ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

Факультет кибербезопасности и управления

Кафедра программной инженерии

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №** 2

по дисциплине Операционные системы и оболочки

Взаимодействие с памятью ОС Linux

|  |  |  |
| --- | --- | --- |
| **ВЫПОЛНИЛ** | | |
| студент | РПИС-11  (группа) | Яковлев П. Н.  (ФИО) |
|  | **ПРОВЕРИЛ** |  |
| (должность) | | **Баженов А. Э**.  (ФИО) |

Самара

2023

**Цель лабораторной работы** – изучение принципов взаимодействия программного обеспечения с адресным пространством в операционной системе Linux с использованием средств языков программирования Java и Python и системных вызовов.

**Задание**: Все варианты должны иметь графический интерфейс. Разработать приложение, которое позволяет чтение и запись данных из адресного пространства процесса. Пользователь самостоятельно указывается адресное пространство в программе.

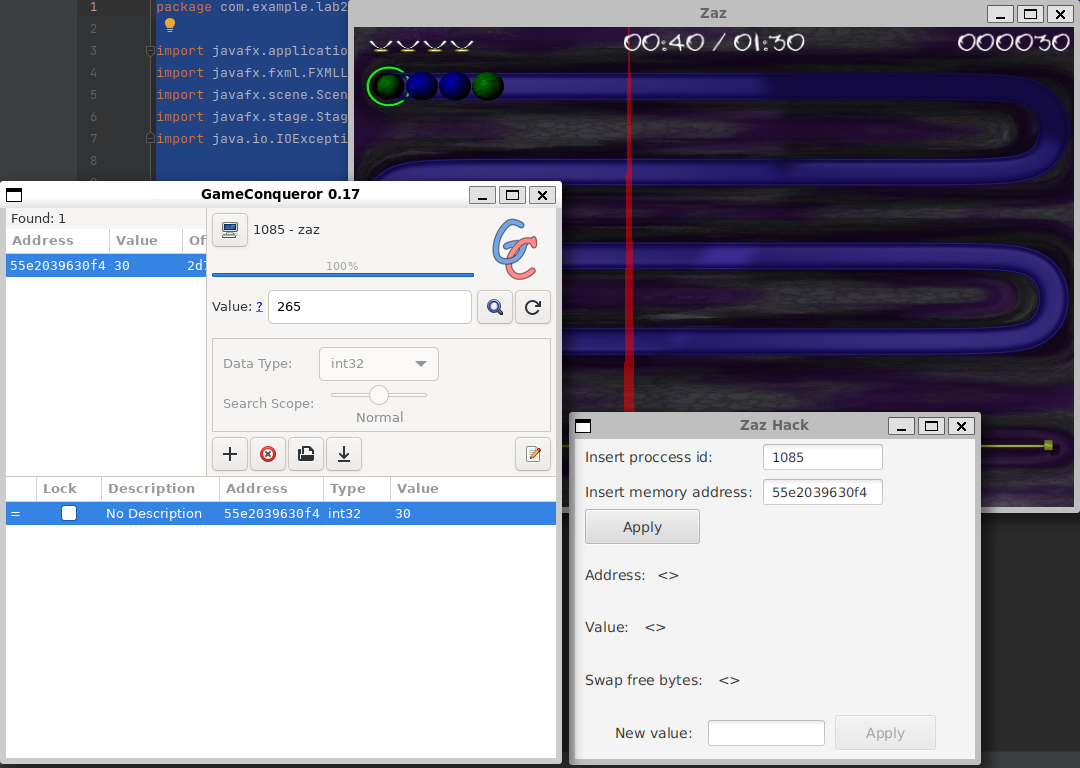
Для поиска нужного адреса рекомендуется использовать бесплатную программу scanmem или gameconqueror.

Четные номера зачетных книжек работают с блокнотом, **нечетные с любой компьютерной игрой**.

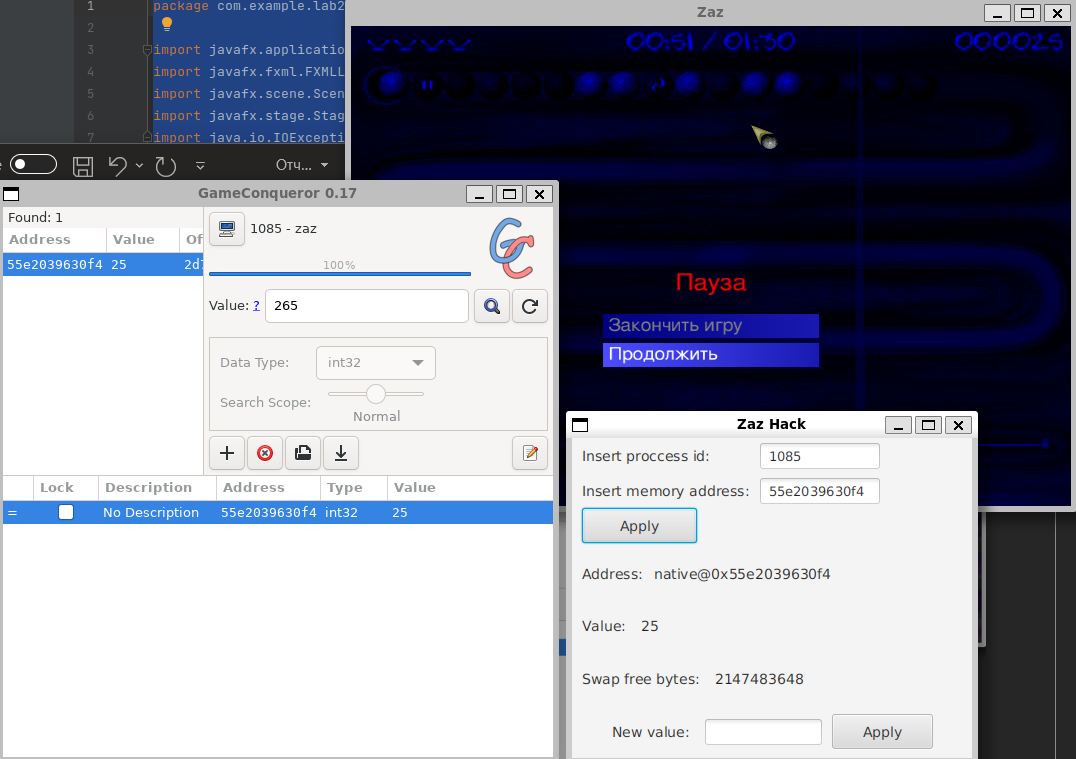
Добавить в приложение информацию, правильность которой необходимо подтвердить и объяснить:

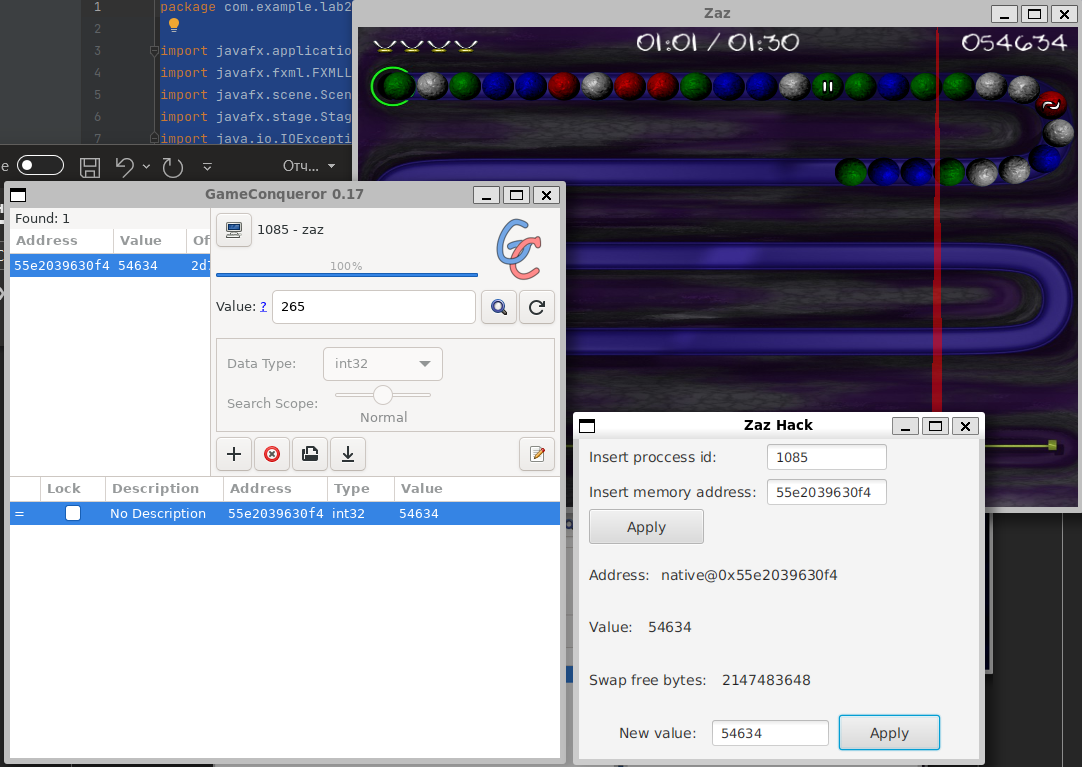
**Вариант 5. Количество свободных байтов файла подкачки.**

Для создания графического интерфейса была использована библиотека javaFX с использованием SceneBuilder. Чтение и запись в адрес процесса осуществлялось с помощью метода ptrace и соответствующих вызовов. Для доступа к этому методу была скачана и импортирована библиотека Java Native Access (JNA).

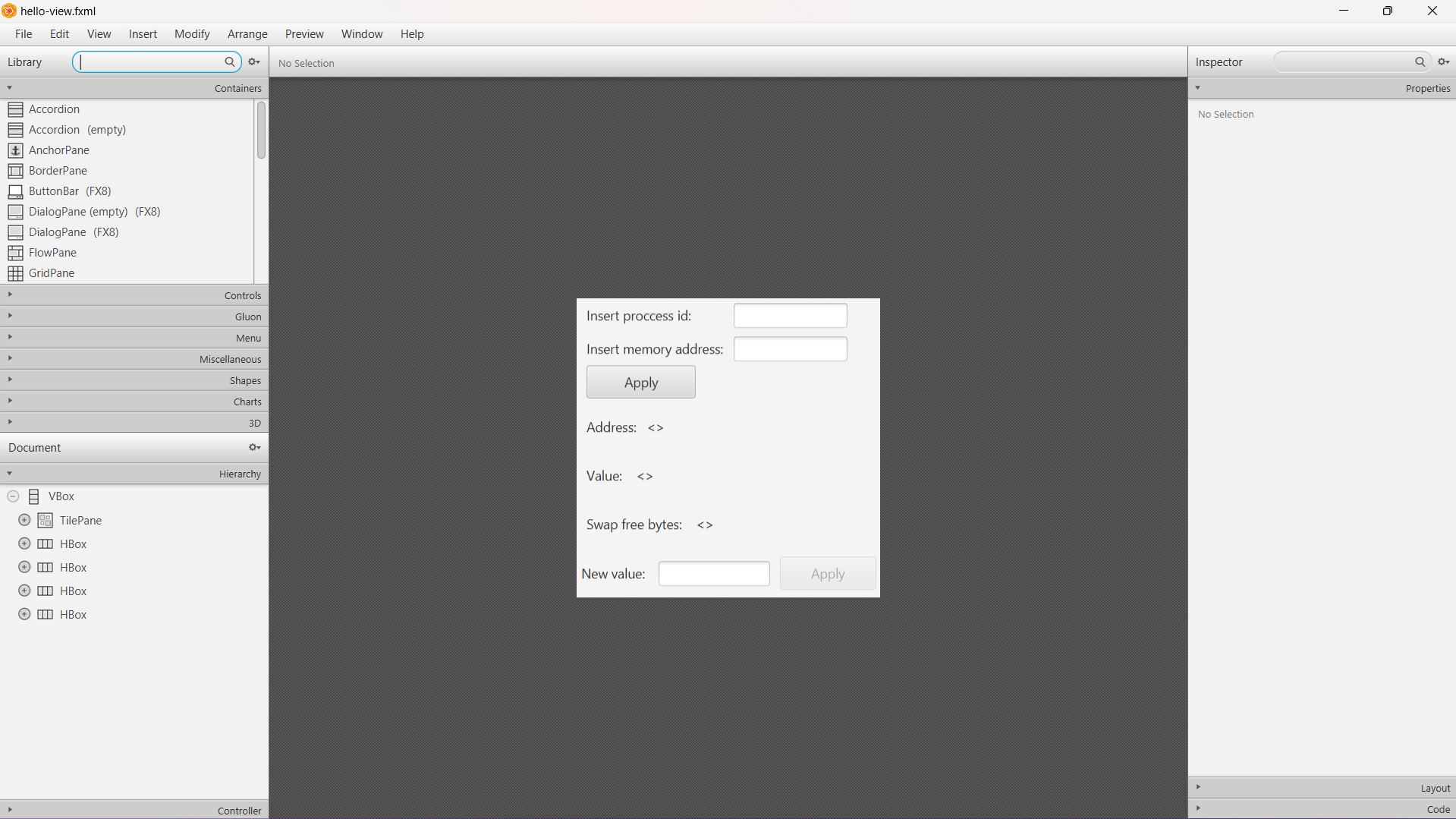
Пример работы программы: 

Считывание из адреса памяти:



Изменение значения в адресе:

**Дизайн программы в SceneBuilder**



Листинг:

HelloApplication (Main):

package com.example.lab2;

import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.scene.Scene;

import javafx.stage.Stage;

import java.io.IOException;

public class HelloApplication extends Application {

@Override

public void start(Stage stage) throws IOException {

FXMLLoader fxmlLoader = new FXMLLoader(HelloApplication.class.getResource("hello-view.fxml"));

Scene scene = new Scene(fxmlLoader.load(), 400, 320);

stage.setTitle("Zaz Hack");

stage.setScene(scene);

stage.show();

}

public static void main(String[] args) {

launch();

}

}

Controller:

package com.example.lab2;

import javafx.fxml.FXML;

import javafx.scene.control.Alert;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.layout.HBox;

import javafx.scene.layout.TilePane;

import java.io.\*;

public class Controller {

int pid, readed;

String addressSt;

@FXML

private TilePane paneInsert;

@FXML

private Button btnApplyAddress, btnApplyValue;

@FXML

private Label lblAddress, lblValue, lblSwap;

@FXML

private TextField textFieldInsAddress, textFieldInsPid, textFieldInsertValue;

@FXML

private HBox hBoxInsertValue, hBoxAddress, hBoxValue;

@FXML

protected void OnClickApplyAddress() throws IOException {

try {

btnApplyAddress.setText("Apply");

pid = Integer.parseInt(textFieldInsPid.getText());

addressSt = textFieldInsAddress.getText();

MemMethods mMethod = new MemMethods(pid, addressSt);

readed = mMethod.read();

lblValue.setText(String.valueOf(readed));

lblAddress.setText(mMethod.getAddress());

infSys inf = new infSys();

lblSwap.setText(String.valueOf(inf.getSwap()));

btnApplyValue.setDisable(false);

} catch (NumberFormatException e){

NotANumberMessage();

}

}

@FXML

protected void OnClickApplyInsValue(){

try {

MemMethods mMethod = new MemMethods(pid, addressSt);

mMethod.write(Integer.parseInt(textFieldInsertValue.getText()));

readed = mMethod.read();

lblValue.setText(String.valueOf(readed));

}catch (NumberFormatException e){

NotANumberMessage();

}

}

@FXML

void NotANumberMessage(){

Alert alert = new Alert(Alert.AlertType.WARNING);

alert.setTitle("Enter warning");

alert.setHeaderText("Enter a number in the field");

alert.showAndWait();

}

}

**MemMethods**:

package com.example.lab2;

import com.sun.jna.\*;

public class MemMethods {

Pointer address;

int pid;

CLibrary libc = (CLibrary) Native.loadLibrary("c", CLibrary.class);

public MemMethods(int pid, String adrStr){

this.pid = pid;

address = new Pointer(Long.parseLong(adrStr, 16));

}

public int read(){

libc.ptrace(CLibrary.PTRACE\_ATTACH, pid, null,0);

int peek = libc.ptrace(CLibrary.PTRACE\_PEEKDATA, pid, address, 0);

libc.ptrace(CLibrary.PTRACE\_DETACH, pid, null,0);

return peek;

}

public void write(int value){

libc.ptrace(CLibrary.PTRACE\_ATTACH, pid, null,0);

libc.ptrace(CLibrary.PTRACE\_POKEDATA, pid, address, value);

libc.ptrace(CLibrary.PTRACE\_DETACH, pid, null,0);

}

public String getAddress(){

return address.toString();

}

}

interface CLibrary extends Library

{

int PTRACE\_ATTACH = 16;

int PTRACE\_DETACH = 17;

int PTRACE\_PEEKDATA = 2;

int PTRACE\_POKEDATA = 5;

int ptrace(int request, int pid, Pointer addr, int data);

}

**infSys**:

package com.example.lab2;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import java.io.\*;

public class infSys {

public long getSwap() throws IOException {

Pattern pattern = Pattern.compile("([\\/A-Za-z0-9]+)[\\s]+([a-z]+)[\\s]+([0-9]+)[\\s]+([0-9]+)[\\s]+([\\-0-9]+).\*");

BufferedReader reader = new BufferedReader(new FileReader("/proc/swaps"));

String s = reader.readLine();

while (s != null) {

Matcher matcher = pattern.matcher(s);

if (matcher.matches()) {

System.out.println(s);

return Long.parseLong(matcher.group(3)) \* 1024L;

}

s = reader.readLine();

}

reader.close();

return 0;

}

}

Hello-view.fxml:

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>

<?import javafx.scene.control.Button?>

<?import javafx.scene.control.Label?>

<?import javafx.scene.control.TextField?>

<?import javafx.scene.layout.HBox?>

<?import javafx.scene.layout.TilePane?>

<?import javafx.scene.layout.VBox?>

<?import javafx.scene.text.Font?>

<VBox maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="315.0" prefWidth="320.0" xmlns="http://javafx.com/javafx/19" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.lab2.Controller">

<children>

<TilePane fx:id="paneInsert" hgap="10.0" prefHeight="138.0" prefWidth="320.0" tileAlignment="CENTER\_LEFT">

<children>

<Label text="Insert proccess id: ">

<font>

<Font size="14.0" />

</font>

</Label>

<TextField fx:id="textFieldInsPid" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="26.0" prefWidth="120.0" />

<Label layoutX="15.0" layoutY="18.0" text="Insert memory address:">

<font>

<Font size="14.0" />

</font>

</Label>

<TextField fx:id="textFieldInsAddress" layoutX="165.0" layoutY="15.0" maxWidth="-Infinity" prefHeight="26.0" prefWidth="120.0" />

<Button fx:id="btnApplyAddress" mnemonicParsing="false" onAction="#OnClickApplyAddress" prefHeight="35.0" prefWidth="115.0" text="Apply">

<font>

<Font size="14.0" />

</font>

</Button>

</children>

<padding>

<Insets left="10.0" />

</padding>

</TilePane>

<HBox fx:id="hBoxAddress" alignment="CENTER\_LEFT" prefHeight="80.0" prefWidth="200.0" spacing="10.0">

<children>

<Label text="Address:">

<font>

<Font size="14.0" />

</font>

</Label>

<Label fx:id="lblAddress" layoutX="20.0" layoutY="20.0" text="&lt;&gt;">

<font>

<Font size="14.0" />

</font>

</Label>

</children>

<padding>

<Insets left="10.0" />

</padding>

</HBox>

<HBox fx:id="hBoxValue" alignment="CENTER\_LEFT" prefHeight="80.0" prefWidth="200.0" spacing="10.0">

<children>

<Label text="Value: ">

<font>

<Font size="14.0" />

</font>

</Label>

<Label fx:id="lblValue" text="&lt;&gt;">

<font>

<Font size="14.0" />

</font>

</Label>

</children>

<padding>

<Insets left="10.0" />

</padding>

</HBox>

<HBox fx:id="hBoxValue1" alignment="CENTER\_LEFT" layoutX="10.0" layoutY="196.0" prefHeight="80.0" prefWidth="200.0" spacing="10.0">

<children>

<Label text="Swap free bytes: ">

<font>

<Font size="14.0" />

</font>

</Label>

<Label fx:id="lblSwap" text="&lt;&gt;">

<font>

<Font size="14.0" />

</font>

</Label>

</children>

<padding>

<Insets left="10.0" />

</padding>

</HBox>

<HBox fx:id="hBoxInsertValue" alignment="CENTER" layoutX="10.0" layoutY="110.0" prefHeight="80.0" prefWidth="200.0" spacing="10.0">

<children>

<Label text="New value: ">

<font>

<Font size="14.0" />

</font>

</Label>

<TextField fx:id="textFieldInsertValue" prefHeight="26.0" prefWidth="117.0" />

<Button fx:id="btnApplyValue" disable="true" mnemonicParsing="false" onAction="#OnClickApplyInsValue" prefHeight="35.0" prefWidth="101.0" text="Apply">

<font>

<Font size="14.0" />

</font>

</Button>

</children>

</HBox>

</children>

</VBox>