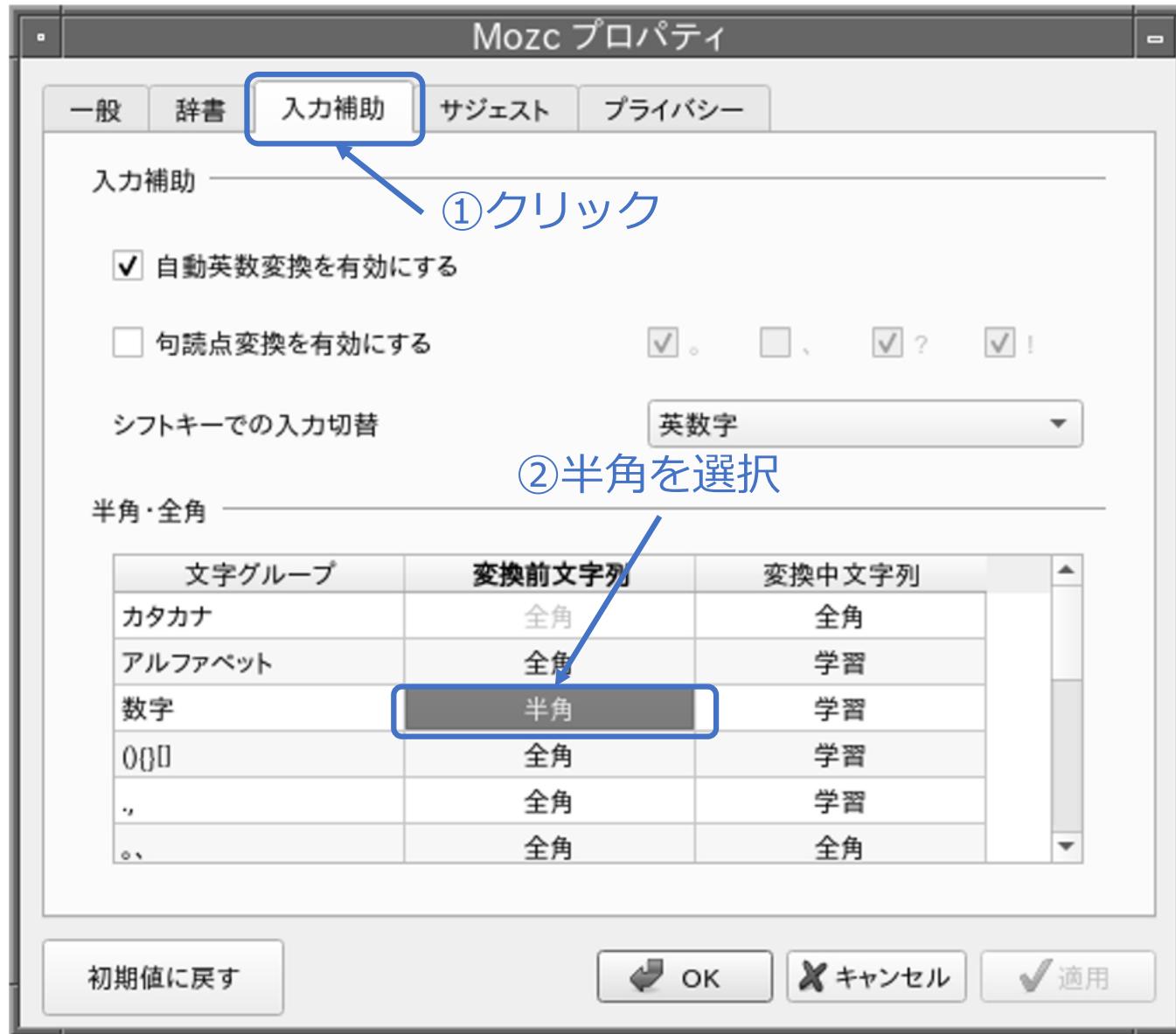


```
pcuser@pc100001:~ % env QT_IM_MODULE= mozc_tool config_dialog
```

env QT\_IM\_MODULE= mozc\_tool config\_dialog  
と入力し、Enterキーを押す。

※ env QT\_IM\_MODULE= で一時的に環境変数をクリアするのがポイント

## 続き | 11-1. mozcのインストールと設定



このページはお好みに  
よって変更してください。

## 続き | 11-1. mozcのインストールと設定



## 11-2. Macのユーザー辞書をMozcにインポートしたい

- Macのユーザー辞書をMozcにインポートしたい

Mac側の作業（概要） ※詳細な手順は一部省いて説明しています。

1. 画面右上の入力メニュー「A」→ ユーザ辞書を編集... をクリックする。
2. 「ユーザ辞書」タブをクリックする（表示されない場合は次へすすむ）。
3. command+A キーを押す。
4. 選択された範囲を、デスクトップにドラッグ&ドロップする。
5. ユーザ辞書.plist が作成されるのを確認する。（中身はXML形式のファイル）
6. ユーザ辞書.plist をFreeBSDにコピーする。

ファイル共有でコピーすることを推奨  
(scpコマンドでは、「ザ」が「サ」扱いとなる)

## 続き | 11-2. Macのユーザー辞書をMozcにインポートしたい

FreeBSD側の作業（ユーザ辞書.plistをmozcのユーザー辞書ファイル形式に変換する）

変換スクリプトの作成

```
% vim conv_dic_mozc.py
```

※本スクリプトは生成AIにて作成後、若干手直しした。

```
import plistlib

# ユーザー辞書を読み込む
with open('ユーザ辞書.plist', 'rb') as f:
    plist_data = plistlib.load(f)

# よみがなと単語を抽出し、標準出力に表示する
for item in plist_data:
    reading = item.get("shortcut", "")
    word = item.get("phrase", "")
    # 整形して出力する
    print(f'{reading} {word} 名詞')
```

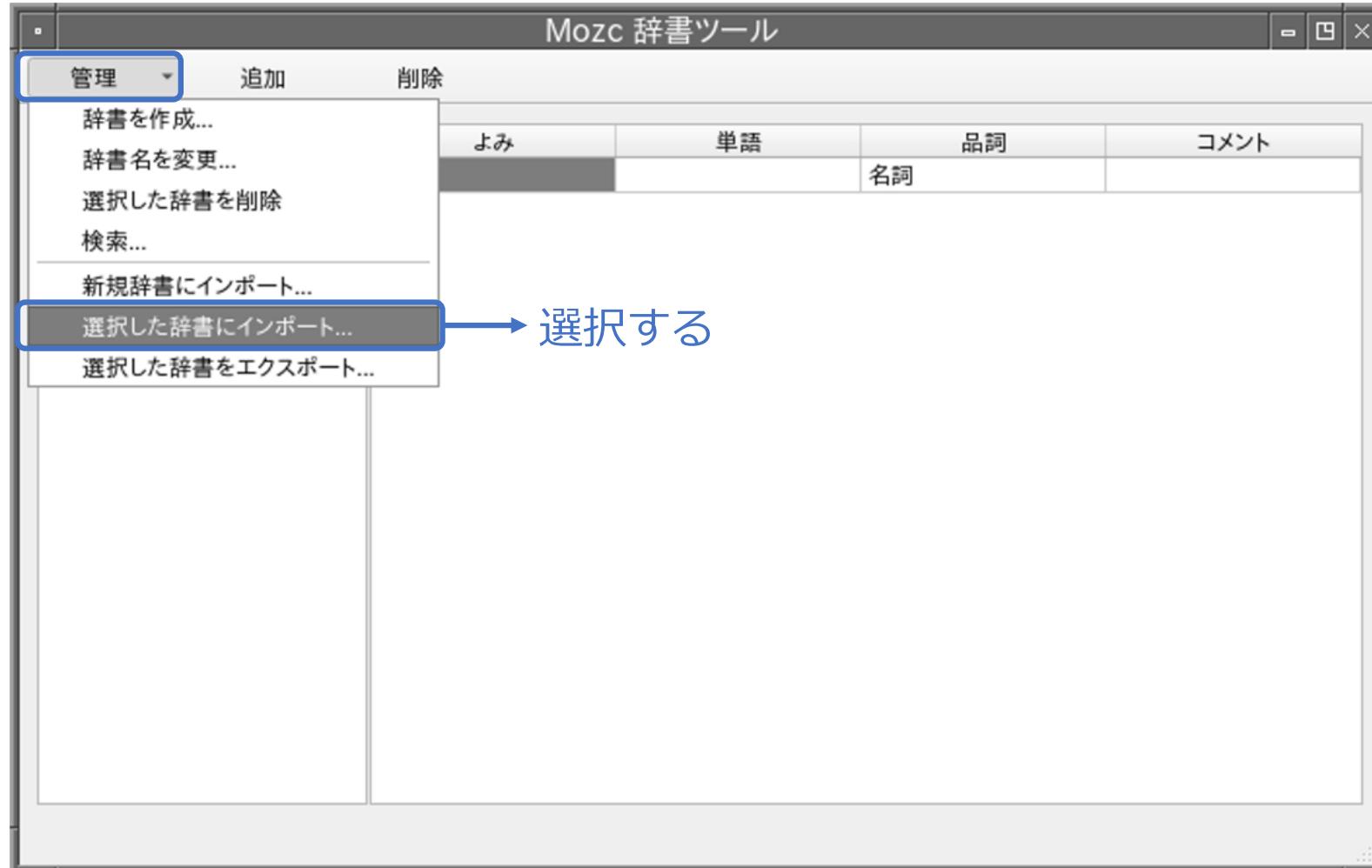
変換スクリプトの実行

```
% python3.11 conv_dic_mozc.py > mozcdic.txt
```

## 続き | 11-2. Macのユーザー辞書をMozcにインポートしたい

辞書ツールを立ち上げる

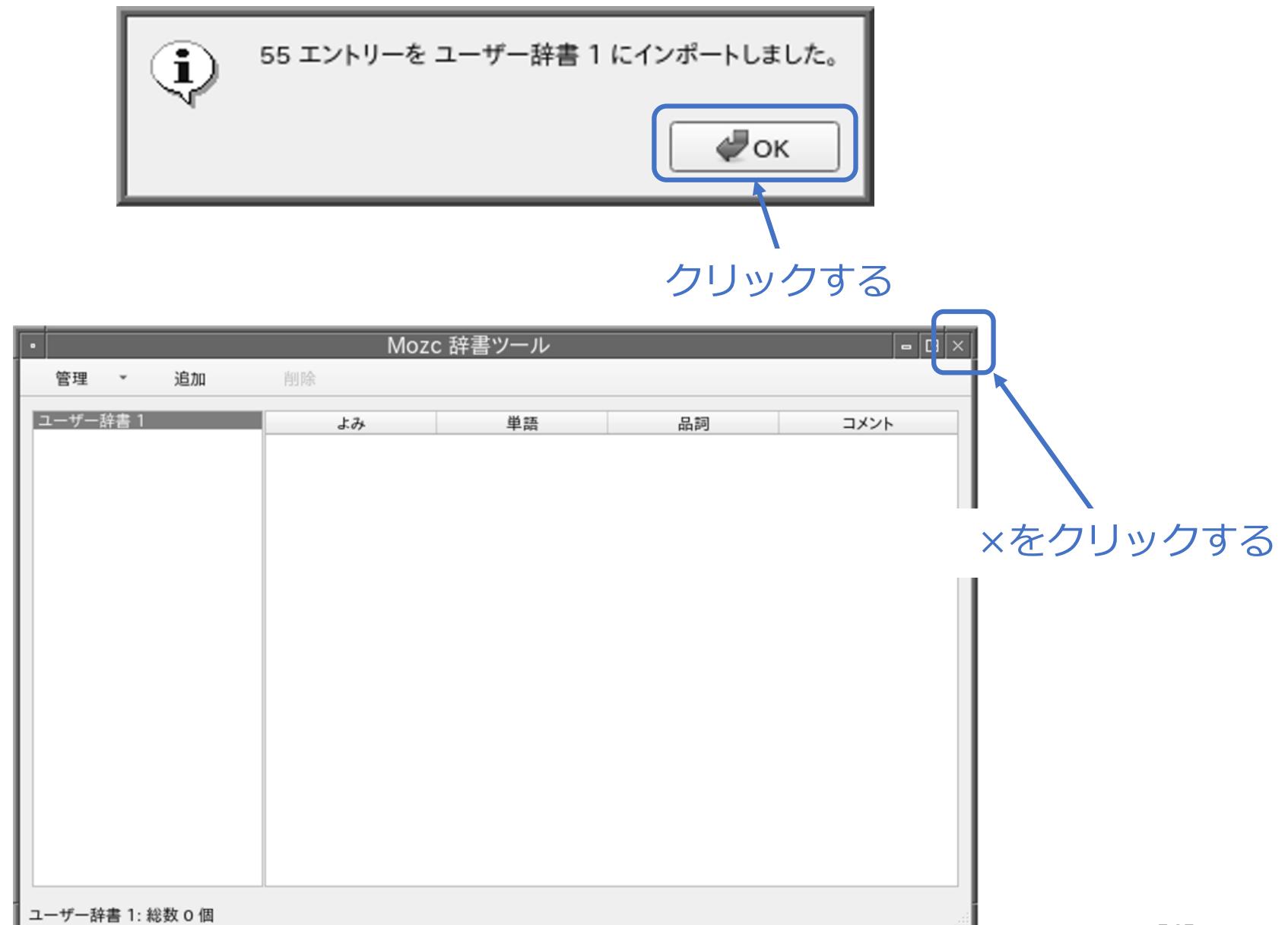
```
% env QT_IM_MODULE= mozc_tool dictionary_tool
```



## 続き | 11-2. Macのユーザー辞書をMozcにインポートしたい



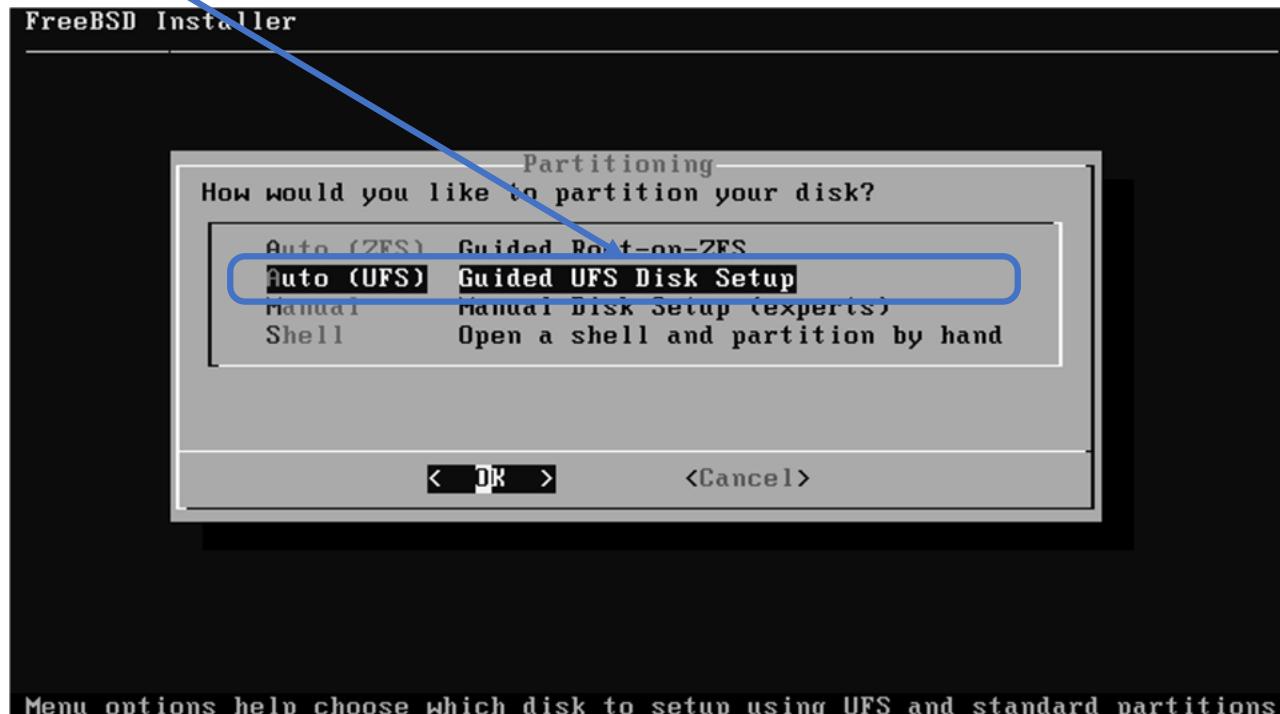
## 続き | 11-2. Macのユーザー辞書をMozcにインポートしたい



## 12-1. UFSファイルシステムを使用したい

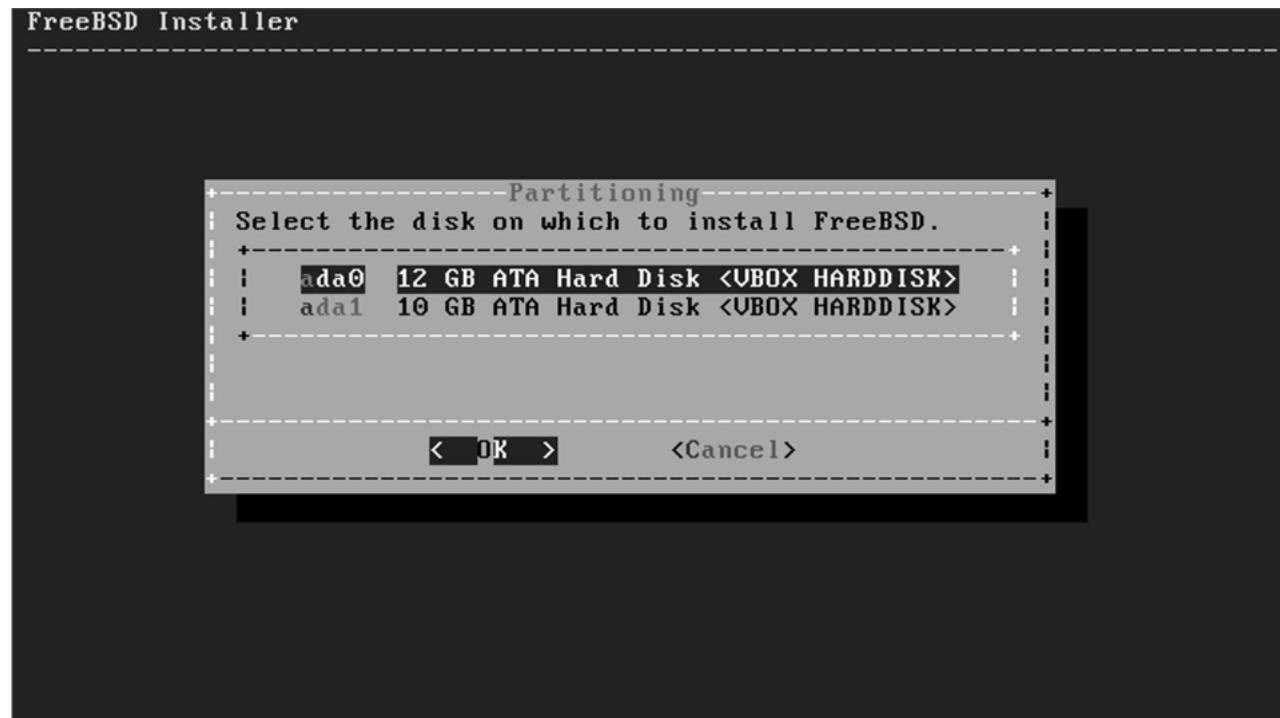
UFSファイルシステムを使用した場合は、当インストール&活用ガイドの47～54ページを、546～552ページに読み替えてください。

- ① Auto (UFS) を選択し、Enterキーを押す。



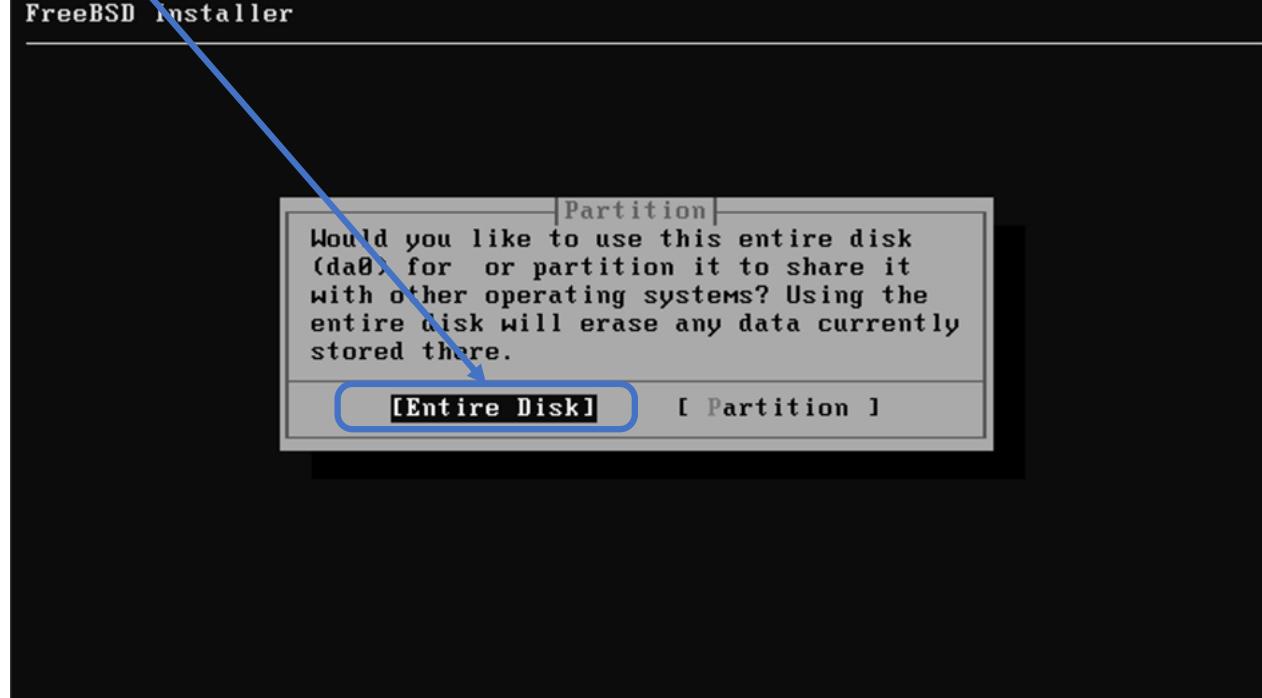
## 続き | 12-1. UFSファイルシステムを使用したい

- ① 内蔵ストレージが複数あるなど、もし、このような画面が表示された場合は、↑↓キーでインストール先ディスク選択し、Enterキーを押す。



## 続き | 12-1. UFSファイルシステムを使用したい

① Entire Diskを選択し、Enterキーを押す。



# 続き | 12-1. UFSファイルシステムを使用したい

① もし、この確認画面が表示された場合は、Enterキーを押す。



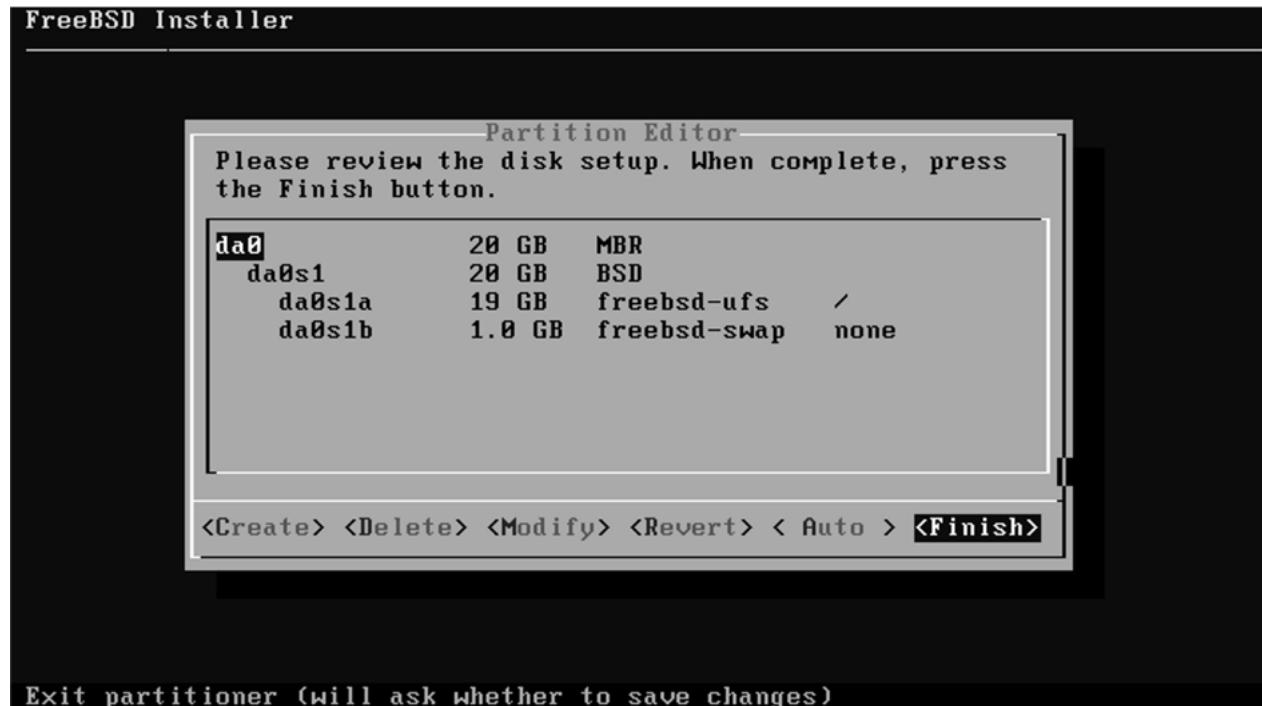
# 続き | 12-1. UFSファイルシステムを使用したい

① GPTまたはMBRのどちらかが選択されていることを確認し、そのままEnterキーを押す。



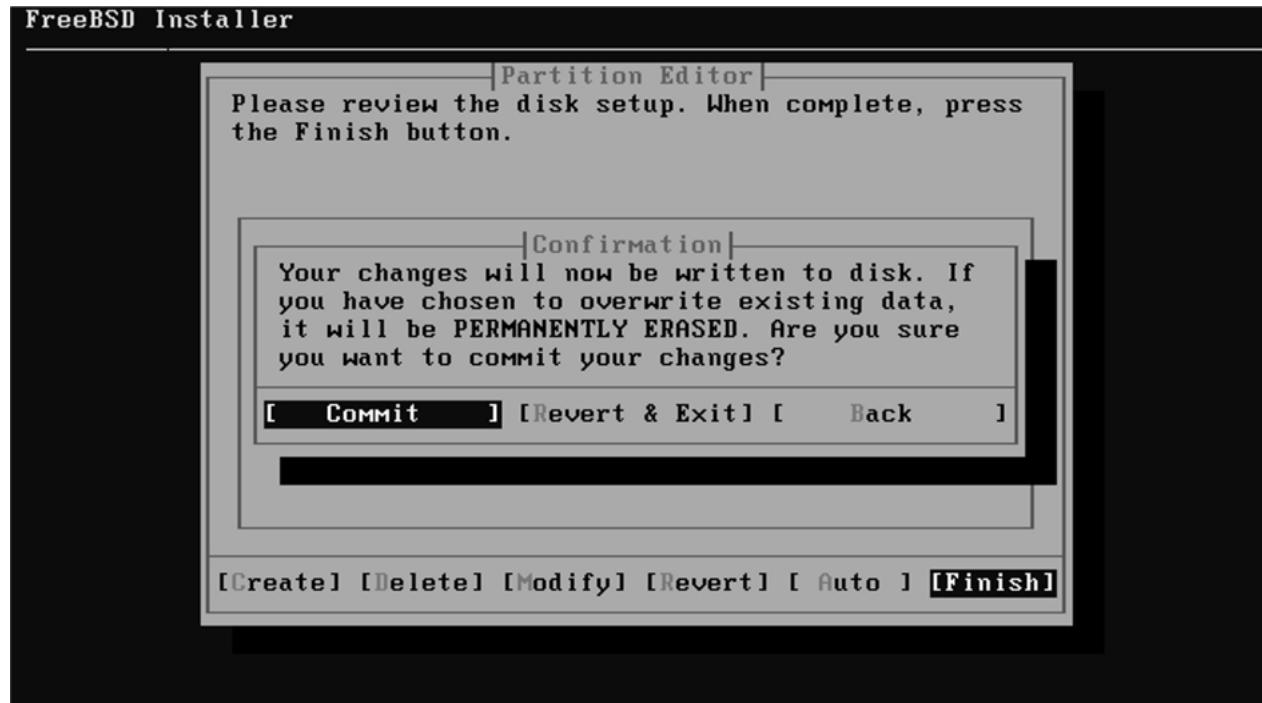
# 続き | 12-1. UFSファイルシステムを使用したい

① Enterキーを押す。



# 続き | 12-1. UFSファイルシステムを使用したい

① Enterキーを押す。



② しばらく待つ。

## 12-2. ZFSスナップショットを使用したい

### ■ ZFSスナップショットを取得したい

スナップショットとは、パソコンのファイルを指定した時点で保存する機能です。これを使えば、ファイルの変更や削除があっても、元の状態に戻すことができます。

#### □ 現在のディスクの状況を見る

```
% zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
zroot	13.3G	38.5G	96K	/zroot
zroot/ROOT	7.93G	38.5G	96K	none
zroot/ROOT/default	7.93G	38.5G	7.93G	/
zroot/home	5.36G	38.5G	96K	/home
zroot/home/pcuser	5.36G	38.5G	3.93G	/home/pcuser
zroot/tmp	5.98M	38.5G	5.79M	/tmp
zroot/usr	288K	38.5G	96K	/usr
zroot/usr/ports	96K	38.5G	96K	/usr/ports
zroot/usr/src	96K	38.5G	96K	/usr/src
zroot/var	1.15M	38.5G	96K	/var
zroot/var/audit	96K	38.5G	96K	/var/audit
zroot/var/crash	96K	38.5G	96K	/var/crash
zroot/var/log	652K	38.5G	436K	/var/log
zroot/var/mail	136K	38.5G	136K	/var/mail
zroot/var/tmp	104K	38.5G	96K	/var/tmp

## 続き | 12-2. ZFSスナップショットを使用したい

- zroot/home/pcuserのスナップショットを取得したい

```
% sudo zfs snapshot zroot/home/pcuser@20250826_1200
```

わかりやすいように、スナップショットを取得する日付、時刻を記入する。

- 取得したスナップショットを参照したい

```
% cd /home/pcuser/.zfs/snapshot
```

```
% ls
```

```
20250826_1200
```

```
% cd 20250826_1200
```

```
% ls
```

```
{
```

.zfsのディレクトリ以下にスナップショットを取得した時点のファイルが保存される。

この先、スナップショットを取得した時点のファイルが参照できる（コピーも可）。

## 続き | 12-2. ZFSスナップショットを使用したい

- スナップショットの一覧を見たい

```
% zfs list -t snapshot
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
zroot/home/pcuser@20250826_1200	61.6M	-	3.93G	-

- スナップショットの詳細を見たい

```
% zfs get creation zroot/home/pcuser@20250826_1200
```

NAME	PROPERTY	VALUE	SOURCE
zroot/home/pcuser@20250826_1200	creation	Tue Aug 26 12:00 2025	-

↑  
スナップショットを取得した、正確な日付  
と時刻が表示される。

## 続き | 12-2. ZFSスナップショットを使用したい

- 既存のファイルとスナップショットとのファイルを比較したい

```
% sudo zfs diff zroot/home/pcuser@20250826_1200
```

```
{
```

先頭に表示される記号の意味（主なもの）

記号	内容
-	ファイルの削除。-と+の組み合わせでファイルの中身の変更。
+	ファイルの追加。-と+の組み合わせでファイルの中身の変更。
M	ファイルの属性（パーミッションなど）の変更。
R	ファイル名の変更（リネーム）。またはファイルの移動（パスの変更）。

## 続き | 12-2. ZFSスナップショットを使用したい

- ロールバックしたい

```
% sudo zfs rollback zroot/home/pcuser@20250826_1200
```

※スナップショット取得以降の全てのファイルの変更が無効になりますので注意！

※最新のスナップショットにしかロールバックできません（強制的に過去のスナップショットにロールバックするオプションもあり。詳細はChatGPTへ質問してください）。

- スナップショットを削除したい

```
% sudo zfs destroy -R zroot/home/pcuser@20250826_1200
```

※スナップショットを削除しない限り、ディスクの容量は増え続けますので、定期的に削除してください。（運用例：一週間に一度、新たにスナップショットを取得し、古いスナップショットを削除するなど）

## 続き | 12-2. ZFSスナップショットを使用したい

### □ スナップショット運用上の注意点

- (前ページにも記載済みですが) スナップショットを削除しない限り、ディスクの容量は増え続けますので、定期的に削除してください（運用例：一週間に一度、新たにスナップショットを取得し、古いスナップショットを削除するなど）。
- freebsd-update installコマンドを実行すると、自動的に「zroot/ROOT/default@YYYY-MM-DD-HH:MM:SS-0」の形式でスナップショットが取得されます。これも、必要がなくなったら削除してください。

## 12-3. Windowsからもスナップショットを見たい

あらかじめ「7-3.Windowsやmacとファイル共有したい（SMB）」が設定されている必要があります。

### FreeBSD側の設定

#### ① 設定ファイルの変更

```
% sudo vi /usr/local/etc/smb4.conf
}
read only = No

vfs objects = shadow_copy2
shadow:snapdir = .zfs/snapshot
shadow:sort = desc
shadow:format = %Y%m%d_%H%M
shadow:localtime = yes
```

追加する

ファイル共有で、対象とするスナップ  
ショットの名称フォーマット

#### ② 再起動

```
% sudo shutdown -r now
```

## 続き | 12-3. Windowsからもスナップショットを見たい

### Windows側の設定

- Windowsから共有サーバーに接続する

- ① エクスプローラーを起動する
- ② アドレス欄に ¥¥192.168.1.8 と入力する
- ③ ネットワーク資格情報の入力画面にて、ユーザー名に「pcuser」、パスワードにパスワードを入力し、OKを押す。

FreeBSDパソコンのIPアドレス

このパスワードは pdbedit で入力したものに入力する

- ④ pcuser\_shareを右クリック→プロパティ(R)

- ⑤ "以前のバージョン"タブをクリック

- ⑥ 取得したスナップショットが表示される

- ⑦ ダブルクリックすることで、過去のファイルが閲覧できる

(アドレス欄のフォルダ名に、スナップショットの日時が表示される)

この日時は厳密なものではなく。zfsのスナップショット名を表示している。

## 12-4. 外付けHDDをフォーマットしたい

以下、

※※※ デバイス名を間違えないように注意してください ※※※

- ・USB接続のHDD
  - ・全領域をUFSファイルシステムでフォーマット（初期化）する
  - ・パーティションはMBR
- の例で説明します。

①USB接続の外付けHDDを、パソコンに接続する。

②デバイス名を確認する。

```
% dmesg | tail
```

```
umass0 on uhub1
umass0: <SABRENT SABRENT, class 0/0, rev 3.00/2.04, addr 3> on usbus0
umass0: SCSI over Bulk-Only; quirks = 0xc105
umass0:5:0: Attached to scbus5
da0 at umass-sim0 bus 0 scbus5 target 0 lun 0
da0: <SABRENT 0204> Fixed Direct Access SPC-4 SCSI device
da0: Serial Number
da0: 400.000MB/s transfers
da0: 305245MB (625142448 512 byte sectors)
da0: quirks=0x2<NO_6_BYTE>
```

③デバイス名をメモしておく。

## 続き | 12-4. 外付けHDDをフォーマットしたい

### ④パーティション情報を削除する

```
% sudo gpart destroy -F da0  
da0 destroyed
```

前ページでメモしたデバイス名

→ 完全消去されたHDDなどはエラーとなるので次に進む

### ⑤パーティション情報をMBRに設定する

```
% sudo gpart create -s MBR da0  
da0 created
```

### ⑥パーティションを作成する

```
% sudo gpart add -t freebsd da0  
da0s1 added
```

### ⑦フォーマットする

```
% sudo newfs /dev/da0s1
```

## 続き | 12-4. 外付けHDDをフォーマットしたい

### ⑧マウントする

```
% sudo mount /dev/da0s1 /mnt  
% cd /mnt  
% ls  
% df
```

### ⑨アンマウントする

```
% cd  
% sudo umount /mnt
```

## 12-5. 外付けHDDにバックアップを取りたい

以下、

- rsyncコマンドを使用
- 全ディレクトリが対象
- 12-4. でフォーマットした外付けHDDを使用する  
の例で説明します。

①バックアップ用の外付けHDDを接続する

```
% sudo mount /dev/da0s1 /mnt
```

②バックアップツールをインストールする

```
% sudo pkg install -y rsync
```

## 続き | 12-5. 外付けHDDにバックアップを取りたい

### ③バックアップを実行する

```
% sudo rsync -av --exclude=/mnt / /mnt/backup_all/
```

ループしないようにマウントしているディレクトリを除外する

最後の / を忘れない

### ③アンマウントする

```
% cd  
% sudo umount /mnt
```

## 12-6. ファイル・ディレクトリを指定してバックアップを取りたい

■ 外付けHDDに、ファイル・ディレクトリを指定してバックアップを取りたい

以下、

- rsyncコマンドを使用
- 指定したファイル・ディレクトリがバックアップ対象
- 12-4. でフォーマットした外付けHDDを使用

の例で説明します。

## 続き | 12-6. ファイル・ディレクトリを指定してバックアップを取りたい

### ① バックアップ対象 設定ファイルの作成

```
% vim ~/backup_config

# 対象ディレクトリ・ファイル
home/pcuser
etc/rc.conf
etc/ntp.conf
etc/wall_cmos_clock
usr/local/etc/X11/xorg.conf.d/vmmouse.conf
etc/gettytab
etc/ttys
etc/motd.template
etc/motd.template.org
boot.config
boot/loader.conf
etc/wpa_supplicant.conf
etc/ssh/sshd_config
usr/local/etc/smb4.conf
usr/local/lib/firefox/distribution/policies.json
usr/local/share/ghostscript/10.05.0/Resource/Init/cidfmap
etc/pkg/FreeBSD.conf
usr/local/etc/pkg/repos/quarterly.conf
etc/pf.conf
usr/local/etc/nginx/nginx.conf
etc/jail.conf
```

## 続き | 12-6. ファイル・ディレクトリを指定してバックアップを取りたい

### ② バックアップ除外 設定ファイルの作成

```
% vim ~/.backup_exclude_config  
# 除外ディレクトリ・ファイル  
home/pcuser/.cache  
home/pcuser/.i386-wine-pkg  
home/pcuser/.wine  
*.core  
.~
```

## 続き | 12-6. ファイル・ディレクトリを指定してバックアップを取りたい

③ バックアップ用の外付けHDDを接続する

```
% sudo mount /dev/da0s1 /mnt
```

④ バックアップを実行する

```
% sudo rsync -avr $  
--files-from=/home/pcuser/.backup_config $  
--exclude-from=/home/pcuser/.backup_exclude_config $  
/ /mnt/backup_20250826/
```

-a オプションには通常、-r オプションが含まれるが、  
--files-from オプションが使われると無効になるので、明示  
する。

⑤ アンマウントする

バックアップ取得日

```
% cd  
% sudo umount /mnt
```

## 12-7. ディレクトリの中身を同期（ミラーリング）させたい

- 外付けHDDと、ディレクトリの中身を同期（ミラーリング）させたい

以下、

- rsyncコマンドを使用
  - /home/pcuser以下が同期の対象
  - 12-4. でフォーマットした外付けHDDを使用
- の例で説明します。

①バックアップ用の外付けHDDを接続する

```
% sudo mount /dev/da0s1 /mnt
```

②バックアップを実行する

```
% sudo rsync -av --delete /home/pcuser/ /mnt/backup_sync/
```

③アンマウントする

コピー元にファイルがない場合はコピー先のファイルを削除する  
(つまり、同期するという意味)

```
% cd  
% sudo umount /mnt
```

## 12-8. スナップショット領域をバックアップしたい

- 外付けHDDに、ZFSスナップショット領域をバックアップしたい

以下、

- rsyncコマンドを使用
- /home/pcuser/.zfs/snapshot/20250826\_1200がバックアップの対象
- 12-4. でフォーマットした外付けHDDを使用

の例で説明します。

①バックアップ用の外付けHDDを接続する

```
% sudo mount /dev/da0s1 /mnt
```

②バックアップを実行する

```
% sudo rsync -av /home/pcuser/.zfs/snapshot/20250826_1200/ ¥  
/mnt/backup_20250826_1200/
```

③アンマウントする

```
% cd  
% sudo umount /mnt
```

## 12-9. 外付けHDDを完全消去したい

### ■外付けHDDを完全消去したい

※※※ デバイス名を間違えないように注意してください ※※※

① アンマウントしておく

② 消去する（ゼロデータを書き込む）

```
% sudo dd if=/dev/zero of=/dev/da0 bs=1M
```

※補足：消去時間は、320GBのHDD（5400rpm）にて、70分程度でした。

## 12-10. 外付けHDDをumount後、安全に取り外したい

### ■外付けHDDをumount後、安全に取り外したい

① アンマウントする

② HDDの回転を止めても良い、というコマンドを実行する

```
% sudo camcontrol stop /dev/da0
```

このコマンドでHDDのディスク回転音が聞こえなくなると思います。

③ USBケーブルをパソコン本体や外付けHDDから外す

④ HDDを保管する

## 12-11. SSDやHDDの型番、シリアル番号を見たい

### ■SSDやHDDの型番、シリアル番号を見たい

#### ① 接続されているデバイス名を得る

```
% sysctl -n kern.disks  
da0 ada0 ada1 nda0
```

FreeBSDパソコンに接続されているデバイス名

#### ② 型番とシリアル番号を得る (SATA向け)

```
% sudo camcontrol identify ada0 | grep -e 'device model' ¥  
-e 'serial number'
```

#### ③ 型番とシリアル番号を得る (NVMe向け)

```
% sudo camcontrol identify nda0 | grep -e 'Model Number' ¥  
-e 'Serial Number'
```

## 12-12. CrystalDiskInfoのような情報を見たい

### ■CrystalDiskInfoのような情報を見たい

#### ① ツールのインストール

```
% sudo pkg install -y smartmontools
```

#### ② 実行する

```
% sudo smartctl -a /dev/ada0
```

## 13-1. isoファイルをマウントしたい

マウントする

```
% sudo mount -t cd9660 /dev/`sudo mdconfig -f filename.iso` /mnt
```

アンマウントする

```
% cd  
% sudo umount /mnt
```

## 13-2. 毎日23:00に、自動的にパソコンをシャットダウンしたい

設定

```
% crontab -e
```

```
0 23 * * * sudo shutdown -p now ← 新規に追加する
```

※上記例では、sudoコマンドを使っているため、事前に「5-3. 特定のコマンドは、パスワードなしでsudoを実行したい」を設定しておく必要があります。

## 13-3. Linux等のrpmパッケージの中身を確認したい

rpm2cpioをインストールする

```
% sudo pkg install -y rpm2cpio
```

rpmパッケージを展開する

```
% rpm2cpio package_filename.rpm | cpio -idmv
```

## 13-4. 暗号化イメージを作成し、必要な時のみマウントしたい

### ① 暗号化イメージファイルの作成

※ Macのディスクユーティリティーで作成できるイメージファイル (.dmg) や、スパースディスクイメージ (.sparseimage) ファイルのようなもの。

※ ここでは8Mバイトのイメージファイルを作成する例

### イメージファイルの作成

```
% dd if=/dev/zero of=file.img bs=1M count=8
```

### 暗号化の初期化

```
% sudo mdconfig -a -t vnode -f file.img ← イメージファイルの使用宣言  
md0 ←  
% sudo geli init -s 4096 /dev/md0 ← 暗号化の初期化  
Enter new passphrase: *****  
Reenter new passphrase: *****  
  
% sudo geli attach /dev/md0 ← 暗号化デバイスの接続  
Enter passphrase: *****
```

### ファイルシステムの作成

```
% sudo newfs /dev/md0.eli
```

## 続き | 13-4. 暗号化イメージを作成し、必要な時のみマウントしたい

マウント

```
⌚ sudo mount /dev/md0.eli /mnt
```

確認

```
⌚ cd /mnt  
⌚ ls
```

アンマウント

```
⌚ cd  
⌚ sudo umount /mnt  
⌚ sudo geli detach /dev/md0
```

## 続き | 13-4. 暗号化イメージを作成し、必要な時のみマウントしたい

### ② 暗号化イメージファイルの利用 マウント

※ 実際の利用はシェルスクリプトを組むことを推奨  
※ umountし忘れ、突然の電源OFFなどの耐性は要確認

```
% sudo mdconfig -a -t vnode -f file.img
md1
%
% sudo geli attach /dev/md1
Enter passphrase: *****
%
% sudo mount /dev/md1.eli /mnt
```

読み書きできるようにパーティションを変更する

```
% cd /mnt
% sudo chmod 775 .
```

中身の確認

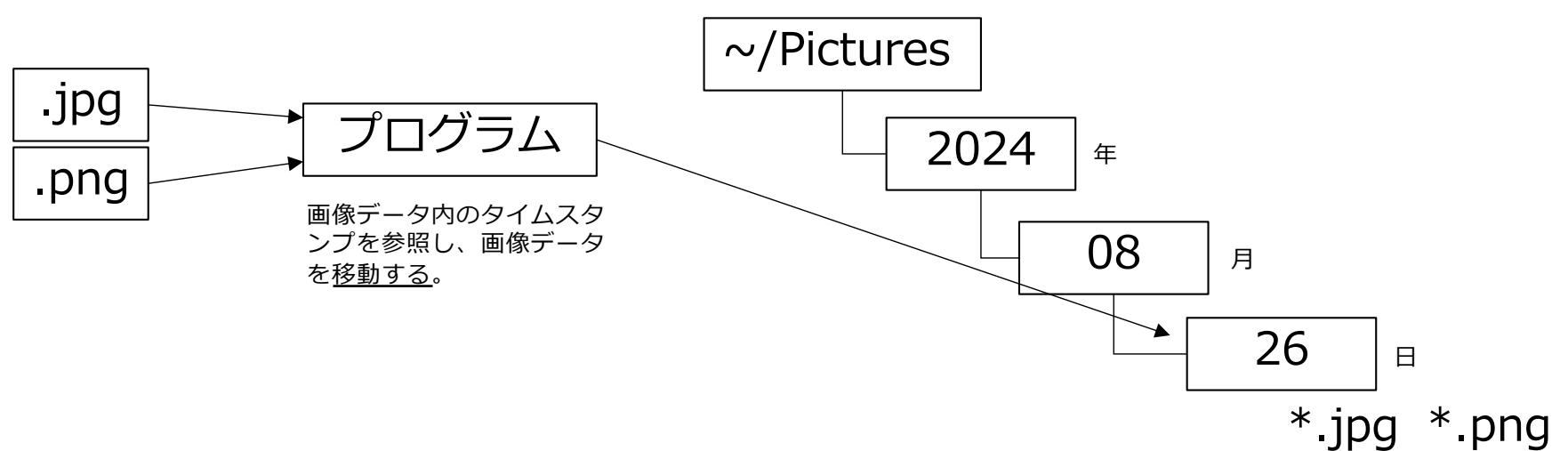
```
% ls
```

アンマウント

```
% cd
% sudo umount /mnt
% sudo geli detach /dev/md1
```

## 13-5. デジカメなどの画像データを、撮影日（年/月/日）別に整理したい

処理概要図：



exiftoolのインストール

```
% sudo pkg install -y p5-Image-ExifTool
```

保存先のディレクトリ作成

```
% mkdir ~/Pictures
```

## 続き | 13-5. デジカメなどの画像データを、撮影日（年/月/日）別に整理したい

### シェルスクリプトの作成

```
% vim move_picture.tcsh

#!/bin/tcsh

# 保存先のディレクトリを設定する
set base_dir="$HOME/Pictures"

# ファイル名が指定されていない場合は終了する
if ($#argv == 0) then
    echo "ファイルを指定してください。"
    exit 1
endif

# 各ファイルを処理
foreach file ($argv)
    # JPEGまたはPNGファイルのみ処理
    if (! -f "$file" || ("$file:e" != "jpg" && "$file:e" !=
"jpeg" && "$file:e" != "png" && "$file:e" != "JPG" &&
"$file:e" != "JPEG" && "$file:e" != "PNG")) then
        echo "$file は JPEG/PNG ファイルではありません。"
        continue
    endif

    # 作成日時を取得する
    set date_str=`exiftool -DateTimeOriginal -d "%Y/%m/%d" -s3
"$file"`
    if ("$date_str" == "") then
        echo "$file のタイムスタンプが見つかりません。"
        continue
    endif
```

```
# 年・月・日に分離する
set year=`echo $date_str | cut -d'/' -f1`
set month=`echo $date_str | cut -d'/' -f2`
set day=`echo $date_str | cut -d'/' -f3`

# 移動先ディレクトリを作成する
set target_dir="$base_dir/$year/$month/$day"
if (! -d "$target_dir") mkdir -p "$target_dir"

# 移動先ファイル名を設定する
set filename=`basename "$file"`
set target_file="$target_dir/$filename"

# ファイル名が重複する場合、連番を追加する
set count = 1
while (-e "$target_file")
    set target_file =
"$target_dir/${filename:r}_${count}.${filename:e}"
    @ count++
end

# ファイルを移動する
mv "$file" "$target_file"

# 移動したファイルのパーミッションを読み取り専用に設定する
chmod 440 "$target_file"
end
```

※本スクリプトは生成AIにて作成後、若干手直しました。

### 実行権限の付与

```
% chmod +x move_picture.tcsh
```

## 続き | 13-5. デジカメなどの画像データを、撮影日（年/月/日）別に整理したい

シェルスクリプトの実行

```
% ./move_picture.tcsh *.jpeg
```

## 13-6. Macで作成した .dmg / .sparseimage ファイルの中身を参照したい

HFSExplorerをインストールする

```
% sudo pkg install -y hfsexplorer
```

実行する

```
% hfsexplorer
```

使い方については、省略。

※読み取り可能なイメージファイル

- HFS+でフォーマットされた .dmg / .sparseimage ファイル (APFSは不可)
- 圧縮 .dmg ファイル
- 暗号化 .dmg / .sparseimage ファイル (要パスワード入力)

→ パスワード入力時、入力フォーカスが外れることがあるので、その場合はメニューバーをクリックする。

## 13-7. Macのpkgファイルの中身を確認・展開したい

Macアプリのインストーラーなどでよく使われている.pkgファイルの中身を、確認・展開する方法を説明します。

pkgパッケージの中身を確認する

```
% tar -tf package.pkg
```

pkgパッケージの中身の一部を展開する

```
% tar -xf package.pkg filename
```

Payloadファイルを展開する

```
% cat Payload | gunzip -dc | cpio -idmv
```

## 13-8. アプリがGTK系かQt系かを見分けたい

```
% ldd /usr/local/bin/appname | grep -E 'libgtk|libQt'
```

アプリケーションの名前を入力する。

主なアプリ	系列
OpenSCAD 2025.02.11	Qt5系
QGIS 3.40	Qt5系
Sylpheed 3.7.0	GTK2.0系
GIMP 2.10	GTK2.0系
xpad 3.1	GTK2.0系
uim-pref-gtk3	GTK3.0系

# 13-9. イメージファイルをUSBメモリーに書き込みたい

※※※ USBメモリーの内容は消去され、上書きされます ※※※

①USBメモリーをパソコンに接続する。

②デバイス名を確認する。

```
% dmesg | tail
```

```
umass0 on uhub1
umass0: <Sony Storage Media, class 0/0, rev 3.20/1.10, addr 2> on usbus0
umass0: SCSI over Bulk-Only; quirks = 0x8100
umass0:5:0: Attached to scbus5
da0 at umass-sim0 bus 0 scbus5 target 0 lun 0
da0: <Sony Storage Media PMAP> Removable Direct Access SPC-4 SCSI device
da0: Serial Number
da0: 400.000MB/s transfers
da0: 7416MB (15187968 512 byte sectors)
da0: quirks=0x2<NO_6_BYTE>
```

③デバイス名をメモしておく。

④イメージファイルを書き込む

```
% sudo dd if=imagefile.img of=/dev/da0 bs=1M conv=sync status=progress
```

## 13-10. Windowsで使用しているUSBメモリーを読み取りたい

①USBメモリーをパソコンに接続する。

②デバイス名を確認する。

```
% dmesg | tail
```

```
umass0 on uhub1
umass0: <Sony Storage Media, class 0/0, rev 3.20/1.10, addr 2> on usbus0
umass0: SCSI over Bulk-Only; quirks = 0x8100
umass0:4:0: Attached to scbus4
da0 at umass-sim0 bus 0 scbus4 target 0 lun 0
da0: <Sony Storage Media PMAP> Removable Direct Access SPC-4 SCSI device
da0: Serial Number
da0: 400.000MB/s transfers
da0: 7416MB (15187968 512 byte sectors)
da0: quirks=0x2<NO_6_BYTE>
```

③ デバイス名をメモする。

## 続き | 13-10. Windowsで使用しているUSBメモリーを読み取りたい

### ④ マウントする

```
% sudo mount_msdosfs -L ja_JP.UTF-8 /dev/da0s1 /mnt
```

### ⑤ USBメモリの内容を確認する

```
% cd /mnt  
% ls
```

前のページでメモした  
デバイス名+s1  
を入力する（ここでは  
da0s1）。

エラー表示される場合は、  
デバイス名+p1  
で試す（da0p1など）。

### ⑥ ホームディレクトリに移動する

```
% cd
```

### ⑦ アンマウントする

```
% sudo umount /mnt
```

### ⑧ USBメモリーをパソコンから抜く

## 13-11. (メモ書き) 簡単なデータベースソフトを作りたい・使いたい

ここでは、当方の簡易な検討結果をご参考情報として記載します。なお自作の場合、データベースエンジンは、SQLite3の使用を想定しています。

UI種別	ツール種別	検討結果
TUI	シェル+標準dialogコマンドで自作。	お手軽に使える感じ。キー操作が独特。画面まわりのカスタマイズ性に欠ける。日本語表示・入力はOKだが、曖昧幅の入力が不完全。
TUI	シェル+gumコマンドで自作	クセのあるCUI。
GUI	シェル+yadまたはzenityコマンドで自作。	お手軽に使える感じ。プログラミングがやや面倒か？
GUI	シェル+Xdialogまたはkdialogコマンドで自作。	複数入力の画面が作れなそう。
GUI	KexiやLibre OfficeのBase Database	Accessっぽい感じ。
GUI	Python+Tkinterで自作。 書籍「Pythonでデスクトップアプリを作ろう」を推奨。	プログラミングがやや面倒か？
GUI	Wine+Windowsのシェアウェア「TCARD」。	お手軽に使える。
web	FastAPI + Jinja2 + uvicornで自作。	意外とお手軽に使える。 <a href="#">「15-1. データベースwebアプリを作りたい」</a> を参照のこと。

## 13-12. 自作のmanページを作成したい

Unix系システムで古くから使われている「man」<sup>マン</sup>（コマンドの説明書を読むための標準ツール）の仕組みを利用して、自分専用のドキュメント（TIPS・メモ・使い方集）を作成する方法を解説します。

今回は例として、pkg（パッケージ）コマンドにおいてよく使われるオプションの、manページを作成します。

- ・ディレクトリの作成

```
% mkdir -p ~/man/man7
```

ホームディレクトリにmanというディレクトリがあれば、マニュアルページとして検索対象になる。

# 続き | 13-12. 自作のmanページを作成したい

- manページの作成

```
○ vim ~/man/man7/pkg-tips.7
```

```
.TH PKG-TIPS 7 "2025年11月" "個人メモ" "ユーザーガイド" ← ヘッダーとフッター
.SH 【名前】 ← セクション見出し
pkg-tips ¥- よく使うpkgコマンドの例集
.SH 【使用例】 ← 表示用ハイフン
.TP ← タグ付き段落
インストールしたパッケージを全て表示する:
.EX ←
pkg info ← 等幅で表示（コードなど）
.EE ← htmlで言うところの、
.TP <pre><code>
手動でインストールしたパッケージのみを表示する:
.EX </code></pre>
pkg query -e '%a = 0' '%o'
.EE
.TP
パッケージの依存関係を表示する:
.EX
pkg info -d -r ¥fIパッケージ名¥fR ← ¥fI・・・¥fRで斜体
.EE (実際はアンダーライン表示)
```

- 実行

```
○ man pkg-tips
```

# 続き | 13-12. 自作のmanページを作成したい

## ・他、よく使われるmanマクロの説明

.SH 【オプション】

.TP

.BI "-d " 秒数 ←

指定した秒数後にキャプチャする。

.BI は、太字・斜体を引数ごとに交互に適用する、の意味（実際は  
斜体は、アンダーライン表示となる）

.TP

¥&.isoファイルマウントする：←

¥&. でドットを表す。

## 13-13. Windowsの自己解凍型.exeファイルを展開したい

Windowsでよく使われる自己解凍型の.exeファイルや、.msiファイル、.cabファイル、MS-DOS時代からの圧縮形式（SETUP.IN\_などのファイル名）の中身を確認・展開したい方法について解説します。下記のツールで確認してみてください。

### ■方法1：7-Zipの利用

```
% sudo pkg install -y 7-zip ← 7-Zipのインストール  
% 7z l setup.exe ← 中身の確認  
% 7z x setup.exe ← 展開  
% 7z x SETUP.IN_ ← SETUP.IN_ファイルの展開
```

# 続き | 13-13. Windowsの自己解凍型.exeファイルを展開したい

## ■方法2 : msitoolsの利用

```
% sudo pkg install -y msitools ← msitoolsのインストール
```

```
% msiextract setup.msi ← 展開
```

※7-Zipでも展開できます。

## ■方法3 : cabextractの利用

⇒ 10-11. 拡張子が.cabのファイルを展開したい を参照。

## 13-14. (メモ書き) 日本語入力時の表示と未確定文字の扱い

端末工ミュレータ、およびWine上のテキストエディタでの日本語入力時の表示、および入力時の未確定文字の扱いなど、日本語入力周り関連の検証結果などをメモ書きしておきます。

### ■端末工ミュレータにおける日本語入力時の表示モード一覧

	fcitx	IBus	uim
mlterm (xim経由?)	オーバーザスポット	オーバーザスポット	インライン
xterm	オーバーザスポット	オーバーザスポット	インライン
xfce4-terminal	インライン	インライン	インライン
urxvt	オーバーザスポット	オーバーザスポット	インライン

## 続き | 13-14. (メモ書き) 日本語入力時の表示と未確定文字の扱い

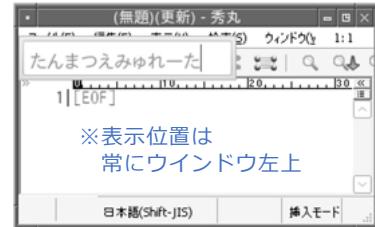
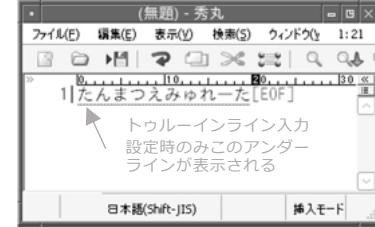
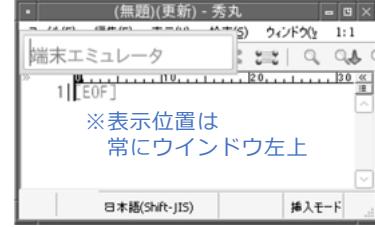
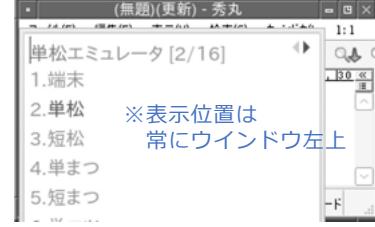
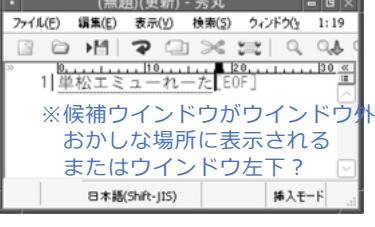
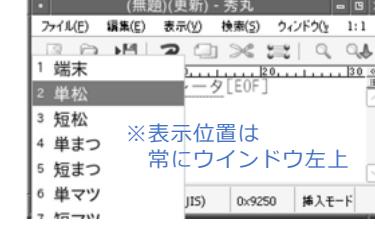
■ mlterm (xim経由?) での日本語入力時の画面 (FreeBSD14.2リリース時のpkg群で検証)

	fcitx オーバーザスポット	IBus オーバーザスポット	uim インライン
入力時			
変換時			
選択時			
確定時			

予測変換等の機能はOFFにしてあります

## 続き | 13-14. (メモ書き) 日本語入力時の表示と未確定文字の扱い

### ■ Wine+秀丸 (トゥルーアインライン入力設定時) での日本語入力時の画面

	fcitx	IBus	uim
入力時	 <p>※表示位置は 常にウインドウ左上</p>		 <p>トゥルーアインライン入力 設定時のみこのアンダーラインが表示される</p>
変換時	 <p>※表示位置は 常にウインドウ左上</p>		
変換候補 表示時	 <p>※表示位置は 常にウインドウ左上</p>	 <p>※候補ウインドウがウインドウ外など おかしな場所に表示される またはウインドウ左下？</p>	 <p>※表示位置は 常にウインドウ左上</p>
確定時			

## 続き | 13-14. (メモ書き) 日本語入力時の表示と未確定文字の扱い

### ■他、日本語メソッド・端末エミュレータに関する雑記

種別	内容
IBus	<ul style="list-style-type: none"><li>ウインドウシステム立ち上げ後に、毎回「直接入力」から「ひらがな」にマウスで変更する必要あり。⇒これが原因で本ガイドではuimを採用した経緯あり。</li><li>上記問題点、openSUSEでは新たに入力方式？を設けて対応か？？？</li></ul>
wineでの日本語入力	<ul style="list-style-type: none"><li>Linuxの一部ディストロでは日本語入力のオンライン化をいさぎよく諦め？、レジストリにて HKEY_CURRENT_USER\Software\Wine\X11 DriverのInputStyleにrootを設定⇒ウインドウ左下外に入力中の文字が表示される。</li></ul>
IBus/Fcitx	<ul style="list-style-type: none"><li>xmodmapでCAPSキーを[半角/全角キー]に置き換えるも、Webブラウザなどでまれに設定が効かないことがある（IBusやFcitx側でキーを優先的にフックしている模様）。⇒本ガイドではuimを使用し、さらにxkbcompでキーの置き換えを実施。</li></ul>
曖昧幅	<ul style="list-style-type: none"><li>曖昧幅を全角扱いできるのは、mltermやxfce4-terminal。</li></ul>
結合文字	<ul style="list-style-type: none"><li>結合文字を分解表示できるのは、mlterm。</li></ul>
罫線	<ul style="list-style-type: none"><li>曖昧幅を全角のまま、罫線のみを半角表示できるのは、mlterm (.mlterm/mainに要記述)</li></ul>

# 14-1. chroot環境を作つてGUIアプリを動作させたい

## ■ chroot環境を作つて X Window System のGUIアプリを動作させたい

chrootは、指定したディレクトリを一時的にルートディレクトリ (/) とすることで、仮想的な隔離環境を提供する技術です。FreeBSD上でこの機能を利用することで、本体のシステムとはファイル的に分離された「ミニFreeBSD環境」を構築し、その中でアプリケーションを動作させることができます。

今回この仕組みを応用し、ミニ環境内からFirefoxを起動し、その画面出力をXサーバ（ホスト側）に表示させる手順を説明します。

- chroot用のディレクトリ、システムファイルを構築する

```
% su  
Password: *****  
# mkdir -p /srv/chroot  
# cd /srv/chroot  
# fetch https://download.freebsd.org/ftp/releases/amd64/14.3-RELEASE/base.txz  
# tar -xvpf base.txz
```

FreeBSD14.3のbaseをダウンロードし、インストールしている

## 続き | 14-1. chroot環境を作つてGUIアプリを動作させたい

- ・ネットワーク、タイムゾーン、Xサーバー・クライアント環境を設定する

```
# cp /etc/resolv.conf /srv/chroot/etc ← ホストOS側からコピーしている
# cp /etc/localtime /srv/chroot/etc
# mkdir -p /srv/chroot/tmp/.X11-unix
```

- ・chroot環境に移る

```
# chroot /srv/chroot /bin/sh
```

- ・ネットワーク、タイムゾーン、OSアップデート、pkg環境を構築する

```
[chroot]# freebsd-update fetch ← 待ち状態になった場合、Qキーを何度か押す
[chroot]# freebsd-update install
[chroot]# pkg
... [y/N]: y
```

便宜上このように表現していますが、  
実際は[chroot]は表示されません。

# 続き | 14-1. chroot環境を作つてGUIアプリを動作させたい

- ・アプリ、フォントなどをインストールする

```
[chroot]# pkg install -y firefox mesa-libs ja-font-ipa xset
```

パッケージの環境は、特に指定していないので、quarterly を参照しています。

- ・ユーザーを設定する

```
[chroot]# adduser
```

※72～73ページと同じ操作を行う。

- ・ユーザーの環境変数を設定する

```
[chroot]# vi /home/pcuser/.cshrc
```

```
setenv LANG ja_JP.UTF-8  
setenv LC_ALL ja_JP.UTF-8  
  
setenv GTK_IM_MODULE uim  
setenv QT_IM_MODULE uim  
setenv XMODIFIERS "@im=uim"  
  
setenv DISPLAY :0
```

日本語設定、および  
キー入力周りの設定が必要

先頭部分に追加する

画面表示先の設定

## 続き | 14-1. chroot環境を作つてGUIアプリを動作させたい

- ・ユーザーの環境変数を設定する（続き）

```
[chroot]# vi /home/pcuser/.login  
  
xset r rate 250 40
```

} 最後の方に追加する

キー入力周りの設定が必要

- ・chroot環境から抜ける

```
[chroot]# exit  
# exit  
%
```

# 続き | 14-1. chroot環境を作つてGUIアプリを動作させたい

## □ chroot環境のFirefoxを起動する

```
% xhost +local: ← 画面使用許可を与える。ログアウトするまで有効。  
% sudo mount -t nullfs /tmp/.X11-unix /srv/chroot/tmp/.X11-unix  
% sudo mount -t devfs devfs /srv/chroot/dev ↑  
↑  
X Window System、chroot環境を使うための設定。再起動するまで有効。  
% sudo chroot /srv/chroot su - pcuser -c 'firefox'
```

≡ → ヘルプ → Firefoxについて(A) を選択すると、異なったバージョンのFirefoxが表示されると思います。

### ・ 使用後、アンマウントする

```
% sudo umount /srv/chroot/tmp/.X11-unix  
% sudo umount /srv/chroot/dev
```

## 続き | 14-1. chroot環境を作つてGUIアプリを動作させたい

### □chroot環境を完全消去する

```
% sudo chflags -R noschg /srv/chroot  
% sudo rm -rf /srv/chroot
```

## 14-2. 仮想環境を使って、FreeBSD内にLinuxをインストールしたい

FreeBSDに標準搭載されたハイパーバイザ型仮想化技術「bhyve（ビーハイブ）」を用い、FreeBSD上でLinuxやWindowsなどのOSを仮想マシンとして実行することができます。通常はSSHなどを通じてCUIベースで利用しますが、VNCを通じてデスクトップ環境を利用することもできます。

ここでは、Ubuntuをインストールし、デスクトップ環境を構築する手順について説明します。

- ・ファームウェアをインストールする

```
% sudo pkg install -y bhyve-firmware
```

- ・仮想ネットワークを設定する

```
% sudo vi /etc/rc.conf
```

```
cloned_interfaces="bridge0 tap0"
ifconfig_bridge0="addm em0 addm tap0 up"
ifconfig_tap0="up"
```

## 続き | 14-2. 仮想環境を使って、FreeBSD内にLinuxをインストールしたい

- 前のページで設定した仮想ネットワークの設定を有効にする

```
% sudo shutdown -r now
```

- ログインする
- ディスクイメージ用のディレクトリを作成する

```
% mkdir -p ~/vm/ubuntu  
% cd ~/vm/ubuntu
```

- UbuntuのインストーラISOイメージをダウンロードする

```
% fetch https://releases.ubuntu.com/22.04/ubuntu-22.04.5-desktop-amd64.iso
```

- Ubuntu用のディスクスペースを確保する（ディスクイメージを作成する）

```
% truncate -s 20G ubuntu.img
```

↑  
ディスク容量  
20GByte

## 続き | 14-2. 仮想環境を使って、FreeBSD内にLinuxをインストールしたい

- UbuntuのインストーラーISOイメージを指定して仮想環境を起動する

```
% sudo bhyve ¥  
-c 2 ¥ ← 仮想CPU数  
-m 4G ¥ ← メモリ  
-s 0:0,hostbridge ¥  
-s 1:0,lpc ¥  
-s 2:0,ahci-cd,./ubuntu-22.04.5-desktop-amd64.iso ¥ ← CDデバイス  
-s 3:0,virtio-blk,./ubuntu.img ¥ ← 仮想ディスク  
-s 4:0,virtio-net,tap0 ¥ ← ネットワーク  
-s 29,fbuf,tcp=0.0.0.0:5900,w=1024,h=768 ¥ ← 画面出力先  
-s 30,xhci,tablet ¥  
-l bootrom,/usr/local/share/uefi-firmware/BHYVE_UEFI.fd ¥  
-AHP -W ¥  
ubuntu ← 仮想環境名
```

## 続き | 14-2. 仮想環境を使って、FreeBSD内にLinuxをインストールしたい

- (別のウインドウで) VNCのインストール

```
% sudo pkg install -y tigervnc-viewer
```

- Ubuntuの画面に接続する

```
% vncviewer localhost:5900
```

- インストール作業を行う

1. 日本語を選択
2. Ubuntuをインストールをクリック
3. キーボードレイアウトは英語キーボードの場合「English(US)」「English(US)」、日本語キーボードの場合「Japanese」「Japanese」を選択して、続けるをクリック
4. アップデートと他のソフトウェアは、続けるをクリック
5. インストールの種類は、インストール(I)をクリック
6. ディスクに変更を書き込みますか？は、続けるをクリック
7. どこに住んでいますか？は、Tokyoが選択されていることを確認し、続けるをクリック
8. あなたの情報を入力してください  
あなたの名前、pcuser　コンピュータの名前、vm100001  
ユーザー名の入力、pcuser　パスワードの入力、入力する　パスワードの確認、入力する  
続けるをクリック

## 続き | 14-2. 仮想環境を使って、FreeBSD内にLinuxをインストールしたい

- ・インストール完了後、仮想環境を停止する

「インストールが完了しました」の画面が表示されたら、下記を実行してください。

```
% sudo bhyvectl --destroy --vm=ubuntu
```

## 続き | 14-2. 仮想環境を使って、FreeBSD内にLinuxをインストールしたい

### □ 仮想環境でUbuntuを起動する

```
% cd  
% sudo bhyve -c 2 -m 4G -s 0:0,hostbridge -s 1:0,lpc -s 3:0,virtio-blk,/vm/ubuntu/ubuntu.img -s 4:0,virtio-net,tap0 -s 29,fbuf,tcp=0.0.0.0:5900,w=1024,h=768 -s 30,xhci,tablet -l bootrom,/usr/local/share/uefi-firmware/BHYVE_UEFI.fd -AHP -W  
ubuntu
```

~などは使えないで相対パスか絶対パス

- Ubuntuの画面に接続する

```
% vncviewer localhost:5900
```

## 続き | 14-2. 仮想環境を使って、FreeBSD内にLinuxをインストールしたい

- ubuntuの終了

画面右上の電源ボタン→ 電源オフ/ログアウト → 電源オフ... → 電源オフ

- 仮想環境を停止する

```
% sudo bhyvectl --destroy --vm=ubuntu
```

※補足：bhyveにおいて、仮想環境のサウンド機能は、提供されません。

## 14-3. 過去のパッケージ群を使用したい

FreeBSDでは通常、安定版のパッケージ群（`pkg install`でインストールできるもの）を参照するように設定されていますが、リリース当時のパッケージ群もサーバーに残してしています\*1。パッケージ群が不安定で困っている場合は、下記を試してみてください（本メモでは、84ページにて設定済みです）。

- 確認作業
  - webブラウザで、<http://pkg.freebsd.org/FreeBSD:14:amd64/>にアクセスする

### Index of /FreeBSD:14:amd64/

File Name ↓	File Size ↓	Date ↓
<a href="#">Parent directory/</a>	-	-
<a href="#">base_latest/</a>	-	2026-Jan-22 00:04
<a href="#">base_release_0/</a>	-	2024-Sep-20 00:06
<a href="#">base_release_1/</a>	-	2025-Feb-21 12:06
<a href="#">base_release_2/</a>	-	2025-Oct-01 00:05
<a href="#">base_release_3/</a>	-	2025-Dec-17 12:07
<a href="#">base_weekly/</a>	-	2026-Jan-18 12:05
<a href="#">hash_latest/</a>	-	2025-Nov-16 15:46
<a href="#">kmods_latest/</a>	-	2026-Jan-22 00:04
<a href="#">kmods_latest_2/</a>	-	2025-Oct-14 12:05
<a href="#">kmods_latest_3/</a>	-	2026-Jan-22 00:04
<a href="#">kmods_quarterly/</a>	-	2026-Jan-22 00:04
<a href="#">kmods_quarterly_2/</a>	-	2025-Oct-14 12:05
<a href="#">kmods_quarterly_3/</a>	-	2026-Jan-22 00:04
<a href="#">latest/</a>	-	2026-Jan-20 16:55
<a href="#">quarterly/</a>	-	2026-Jan-20 05:10
<a href="#">release_1/</a>		2024-May-19 04:31
<a href="#">release_2/</a>	-	2024-Nov-16 02:27
<a href="#">release_3/</a>	-	2025-May-13 20:56
<a href="#">release_0</a>	46	2023-Oct-18 18:52

この `release_3` が、FreeBSD14.3リリース時点の  
パッケージ群と思われる

\*1 ただし、バージョンのサポートが切れると、削除されます（すでにFreeBSD12のものは無いです）。

## 続き | 14-3. 過去のパッケージ群を使用したい

- ・設定作業

```
% sudo vi /etc/pkg/FreeBSD.conf  
:  
url: "pkg+https://pkg.FreeBSD.org/${ABI}/release_3",  
:
```

※本作業を行うと、各パッケージのバージョンアップは一切行われません。

↑  
変更する

## 続き | 14-3. 過去のパッケージ群を使用したい

- リポジトリを指定して、パッケージのインストールを行いたい

前のページまでの方法では、各パッケージのバージョンアップは一切行われません。しかし、明示的に新しいパッケージをインストールしたい場合もあると思いますので、その方法について解説します。  
(注記：下記の例では Chromium 最新版のインストール方法を紹介していますが、2026年2月時点では大幅にパッケージの内容が更新されており、依存関係も大幅に変わっているため、推奨しません)

- pkgリポジトリの初期設定

```
% sudo mkdir -p /usr/local/etc/pkg/repos/  
% sudo cp /etc/pkg/FreeBSD.conf /usr/local/etc/pkg/repos/quarterly.conf  
% sudo vi /usr/local/etc/pkg/repos/quarterly.conf
```

quarterly: { ← 变更する  
 url: "pkg+https://pkg.FreeBSD.org/\${ABI}/quarterly", ← 变更する  
 mirror\_type: "srv",  
 signature\_type: "fingerprints",  
 fingerprints: "/usr/share/keys/pkg",  
 enabled: no ← 变更する  
}

## 続き | 14-3. 過去のパッケージ群を使用したい

- pkgリポジトリを指定してパッケージをインストール
- リポジトリを有効にする

```
% sudo vi /usr/local/etc/pkg/repos/quarterly.conf
```

```
:
```

```
enabled: yes ← 変更する
```

- リポジトリを更新し、リポジトリ指定し検索する

```
% sudo pkg update
```

```
% pkg search -r quarterly chromium
```

```
chromium-144.0.7559.109
```

```
Google web browser based on WebKit
```

```
:
```

新しいバージョン

## 続き | 14-3. 過去のパッケージ群を使用したい

- リポジトリを指定し、インストールする

```
% sudo pkg install -r quarterly chromium
```

```
:  
Number of packages to be removed: 12  
Number of packages to be installed: 17  
Number of packages to be upgraded: 42
```

```
The operation will free 432 MiB.  
426 MiB to be downloaded.
```

```
:  
Proceed with this action? [y/N]: y
```

リポジトリを明示的に指定する \*1

依存関係が少なく、影響範囲が少ないとと思われるのならば、インストールしても問題ないと思われるが、依存関係が多く、更には、無関係のパッケージまで削除される場合があるので、理解した場合のみ実施のこと。

- リポジトリを無効に戻す（※忘れないこと※ \*1）

```
% sudo vi /usr/local/etc/pkg/repos/quarterly.conf
```

```
:  
enabled: no ← 変更する
```

\*1 この指定を忘れた場合、pkgは /etc/pkg および /usr/local/etc/pkg/repos の.confファイルで enabled=yes になっているパッケージ全て検索してしまう。通常運用下では忘れてしまう可能性が大なので、使用後は明示的に enabled=no にするようにしている。

## 14-4. インストール済みのパッケージをバックアップしたい

### ■ インストール済みのパッケージをバックアップしたい

```
% sudo pkg fetch -d -o /home/pcuser/packages `pkg query '%n'`
```

:

↑  
↑ バックアップ先ディレクトリ  
↓ 依存関係も含める

↑  
インストールされている  
全てのパッケージ名

```
Proceed with fetching packages? [y/N]: y
```

### □ すべてのパッケージを削除したい

- rootでログインする。

↓ すべてのパッケージを対象にする

```
# pkg delete -af
```

:

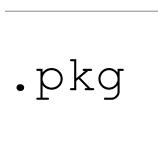
↑ 強制削除

```
Proceed with deinstalling packages? [y/N]: y
```

## 続き | 14-4. インストール済みのパッケージをバックアップしたい

- バックアップしたパッケージからパッケージを再インストールしたい
  - ・ rootでログインする。
  - ・ pkgをインストールする

```
# cd /tmp
# ls /home/pcuser/packages/All/pkg*
/home/pcuser/packages/All/pkg-2.1.2.pkg
/home/pcuser/packages/All/pkgconf-2.3.0,1.pkg
# tar -xvf /home/pcuser/packages/All/pkg-2.1.2.pkg
# cd usr/local/sbin ←———— /tmp/usr/local/sbinのこと
# ls
pkg          pkg-static
# mkdir /usr/local/sbin
# cp pkg-static /usr/local/sbin
```



パッケージファイル

## 続き | 14-4. インストール済みのパッケージをバックアップしたい

(続き)

- ・バックアップしたパッケージからインストールする

```
# cd /home/pcuser/packages/All  
# pkg-static add ./*.pkg  
  
# shutdown -r now
```

## 14-5. VMware Workstation 17 Playerでの利用方法

### ■ VMware Tools をインストールする

```
% sudo pkg install -y open-vm-tools  
% sudo pkg install -y xf86-input-vmmouse
```

### ■ 時刻を修正する

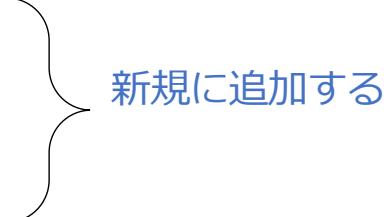
```
% sudo touch /etc/wall_cmos_clock
```

### ■ ホスト ⇄ ゲストOS間を、マウスで行き来できるように設定する

```
% sudo vi /usr/local/etc/X11/xorg.conf.d/vmmouse.conf
```

```
Section "InputClass"  
    Identifier      "Mouse0"  
    Driver          "vmmouse"  
    MatchIsPointer "on"  
EndSection
```



### ■ ホスト ⇄ ゲストOS間で、コピペできるように設定する

```
% vim ~/.xinitrc  
    {  
        vmware-user-suid-wrapper & } この位置に追加する  
  
exec fvwm3
```

## 14-6. Jail（軽量コンテナ）を用いたWebサーバーを構築したい

ジエイル

Jailは、FreeBSDが標準で備えるOSレベルの仮想化機能であり、プロセスやファイルシステム、ネットワーク環境を独立した空間として分離する技術です。この仕組みを利用することで、1台のFreeBSD上に複数の「小さなFreeBSD環境」を構築し、それぞれを安全かつ独立して運用することができます。

今回は、このJail機能を用いてWebサーバー環境を構築し、ホストシステムと分離した状態で動作させる手順を説明します。

- Jail用のディレクトリ、システムファイルを構築する

```
% su  
Password: *****  
# mkdir -p /home/jail/j1-web  
# cd /home/jail/j1-web  
# fetch https://download.freebsd.org/ftp/releases/amd64/14.3-RELEASE/base.txz  
# tar -xvpf base.txz
```

## 続き | 14-6. Jail (軽量コンテナ) を用いたWebサーバーを構築したい

- ・ネットワーク、タイムゾーンを設定する（ホスト側の設定を流用）

```
# cp /etc/resolv.conf /home/jail/jl-web/etc  
# cp /etc/localtime /home/jail/jl-web/etc
```

- ・pkgファイルの参照先の固定化（ホスト側の設定を流用）

```
# cp /etc/pkg/FreeBSD.conf /home/jail/jl-web/etc/pkg
```

- ・Jail内のFreeBSDのパッチレベルを最新にする\*1

```
# freebsd-update -b /home/jail/jl-web fetch  
# freebsd-update -b /home/jail/jl-web install
```

\*1 ホスト側も最新にしておいてください。

## 続き | 14-6. Jail (軽量コンテナ) を用いたWebサーバーを構築したい

- Jail設定ファイルの作成

```
# vi /etc/jail.conf
      1 (エル)
jl-web {
    host.hostname = "jl100001.local"; ← ホスト名
    ip4.addr = 192.168.1.108; ← jail用のIPアドレス
    interface = em0; ← ネットワークインターフェース名
    path = "/home/jail/jl-web"; ← Jailのディレクトリ
    mount.devfs;
    exec.start = "/bin/sh /etc/rc";
    exec.stop = "/bin/sh /etc/rc.shutdown";
    persist;
}
```

## 続き | 14-6. Jail (軽量コンテナ) を用いたWebサーバーを構築したい

- FreeBSD起動時にJail機能が使えるように設定する

```
# vi /etc/rc.conf  
  
jail_enable="YES"
```

- 再起動する

```
# shutdown -r now
```

- pcuserでログインして、startxする。

- 起動確認

```
% jls  
 JID  IP Address      Hostname          Path  
 1   192.168.1.108  jl100001.local  /home/jail/jl-web
```

## 続き | 14-6. Jail (軽量コンテナ) を用いたWebサーバーを構築したい

- Jail内にログイン

```
% sudo jexec jl-web /bin/tcsh  
root@jl100001:/ #
```

- パッケージの初期化 (Jail内)

```
root@jl100001:/ # pkg  
· · · [y/N]: y
```

## 続き | 14-6. Jail (軽量コンテナ) を用いたWebサーバーを構築したい

Webサーバーのnginxのインストール (Jail内)

```
root@j1100001:/ # pkg install -y nginx
```

起動設定 (Jail内)

```
root@j1100001:/ # vi /etc/rc.conf
```

```
nginx_enable="YES" ← 新規に追加する
```

公開するディレクトリの作成 (Jail内)

```
root@j1100001:/ # mkdir -p /home/www
```

実際の運用の際には、このディレクトリのオーナー・グループ・パーミッション設定については要調整。

## 続き | 14-6. Jail (軽量コンテナ) を用いたWebサーバーを構築したい

テスト用の index.html ファイルを作成する (Jail内)

```
root@j1100001:/ # vi /home/www/index.html
```

```
<h1>Hello Jail World!</h1>
```

新規に追加する

jail環境内には、  
vimをインストールしていないので  
viで編集する。

## 続き | 14-6. Jail (軽量コンテナ) を用いたWebサーバーを構築したい

### nginx設定ファイルの編集 (Jail内)

```
root@j1100001:/ # mv /usr/local/etc/nginx/nginx.conf /usr/local/etc/nginx/nginx.conf.org  
root@j1100001:/ # vi /usr/local/etc/nginx/nginx.conf  
  
worker_processes 1;  
  
events {}  
  
http {  
    server {  
        listen 80;  
        server_name _;  
  
        root /home/www;  
        index index.html;  
    }  
}
```



新規に追加する

## 続き | 14-6. Jail (軽量コンテナ) を用いたWebサーバーを構築したい

Jailから抜ける

```
root@j1100001:/ # exit
```

ファイアウォールの設定（ホスト側を編集する）

```
% sudo vi /etc/pf.conf
```

```
pass in on em0 proto tcp to 192.168.1.108 port { 80 } keep state
```

↑ 最終行に追加する  
Jail用のIPアドレスに対して設定する

- 再起動する

```
% sudo shutdown -r now
```

- pcuserでログインして、startxする。

## 続き | 14-6. Jail (軽量コンテナ) を用いたWebサーバーを構築したい

Webブラウザから確認する

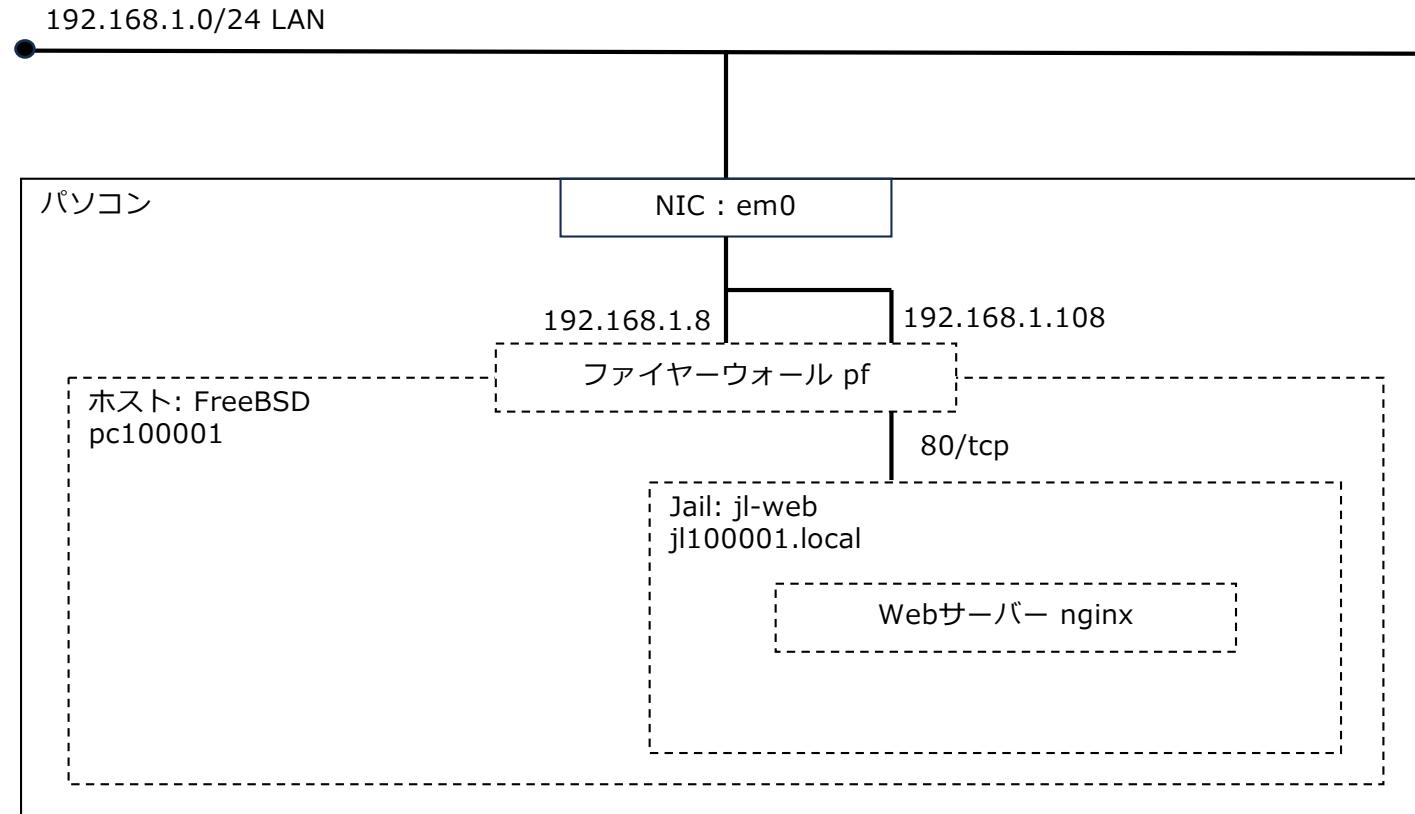
<http://192.168.1.108>

他のパソコンのWebブラウザーから確認する

<http://192.168.1.108>

## 続き | 14-6. Jail (軽量コンテナ) を用いたWebサーバーを構築したい

(メモ書き) 本設定におけるシステム・ネットワーク構成図



## ■ Jail環境の削除（一部概要のみ説明）

- ホストの /etc/rc.conf に設定した jail\_enable="YES" をコメントアウトまたは削除する。

- 再起動して、pcuserでログインする。

- ディレクトリの削除

```
% su  
Password: *****  
# chflags -R noschg,nouchg /home/jail/j1-web  
# rm -rf /home/jail/j1-web
```

- /etc/jail.conf を削除する。

## 14-7. サービスをすぐに起動したい

本マニュアルでは、サービスの起動方法については「/etc/rc.confの編集」→「再起動」と統一して記載をしていますが、再起動をせずにサービスの起動、およびコマンドラインにて/etc/rc.confを編集することができます。

例：ファイアウォール pf

```
% sudo service pf enable ← /etc/rc.confにpf_enable="YES"を設定
% sudo sysrc pf_rules+=/etc/pf.conf ← /etc/rc.confにpf_rules=/etc/pf.confを設定
% sudo service pf start ← pfサービスを起動する
```

# 15-1. 簡単なデータベースwebアプリを作りたい

今回は例として、「行きたい場所」を管理するデータベースwebアプリを作成します。データベースには「場所」「緯度」「経度」「ステータス」の4項目を1つのレコードとして、追加・修正・削除できるようにwebアプリを構築します。

システム的には、画面（HTML）周りを Jinja2、処理の受付や振り分けを FastAPI、データの保存に SQLite、Webサーバーとして uvicorn を使用しています。

- web・データベース関連ツールのインストール

```
% sudo pkg install -y py311-fastapi py311-uvicorn \
py311-sqlite3 py311-jinja2
```

v (ブイ)

- アプリを保存するディレクトリの作成と移動

```
% mkdir ~/web
% cd ~/web
```

# 続き | 15-1. 簡単なデータベースwebアプリを作りたい

- ・サンプルプログラムをダウンロードする

```
% fetch ¥  
https://raw.githubusercontent.com/moginwc/freebsd143/refs/heads/main/web/index.html  
% fetch ¥  
https://raw.githubusercontent.com/moginwc/freebsd143/refs/heads/main/web/main.py  
% fetch ¥  
https://raw.githubusercontent.com/moginwc/freebsd143/refs/heads/main/web/style.css
```

ファイル	内容
index.html	行きたい場所の表示や、追加・検索・更新・削除のHTMLフォーム。
main.py	HTMLのフォームと連動して、データベース操作（表示・追加・更新・削除・検索）を行う。
style.css	index.htmlのスタイルシート

※詳しくは各ソースコードをご覧ください。

※上記プログラムは生成AIにて大枠を作成しています。

# 続き | 15-1. 簡単なデータベースwebアプリを作りたい

- ・ webサーバーを実行する → コロン

```
% uvicorn main:app --reload --host 0.0.0.0 --port 8080
```

※終了は、CTRL+C

- ・ webブラウザからデータベースwebアプリにアクセスする

```
http://localhost:8080/
```

適当に情報を登録した例

一覧表示

GoogleMap の表示

ダウンロード (全件)

普段ダウンメニュー

追加

検索

修正・削除

世界測地系10進形式

	場所	緯度	経度	状況	操作
地図	A点	35.6650016	139.6963361	未訪問	更新 削除
地図	B点	35.710063	139.8107	未訪問	更新 削除
地図	C点	36.069147	139.627710	未訪問	更新 削除

CSVダウンロード

# 続き | 15-1. 簡単なデータベースwebアプリを作りたい

【補足】他のパソコンからwebアクセスする場合

ファイヤーウォールの設定

```
% sudo vi /etc/pf.conf
```

```
pass in on em0 proto tcp to port { 22, 80, 445, 3389, 8080 } keep state
```

追加する

- ・再起動する

```
% sudo hutdown -r now
```

- ・pcuserでログインして、startxする。

- ・webサーバーを実行する

```
% cd ~/web
```

```
% uvicorn main:app --reload --host 0.0.0.0 --port 8080
```

- ・他のパソコンのwebブラウザからデータベースwebアプリにアクセスする

```
http://192.168.1.8:8080/
```

## 15-2. 駅発車標表示専用パソコンにしたい

電車の発車時刻や行き先を表示するための専用パソコンとして設定します（駅の雰囲気を味わいたい人向けです）。以下、概略を説明します。

### ① FreeBSDのインストール

本 [インストール&活用ガイド] の、

1. インストーラーのダウンロード
2. インストール
3. 初期設定

を実行する。

なお、ディスク設定は「UFS」で行う（12-1. 従来通り、UFSファイルシステムを使用したいを参照）。

## 続き | 15-2. 駅発車標表示専用パソコンにしたい

② パソコンの電源投入後、自動的にログイン・ウインドウシステムが立ち上げるための設定

- 再起動
- pcuser でログイン
- startx でウインドウシステムを立ち上げる。

その後、本 [インストール&活用ガイド] の、  
5-1. 自動的にログインできるようにしたい  
5-2. ログイン後、自動的にウインドウマネージャーを起動したい  
を実行する。

③ サンプルプログラム、時刻表サンプルのダウンロード

```
% mkdir ~/bin
% cd ~/bin
% fetch ¥
https://raw.githubusercontent.com/moginwc/freebsd143/refs/heads/main/bin/timetable.py
% fetch ¥
https://raw.githubusercontent.com/moginwc/freebsd143/refs/heads/main/bin/timetable.csv
```

1366x768モニター専用  
サンプルプログラム

※上記プログラムは生成AIにて大枠を作成しました。なお、あまり動作検証していませんので各自チェックをお願いします。

時刻表サンプル

# 続き | 15-2. 駅発車標表示専用パソコンにしたい

## ④ 発車標プログラムを立ち上げるための設定

コメントアウト

```
% cd  
% vim ~/.xinitrc  
  
:  
#uim-toolbar-gtk3 -1-1 &  
:  
#xclock -update 1 -d -strftime '%Y-%m-%d(%a) %T' -geometry -1+1 -bg ¥#f8f8f7 &  
#xterm -geometry 109x5+1+1 -e tail -f /var/log/messages &  
#xfce4-terminal --geometry=80x24+1+97 &  
:  
xset s off -dpms s noblank      ← 追加 (一定時間操作がないと画面がブラックアウトするのを防ぐ設定)  
python3.11 ~/bin/timetable.py  ← 追加 (プログラムの起動)  
exec fvwm3
```

## ⑤ Tkinterをインストールする

```
% sudo pkg install -y py311-tkinter
```

# 続き | 15-2. 駅発車標表示専用パソコンにしたい

## ⑥ 再起動

```
% sudo shutdown -r now
```

## ⑦ FreeBSD起動後、発車標が画面に表示される

発車1分前に「まもなく電車が来ます」と表示される。  
プログラムは [ESC] キーで終了する。

## ※補足：時刻表の形式について

```
% vim ~/bin/timetable.csv
```

みやま線 こぶし駅 発車案内 ← 1行目は見出し  
ゆの町, 05:45 ← 行き先, 時刻 (HH:MM形式) が時刻通りに並んでいること  
ひら坂, 06:20  
やま辺, 07:05  
:

## 15-3. デジタルサイネージ専用パソコンにしたい

### ■デジタルサイネージ（デジタルフォトフレーム）専用パソコンにしたい

電源を入れると、複数の画像の表示を繰り返すデジタルサイネージ（デジタルフォトフレーム）専用パソコンとして設定します。なお、18:00に自動的に電源が切れるように設定します。以下、概略を説明します

#### ① FreeBSDのインストール

本 [インストール&活用ガイド] の、

1. インストーラーのダウンロード
2. インストール
3. 初期設定

を実行する。

なお、ディスク設定は「UFS」で行う（12-1. 従来通り、UFSファイルシステムを使用したいを参照）。

## 続き | 15-3. デジタルサイネージ専用パソコンにしたい

② パソコンの電源投入後、自動的にログイン・ウインドウシステムが立ち上げるための設定

- shutdown -r now で再起動
- pcuser でログイン
- startx でウインドウシステムを立ち上げる。

その後、本 [インストール&活用ガイド] の、

- 5-1. 自動的にログインできるようにしたい
- 5-2. ログイン後、自動的にウインドウマネージャーを起動したい
- 5-3. 特定のコマンドは、パスワードなしでsudoを実行したい

を実行する。

} 再起動はしないでください

さらに、

- 13-2. 毎日23:00に、自動的にパソコンをシャットダウンしたい  
を参考に、18:00に電源が切れるように設定する。

# 続き | 15-3. デジタルサイネージ専用パソコンにしたい

## ③ 画像データのコピー

```
% mkdir ~/Pictures
```

- ・上記フォルダーに表示したい画像ファイルをコピーする。

## ④ スライドショープログラムのインストール

```
% sudo pkg install -y feh
```

# 続き | 15-3. デジタルサイネージ専用パソコンにしたい

## ⑤ スライドショープログラムを立ち上げるための設定

コメントアウト

```
% vim ~/.xinitrc

:
#uim-toolbar-gtk3 -1-1 &
:
#xclock -update 1 -d -strftime '%Y-%m-%d(%a) %T' -geometry -1+1 -bg ¥#f8f8f7 &
#xterm -geometry 109x5+1+1 -e tail -f /var/log/messages &
#xfce4-terminal --geometry=80x24+1+97 &
:
xset s off -dpms s noblank           ← 追加 (一定時間操作がないと画面がブラックアウトするのを防ぐ設定)
feh -. -Z -F -D 5 -Y ~/Pictures/*.*   ← 追加 (プログラムの起動)
exec fvwm3
```

※オプションについては、8-9. 画面スライドショーをしたい、を参照

## ⑥ 再起動

```
% sudo shutdown -r now
```

## ⑦ FreeBSD起動後、5秒間隔で画像が画面に表示され続ける

プログラムは [ESC] キーで終了する。なお、18:00になると、自動的に電源がOFFになる。

## 続き | 15-3. デジタルサイネージ専用パソコンにしたい

### 【補足】

15-2.および15-3.にて設定を間違えてしまい、コンソールからログインするのが困難な場合は、「7-1. リモートからログイン、ファイル転送したい」を参考に、リモートからSSH接続して設定等を確認してください。

## 15-4. ドット絵を描いて動かしたい

ドット絵でキャラクターを描き、カーソルキーの上下左右を押すことによって、キャラクターが動くプログラムの例を紹介します。

### ① テキストエディタを使って、ドット絵でキャラクターを描く

```
% vim character.xpm
```

※本キャラクターは生成AIにて作成しました

```
/* XPM */
static char *XPM_X[] = {
    /* width height num_colors chars_per_pixel */
    "8 8 5 2",
    ". c None",      /* 透明背景 */
    "s c #00ccff",   /* スライム色（水色） */
    "w c #ffffff",   /* 白目 */
    "b c #000000",   /* 黒目・輪郭 */
    "r c #ff6666",   /* 赤い口 */
    /* pixels */
    ". . s s s s . .",
    ". . s s s s s .",
    "s s w b b w s s",
    "s s s s s s s s",
    "s s s s s s s s",
    "s s r r r s s s",
    ". . s s s s s . .",
    ". . s s s s s . "
}
```

5色を使用する宣言

1ドットを2文字で表す

8x8ドット

色の設定

縦8行（行とドットが対応）

横16文字だが、上記の設定により2文字で1ドットとし、8ドットとしている

## 続き | 15-4. ドット絵を描いて動かしたい

### ② キャラクターの.png化と拡大

8x8ドットでは小さすぎるので、例として3倍に拡大（24x24ドット）

```
% magick character.xpm -scale 300% character.png
```

### ③ ライブラリのインストール

```
% sudo pkg install -y py311-pillow py311-tkinter
```

# 続き | 15-4. ドット絵を描いて動かしたい

## ④ プログラムの作成

```
% vim character.py

import tkinter as tk
from PIL import Image, ImageTk # Pillowライブラリが必要 (pkg install py311-pillow)

class DotCharacterApp:
    def __init__(self, root):
        self.root = root
        self.root.title("ドット絵キャラ移動")

        # Canvasのサイズ
        self.canvas_width = 448
        self.canvas_height = 576

        # Canvas作成
        self.canvas = tk.Canvas(root, width=self.canvas_width, height=self.canvas_height, bg="white")
        self.canvas.pack()

        # PNG画像読み込み (Pillow経由)
        self.image = Image.open("character.png")
        self.tk_image = ImageTk.PhotoImage(self.image)

        # 初期位置
        self.x = 100
        self.y = 100

        # 画像をCanvasに配置
        self.image_id = self.canvas.create_image(self.x, self.y, image=self.tk_image, anchor="nw")

        # キー入力バインド
        self.root.bind("<KeyPress>", self.on_key_press)
```

次ページへ

# 続き | 15-4. ドット絵を描いて動かしたい

```
def on_key_press(self, event):
    # 移動量（必要ならここで変更可能）
    dx, dy = 0, 0
    step = 8 # 移動ステップ（ピクセル単位）

    if event.keysym == "Up":
        dy = -step
    elif event.keysym == "Down":
        dy = step
    elif event.keysym == "Left":
        dx = -step
    elif event.keysym == "Right":
        dx = step

    self.x += dx
    self.y += dy

    # Canvas上で画像の位置を更新
    self.canvas.move(self.image_id, dx, dy)

if __name__ == "__main__":
    root = tk.Tk()
    app = DotCharacterApp(root)
    root.mainloop()
```

前ページより

## ⑤ プログラムの実行

※上記プログラムは生成AIにて作成しました

```
% python3.11 character.py
```

カーソルキーでキャラクター（スライム？）が移動します。

※終了は、画面右上の×

参考：note掲載の「Pythonで作ったMSX風ミニゲーム」などのソースも参考になります。[https://note.com/hirose\\_t/n/n54cf93776b21](https://note.com/hirose_t/n/n54cf93776b21)

## 15-5. オリジナルのビットマップフォントを作りたい

ドット絵みたいな感じで、ビットマップフォント（BDF）を作成する方法を解説します。ここでは例として、8x16ドット（半角）の「A」、および16x16ドット（全角）の「あ」の2文字のみのフォントファイルを作成します。

```
% vim myfont.bdf
```

- ・フォントファイルをテキストファイルで作成する

```
STARTFONT 2.1
FONT -misc-myfont-medium-r-normal--16-160-75-75-c-80-iso10646-1 ← フォント名
SIZE 16 75 75
FONTBOUNDINGBOX 16 16 0 -2
STARTPROPERTIES 2
FONT_ASCENT 14
FONT_DESCENT 2
ENDPROPERTIES
CHARS 2 ← このフォントファイル内のフォント数
```

次のページへ

※ コメントのない箇所は、基本的に16x16の場合はそういうものだと思ってください。詳細については、ChatGPTまで。

# 続き | 15-5. オリジナルのビットマップフォントを作りたい

STARTCHAR A ← STARTCHARの後はコメント（わかりやすく表記）  
ENCODING 65 ← 「A」のU+0041unicodeを10進数で表す  
SWIDTH 500 0  
DWIDTH 8 0 ← セル幅は8px  
BBX 8 16 0 -2 ← 実際のサイズと描画原点のオフセット  
BITMAP  
00  
00  
00  
00  
00  
18  
24  
42 ← 8x16ドットを16進数で表記  
42  
7E  
42  
42  
42  
42  
42  
42  
00  
ENDCHAR

STARTCHAR あ ← STARTCHARの後はコメント  
ENCODING 12354 ← 「あ」 U+3042のunicodeの10進数  
SWIDTH 1000 0  
DWIDTH 16 0 ← セル幅は16px  
BBX 16 16 0 -2 ← 実際のサイズと描画原点のオフセット  
BITMAP  
0000  
0000  
0200  
02E0  
1F00  
0640  
07E0  
06D8 ← 16x16ドットを16進数で表記  
1C84  
1584  
2704  
2604  
2E08  
3270  
0000  
0000  
ENDCHAR  
ENDFONT

## 続き | 15-5. オリジナルのビットマップフォントを作りたい

- ・フォントファイル (.BDF) を、バイナリ形式 (.PCF) に変換する

```
% sudo pkg install -y bdftopcf  
% bdftopcf -o myfont.pcf myfont.bdf
```

- ・システムで使用できるように登録する

```
% mkdir ~/.fonts → すでに存在するかもしれません  
% mv myfont.pcf ~/.fonts  
% cd ~/.fonts  
% sudo pkg install -y mkfontscale  
% mkfontdir → フォント名とフォントファイル名の対応表を作成する  
% xset +fp ~/.fonts → システムのフォント検索対象ディレクトリに追加する  
% xset fp rehash → 上記設定を有効にする
```

- ・確認する

```
% xterm -fn '-misc-myfont-medium-r-normal--16-160-75-75-c-80-iso10646-1'
```

※ほとんどの文字は表示されないが、「A」「あ」は表示される。

## 15-6. 色付き丸ボタンのクリックでプログラムを起動できるようにしたい

ここでは、画面上に色付きの丸ボタンを配置し、マウスでクリックすることにより、プログラムを起動させる簡単なランチャーを作成します。

Tkinterのインストール

```
% sudo pkg install -y py311-tkinter
```

ランチャーの作成

```
% vim ~/bin/color_launcher.py
```

```
import tkinter as tk
import subprocess

# コマンド設定
COMMANDS = {
    'blue': ['xterm', '-e', 'tcsh', '-c', 'echo 青色が押されました.; sleep 3'],
    'red': ['xterm', '-e', 'tcsh', '-c', 'echo 赤色が押されました.; sleep 3'],
    'green': ['xterm', '-e', 'tcsh', '-c', 'echo 緑色が押されました.; sleep 3'],
    'yellow': ['xterm', '-e', 'tcsh', '-c', 'echo 黄色が押されました.; sleep 3'],
}

def run_command(color):
    cmd = COMMANDS[color]
    subprocess.Popen(cmd)

root = tk.Tk()
root.title("カラーランチャー")
root.geometry("1050x300")
```

起動するプログラムを定義する

次ページへ

## 続き | 15-6. 色付き丸ボタンのクリックでプログラムを起動できるようにしたい

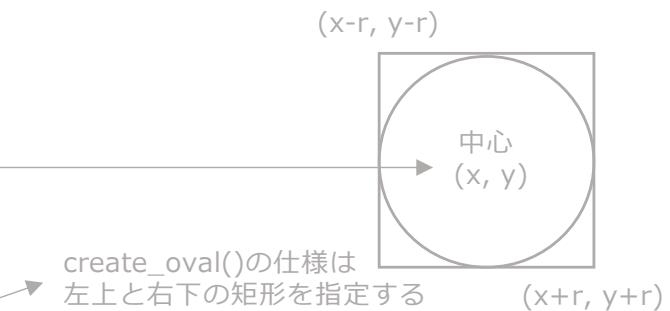
前のページより

```
canvas = tk.Canvas(root, width=1050, height=300)
canvas.pack()

# 円の位置と色
buttons = [
    (150, 150, 'blue'),
    (400, 150, 'red'),
    (650, 150, 'green'),
    (900, 150, 'yellow'),
]

# 円を描画してイベントを設定
for x, y, color in buttons:
    r = 100
    item = canvas.create_oval(x-r, y-r, x+r, y+r, fill=color, outline="")
    canvas.tag_bind(item, "<Button-1>", lambda e, c=color: run_command(c))

root.mainloop()
```



実行

```
% python3.11 ~/bin/color_launcher.py
```

※ プログラムは、[https://raw.githubusercontent.com/moginwc/freebsd143/refs/heads/main/bin/color\\_launcher.py](https://raw.githubusercontent.com/moginwc/freebsd143/refs/heads/main/bin/color_launcher.py) にもあります。

## 15-7. ファイル作成をトリガに二重起動を防いで処理を実行したい

ここでは、特定のファイルが作成されたことを合図にバッチ処理を起動し、かつ二重起動を防止する方法について解説します。

- ・特定のファイルが作成されたかどうかを監視し、バッチ処理を起動するシェルスクリプトの作成

```
% mkdir ~/job
% vim ~/job/watcher.tcsh

#!/bin/tcsh -f

set eojfile = /home/pcuser/job/frontjob.end
set lock    = /home/pcuser/job/mainjob.lock
set script  = /home/pcuser/job/mainjob.tcsh

while (1)
    if ( -f $eojfile ) then
        lockf -s -t 0 $lock $script start
    endif
    sleep 5
end
```

} frontjob.end というファイルが確認できたら、mainjob.tcsh を排他的に実行する。

### 【注意】

lockf は lockf 経由で起動されたプロセスしか二重起動を防ぎません。別の端末から \$script を直接実行することは可能です。よって不用意に実行されないように、引数に "start" がないと起動しないようにしています。（次ページも参照）

本シェルスクリプトは下記にあります。

<https://raw.githubusercontent.com/moginwc/freebsd143/refs/heads/main/job/watcher.tcsh>

## 続き | 15-7. ファイル作成をトリガに二重起動を防いで処理を実行したい

- ・バッチ処理シェルスクリプトの作成

```
% vim ~/job/mainjob.tcsh

#!/bin/tcsh -f

if ("$1" != "start" ) then
    echo "This shell script must not be executed from the command line."
    exit 1
endif

set dt = `date '+%Y-%m-%d %H:%M:%S'`
echo "main job start. ${dt}"

# 実処理
sleep 10

if ($status != 0) then
    # 異常終了
    touch /home/pcuser/job/mainjob.abn
    set dt = `date '+%Y-%m-%d %H:%M:%S'`
    echo "main job abnormal end. ${dt}"
    exit 1
else
    # 前処理終了ファイルを消す
    rm -f /home/pcuser/job/frontjob.end

    # 正常終了
    touch /home/pcuser/job/mainjob.end
    set dt = `date '+%Y-%m-%d %H:%M:%S'`
    echo "main job end. ${dt}"
    exit 0
endif
```

本シェルスクリプトは下記にあります。  
<https://raw.githubusercontent.com/moginwc/freebsd143/refs/heads/main/job/mainjob.tcsh>

引数に"start"がないと実行させないようにしている。

## 続き | 15-7. ファイル作成をトリガに二重起動を防いで処理を実行したい

- 動作確認

```
% chmod +x ~/job/watcher.tcsh  
% chmod +x ~/job/mainjob.tcsh  
  
% ~/job/watcher.tcsh
```

終了はCTRL+C

- 別の端末エミュレータにて

```
% touch ~/job/frontjob.end
```

- すると、mainjob.tcsh が起動します。

## 15-8. PDFファイルに文書番号・ページ番号などを付与したい

あらかじめ、「8-8. 日本語を含む PostScript ファイルを、PDF ファイルに変換したい」を参照し、[Ghostscript の設定が終わっていることを前提](#)とします。

- qpdfのインストール

```
% sudo pkg install -y qpdf
```

### ■ PDFファイルの1ページ目の左上付近に文書番号を挿入する (A4縦サイズを前提)

```
% vim ~/bin/pdf_docno.tcsh
#!/bin/tcsh

set strings = "技26-001" ← 文書番号

# 文書番号のみのPDFファイルを作成する
gs -q -dBATCH -dNOPAUSE -sDEVICE=pdfwrite ¥ ← 出力はpdfファイル
    -sPAPERSIZE=a4 ¥ ← A4縦
    -sOutputFile=docno.pdf - << EOF ← docno.pdfという、文書番号のみが記載されたPDFファイルをいったん作成する

2.834645669 2.834645669 scale %単位をmmにする ← PostScriptのデフォルト単位はインチなので
/IPAGothic-UniJIS-UTF8-H findfont 6.35 scalefont setfont ← わかりやすいようにミリに変更する

10 283 moveto ← 次ページで説明
($strings) show ← 文字列を描く ← フォントの設定（次ページにて追加説明あり）

showpage ← 実際に描画する
EOF

# 元のPDFファイルと、文書番号のPDFファイルを合成する
qpdf in.pdf --overlay docno.pdf -- out.pdf ← 文書番号のみが記載されたPDFファイル docno.pdf と、元のPDFファイルを合成する
```

本シェルスクリプトは下記にあります。

[https://raw.githubusercontent.com/moginwc/freebsd143/refs/heads/main/bin/pdf\\_docno.tcsh](https://raw.githubusercontent.com/moginwc/freebsd143/refs/heads/main/bin/pdf_docno.tcsh)

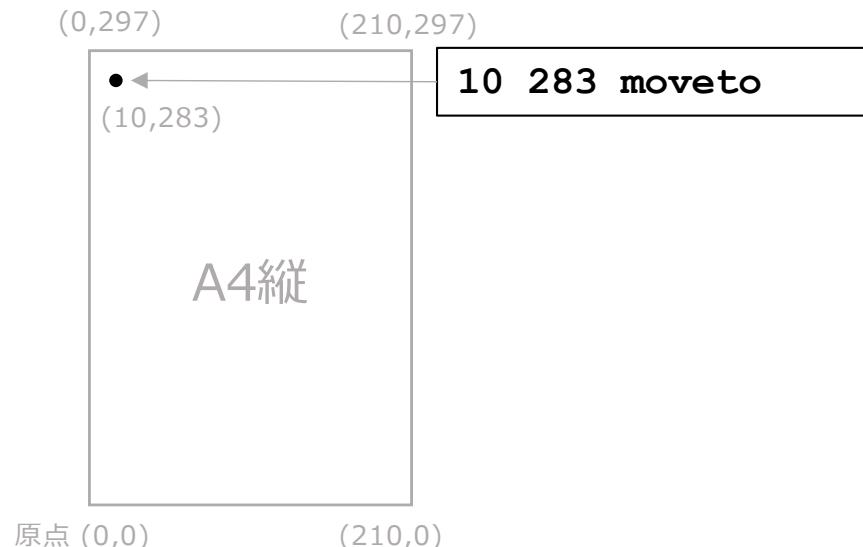
## 続き | 15-8. PDFファイルに文書番号・ページ番号などを付与したい

- 追加説明（ポイントとmmの対応表）

ここでは、18ポイントを設定している

```
/IPAGothic-UniJIS-UTF8-H findfont 6.35 scalefont setfont
```

- 追加説明（PDFファイルの座標：A4縦サイズ）

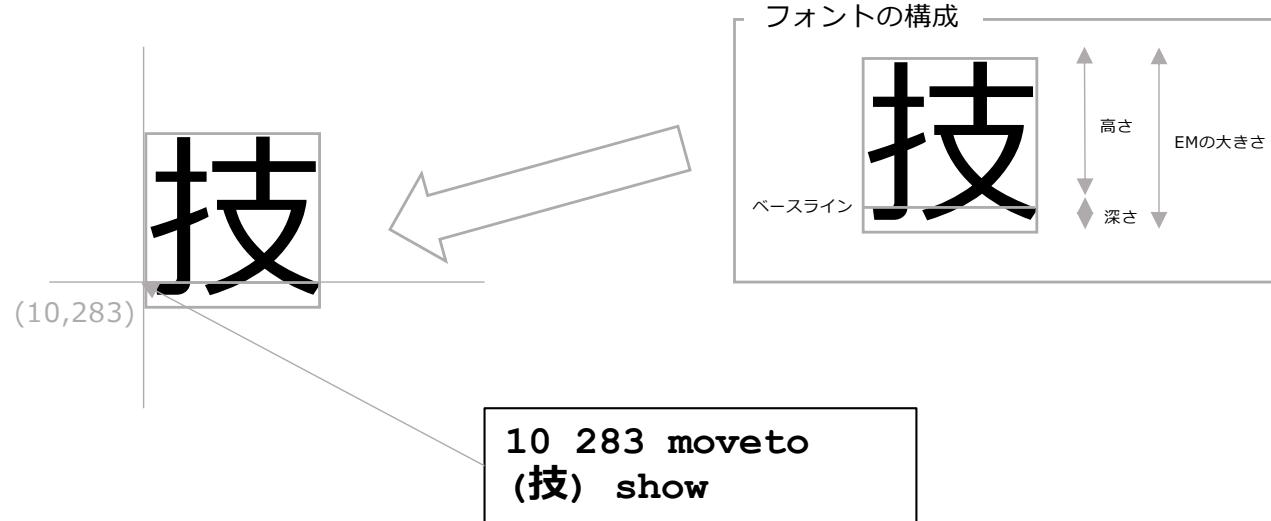


原点は左下となる。

pt	mm
9	3.175
10	3.528
10.5	3.704
12	4.233
14	4.939
16	5.645
18	6.35
20	7.056

## 続き | 15-8. PDFファイルに文書番号・ページ番号などを付与したい

- 追加説明（座標とフォントの関係）



- フォントのベースラインを基準に、movetoされる（フォントの左下が基準ではない）。
- ベースラインの位置はフォントごとに異なる。
- 思い通りに配置するには、Y座標の微妙な調整が必要。

- 実行権限の付与と実行

```
% chmod +x ~/bin/pdf_docno.tcsh
% ~/bin/pdf_docno.tcsh
```

## 続き | 15-8. PDFファイルに文書番号・ページ番号などを付与したい

### ■ PDFファイルの右上付近にページ数を挿入する (A4縦サイズを前提)

```
% vim ~/bin/pdf_pageno.tcsh
#!/bin/tcsh

# 元のPDFファイルの総ページ数を把握する
set total = `qpdf --show-npages in.pdf` ← 総ページ数を取得する

# ページ番号のみのPDFファイルを作成する
gs -q -dBATCH -dNOPAUSE -sDEVICE=pdfwrite \
  -sPAPERSIZE=a4 \
  -sOutputFile=pageno.pdf - << EOF

1 1 $total {
  2.834645669 2.834645669 scale ← 単位をインチからミリに変更する
  /Times-Roman findfont 3.5 scalefont setfont ← 10ポイント程度
  /pageno exch def ← スタックに積まれている for ループの現
  200 287 moveto 在値を pageno に代入する
  pageno 3 string cvs show
  showpage ← 数値から文字列に変換 (convert to string)
} for ← 長さ3の空の文字列
EOF

# 元のPDFファイルと、ページ番号のみのPDFファイルを合成する
qpdf in.pdf --overlay pageno.pdf -- out.pdf ← ページ数のみが記載されたPDFファイル pageno.pdf
                                               と、元のPDFファイルを合成する
```

本シェルスクリプトは下記にあります。

[https://raw.githubusercontent.com/moginwc/freebsd143/refs/heads/main/bin/pdf\\_pageno.tcsh](https://raw.githubusercontent.com/moginwc/freebsd143/refs/heads/main/bin/pdf_pageno.tcsh)

© MOGI NETWORK CENTRE

## 続き | 15-8. PDFファイルに文書番号・ページ番号などを付与したい

- 実行権限の付与と実行

```
% chmod +x ~/bin/pdf_pageno.tcsh  
% ~/bin/pdf_pageno.tcsh
```

- PDFファイルの右上付近に「ページ数 / 総ページ数」（例：1 / 88）を挿入する

前のページの下記の部分を修正します。

```
/pageno exch def  
190 287 moveto ← 座標を若干修正する  
pageno 3 string cvs show  
( / ) show ← 半角スペース+スラッシュ+半角スペース  
$total 3 string cvs show ← 総ページ数  
  
showpage
```

# appendix. 参考文献・資料・謝辞

インストール・設定関連：

1. Solaris（サン・マイクロシステムズ（当時））、SunView・OpenWindows・CDEデスクトップ画面
2. FreeBSD インストール系のwebサイト全般
3. 入力メソッド、日本語入力システム系のwebサイト全般
4. mlterm系のwebサイト全般
5. macOSのUI全般
6. ChatGPT

メモ作成関係：

1. パワーポイントの品質と生産性を向上させるデザイン・テンプレート  
<http://ppt.design4u.jp/template/>

※他にも各種情報を活用させていただきました。ありがとうございます。

謝辞：

FreeBSDをはじめとした数々の素晴らしいソフトウェアの提供、および支えてきた方々に対し、深く感謝します。

# appendix. 残件・未解決問題・気になる点

種別	内容	考察／メモ／暫定対応方法
未解決問題	アクティブウインドウを ALT+PrtSc でキャプチャさせようとし、.fvwm2rc に「key Print A M Exec scrot -ub」と記載したが、動作しない。	key Print A A …は問題なし。 Metaキーの問題なのか？ しかし key F4 A M は動作する。 当方のPC固有の問題か？ ⇒他のPCでは問題なし
未解決問題	Chromiumにおいて、NHKプラスは「操作の実行中にエラーが発生しました。2205000000」で表示されない。（Firefoxは仕様でそもそも視聴できない）	「14-2. 仮想環境を使って、FreeBSD内にLinuxをインストールしたい」を実施し、さらにubuntu内にchromeをインストールすれば、視聴可能。 <u>しかし、この仕組み（bhyve）では音声が出ない。</u>
(済)	Windowsとのデュアルブート環境において、USBマウスのホイールの動きが、過大に動く。	USB端子を抜き差しする。→別のマウスに変更したところ、改善した。
未解決問題	日本語変換効率や半角英数字入力時の利便性を考えると uim-mozc を使用したいが、mozcのユーティリティー類が全く起動しない（Qtライブラリと衝突している模様）。	本メモでは、左記不具合を暫定的に回避して使用できるように記載。

# appendix. 残件・未解決問題・気になる点

種別	内容	考察／メモ／暫定対応方法
未解決問題	xpadの最新版において、個々の付箋の背景色を設定しても、再起動した際に元に戻ってしまう。	xpadのバージョン3を使用する。 (ただし、undo操作が効かない)
未解決問題	fwwmbuttons にて、Action(Mouse 1) と記載したが、リモートデスクトップ経由だとクリックできない。	不明。
未解決問題	YouTube最大化表示時など、uimのツールバーが表示されたままである。	放置
(済)	Firefoxにおいて、webサイト上の「コードをコピー」ができない。Chromiumは問題ない。	Firefoxの問題ではない。 便利な使い方 9-10. を参照
未解決問題	Firefoxにおいて、初回起動時・バージョンアップ時？・異常終了後？に、日本語表示モードが外れてしまう。	不明。
未解決問題	Firefoxにおいて、起動前に一通りの設定（日本語化、ブックマーク登録、トップページ設定など他）をすることができない。（厳密には、システム全体の設定となってしまい、個人設定ができない）	システム全体の設定ではあるものの、ある程度設定可能となった。 8-10. を参照。
未解決問題	Firefoxにおいて、前回のセッションを復元しますか？の表示を防ぎたいが、policies.jsonでは設定できない。	
未解決問題	Chromiumにおいて、YouTube動画再生中に一瞬途切れることが多々ある。CPU負荷がかかっている様子は無し（Firefoxはこのような現象は無し）。	不明。FreeBSD14.2でも、FreeBSD14.3でも解決せず。 他のPCでも解決せず。

# 続き | appendix. 残件・未解決問題・気になる点

種別	内容	考察／メモ／暫定対応方法
(済)	無線LANの設定方法を本メモに記載する。	(記載済み)
(済)	イヤホンの使用方法を本メモに記載する。	(いつの間にか使えるようになっていた)
(済)	コンボジヤック（4極端子）に接続したイヤホンマイクの、MICが機能しない（PC本体の内蔵マイクを使っている感じ）	(いつの間にか使えるようになっていた)
未解決問題	VirtualBoxをインストールしたが、仮想マシンが起動しない。	不明。vboxがロードできていない感じ。
(済み)	ファイルシステムにZFSを採用し、スナップショット・暗号化する方法を本メモに記載する。	(記載済み。なお暗号化はZFSのものではなく、FreeBSD標準のもの)
気になる点	Sansフォント？の数字の「1」の字体。ひと回り大きく感じる。	リモートデスクトップ経由で接続した場合や仮想環境では感じない。レンダリングエンジンの違いか？
気になる点	たまに、pkg install で以前インストールできたパッケージがインストールできないことがある。（本マニュアル作成中にも、Firefox と ImageMagick がインストールできないことがあった）	FirefoxはDVDイメージに含まれているものをインストールした。 → 14-3.過去のパッケージ群を使用したい、も参照のこと。

# 続き | appendix. 残件・未解決問題・気になる点

種別	内容	考察／メモ／暫定対応方法
(済)	GTK2.0系アプリのフォントのアンチエイリアス処理にて、グレースケール方式を指定しても反映されない。	9-14. にて解決
(済)	Macのリモートデスクトップ（Windows App）からFreeBSDにrdp接続した場合、日本語入力のON/OFFができなくなってしまった。	原因不明。調査中。→何もしていないが、解決した。
未解決問題	Mac側で英語キーボードを使い、CapsLockでIMEをON/OFFに設定している場合、FreeBSDからTigetVNC経由で接続してもON/OFFしない。	
(済)	Zoomが利用できるかどうかの確認。	8-19.webカメラを使いたい、にて記載済み。
未解決問題	tint2（タスクバー）が全く動作しない。	不明。
(済)	ファイヤーウォール関係の設定。	本マニュアルにて記載済み。
未解決問題	MacのFinderで書き込んだブルーレイディスク（BD-R DL / UDF 2.01）が読み込みできない。	不明。認識していない気がする。
(済)	GTK系アプリにおいて、日本語とアルファベットの高さが微妙にズれる。	日本語フォントを明示的に指定する。

# 続き | appendix. 残件・未解決問題・気になる点

種別	内容	考察／メモ／暫定対応方法
(済)	プリンター関連の設定。	6-5.を参照
気になる点	Wineのバージョンアップ時、Wine本体のバージョンと、32bit環境のバージョン不一致が発生し、Wineが一切動作しなくなることがある。（過去2度ほど経験）	「FreeBSD Wineあるある」とのこと。対策は、パッケージが整うまではしばらく待つ。
気になる点	あるパッケージをインストールすると、別のパッケージが削除されてしまうことがたまにある。また、旧バージョンのパッケージインストールされることがある。	依存関係？
未解決問題	FreeBSD14系の2025年5月以降のpkg quarterlyのパッケージ群において、uim-ximとmltermの相性問題なのか、mltermにてウインドウの外にカーソルがある状態でキー入力を行うとハングアップする。	この問題はネット上の記事の「mltermが急に激重くなる問題を解決する」の症状に似ている。本マニュアルでは、デフォルトの端末エミュレータをxfce4-terminalとし、問題対策は8-25.にて記載。
未解決問題	Wine10において、Windowsアプリケーションのダイアログボックスを移動すると、ボタン等が反応しなくなる（画面上ではダイアログボックスが移動しても、ボタン等は元の座標にある模様）。この問題は、fvwm3にて発生する。	Windowsアプリは、ダイアログボックスを移動できないように、.fvwm2rcにStyle "* .exe" !DecorateTransient を指定。

# 続き | appendix. 残件・未解決問題・気になる点

種別	内容	考察／メモ／暫定対応方法
残件	本インストール&活用ガイド、肥大化対策（テストが大変、パワポの限界、githubとの非連動による不整合多発）	本インストール&活用ガイドのweb化を検討？⇒他力本願
未解決問題	FreeBSD標準のftpdを使用した場合のファイヤーウォールの設定。	
残件	本インストール&活用ガイド。そもそもインストールが大変すぎるのではないか。	<a href="https://github.com/moginwc/freebsd143">https://github.com/moginwc/freebsd143</a> に「お急ぎの方向け設定」を記載済み。 将来は、ディストリビューション化？⇒他力本願
残件	高DPI環境（超高解像度モニター）の対応	対応しない

# appendix. 改訂履歴

版数	発行日	改訂履歴
第1版	2022年8月24日	初版発行
第2版	2022年11月4日	<ol style="list-style-type: none"><li>mozc辞書ツールやlibreofficeで日本語入力ができないため、日本語入力モジュールをfcitx-mozcから、ibus-mozcに変更した。</li><li>Firefoxのインストールを追加した。</li><li>他、細かい部分を修正した。</li></ol>
第3版	2023年2月8日	<ol style="list-style-type: none"><li>第2版の課題であった、①かな漢字変換のINLINE化、②入力モード変更の手間削減、およびurxvtにおいて、①「※」、「℃」、丸数字が表示されない、②かな漢字変換をINLINEにした場合、行間を空けると入力時の文字のフォントがおかしくなる、③★■などいわゆる曖昧幅の文字が適切に表示されない、などの現象に対応するため、端末工ミュレータをmltermに、入力メソッド・かな漢字変換モジュールをuim-anthyに変更した。</li><li>FreeBSDのバージョンを12.4に変更した。</li><li>起動時のコンソールを削除し、時計を表示させるようにした。</li><li>スクリーンショットツール、画像表示ツールのインストールを追加した。</li><li>他、細かい部分を修正した。</li></ol> <p>【既知問題点】</p> <ol style="list-style-type: none"><li>mltermの行間とフォントサイズの組み合わせによっては、①アンダーバーが表示されないことがある、②かな漢字入力中の文字左上にゴミが表示される。</li><li>アプリによっては、ALT+`入力で、メニューバーに反応してしまうことがある。</li><li>実機環境にて、X Window終了後に画面が元に戻らないことがある。</li></ol>

# 続き | appendix. 改訂履歴

版数	発行日	改訂履歴
第4版	2023年6月19日	<ol style="list-style-type: none"><li>FreeBSDのバージョンを 13.2 に変更した。</li><li>画面ロックツールのインストールを追加した。</li><li>fvwm2を起動すると、日本語キーボードの配列が無効になるのを改善した。</li><li>他、細かい部分を修正した。 【既知問題点】<ol style="list-style-type: none"><li>第3版の既知の問題点は、解決していない。</li><li>実機環境にて、画面ロック解除後にウインドウを選択できなくなる場合がある。~/.fvwm2rc の style * ClickToFocus をコメントアウトすることで対処可能。</li></ol></li></ol>
第5版	2024年4月7日	<ol style="list-style-type: none"><li>FreeBSDのバージョンを 14.0 に変更した。</li><li>[CapsLock]キーで、かな漢字変換をON/OFFできるように変更した。</li><li>ルートメニューに「パフォーマンスマネージャ...」を追加し、topコマンドを実行するようにした。</li><li>ルートメニューのスクリーンショットのオプションを変更した。</li><li>グラフィックドライバーのインストール説明を追加した。（第3版の問題点3を解決）</li><li>本説明書に「便利な使い方」を追加した。他、細かい部分を修正した。 【既知問題点】<ol style="list-style-type: none"><li>[CapsLock]キーで、かな漢字変換をON/OFFできるようにしたが、アプリやアプリの画面状態によっては効かず、本来のCapsLockが機能してしまうことがある。</li><li>USBマウスのホイールの動きが、過大に動くことがある。USB端子の抜き差しで改善される。（Windowsとのデュアルブート環境だと問題が発生する？）</li></ol></li></ol>

# 続き | appendix. 改訂履歴

版数	発行日	改訂履歴
第6版	2024年5月8日	<ol style="list-style-type: none"><li>1. ntpdを起動させるようにした。</li><li>2. [CapsLock]キーで日本語入力のON/OFFを切り替える設定を、別の方に切り替えた（第5版の既知問題点1に対応）。</li><li>3. タイトルバー、ルートメニューなどのフォントを変更した。</li><li>4. タイトルバーボタンのデザイン変更と、閉じるボタンを新規に追加した（fvwm3を参考にした）。</li><li>5. 便利な使い方を充実させた。</li><li>6. 他、細かい部分を修正した。</li></ol>
第7版	2024年8月20日	<ol style="list-style-type: none"><li>1. FreeBSDのバージョンを 14.1 に変更した。</li><li>2. 第6版において、FirefoxやChromiumなどを複数起動させ、アイコン化→復元すると、fvwm2が異常終了する現象が確認できたので、ウィンドウマネージャを fvwm3 へ変更した。</li><li>3. 上記に伴い、~/.fvwm2rc の記載内容を変更した。</li><li>4. xpad（付箋）、lupe（拡大鏡）をインストールするようにした。</li><li>5. mlterm を132桁x24行でも開けるようにした。</li><li>6. 便利な使い方を充実させた。</li><li>7. 他、細かい部分を修正した。</li></ol>

# 続き | appendix. 改訂履歴

版数	発行日	改訂履歴
第8版	2024年12月8日	<ol style="list-style-type: none"><li>FreeBSDのバージョンを 14.2 に変更した。</li><li>インストール時にはIPv6は設定しないように変更した。</li><li>インテルのグラフィックドライバーは、5.10のものを利用するようにした。</li><li>便利な使い方を大幅に充実させた。</li><li>他、細かい部分を修正した。</li></ol> <p>【早期に修正していただきたい既知問題点】</p> <ul style="list-style-type: none"><li>uim-mozcを使用すると、Qt系アプリが動作不良を起こす（起動しない、セグメントエラーなど）。</li><li>wine + uim-anthy（またはmozc）において、日本語入力変換中、変換候補がウインドウ左上に表示されてしまう。</li></ul>
第9版	2025年5月21日	<ol style="list-style-type: none"><li>FreeBSD14.2リリース当時のパッケージ群を固定して使うように変更した（安定性確保のため）。</li><li>fwmbuttonsのランチャーの機能を、プログラムが動作しているときはフォーカス、動作していないときは起動するように変更した。mltermのみ、右クリックで常に新規起動するようにした。</li><li>ディスクは暗号化するようにした。</li><li>便利な使い方を大幅に充実させた。</li><li>他、細かい部分を修正した。</li></ol>

# 続き | appendix. 改訂履歴

版数	発行日	改訂履歴
第10版	2025年10月8日	<ol style="list-style-type: none"><li>引き続き、（サポートは終了しているが）安定しているFreeBSD14.2、およびリリース当時のパッケージ群を使用した。</li><li>fvwmbuttonsのランチャーに、右クリック時の動作を追加した。</li><li>fvwm3のフォントサイズ、ランチャーのアイコンサイズを少し小さくした。</li><li>ファイヤーウォールの設定を追加した。</li><li>sudoコマンドについて、記載内容に誤りがあったので修正した。</li><li>便利な使い方を充実させた。</li><li>他、細かい部分を修正した。</li></ol>
第11版	2026年2月4日	<ol style="list-style-type: none"><li>FreeBSDのバージョンを 14.3 に変更した。</li><li>AMDのグラフィックドライバーを使う場合について、メモ書きを追加した。</li><li>端末エミュレータを、mlterm から xfce4-terminal に変更した。これに伴い、関連箇所を修正した（修正できない部分はmltermのままとした）。</li><li>Wine10.0化に伴い、fvwm3との組み合わせでウインドウ周りに不具合が多発するので、①ダイアログボックス表示時にはウインドウを移動させない設定に、②VB6系のアプリは、仮想デスクトップ内で動作させるように説明書きした。</li><li>本マニュアル内（特にコマンドのオプション）で、半角ハイフンマイナスが大量にENダッシュに置き換わっていたのを修正した。</li><li>fvwm3起動時にコンソール（厳密には /var/log/messages ）を表示させるようにした。</li><li>音声出力トラブル関連の対策を追記した。</li><li>便利な使い方を充実させた。</li><li>他、細かい部分を修正した。</li></ol>

お疲れ様でした。