Technical documentation

Lightweight Terminal Application
with QML and Qt framework

Introduction

Terminix is a lightweight terminal emulator designed to manage multiple terminal sessions within a single application window. It is meant to be a standalone app that can be used out-of the box with a GNU/Linux-based operating system.

Scope

Terminix provides the command-line interface and has the ability to run multiple terminal instances and arrange them in a grid-like structure inspired by that of GNOME Terminator.

Target users

Developers who prefer a minimal development environment (me). System administrator who requires multitasking capability.

Installation and preview

For developer setup instruction and requirements, as well as product preview and latest development progress, see: https://github.com/lanphgphm/terminix

References

This product is inspired by GNOME Terminator: https://gnome-terminator.org/

Requirements

Functional Requirements	- **Command-line interface:** users can run commands and see output of the job. - **Multiple sessions**: open multiple terminal sessions within a single Terminix window - **Window tiling**: tile terminal in the window horizontally and vertically - **Terminal resizing**: each terminal session must be resizable - **Keyboard shortcuts**: support keybinding and common keyboard shortcuts
Non-Functional Requirements	- **Usability**: Interface shall be intuitive with minimal training for users familiar with standard terminal - **Compatibility**: compatible with major Linux distributions
Interface Requirements	- **User Interface**: UI is clear (current active terminal visually stands out), minimal (avoid visual overload) - **Operating System Interface**: integrate well with command-line interface of host operating system

Why targeting high POSIX-compliance OS?

Because it's the new standard that most modern systems are moving towards. From man-page of pty.h:

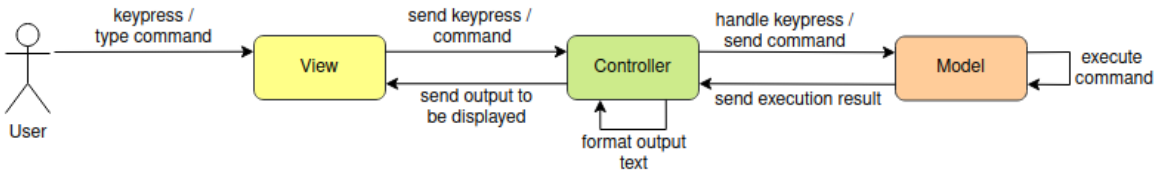
```
Linux  provides  both BSD-style and (standardized) System
V-style pseudoterminals.  System  V-style  terminals  are
commonly called UNIX 98 pseudoterminals on Linux systems.

Since  Linux 2.6.4, BSD-style pseudoterminals are consid-
ered deprecated: support can be  disabled  when  building
the  kernel  by  disabling the CONFIG_LEGACY_PTYS option.
(Starting with Linux 2.6.30, that option is  disabled  by
default in the mainline kernel.)  UNIX 98 pseudoterminals
should be used in new applications.
```

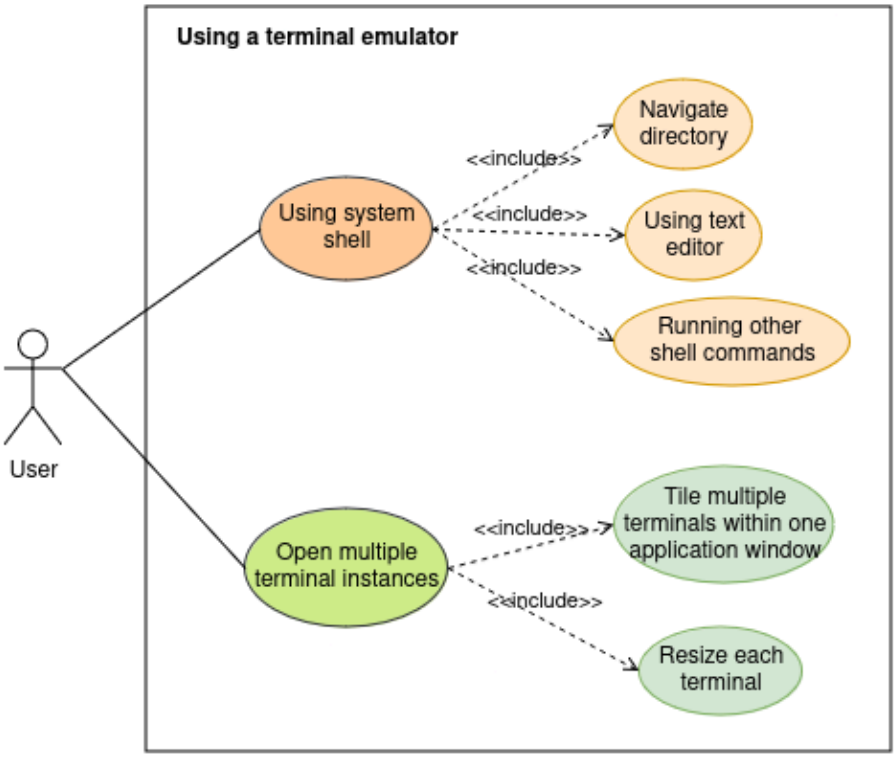
The point where this choice matters is the pty.h module which provides the pseudo-terminal to the terminal emulator. If your system is not high in POSIX compliance but has this module anyways, you can try.

System Architecture

Model-View-Controller design is applied as shown in the class diagram below.



Use Case diagram



Class Diagram