

# AI専用PCの Liveインストールイメージの作成

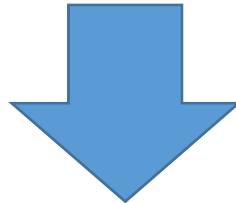
20 千葉 桂大

指導教員 飯坂 ちひろ

# 研究の背景

---

- AI専用PCの問題点
  - メンテナンスをする人がいない
  - リカバリメディアが存在しない



解決する手段として、

Debianを用いたLiveインストールイメージの作成を検討

# Liveインストールイメージとは

---

- Liveインストールイメージは、OSをインストールする前にUSBメモリやDVDからシステムを試すことが出来るものである。今回作成するイメージは、付属のインストーラーを使ってそのままインストールできるようにする。



# なぜDebianを選んだか

---

- ・様々なディストリビューションがある中で、Debianは安定性において非常に高い評価を受けている。
- ・また、授業で扱うディストリビューションもDebian系のため、他のディストリビューションに比べて扱いやすいと考え、Debianを使用することにした。



ディストリビューションの一例

# 他のディストリビューションとの比較

|           | Debian      | RedHat Enterprise Linux  | Arch Linux |
|-----------|-------------|--------------------------|------------|
| 安定性       | 高い          | 高い                       | 低い         |
| パッケージの新しさ | 古い          | 古い                       | 新しい        |
| リリースモデル   | 固定リリース      | 固定リリース                   | ローリングリリース  |
| パッケージ管理   | DEB (apt)   | RPM (yum, dnf)           | Pacman     |
| 主なターゲット   | サーバー・デスクトップ | サーバー                     | デスクトップ     |
| 情報の多さ     | とても多い       | まあまあ多い                   | まあまあ多い     |
| 価格        | 無料          | 有料(最安で<br>1年/¥125,700 ~) | 無料         |

# Debianの特徴

---

- ・安定性に重きを置いている分、パッケージのバージョンが他のディストリビューションよりも古いことが多い  
→ 繼続的なメンテナンスが期待できないため、新しいパッケージを利用するよりも、安定性を取るべきだと考えた
- ・また、Dockerを用いることで間接的に新しいバージョンのパッケージを利用することが出来る

# 公式のDebianとの差異

---

- Nvidia製のGPUをLinux上で動作させる
  - 煩雑な設定が必要
- もしシステムに問題が発生した場合でも…
  - 再インストールが容易
  - 初期設定の手間も減らすことができる



# ライブイメージ開発用PCの環境

|       |                   |
|-------|-------------------|
| 機種名   | PC-VK540          |
| CPU   | AMD Ryzen 5 5500U |
| OS    | Debian 12         |
| 使用ツール | live-build        |

# AI専用PCの構成

|       |                        |
|-------|------------------------|
| OS    | Ubuntu 18.04 / 24.04   |
| CPU   | Intel Xeon W-2223      |
| GPU   | NVIDIA GeForce RTX3090 |
| メモリ   | Kingston 8GBx4 (32GB)  |
| ストレージ | SSD 1TB x 2            |

# 環境構築

---

- ・当初はWindowsのWSL上にDebianをインストールする予定  
→ ライブイメージ開発用PCに直接インストールするように変更
- ・これによってファイルアクセス速度が向上、より作業が快適に
- ・また、インストールしたDebianに今回使用するパッケージ（live-build）をインストールする。

# 動作確認

---

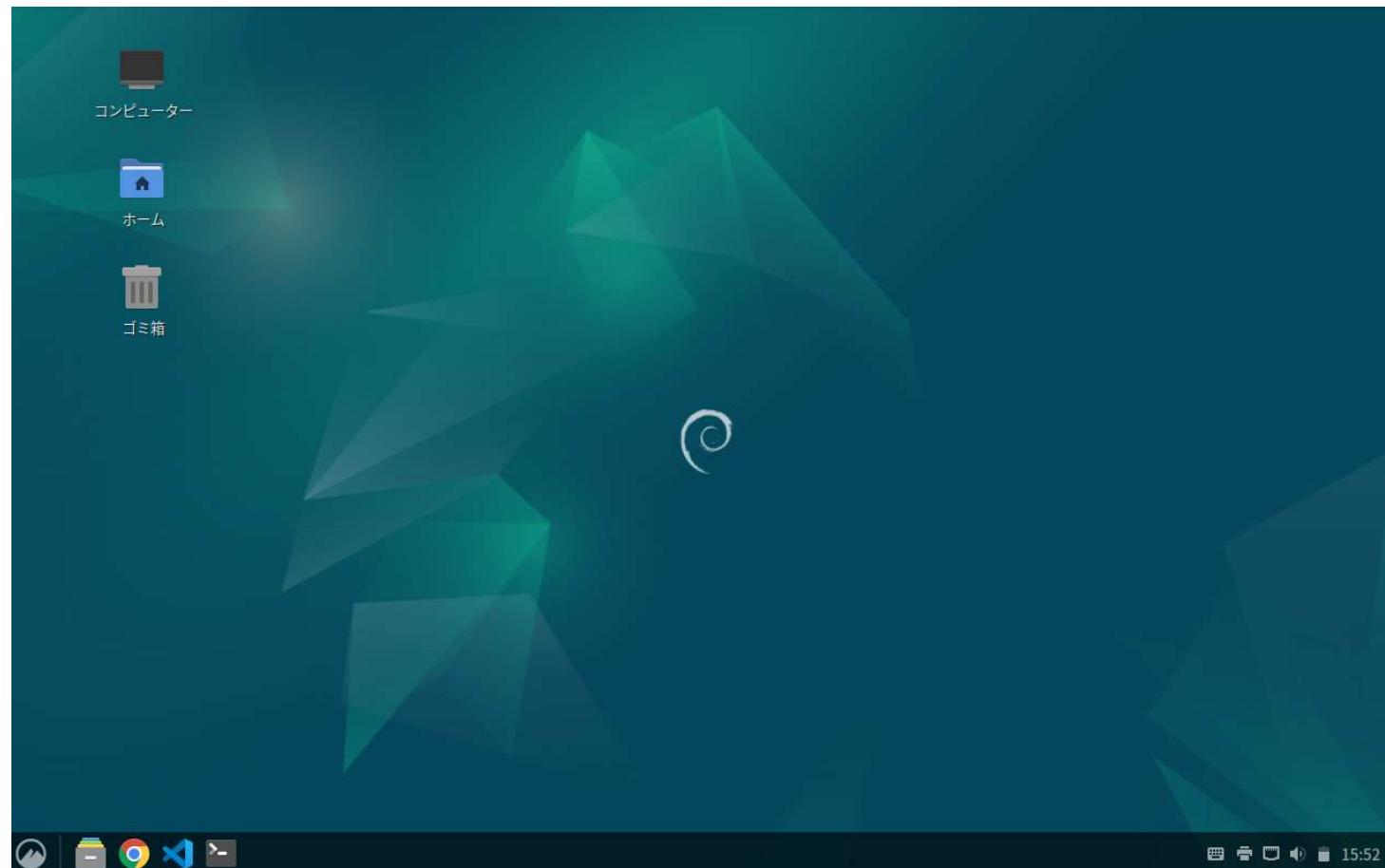
- ・ハードウェアに起因しない箇所は仮想環境上で動作確認を行う
  - 動作確認をすぐに行うことができる
- ・また、仮想環境での動作確認でインストールが止まってしまう不具合が発生した

# 不具合の修正

---

- ・動作確認で発生した不具合を修正する。  
→ 不具合の元となっていたパッケージを特定し、原因を修正した
- ・その後、AI専用PC上での動作確認として、最低限GUI環境に必要なパッケージをインストールしたイメージを作成、動作確認を行った

# デスクトップのイメージ



# パッケージの追加

---

- ・元々の環境に入っていたパッケージを参考に、その他のパッケージを追加した
  - 出来るだけDebian公式のリポジトリから追加
- ・元々の環境にインストールされていて、VSCodeやGoogle ChromeなどのDebian公式のリポジトリに存在しないソフトウェア
  - 各ソフトウェア公式のリポジトリを登録し、インストール

# パッケージの追加

---

- DockerはDebian公式のリポジトリにも存在するが、後の動作確認でバージョンが古いことによる不都合が発生した  
→ VSCodeやGoogle Chromeと同じようにソフトウェア公式のリポジトリを追加してインストールする
- Nvidiaのドライバは、致命的な不具合の原因になりかねないため、最も安定しているDebian公式のリポジトリからインストールする

# その他の設定

---

- ・ブラウザのログイン情報の保存を無効化
  - 不特定多数の利用者がいる中で、プライバシーを強化
- ・シェルの自動補完などの設定を一部変更
  - CLIの操作やDockerの操作をより便利に

# アプリの作成

---

- 学内ストレージ（Pドライブ・Sドライブ）への接続をGUIで行えるアプリを作成
- AI専用PCから手軽にPドライブ・Sドライブに接続できる



# アプリ開発用PCの環境

---

|        |                               |
|--------|-------------------------------|
| 言語     | GUI : Python<br>処理 : シェルスクリプト |
| エディタ   | Vim                           |
| ライブラリ  | python3-gi<br>gir1.2-gtk3.0   |
| GUIビルダ | Glade Interface Designer      |

# アプリの作成

---

- ・言語はGUI部分にPython、処理部分にシェルスクリプトを用いている。  
→ 当初はRustを使用していたが、バージョンの互換性の問題でPythonに変更した
- ・GUI部分にはGTKを使用している。各ライブラリはPythonでGTKを扱うために必要なものである。

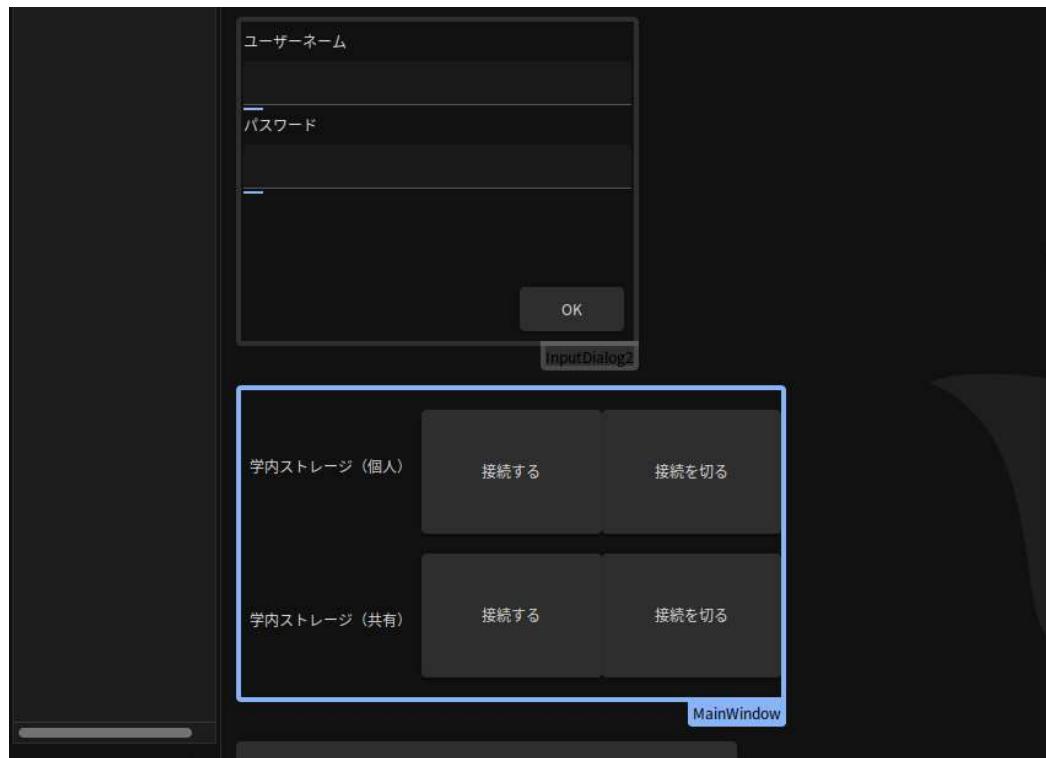
# アプリの作成

---

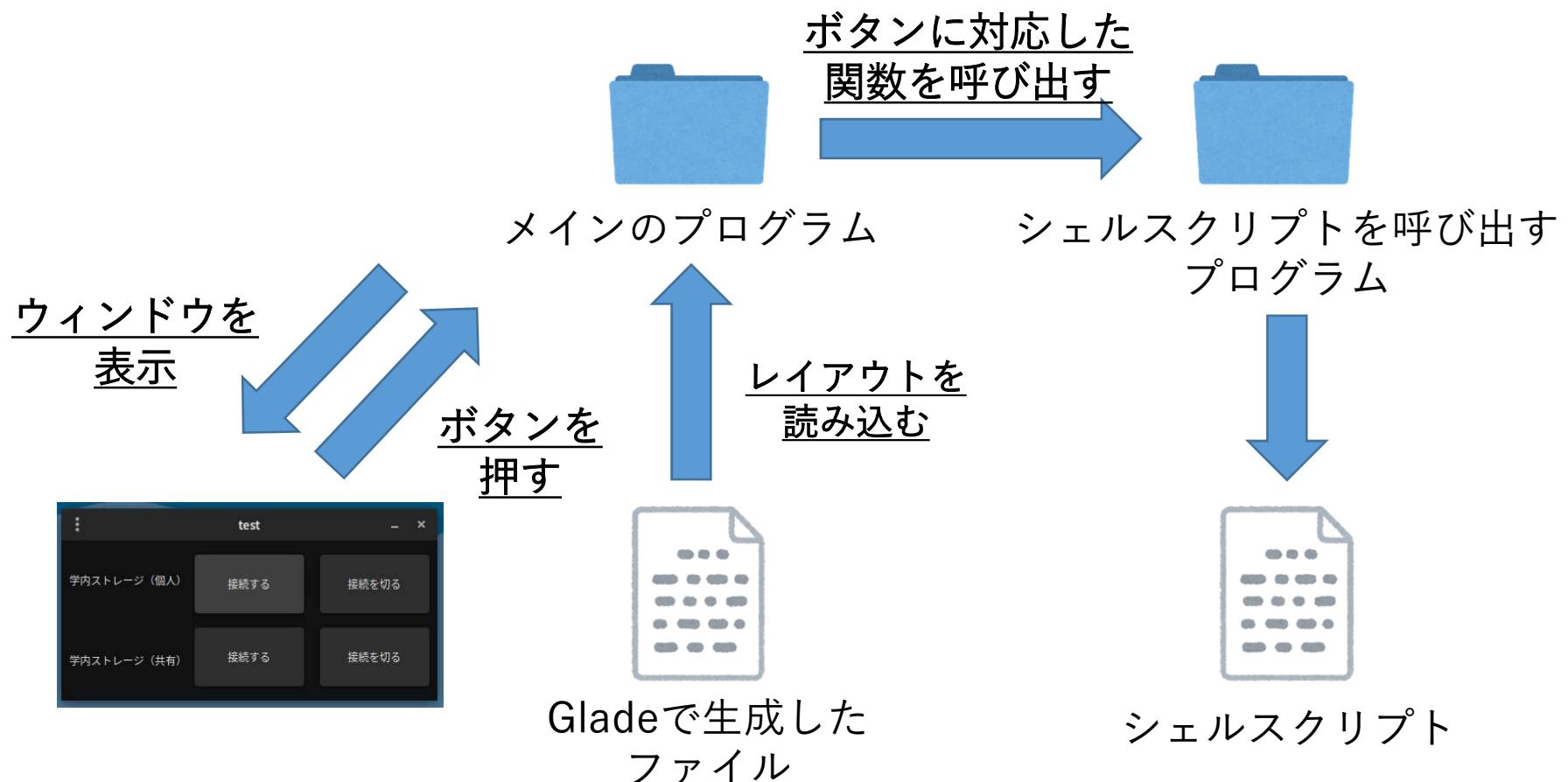
- GUIビルダーのGlade Interface Designer ( Glade )は、GTKを用いたアプリケーションのGUIを視覚的に定義することが出来るアプリケーションである。
- 定義したレイアウトをXMLファイルとして生成し、GUI表示用のプログラムから読み込む。

# アプリの作成

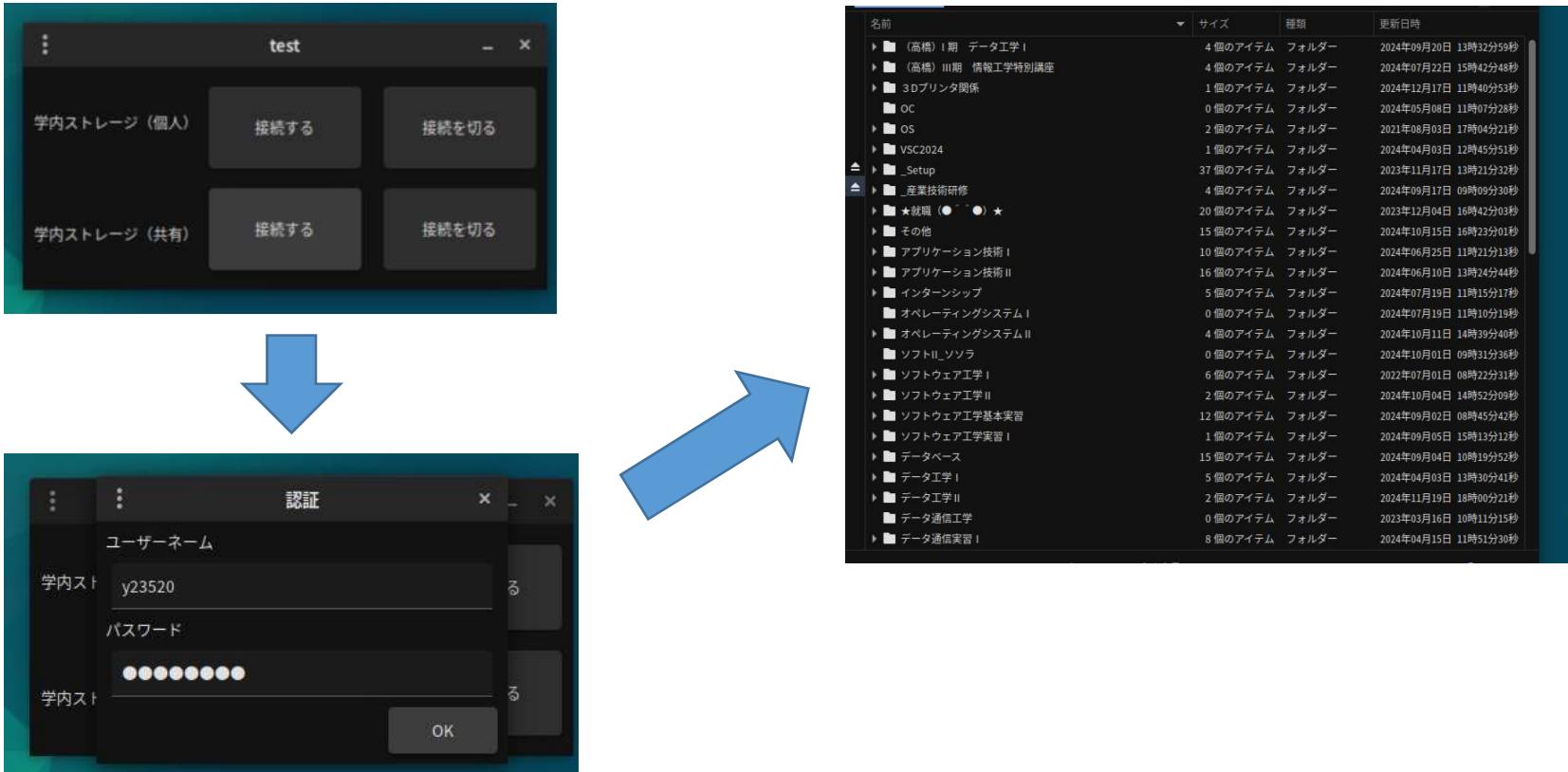
## Gladeの編集画面



# アプリの構造



# アプリの構造



# インストール後の動作確認

---

- ・ここまでの大体の部分が完成したため、実際にAI専用PCにインストールして動作を確認  
→ ライブ環境では正常に動作
- ・しかし、インストール後にGUI環境が起動しない不具合が発生



# 不具合の修正

---

- GUI環境を描画するソフトウェア（X11）のログを確認  
→ Nvidiaのドライバの読み込みでエラーが発生していた
- そのため、GUI環境を描画するソフトウェアがモジュールを読み込む先に、Nvidia ドライバのシンボリックリンクを作成  
→ GUI環境が起動するようになった

# おわりに

---

- ・多少不具合等がありながらも、当初の予定であった
  - ・設定をせずともNvidia製のグラフィックボードが動作する
  - ・再インストールが容易
- を満たすものを作成することが出来た。
- ・今回の成果が、将来的にAI専用PCを用いた研究に貢献することを期待したい。