MINI PROJECT ON

**KEYBOARD MACRO UTILITY**

**FINAL PROJECT REPORT**

submitted by

**AKASH AGRAWAL(B15303)**

**RIYANSH GOYAL(B15329)**

**RAJANISH KUMAR UPADHAYAY(B15126)**

**AKASH KATH(B15105)**

for course CS-307

**System Practicum**



**SCHOOL OF COMPUTING AND ELECTRICAL ENGINEERING**

**INDIAN INSTITUTE OF TECHNOLOGY MANDI**

**MAY 2018**

**Introduction**

This project has been developed under the mini-projects alloted as part of the CS-307 (System Practicum) course. The aim of the project was to make a utility by making changes to linux Kernel which helps the user create, edit, view and delete keyboard macros. The objective was to work at the kernel level so that the utility functions across user applications and the user need not to install additional packages or softwares for the same.

In day-to-day use, an average user types multiple phrases, clauses and sentences which are repetitively typed over and over again. This process becomes tedious. What is tedious eventually becomes erroneous. This poses a substantial problem which can be solved using some ingenuity and a fair bit of knowledge about the linux Kernel, especially the device drivers. In order to motivate a solution, consider the following related problem: while using a PC, a number of tasks can be performed by moving the mouse(or touchpad) pointer to a specific location and then performing one or more clicks and movements on the screen. While this process is interactive and feels intuitive, it is rather slow, sometimes time-consuming and occasionally annoying. In order to boost user-productivity, user application designers have come up with keyboard shortcuts which reduce the complex combination of multiple movements and clicks to a few keypresses. Building upon the same idea we have developed keyboard macros which are essential key-combinations which when pressed print on the current application window the text stored by the user corresponding to the typed keys. This helps boost the user productivity by manifolds since the tedious process of typing a hundred keys gets reduced to essentially typing three keys.

Furthermore since the change is at the Kernel level, there’s no extra process running in the background for logging the key presses; using lesser system resources for process maintenance overheads.

**Problem Statement**

Modify the Linux keyboard device driver to implement keyboard macros. Special key combinations start and end recording of a macro. The macros are assigned by some key combination. Write a utility to view, delete, edit and assign macros.

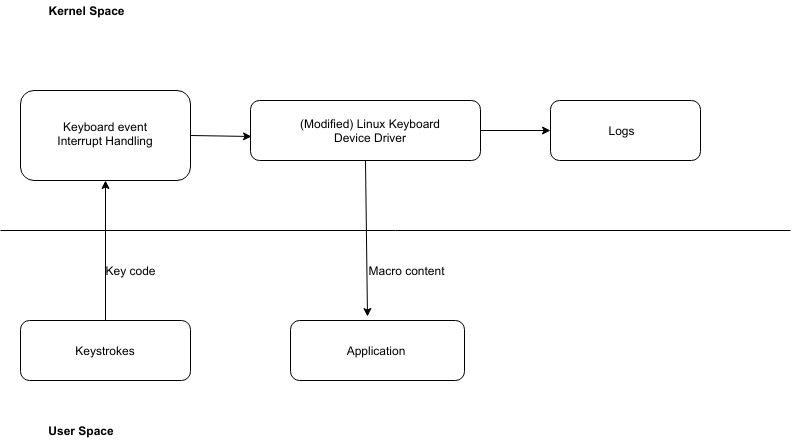
**What is a macro ?**:

A macro is a saved sequence of commands or keyboard strokes that can be stored and then recalled with a single command or keyboard stroke.

**System Design**

We have changed the existing Linux Keyboard Device Drivers rather making a new module.

This change in the linux keyboard device driver has been tested on the Linux Kernel version 4.16.8.



**Fig1. Our Design**

**Limitations:**

The number of macros which can be stored using our utility is limited to 10.

There is no GUI which can tell user whether he is in recording macro mode, editing mode or normal operation mode. User can not automatically see whether a recording, editing or deleting attempt is successful. He has to manually check by displaying the macros.

Currently, the macros aren’t stored on a persistent storage i.e. if the system reboots than the user has to again define all the macros from scratch.

There is a limit to number of characters or keys which can be stored in a macro.

**Conclusion:**

In this project we successfully recorded, edited, deleted and displayed macros using our utility.

The key combinations used to enter recording ,editing, deleting modes are unique and doesn’t interfere with the functionality of any user application.

In the future, we will try to store macros in a persistent storage so that user doesn’t have to re-record macros. Additionally, some measures will be taken so that the user gets to know if he is in recording, editing or deleting mode.

**References:**

[1] <https://www.cyberciti.biz/faq/debian-ubuntu-building-installing-a-custom-linux-kernel/>

[2] Robert Love, Linux Kernel Development (3rd Edition), Addison-Wesley Professional

[3] Jonathan Corbet, Allessandro Rubini & Greg Kroah-Hartman, Linux Device Drivers (3rd Edition), O'Reilly

[4] <https://whatis.techtarget.com/definition/macro>, 27/05/2018

[5]<https://elixir.bootlin.com/linux/v4.16.8/source/drivers/input/keyboard/atkbd.c> , 26/05/2018

[6] <https://www.infradead.org/~mchehab/kernel_docs_pdf/linux-input.pdf>, 20,05,2018