🛠️ Dokumentation af UI-applikation

(Koncept indtil videre)

## Ikke teknisk beskrivelse

Dette system er en automatiseret måde at bygge og pakke et lille program på, kaldet **minimal-ui**. Det bruger en såkaldt *Makefile*, som er en slags opskrift, der fortæller computeren, hvordan den skal samle programmet og dets nødvendige dele.

Dette system gør det nemt for både udviklere og brugere at bygge, pakke og installere et program uden at skulle dykke ned i komplekse tekniske detaljer. Det sikrer, at programmet fungerer korrekt og er let at dele og installere på forskellige computere.

Denne løsning er designet til at gøre det nemt at udvikle og teste en lille, selvstændig brugerflade (kaldet "minimal-ui") direkte i din webbrowser. Det betyder, at du kan arbejde med og afprøve din applikation lokalt, uden at skulle installere den på en maskine/robot.

Når du er tilfreds med, hvordan det hele ser ud og fungerer i din browser, kan du pakke det hele sammen til en komplet installationspakke. Denne pakke indeholder alt, hvad der er nødvendigt for at køre applikationen på en hvilken som helst computer, uden behov for yderligere installationer eller internetforbindelse.

# Teknisk:

Makefile i koden er designet til at bygge og installere en minimal grafisk brugerflade (UI) applikation ved hjælp af C-programmering og biblioteker som GTK og WebKitGTK. Formålet er at skabe en lille, selvstændig applikation, der kan køre på vores 32-bit Linux-system uden behov for yderligere afhængigheder.

## Struktur og funktioner

Makefilen indeholder følgende hovedmål (targets):

* **all**: Bygger applikationen.
* **minimal-ui**: Kompilerer kildekoden til en eksekverbar fil.
* **bundle**: Samler applikationen og dens afhængigheder i en distributionsmappe.
* **installer**: Opretter et selvudpakkende installationsscript.
* **clean**: Rydder op ved at fjerne bygget filer og mapper.

## Hvordan det virker

### 1. Bygning af applikationen

Når du kører:

”make”

vil Makefile bruge gcc til at kompilere main.c med de nødvendige flags og biblioteker, der er angivet i CFLAGS og LDLIBS.

### 2. Samling af afhængigheder

Målet bundle opretter en mappe kaldet dist, hvor applikationen og dens afhængigheder kopieres. Dette inkluderer dynamiske biblioteker og nødvendige loader-moduler. Unødvendige biblioteker som lyd- og videobiblioteker fjernes for at reducere størrelsen.

### 3. Oprettelse af installationspakke

Målet installer pakker indholdet af dist-mappen i en .tar.xz-fil og genererer et installationsscript (install.sh). Dette script udpakker applikationen til brugerens hjemmemappe.

### 4. Rydning af bygget filer

Målet clean fjerner de genererede filer og mapper, så du kan starte forfra med en ren opbygning.