МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ  
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

Поволжский Государственный Университет Телекоммуникаций и Информатики

Отчет по практической работе № 4

по дисциплине «Операционные системы и оболочки»

Выполнил:  
студент группы ИВТ-34У  
Горлов А.А.

Самара, 2024

Задание

1 задание. Для нечетных номеров списка группы реализовать №1 (получение координат курсора мыши и нажатая клавиша клавиатуры).

2 задание. Получение и вывод информации об устройствах компьютера.

Вариант 7: Информация о видеокарте: Команда «lspci -v | grep -i vga» и Информация о подключенных мониторах: Команда «xrandr».

Main.py

from gui import DeviceInfoApp

if \_\_name\_\_ == "\_\_main\_\_":

    app = DeviceInfoApp()

    app.mainloop()

gui.py

import tkinter as tk

from input\_devices import get\_mouse\_position, listen\_keyboard, stop\_mouse\_tracking

from system\_info import get\_gpu\_info, get\_monitor\_info

class DeviceInfoApp(tk.Tk):

    def \_\_init\_\_(self):

        super().\_\_init\_\_()

        self.title("Информация о ПК и вводе данных")

        self.geometry("600x400")

        # Метка для отображения позиции мыши

        self.mouse\_position\_label = tk.Label(self, text="Позиция мыши: ", font=("Helvetica", 12))

        self.mouse\_position\_label.pack(pady=10)

        # Метка для отображения последней нажатой клавиши

        self.key\_label = tk.Label(self, text="Последняя нажатая клавиша: ", font=("Helvetica", 12))

        self.key\_label.pack(pady=10)

        # Кнопка для начала отслеживания мыши и клавиатуры

        btn\_start\_listeners = tk.Button(self, text="Начать отслеживание мыши и клавиатуры", command=self.start\_listeners)

        btn\_start\_listeners.pack(pady=10)

        # Кнопка для остановки отслеживания мыши

        btn\_stop\_mouse\_tracking = tk.Button(self, text="Остановить отслеживание мыши", command=self.stop\_mouse\_tracking)

        btn\_stop\_mouse\_tracking.pack(pady=10)

        # Метка для отображения информации о видеокарте

        self.gpu\_info\_label = tk.Label(self, text="Информация о видеокарте: ", font=("Helvetica", 12), justify=tk.LEFT)

        self.gpu\_info\_label.pack(pady=10)

        # Кнопка для получения информации о видеокарте

        btn\_get\_gpu\_info = tk.Button(self, text="Получить информацию о видеокарте", command=self.display\_gpu\_info)

        btn\_get\_gpu\_info.pack(pady=10)

        # Метка для отображения информации о мониторах

        self.monitor\_info\_label = tk.Label(self, text="Информация о мониторах: ", font=("Helvetica", 12), justify=tk.LEFT)

        self.monitor\_info\_label.pack(pady=10)

        # Кнопка для получения информации о мониторах

        btn\_get\_monitor\_info = tk.Button(self, text="Получить информацию о мониторах", command=self.display\_monitor\_info)

        btn\_get\_monitor\_info.pack(pady=10)

    # Запуск отслеживания мыши и клавиатуры

    def start\_listeners(self):

        get\_mouse\_position(self.update\_mouse\_position)

        listen\_keyboard(self.update\_key\_press)

    # Остановка отслеживания мыши

    def stop\_mouse\_tracking(self):

        stop\_mouse\_tracking()

    # Обновление позиции мыши

    def update\_mouse\_position(self, position):

        self.mouse\_position\_label.config(text=f"Позиция мыши: {position}")

    # Обновление информации о последней нажатой клавише

    def update\_key\_press(self, key):

        self.key\_label.config(text=f"Последняя нажатая клавиша: {key}")

    # Отображение информации о видеокарте

    def display\_gpu\_info(self):

        gpu\_info = get\_gpu\_info()

        self.gpu\_info\_label.config(text=f"Информация о видеокарте:\n{gpu\_info}")

    # Отображение информации о мониторах

    def display\_monitor\_info(self):

        monitor\_info = get\_monitor\_info()

        self.monitor\_info\_label.config(text=f"Информация о мониторах:\n{monitor\_info}")

input\_devices.py

import pyautogui

import keyboard

from threading import Thread

# Глобальная переменная для управления отслеживанием мыши

tracking\_mouse = False

# Функция для получения координат мыши

def get\_mouse\_position(callback):

    global tracking\_mouse

    tracking\_mouse = True

    def track\_mouse():

        while tracking\_mouse:

            pos = pyautogui.position()

            callback(pos)

    thread = Thread(target=track\_mouse)

    thread.daemon = True

    thread.start()

# Функция для остановки отслеживания мыши

def stop\_mouse\_tracking():

    global tracking\_mouse

    tracking\_mouse = False

# Функция для отслеживания нажатий клавиш

def listen\_keyboard(callback):

    def on\_key\_event(e):

        callback(e.name)

    thread = Thread(target=lambda: keyboard.hook(on\_key\_event))

    thread.daemon = True

    thread.start()

system\_info.py

import subprocess

# Функция для получения информации о видеокарте

def get\_gpu\_info():

    try:

        gpu\_info = subprocess.check\_output("lspci -v | grep -i vga", shell=True).decode('utf-8')

        return gpu\_info

    except Exception as e:

        return f"Ошибка получения информации о видеокарте: {str(e)}"

# Функция для получения информации о подключённых мониторах

def get\_monitor\_info():

    try:

        # Используем xrandr для получения информации о мониторах

        monitor\_info = subprocess.check\_output("xrandr --listmonitors", shell=True).decode('utf-8')

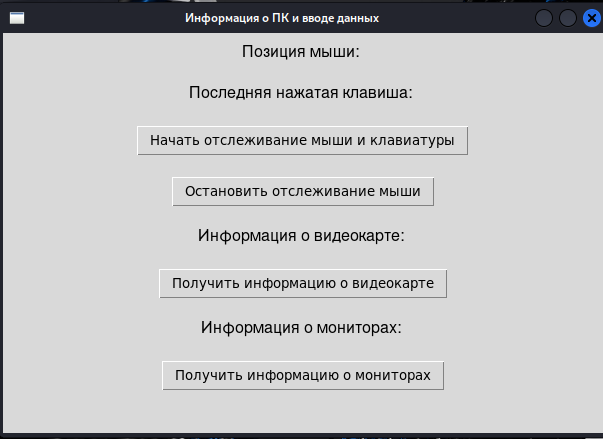
        return monitor\_info

    except Exception as e:

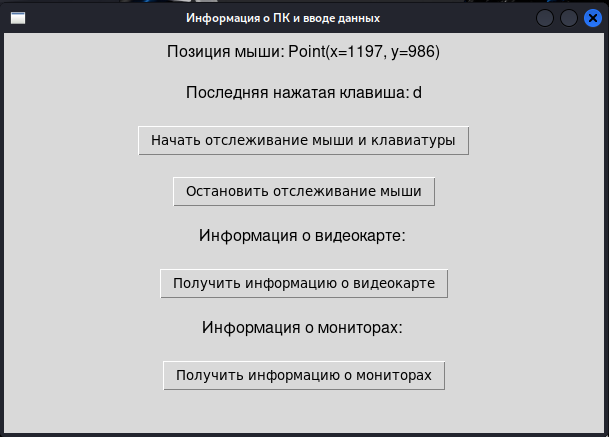
        return f"Ошибка получения информации о мониторах: {str(e)}"

Описание работы

Окно программы



Отслеживание мыши и клавы



Аппаратная информация  
