INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS…..

Device Driver Wrapper enables the use of Microsoft Windows drivers for USB Webcams on Linux based systems

TARUN DHAR DIWAN[1], KAMLESH LEHRE [2], SWEETI RAKESH[3]

[1]Dr.C.V.RAMAN UNIVERSITY, BILASPUR, INDIA
DEPTT.OF ENGINEERING
ASSISTANT PROFESSOR
taruncsit@gmail.com

[2]DR.C.V.RAMAN UNIVERSITY, BILASPUR, INDIA,
DEPTT.OF ENGINEERING
ASSISTANT PROFESSOR
lahrekamlesh@gmail.com

[3]DR.C.V.RAMAN UNIVERSITY, BILASPUR, INDIA,
M.TECH SCHOLAR
sweetirakesh@yahool.in

Abstract: *A Device Driver Wrapper is based on software system that functions perform between an operating system and a Device Driver, such as a device driver, that was not derivative for that operating system. It cans enactment the operating system to use system technologies for which no native's implement existing. Our project objective is to 'wrap' Windows based video device drivers so that they can be used as drivers in Linux. Wrapper works by emulating the Windows kernel and APIs, and dynamically linking the driver to this implementation. It implements Windows kernel API and the hardware device specific API within Linux kernel. A Windows driver for the device is then attaching to this implementation so that the driver runs natively, as though it is in Windows, without binary emulation. A device driver wrapper consists of emulation code of the framework for which the device driver was written for. When this driver wrapper is integrated into the kernel of the operating system which lacks support for framework, it enables the driver to work seamlessly in that operating system. Thus the driver functions as though it is working in the native environment. Wrapper works by implementing the native operating system APIs, and dynamically linking the driver to this implementation.*

Keywords: *Driver Wrapper, communications subsystem, Linux architectures, kernel streaming, Wavelet Transform.*

1. INTRODUCTION

A Device Driver Wrapper strategy for software testing integrates functions as an adapter between an operating system and a driver software test cases design methods into well planned series of steps that result in the successful construction of software .This strategy provides a road map that describe the steps to be conducted as part testing, when these steps are planned and undertaken, and how much efforts, time and resources will be required that was not designed for that operating system. In computing, a device driver or software driver is a computer program allowing higher-level computer programs to interact with a hardware device [1].A driver typically communicates with the device through the computer bus or communications subsystem to which the hardware is connected. When a calling program invokes a routine in the driver, the driver issues commands to the device. Once the device sends data back to the driver, the driver may invoke routines in the original calling program [2]. Drivers are hardware-dependent and operating-system-specific. They usually provide the interrupt handling required for any necessary asynchronous time-dependent hardware interface.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS…..

1.1Principle behind Device Driver Wrapper

As this research covers both Windows and Linux architectures an extensive research of the entities involved in both architectures was done. In the literature survey carried out, a large number of books, papers, journals and project wikis were referred [3]. The major entities that have been discussed are as follows.

Device Driver:

In technology, The Windows Driver Model had a number of limitations, primary being the complexity and difficulty involved in writing a driver. As a result Microsoft introduced the Windows Driver Foundation for writing driver foe Windows 2000 and later versions. WDF functions as an abstraction layer between WDM and the WDF driver that simplifies the task of implementing robust, secure, and efficient drivers.WDF provides a framework that handles the key tasks of a WDM driver: it receives and handles IRPs, manages Plug and Play and power state changes, and so on. The framework calls on the client WDF driver to provide device-specific functionality [4,5]. Lower filter drivers see IRPs that the function driver is trying to send to the bus driver. This schema of layering and processing requests is shown in Fig1

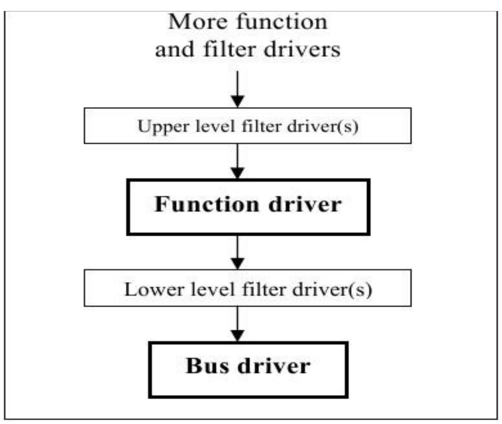

Figure1: This Schema of Layering and Processing Requests.

Drivers are hardware system dependent and operating system specific function. They easily provide the interrupt handling required function for any necessary asynchronous time dependent hardware interface function.

2. LITERATURE SURVEY

Device Driver Wrapper Features of AV Stream system required small driver writers to produce less code, provide a unified kernel streaming classes system for both audio and video mini drivers [6].

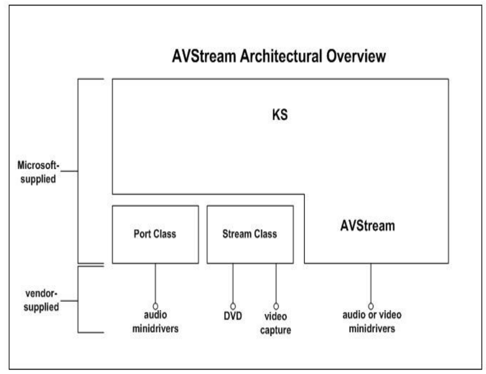

Figure 1: Av stream Architectural Overview

AV Stream manages a construct referred to as an object bag for each AV Stream object visible to the smaller driver. An object bag is a generic container for tacking dynamically allocated memory related with a given aim.

2.1 PURPOSE of RESEARCH:

This research being a Systems dealing with both Windowsand Linux architectures, a detailed plan was necessary for understanding the APIs of both architectures. The complete analysis of all functions within the AV Stream and USBCAMD model in Windows as well as the mapping of those functions within the constraints of V4L API in Linux forms The Windows API being closed source, a few weeks are to be devoted for tracing the system calls from the Windows kernel so as to make sure that the Windows based driver works seamlessly in Linux using V4L.Once the calls have been traced it is then possible to implement emulation functions of those calls [7].

3. EXPERIMENT DESIGN SPECIFICATION

The Wavelet Transform provides a time frequency module respect of the signal. It cans implementation to overcome the short coming of the Short Time Fourier Transform, which can also be used to analyze non-stationary signals. While gives a constant resolution at all frequencies, the Wavelet Transform

ISSN 2320-6802

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS…..

uses dynamically resolution technique by which different frequencies are analyzed the respect of different resolutions.

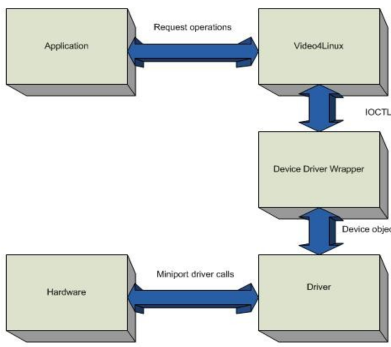

Figure 3: High Level System Architecture Diagram

The application requests data from the device through Video4Linux. Video4Linux attempts to contact the device through the driver, but , as the driver is written according to Windows specification the calls are captured by the device driver wrapper and then are converted into Windows calls and then passed on to the driver [8]. The driver then interacts with the hardware to provide the output to the wrapper which again transforms the output in Linux format to pass it on to the application through Video4Linux.

3.1 Use Case Diagram:

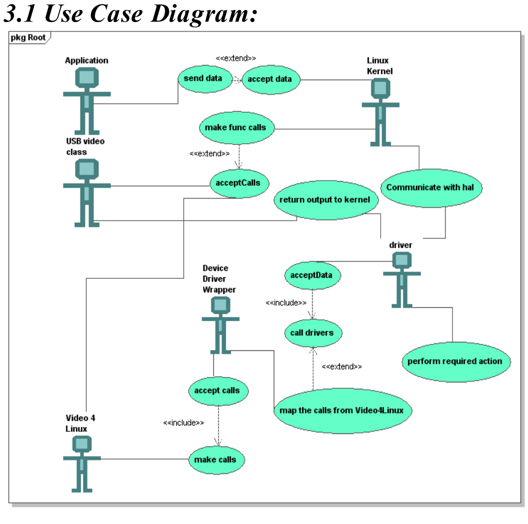

Figure 1: Device driver wrapper Use Case Diagram

USE CASE DESCRIPTION ACTION: [8]

1. Application
   Application requests data from device through V4L.
2. Linux kernel
   The Linux kernel accepts the application data and communicates with either USB Viedo Class or Video4Linux to perform the requested operation.
3. USBVideo Class
   It accepts Linux kernel calls and return output to the kernel from the device directly.
4. Video4Linux
   It accepts Linux kernel calls and communicates with the miniport driver through device driver wrapper.
5. Device driver wrapper
   The calls made by Video4Linux are accepted and mapped to the AVStream or USBCAMD specification and these are sent to the device through the miniport driver.
6. Device Driver
   Data is accepted from device driver wrapper and requested operation is performed through communication with HAL.

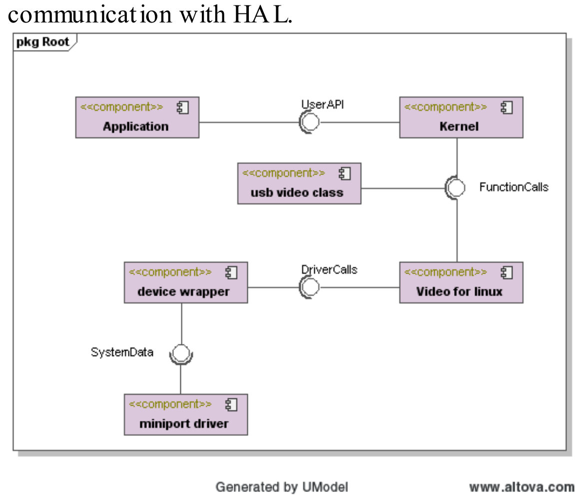

Figure 5: Device Driver Wrapper Component Diagram

COMPONENT DIAGRAM DESCRIPTION: [9]

1. The application program component has to use the API interface provided by the kernel to interact with it.
2. The kernel provides the API interface and the function calls interface so that various modules can interact with it.
3. The UVC and V4L components use the function calls interface to interact with the kernel.
4. The V4L provides a Driver Call interface so that drivers could interact with the kernel.
5. The driver wrapper has to use the device call interface provided by V4L to capture the calls of V4L.

www.ijaret.org Vol. 1, Issue VII, Aug 2013
ISSN 2320-6802

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS…..

The driver wrapper provides the System Data interface so that the miniport driver could use it to perform the requested operation

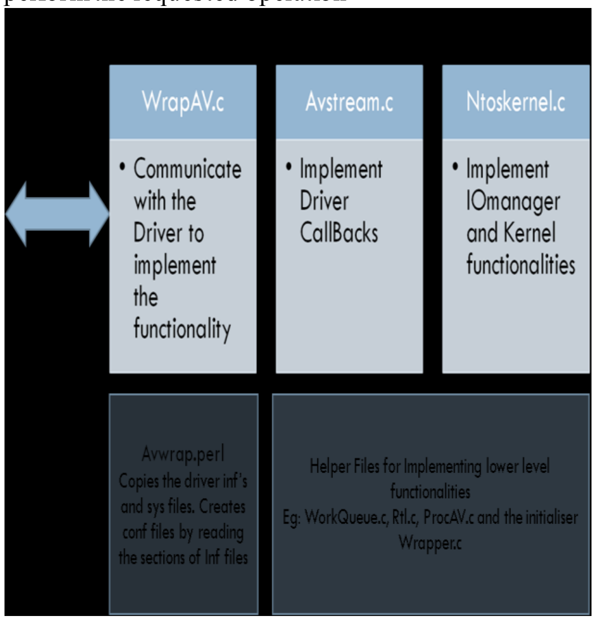

Figure 6: Code Flow Diagram

4. IMPLEMENTATION AND RESULT

Check whether Verification analysis to all the requirements of the client is related to development of the system. There is no execution of code which means that it does not see whether a specific unit of the system works correctly according to the corresponding specified requirement [10]. Instead it only sees whether every part of requirement has a corresponding part in the system or not. Whereas validation analysis checks whether each unit of the system is built according to its respect of requirement or not. It executes the system to see if it's intended behavior tallies with the required intended behavior analysis [11].

.

Test case ID	Test cases	Result
1	Jitters are created	Invalid
2	There is more than one user in front of Webcam	Invalid
3	Environment is minimally illuminated	Valid
4	Images are inverted	Invalid
5	Camera is more than 3 meters	
6	Images are improved	Valid

4.1 TEST CASES [12].

Test case 1

1Use load device command to load device AVwrap.
2Use temporary ioctl file "fuse" at minor number 229.
3Use driver as bml5c that is copied in /etc/AVwrap and provide the vendor id device id and bus number "5".

Result

AVwrap device is created /dev with minor number 229.All the setting wereb copied successfully in load device structure,. ioctl call statement was reached successfully.

Test case2

1. Use Load Driver command to load driver bml5c
2. Provide the vendor driver name and the con file name to be loaded [13].

Result

1. Load _driver structure was filled with all the settings read from con files with detection of the related sys files.
2. The sys file data was mapped to process memory space using mmap and the address was provided in load drive structure.

4.2 Test Name: PERL [14- 15]

Status of Document:

Preliminary

Test purpose:

1. Generation of conf files.
2. Copy of related inf and sys files.

Known issues:

1) Specified sys files not found.
2) Incomplete or corrupt inf file

Preconditions of the test:

1) A windows driver with all the inf files, bin files and sys files are provided.

Test cases:

Test case 1

1) Use AVwrap –i command to install the inf.
2) bml5c Windows driver's path is provided in the argument list.

Result

1.The inf and sys files are copied.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS…..

2.Con files are generated by parsing each and every section of the inf files.

5. CONCLUSION AND FUTURE SCOPE

The purpose of this research was to build an application that would enable Linux users to use the hardware with Windows based drivers seamlessly. In addition, factors like generic Windows kernel simulation, 64 bit compatibility were key points to be taken into account. Our application will also adds one more dimension to existing Linux hardware support, that being, support for all streaming media devices in Linux[16].Studies have shown us that an application like ours can help save a lot of time and effort required in reverse engineering. These observations combined with our research indicate that our application has great scope in the industry. It can greatly help inter-operability between Windows and Linux systems.

Future The AV Stream wrapper can be extended for other streaming media devices with certain extensions other emulation based projects such as WINE and React OS will benefit as the need for driver emulation is eliminated, A generalized wrapper can be made so as to support all windows based drivers on Linux.

APPLICATIONS:

1. The need for reverse engineering will be eliminated [17]

Quality features provided by the specific vendors can be used to the fullest

2. Each video device having a windows based driver can be used with Linux

3. Enhanced streaming media support for Linux

REFERENCES

[1] **Linux Device Drivers, 3rd Edition By Jonathan Courbet,Greg Kroah-Hartman, Alessandro Rubini.**

[2] **Developing Drivers with the Microsof Windows Driver Foundation by Penny Orwick and Guy Smith Microsoft Press 2007**

[3] ***The Linux® Kernel Primer: A Top-Down Approach for x86and PowerPC Architectures*Claudia Salzberg Rodriguez, Gordon Fischer, Steven Smolski**

[4] ***Essential Linux Device Drivers* by Sreekrishnan Venketeshwaran**

[5] ***http:\\www.ndis.sourceforge.net***

[6] ***A comparison of windows and Linux Device driver Architectures* by Melekam Tsegaye and Richard Foss *The Webcam HOW TO* by Howard Shane**

[7] **Kardas-Cinal E, Drozdziel J., Sowinski B.: *Simulation study of a relation between the device drivers coefficient windows and Linux Device driver*, Archives of Transport, vol. 21, issue 1-2, Warsaw 2009.**

[8] **M. K. Bhiwapurkar, V. H. Saran, S. P. Harsha, V. K. Goel, *Influence of Mono-axis Random Vibration on Reading Activity*, Deptt. Of Mechanical & Industrial Engg. Indian Institute of Technology, *Roorkee*, Jun 30, 2010.**

[9] **Hariharan,*Estimation of Ride Index of High Power Diesel Multiple Unit Rail* Coach. *M. Tech* thesis, *IIT Madras*, *India*, 2000.**

[10] **Sriram T. Vengalathur, *Low Cost Fault Detection System for Railcars and Tracks*, University of Bangalore, *India*, Aug 2003.**

[11] **Steven Bleakly and Steven Senini, *Autonomous time Frequency Analysis of Wagon Body Accelerations,* Faculty of Engineering and Physical Systems, *Central Queensland University, Rockhampton, Qld*, *Australia,* 2004.**

[12] **Mirjana Tomicic- Torlakovic, Svetislav Stefanovic, *Railway Noise Vibration-Current European Legislation And Research And Measurements On Serbian Railway,* University of Belgrade, The Faculty of Civil Engineering, *Serbia,* January 2010.**

[13] **C. Madshus, B. Bessason and L. Harvik, *Vibrations From High Speed Railways On Soft Ground,* Norwegian Geotechnical Institute, *Norway,* Nov. 1995.**

[14] **S.K. Mittal, Sandeep Kalra, M.A. Shamshi and B. K. Sharma, *Micro-Controller based Oscillation Monitoring System for the safety of Railway Vehicles with high storage capacity and Real time warning facility,* Central Scientific Instruments Organization, *Chandigarh,* and June 2004.**

[15] **Khairul Anuar Abd Wahid, *Development of Tilt and Vibration Measurement and Detection System using MEMS***

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS…..

Accelerometer as a Sensor, University Sains Malaysia, July 2008.

[16] Simon Wagner, *Derailment Risk Assessment*, James Goldston faculty of Engineering and Physical Systems, Central Queensland University, 2004.

[17] Simon Wagner, *Derailment Risk Assessment*, James Goldston faculty of Engineering and Physical Systems, Central Queensland University, 2004.

AUTHOR BIOGRAPHIES

TARUN DHAR DIWAN RECEIVED HIS MASTER OF ENGINEERING (COMPUTER TECHNOLOGY AND APPLICATION) DEGREE FROM CHHATTISGARH SWAMI VIVEKANANDA TECHNICAL UNIVERSITY –BHILAI, INDIA, AND MASTER OF PHILOSOPHY (GOLD MEDAL LIST) FROM DR. C.V. RAMAN UNIVERSITY. HE IS CURRENTLY AN HOD & MTECH CO-ORDINATOR DEPTT. OF ENGINEERING AT THE DR.C.V.RAMAN UNIVERSITY-BILASPUR, INDIA. HIS CURRENT RESEARCH WORK ARTIFICIAL INTELLIGENT, IMAGE PROCESSING AND SOFTWARE ENGINEER