COSC 439: Operating Systems Project

Title: Advanced Customizable Keyboard Shortcut Mapping Device Driver

Objective: The objective of this project is to develop a device driver that enables users to create customizable keyboard shortcuts at the driver level. This driver will allow for advanced key combinations and complex actions, including multi-key sequences and time-based triggers (e.g., holding a key for 2 seconds). The project will include an intuitive user interface for defining, managing, and executing keyboard shortcuts.

Key Features to Implement:

1. **Advanced Key Combinations**: Implement support for multi-key combinations, sequences of key presses, and time-based actions. Examples include holding a key for a specific duration to trigger an action or pressing a series of keys in sequence to execute a command.
2. **Action Mapping**: Create functionality that allows each keyboard shortcut to trigger specific actions. These actions may include opening applications, launching websites, running scripts, or executing custom commands. For example, the combination **Ctrl + Alt + Y** should open a web browser and navigate to YouTube.
3. **Shortcut Input Interface**: Develop an intuitive interface where users can define their own keyboard shortcuts. The interface should include:
    - **Shortcut Input Field**: A field where users can input the key combination (e.g., Ctrl + Alt + Y).
    - **Action Selection**: A dropdown menu or list that allows users to choose from predefined actions or input custom commands.
4. **Shortcut Management**: Provide a method for users to manage their shortcuts:
    - **Display Registered Shortcuts**: List all registered keyboard shortcuts and their corresponding actions.
    - **Edit/Remove Options**: Allow users to edit or remove previously defined shortcuts through the interface.
5. **User-Friendly Interface**: Create an interface with buttons such as **Save** or **Submit**, allowing users to confirm and store their shortcuts. Ensure that the interface is simple and easy to navigate for all users.

Instructions:

1. **Setup the Driver**: Develop a device driver that integrates with the operating system to capture and map keyboard events. This driver will intercept keyboard inputs and associate them with user-defined actions.
2. **Keyboard Shortcut Registration**: Allow users to define new keyboard shortcuts through the interface. Ensure that the driver captures and records these key combinations.
3. **Action Execution**: Upon detecting a registered keyboard shortcut, the driver should execute the corresponding action. This could involve launching applications, opening files, or executing system commands.
4. **Customizable Actions:** Implement predefined actions such as opening browsers, launching applications, or executing system utilities. Additionally, allow users to specify custom commands that can be mapped to shortcuts.
5. **Testing and Validation:** Test the device driver to ensure accurate detection of multi-key combinations, time-based actions, and the proper execution of mapped actions. Ensure no conflicts arise with existing system shortcuts.
6. **Error Handling**: Ensure robust error handling for invalid key combinations or commands. The system should gracefully inform users of conflicts or issues.

Requirements:

1. **Progress Report:** Submit a progress report outlining encountered challenges, how you have solved them, the current status, and forthcoming steps. Upon submission, feedback will be given for project adjustment based on the provided feedback. **(1 pt)**
2. **Code Implementation:** Implement a working device driver that supports the registration and execution of customizable keyboard shortcuts. Ensure that all required features (e.g., multi-key combinations, time-based actions) are functional and integrated. **(6 pts)**
3. **Technical Report: (5 pts)**
    - **Introduction**: Define the project objectives and the significance of customizable keyboard shortcuts in enhancing productivity.
    - **Feature Description**: Provide details on the implementation of each feature, including keyboard event capturing and action mapping.
    - **Implementation Details**: Explain the technical challenges faced, design decisions, and the methods used to solve these challenges.
    - **Results Analysis**: Display examples of shortcuts created, actions triggered, and the outcomes of the tests performed.
    - **Conclusion**: Summarize key findings and the impact of advanced keyboard shortcuts on user efficiency.
4. **Presentation:** In person presentation that focuses on the technical aspects of the project. Utilize PowerPoint slides to highlight project goals, algorithms employed, implementation details, evaluations, challenges encountered, and insights gained. Additionally, ensure the presentation includes a live demonstration of the project to provide a practical illustration of its functionality. **(5 pts)**
5. **Retrospective and Contribution Report:** Reflect on the Operating Systems (OS) course, summarizing significant lessons learned, their practical relevance, and their impact on understanding OS principles. Additionally, list your own contributions as well as those of your teammates towards the project. **(1 pt)**

Deadlines:

- **Progress Report:** November 17, 2025
- **Presentation and Demo:** December 3–8, 2025 (In person)
    - Presentations will take place in the professor's office (YR 456) or in the library (YR 454).
    - Each group will present together, and all group members must be present.
    - The professor may ask questions or request modifications to the project or source code to verify that the work was done by the students and not generated by GenAI or copied from online sources.
    - Time slots will be provided via Calendly, and students should book a slot according to their convenience.
- **Source Code Submission:** December 8, 2025
- **Technical Report:** December 11, 2025
- **Retrospective and Contribution Report:** December 11, 2025