

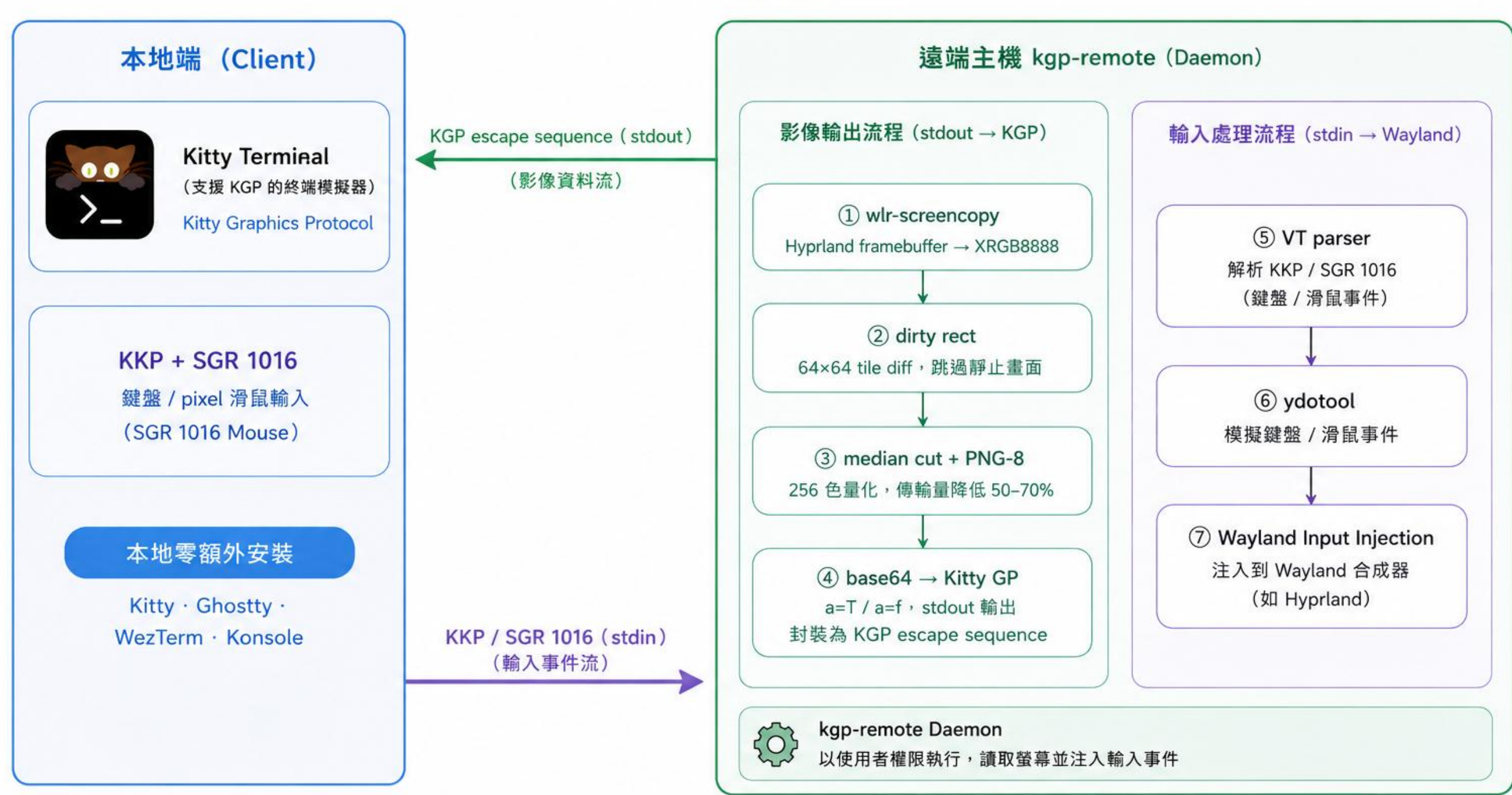
# 基於終端模擬器圖形協議的遠端桌面串流系統

指導教授：李信杰  
專題成員：曾靖棋

## 研究動機與目的

現有遠端桌面方案（VNC、RDP）皆需要在本地端安裝專用客戶端軟體，且對 Wayland 環境的支援程度不一。Kitty Graphics Protocol 是由 Kitty terminal 開發、現已被多個主流終端模擬器採納的圖形顯示標準，其設計允許在終端機視窗內繪製任意解析度的點陣圖圖片。本系統利用此特性，將遠端 Wayland 桌面的畫面透過 SSH 通道以 KGP 格式傳輸至本地終端機繪製，實現零客戶端安裝的遠端桌面體驗。

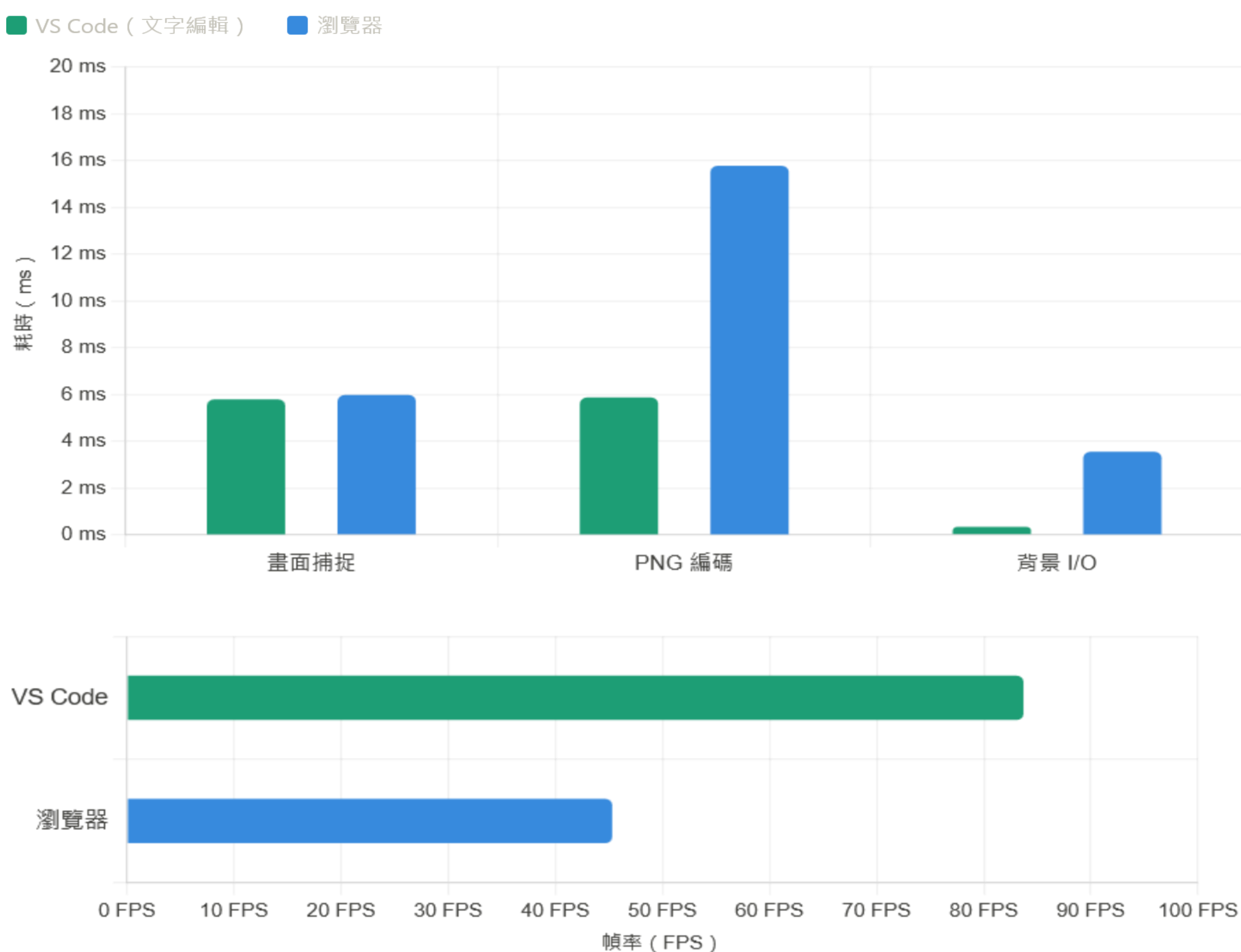
系統架構圖



圖一：系統架構圖

## 效能表現

系統在兩種情境下進行效能量測（圖三）。VS Code 文字編輯場景畫面變動極小，平均每幀處理時間降至約 12 ms，達到 83.7 FPS 且無影格丟失。瀏覽器操作場景因動態內容較多，dirty rect 面積較大，平均影格處理時間約 22 ms，影格損失率約 5%，達到 45.3 FPS。



圖三：效能表現



GitHub原始碼

## 核心技術與系統架構

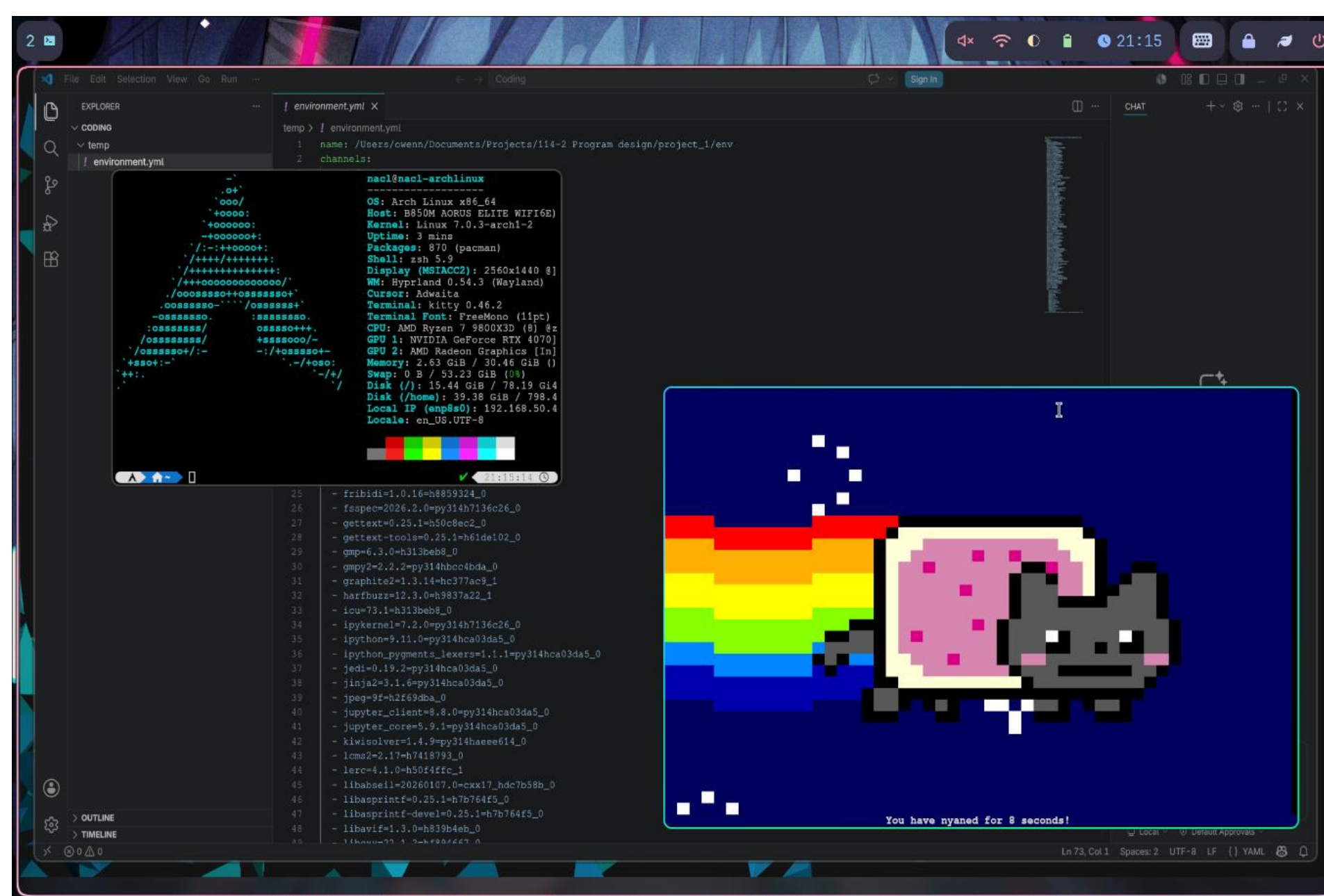
系統架構如圖一所示，主要分為四個部分實現。

**畫面捕捉：**透過 wlr-screencopy Wayland protocol 捕捉 Hyprland compositor 的 framebuffer，取得 XRGB8888 格式的原始像素資料。

**差異偵測：**將畫面切割為 64 x 64 像素的 tile 進行逐行比對，取出所有變動的 bounding box，僅傳輸有變動的畫面區域，降低靜態桌面的傳輸量。

**圖像編碼：**透過 libpng 編碼為 PNG 格式並使用 256 色調色盤（PNG-8）；相較 PNG-24，傳輸資料量降低約 50-70%。

**輸入注入：**解析本地端 Kitty Keyboard Protocol（KKP）的 escape sequence，將按鍵及滑鼠事件透過 ydotool 注入遠端 Wayland 環境，支援 SGR 1016 pixel-level 滑鼠座標。



圖二：系統執行畫面

## 結論與未來展望

系統驗證了以終端圖形協議實現零客戶端遠端桌面的可行性。靜態內容場景（VS Code 文字編輯）FPS 達 83.7，充分展示 Dirty Rect 差異壓縮的效益。

未來方向：改進動態場景壓縮效率、探索硬體加速編碼路徑（VAAPI / AMF）。