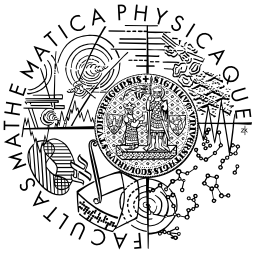

MATEMATICKO-FYZIKÁLNÍ
FAKULTA
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Peter Lakatoš

Analyzátor USB paketů

Katedra distribuovaných a spolehlivých systémů

Vedoucí bakalářské práce: Mgr. Pavel Ježek, Ph.D.
Studijní program: Informatika
Studijní obor: Programování a softwarové systémy

Praha 2021

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne ....

Podpis autora

Poděkování.

Název práce: Analyzátor USB paketů

Autor: Peter Lakatoš

Katedra: Katedra distribuovaných a spolehlivých systémů

Vedoucí bakalářské práce: Mgr. Pavel Ježek, Ph.D., Katedra distribuovaných a spolehlivých systémů

Abstrakt: Abstrakt.

Klíčová slova: klíčová slova

Title: USB Packet Analyzer

Author: Peter Lakatoš

Department: Department of Distributed and Dependable Systems

Supervisor: Mgr. Pavel Ježek, Ph.D., Department of Distributed and Dependable Systems

Abstract: Abstract.

Keywords: key words

Obsah

1. Uvod

1.1 USB

1.1.1 Základné pojmy

vysvetlenie zakladnych pojmov spojenych USB: historia, usb port/conector, plug and play(https://docs.microsoft.com/en-us/windows-hardware/drivers/kernel/introductio to-plug-and-play), host-master, low/full/high speed zariadenia

1.2 Existujúce aplikácie

spomenutie aplikacii ktore sluzia na interpretaciu usb packetov(Wireshark, USBlyzer a pod.)

1.3 Požadované funkcie

Zakladne poziadavky kladene na aplikaciu, co by mala splnat.

1.4 Ciele práce

2. USB a Windows

2.1 USB zbernica

Plug and Play device tree(sposob akym si windows udrziava strom zariadeni na zbernici)(https://docs.microsoft.com/sk-sk/windows-hardware/drivers/gettingstarted/devic nodes-and-device-stacks)

2.2 Device object a device stack

PDO,FDO, Device object(https://docs.microsoft.com/en-us/windows-hardware/drivers/ker to-device-objects) https://docs.microsoft.com/en-us/windows-hardware/drivers/kernel/creatin a-device-object

2.2.1 Drivery

windows driver model(WDM) : https://docs.microsoft.com/en-us/windows-hardware/drivers/kernel/types-of-wdm-drivers bus driver(https://docs.microsoft.com/en-us/windows-hardware/drivers/kernel/bus-drivers), function driver(https://docs.microsoft.com/ us/windows-hardware/drivers/kernel/function-drivers) a filter driver(https://docs.microsoft.co us/windows-hardware/drivers/kernel/filter-drivers)

2.3 Komunikacia s USB zariadenim

sposob komunikacie operacneho systemu so zariadenim pripojenym na USB zbernicu : IRP(https://docs.microsoft.com/en-us/windows-hardware/drivers/gettingstarted/i-o-request-packets) , URB (https://docs.microsoft.com/en-us/windows-hardware/drivers/usbco with-a-usb-device) a pod. https://docs.microsoft.com/en-us/windows-hardware/drivers/kernel/ irps

2.4 USB descriptory

opis zakladnych USB descriporov, hlavne tych ktore neskor aj vyuzivam v program(Device, Interface, Endpoint, Configuration, String, Setup) : https://docs.microsoft.com/e us/windows-hardware/drivers/usbcon/usb-descriptors https://docs.microsoft.com/en-us/windows-hardware/drivers/usbcon/usb-control-transfer

2.4.1 Rozlozenie USB zariadenia z hladiska descriptorov

https://docs.microsoft.com/en-us/windows-hardware/drivers/usbcon/usb-device-layout

2.5 HID zariadenia

hid zariadenie obecne, priklady https://docs.microsoft.com/en-us/windows-hardware/drivers/hid/

2.5.1 Reporty

Input/Output/Feature reporty.

2.5.2 Report Descriptor

Opis report descriptoru, k comu sluzi, pripadne ako z neho vycitat zaujimave data (neskor vyuzite v programe pri parsovani HID Report Descriptoru na naslednu semanticku analyzu dat ktore posiela zariadenie)

3. Analýza technických riešení

3.1 Voľba frameworku

dovod preco som si zvolil qt namiesto inych c++ GUI frameworkov(napriklad sfml)

3.2 Získanie USB packetov

3.2.1 Windows exclusive mód

opisat co to je, a dolezite je spomenut, ze windows otvara v exclusive mode zakladne HID zariadenia ako mys a klavesnica

3.2.2 Známe knižnice

opisat zakladne kniznice na sledovanie USB zbernice a preco som ich nemohol pouzit : libUSB, hidAPI, moufiltr, SetupAPI, WinUSB

3.2.3 Third-party aplikácie

opisat odkial nakoniec ziskavam packety - USBPcap a Wireshark

3.3 Spracovávanie pcap súborov

moznosti ako citat pcap subory : bud pouzit uz existujucu kniznicu : na linuxe Libpcap, windows NPcap(deprecated WinPcap), alebo citat subory manualne : std::istream alebo QFile

3.4 Zobrazenie základných informácií

ako zobrazovat zakladne info o packete : pouzit QListWidget alebo QTableWidget (pripadne nieco ine ako nejaky abstract viewmodel), narok na zakladne funkcionality : lahka rozsiritenlnost o dalsie ”stlpceky” , moznost jednoduchej interakcie(doubleClick na polozku). Mat vsetky info na jednom okne / mat pop-up okna.

3.5 Zobrazenie sémantického významu dát

ako vyzobrazit semanticky vyznam roznych dat - descriptory, usb header, vyznam input dat roznych HID zariadeni

3.6 Hexdump

ako v qt urobit hexdump - do coho zobrazovat data(vytvorit si vlastny viewer dedeny od QAbstractScrollArea, pripadne niecoho ineho) vs najst nieco co uz v qt je a upravit to aby to sedelo poziadavkam. Vziat do uvahy bezne funkcie hexdumpu : selection mody(oznacit naraz hexa a im odpovedajuce printable), logicke oddelenie dat(napriklad farbami)

4. Analýza návrhu

5. Vývojová dokumentácia

5.1 Architekrúra aplikácie

5.2 Jadro aplikácie

5.2.1 USB_Packet_Analyzer

riadi celkovy beh programu, reaguje na input od uzivatela

5.2.2 Item Manager

spracovanie samostatneho packetu a ulozenie dat o nom

5.2.3 DataViewer

trieda ktora ma na starosti vyskakovacie okno po dvojkliku a item a nasledne reaguje na input od uzivatela v okne

5.2.4 TreeItem

reprezentuje jednotlive nody v stromovej strukture ktora sa potom vyuziva na zobrazenie dat v QTreeView

5.3 Modely

5.3.1 AdditionaldataModel

model na spravovanie zvysnych dat(data ktore nie su sucastou hlavicky packetu)

5.3.2 ColorMapModel

vyobrazenie pomocnej mapy na lepsie sa zorientovanie v zvyraznemom hexdumpe

5.3.3 DataViewerModel

model na hexdump - prenasa hex/printable a zaroven o co vlastne ide(konkretny descriptor, interrupt data, ...)

5.3.4 TreeItemBaseModel

model na QTreeView ktorz vyuziva TreeItem

5.3.5 USBPcapHeaderModel

model na QTreeView ale specialne pre USBPcap hlavicku packetu

5.4 Interpretery

5.4.1 BaseInterpreter

abstractna trieda od ktorej dedia vsetkz interpretery

5.4.2 Interpreter factory

facory trieda na pridelenie konkretneho interpreteru za runtimu kvoli jednoduchosti na lepsie rozsirenie programu do buducnosti

5.4.3 Interpretery descriptorov

Config,Device,Setup,String,...

5.4.4 Interrupt transfer interpretery

obecne interrupt transfer interpreter - sluzi skor ako factory na rozne doteraz implementovane HID zariadenia

Joystick interpreter

Mouse interpreter

Keyboard interpreter

5.5 Delegáti

DataViewerDelegate

Qt delegat - stara sa o highlight hexdumpu

5.6 HID

5.6.1 HIDDevices

staticka trieda, drzi vsetky rozpoznane HID zariadenia a obsahuje funkcie specificke nich - parsovanie HID Report descriptoru

5.7 Práca so súbormi

5.7.1 FileReader

praca zo suborom a predavanie precitanych dat, offline/online capture, QFile vs std::istream

5.8 Globálne dáta

5.8.1 ConstDataHolder

staticka trieda na drzanie si konstant ktore su potrebne napriec celym programom. Mapovanie z enumu do jeho stringovej reprezentacie

5.8.2 PacketExternStructs

obsahuje definiciu vsetkych dolezitych USBPcap structov, pcap structov, enumov a vsetkych structov ktore pouzivam v aplikacii

6. Možnosti rozšírenia

Rozobrať čo všetko sa dá urobiť s tými dátami, ktoré už mám uložené v pamati, ale momentálne sa s nimi nič nedeje

6.1 Ukladanie výstupu do súboru

výstup analýzy do súboru(textového)

6.2 Iná vizuálna reprezentácia dát

Momentálne vyzobrazujem dáta prevažne v QTreeView alebo QTableView, ale vdaka tomu ako ich mám uložené + to že nad nimi operuje nejaký model ktorý vie vrátiť dáta na základe indexu, by nemuselo byť taká zložité pridať inú vizualizáciu dát(napríklad obrázkovú ako tu : https://www.usbmadesimple.co.uk/ums_5.htm)

6.3 Pridávanie nových interpreterov pre descripory

pridanie nových druhov descriptorov - pridať nový interpreter do factory

6.4 Pridanie intepreteru na interrupt tranfser

pridanie analyzi interrupt transferu aj pre ine ako hid zariadenia

6.4.1 Pridanie nových HID zariední

nove HID zariadenie - pridanie do interrupt "factory"

6.5 Pridanie analýzy pre isochronous a bulk transfer

semanticka analyza aj inych ako interrupt alebo control transferov - momentalne su rozpoznavane len v hexdumpe

6.6 ?Možnosť rozšírenia na iné platformy?

uprava aplikacie aby bola prenositelna aj na ine platformy, co vsetko by tam bolo treba upravit(pravdepodobne nie vela, kedze qt je prenosne, a prakticky jedine co pouzivam spojene s windowsom su jeho structy na rozne descriptory)

7. Užívateľská dokumentácia

7.1 Inštalácia

nastavenie celkovej aplikácie, ale aj nainstalovanie USBPcap + wireshark a ich kombinácia pre live capture

7.2 Orientácia v GUI aplikácie

popis k jednotlivým tlačidlám gui

7.3 Používanie aplikácie

ako spustit live/offline capture, a celkovo ako pracovať s aplikáciou(popis funkcií - doubleClick na item => zobrazi sa pop-up okno s bližšou analýzou)

8. Záver

8.1 Zhrnutie

celkove zhrnutie prace, ?praca s Qt?

8.2 Budúce plány

Seznam obrázků

Seznam tabulek

Seznam použitých zkratek

Přílohy

.1 První příloha