

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра ИБ

ОТЧЕТ
по лабораторной работе №4
по дисциплине «Модели безопасности компьютерных систем»
Тема: Мандатная модель

Студент гр. 8362

Панфилович А.И.

Преподаватель

Савельев М.Ф.

Санкт-Петербург

2022

Задача.

Написать приложение реализующее следующий функционал:

- Возможность задания для выбранных папок уровней доступа (например, TopSecret, Secret, NonSecret).
- Продемонстрировать основной принцип модель – «Не возможны информационные потоки от объектов с большим уровнем конфиденциальности к объектам с меньшим уровнем конфиденциальности». Должен быть осуществлен запрет копирования файлов из папок с большим уровнем конфиденциальности в папки с меньшим уровнем.

Теоретические сведения.

Мандатная политика безопасности:

- Все субъекты и объекты идентифицированы;
- Задана решетка уровней конфиденциальности информации;
- Каждому объекту системы присвоен уровень конфиденциальности, определяющий ценность содержащейся в ней информации;
- Каждому субъекту системы присвоен доступ, определяющей уровень доверия к нему в компьютерной системе.

Мандатная политика безопасности в автоматной модели безопасности информационных потоков состоит в том, что пользователи с большим уровнем доступа не должны информационно влиять на пользователей с меньшим уровнем доступа.

Разработка программы.

В ходе выполнения задания было разработано GUI приложение для пользователя, позволяющее задавать уровень конфиденциальности для любой папки на диске, также осуществлять перенос файлов из папки с меньшим или равным уровнем конфиденциальности в папку с большим.

Уровни конфиденциальности, заданные пользователем, хранятся в .txt файле (см. рис. 1). Интерфейс программы изображен на рисунке 2.

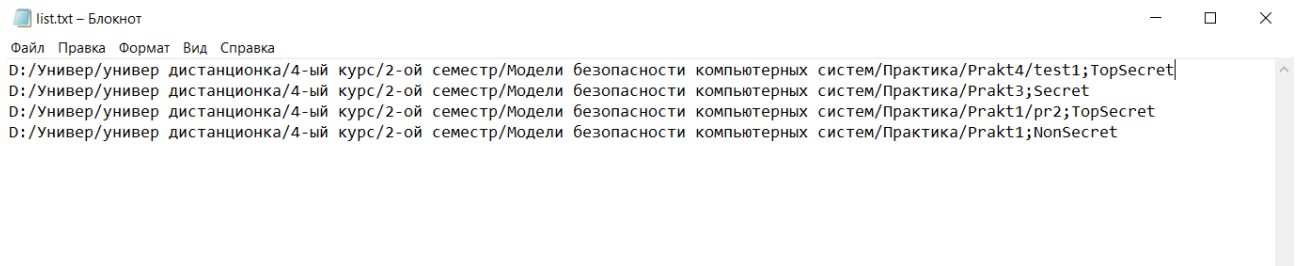


Рисунок 1 – Уровни конфиденциальности в .txt-файле

Интерфейс программы содержит три раздела:

1. «Уровни доступа» – в данном разделе нажатием на кнопку «выбрать папку» пользователь выбирает нужную ему папку через интерфейс проводника Windows, и путь до нее отображается в строке справа. Затем он может изменить/задать уровень доступа к выбранной папке из выпадающего списка справа (в нем также отображается текущий уровень доступа к папке, если он был задан ранее) и зафиксировать изменения нажатием кнопки «изменить уровень доступа». В конце раздела расположено текстовое поле, отображающее файлы, содержащиеся в выбранной папке.

2. «Перенос файлов» – в этой части программы пользователь аналогичным способом выбирает соответствующие папки (сверху-вниз) откуда переносить и куда переносить нажатием на кнопки «выбрать папку», после выбора папок путь до них также отображается в строках справа. После выбора папки-источника пользователь выбирает нужный ему файл из выпадающего списка. Операция перемещения файла выполняется нажатием кнопки «перенести файл».

3. «Статус системы» – в последнем разделе находится текстовое поле куда выводятся все сообщения об успешных завершениях операций и ошибки ввода.

Мандатная политика

Уровни доступа:

Выбрать папку

и компьютерных систем/Практика/Prakt1

NonSecret

Изменить уровень доступа

Содержимое:

pr1
Pr1_Panfilovich_8362.docx
Pr1_Panfilovich_8362.pdf
pr2

Перенос файлов:

Выбрать папку

сности компьютерных систем/Практика/Prakt4/test2

None

test.txt

Выбрать папку

и компьютерных систем/Практика/Prakt4/test1

TopSecret

Перенести файл

Статус системы:

[20:14] Дирректории "D:/Универ/универ дистанционка/4-ый курс/2-ой семестр/Модели безопасности компьютерных систем/Практика/Prakt3" получила статус "Secret"!

[20:14] Дирректории "D:/Универ/универ дистанционка/4-ый курс/2-ой семестр/Модели безопасности компьютерных систем/Практика/Prakt1/pr2" получила статус "TopSecret"!

[20:14] Дирректории "D:/Универ/универ дистанционка/4-ый курс/2-ой семестр/Модели безопасности компьютерных систем/Практика/Prakt1" получила статус "NonSecret"!

Рисунок 2 – Интерфейс программы пользователя

Пример работы программы.

Выполним несколько операций чтобы продемонстрировать работу программы.

1. Пример задания уровня доступа:

Исходное состояние .txt-файла, где хранятся записи об уровнях доступа для различных папок: NonSecret, Secret, TopSecret (см. рис. 3).



Рисунок 3 – Исходное состояния .txt-файла

Теперь в качестве примера зададим уровень конфиденциальности TopSecret для папки Documents (D:/Универ/универ дистанционка/4-ый курс/2-ой семестр/Модели безопасности компьютерных систем/Практика/Prakt4/test1) как показано на рисунке 4:

The screenshot shows a window titled "Мандатная политика" (Mandatory Policy) with standard Windows window controls. The window is divided into several sections:

- Уровни доступа:** (Access Levels) section containing:
 - A "Выбрать папку" (Select folder) button next to a text field containing the path "пьютерных систем/Практика/Prakt4/test1".
 - A dropdown menu currently showing "TopSecret".
 - An "Изменить уровень доступа" (Change access level) button.
- Содержимое:** (Content) section, which is currently empty.
- Перенос файлов:** (File Transfer) section containing:
 - Two "Выбрать папку" (Select folder) buttons, each followed by an empty text field.
 - A "Перенести файл" (Move file) button.
- Статус системы:** (System Status) section containing a text box with the following log entry:

```
[20:20] Дирректории "D:/Универ/универ дистанционка/4-ый курс/2-ой семестр/Модели безопасности компьютерных систем/Практика/Prakt4/test1" полуила статус "TopSecret"!
```

Рисунок 4 – Пример задания уровня доступа

После выполнения операции в .txt-файл была добавлена следующая запись (см. рис. 5):

Рисунок 5 – Добавление новой записи

2. Пример переноса файла:

Продemonстрируем перенос файла test.txt из папки test2 в папку test1 как показано на рисунке 6:

The image shows a window titled "Мандатная политика" (Mandatory Policy) with standard Windows window controls. The window is divided into several sections:

- Уровни доступа:** (Access Levels) section containing a "Выбрать папку" (Select folder) button, an empty text field, a dropdown menu currently showing "TopSecret", and a button labeled "Изменить уровень доступа" (Change access level).
- Содержимое:** (Content) section with a large empty rectangular box.
- Перенос файлов:** (File Transfer) section containing:
 - A "Выбрать папку" (Select folder) button, a text field with the path "ности компьютерных систем/Практика/Prakt4/test2", and a dropdown menu showing "None".
 - A text field containing "test.txt" with a dropdown arrow on the right.
 - Another "Выбрать папку" (Select folder) button, a text field with the path "и компьютерных систем/Практика/Prakt4/test1", and a dropdown menu showing "TopSecret".
 - A button labeled "Перенести файл" (Move file).
- Статус системы:** (System Status) section with a text box containing the message: "[20:53] Файл "test.txt" был перемещен в папку "D:/Универ/универ дистанционка/4-ый курс/2-ой семестр/Модели безопасности компьютерных систем/Практика/Prakt4/test1"."

Рисунок 6 – Разрешенный перенос файла

Данный файл был успешно перемещен, так как он переносился из папки None в папку TopSecret. Покажем, что произойдет если попробовать перенести этот файл обратно (см. рис. 7):

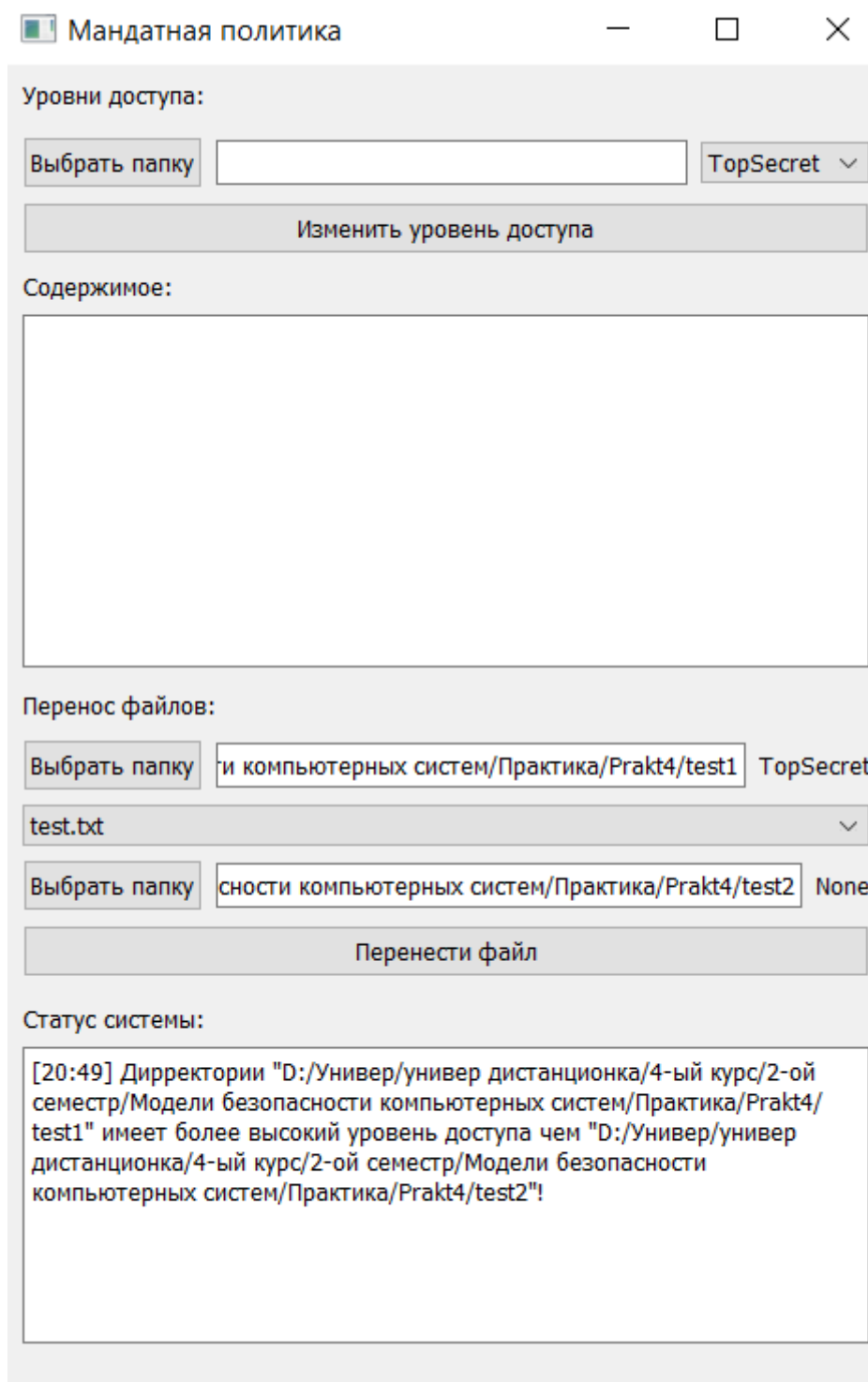


Рисунок 7 – Запрещенный перенос файла

Файл test.txt не будет перенесен обратно так как данный информационный поток запрещен в мандатной модели безопасности.

Выводы.

В результате выполнения лабораторной работы было разработано приложение на языке Python 3.8, GUI приложение пользователя также использовало библиотеку PyQt5. Это приложение эмулирует работу пользователя в системе с мандатной моделью безопасности.

ПРИЛОЖЕНИЕ А

Исходный код программы

Файл main.py:

```
import datetime
import shutil
import PyQt5
import os
import sys

from PyQt5 import QtCore, QtGui, QtWidgets

def check_dir(adr):
    return os.path.isdir(adr)

class Ui_MainWindow(object):

    def __init__(self):
        self.list_dir_mand = {}
        file = open("list.txt", 'r')
        for line in file:
            new_line = line.split(';')
            self.list_dir_mand[new_line[0]] = new_line[1].replace('\n', "")
        file.close()

    def setupUi(self, MainWindow):

        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(474, 745)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.widget = QtWidgets.QWidget(self.centralwidget)
        self.widget.setGeometry(QtCore.QRect(10, 0, 451, 101))
        self.widget.setObjectName("widget")
        self.verticalLayout = QtWidgets.QVBoxLayout(self.widget)
        self.verticalLayout.setContentsMargins(0, 0, 0, 0)
        self.verticalLayout.setObjectName("verticalLayout")
        self.label = QtWidgets.QLabel(self.widget)
        self.label.setObjectName("label")
        self.verticalLayout.addWidget(self.label)
        self.horizontalLayout = QtWidgets.QHBoxLayout()
        self.horizontalLayout.setObjectName("horizontalLayout")
        self.btn_select_dir_1 = QtWidgets.QPushButton(self.widget)
        self.btn_select_dir_1.setObjectName("btn_select_dir_1")
        self.horizontalLayout.addWidget(self.btn_select_dir_1)
        self.path_dir_1 = QtWidgets.QLineEdit(self.widget)
```

```

self.path_dir_1.setObjectName("path_dir_1")
self.horizontalLayout.addWidget(self.path_dir_1)
self.comboBox_1 = QtWidgets.QComboBox(self.widget)
self.comboBox_1.setObjectName("comboBox_1")
self.comboBox_1.addItem("")
self.comboBox_1.addItem("")
self.comboBox_1.addItem("")
self.comboBox_1.addItem("")
self.horizontalLayout.addWidget(self.comboBox_1)
self.verticalLayout.addLayout(self.horizontalLayout)
self.pushButton_2 = QtWidgets.QPushButton(self.widget)
self.pushButton_2.setObjectName("pushButton_2")
self.verticalLayout.addWidget(self.pushButton_2)
self.widget1 = QtWidgets.QWidget(self.centralwidget)
self.widget1.setGeometry(QtCore.QRect(10, 110, 451, 211))
self.widget1.setObjectName("widget1")
self.verticalLayout_2 = QtWidgets.QVBoxLayout(self.widget1)
self.verticalLayout_2.setContentsMargins(0, 0, 0, 0)
self.verticalLayout_2.setObjectName("verticalLayout_2")
self.label_2 = QtWidgets.QLabel(self.widget1)
self.label_2.setObjectName("label_2")
self.verticalLayout_2.addWidget(self.label_2)
self.plain_init = QtWidgets.QPlainTextEdit(self.widget1)
self.plain_init.setObjectName("plain_init")
self.verticalLayout_2.addWidget(self.plain_init)
self.widget2 = QtWidgets.QWidget(self.centralwidget)
self.widget2.setGeometry(QtCore.QRect(10, 330, 451, 156))
self.widget2.setObjectName("widget2")
self.verticalLayout_3 = QtWidgets.QVBoxLayout(self.widget2)
self.verticalLayout_3.setContentsMargins(0, 0, 0, 0)
self.verticalLayout_3.setObjectName("verticalLayout_3")
self.label_3 = QtWidgets.QLabel(self.widget2)
self.label_3.setObjectName("label_3")
self.verticalLayout_3.addWidget(self.label_3)
self.horizontalLayout_3 = QtWidgets.QHBoxLayout()
self.horizontalLayout_3.setObjectName("horizontalLayout_3")
self.btn_select_dir_2 = QtWidgets.QPushButton(self.widget2)
self.btn_select_dir_2.setObjectName("btn_select_dir_2")
self.horizontalLayout_3.addWidget(self.btn_select_dir_2)
self.path_dir_2 = QtWidgets.QLineEdit(self.widget2)
self.path_dir_2.setObjectName("path_dir_2")
self.horizontalLayout_3.addWidget(self.path_dir_2)
self.type1 = QtWidgets.QLabel()
self.horizontalLayout_3.addWidget(self.type1)
self.verticalLayout_3.addLayout(self.horizontalLayout_3)
self.comboBox_init_2 = QtWidgets.QComboBox(self.widget2)
self.comboBox_init_2.setObjectName("comboBox_init_2")
self.verticalLayout_3.addWidget(self.comboBox_init_2)
self.horizontalLayout_2 = QtWidgets.QHBoxLayout()
self.horizontalLayout_2.setObjectName("horizontalLayout_2")

```

```

self.btn_select_dir_3 = QtWidgets.QPushButton(self.widget2)
self.btn_select_dir_3.setObjectName("btn_select_dir_3")
self.horizontalLayout_2.addWidget(self.btn_select_dir_3)
self.path_dir_3 = QtWidgets.QLineEdit(self.widget2)
self.path_dir_3.setObjectName("path_dir_3")
self.horizontalLayout_2.addWidget(self.path_dir_3)
self.type2 = QtWidgets.QLabel()
self.horizontalLayout_2.addWidget(self.type2)
self.verticalLayout_3.addLayout(self.horizontalLayout_2)
self.btn_transfer = QtWidgets.QPushButton(self.widget2)
self.btn_transfer.setObjectName("btn_transfer")
self.verticalLayout_3.addWidget(self.btn_transfer)
self.widget3 = QtWidgets.QWidget(self.centralwidget)
self.widget3.setGeometry(QtCore.QRect(10, 500, 451, 181))
self.widget3.setObjectName("widget3")
self.verticalLayout_4 = QtWidgets.QVBoxLayout(self.widget3)
self.verticalLayout_4.setContentsMargins(0, 0, 0, 0)
self.verticalLayout_4.setObjectName("verticalLayout_4")
self.label_4 = QtWidgets.QLabel(self.widget3)
self.label_4.setObjectName("label_4")
self.verticalLayout_4.addWidget(self.label_4)
self.plain_sys_status = QtWidgets.QPlainTextEdit(self.widget3)
self.plain_sys_status.setObjectName("plain_sys_status")
self.verticalLayout_4.addWidget(self.plain_sys_status)
MainWindow.setCentralWidget(self.centralwidget)
self.menubar = QtWidgets.QMenuBar(MainWindow)
self.menubar.setGeometry(QtCore.QRect(0, 0, 474, 26))
self.menubar.setObjectName("menubar")
MainWindow.setMenuBar(self.menubar)
self.statusbar = QtWidgets.QStatusBar(MainWindow)
self.statusbar.setObjectName("statusbar")
MainWindow.setStatusBar(self.statusbar)

self.retranslateUi(MainWindow)
QtCore.QMetaObject.connectSlotsByName(MainWindow)

self.add_func()

def add_func(self):
    self.btn_select_dir_1.clicked.connect(lambda: self.btn1())
    self.pushButton_2.clicked.connect(lambda: self.pushbtn())
    self.btn_select_dir_2.clicked.connect(lambda: self.btn2())
    self.btn_select_dir_3.clicked.connect(lambda: self.btn3())
    self.btn_transfer.clicked.connect(lambda: self.transfer())

def rank(self, adr):
    ranks = ["NonSecret", "Secret", "TopSecret"]
    keys = self.list_dir_mand.keys()
    if adr in keys and self.list_dir_mand[adr] in ranks:
        return ranks.index(self.list_dir_mand[adr])

```

```

else:
    return 0

def transfer(self):
    adr1 = self.path_dir_2.text()
    adr2 = self.path_dir_3.text()
    if check_dir(adr1):
        if check_dir(adr2):
            r1 = self.rank(adr1)
            r2 = self.rank(adr2)
            if r1 <= r2:
                file = self.comboBox_init_2.currentText()
                fileadr = os.path.join(adr1, file)
                shutil.move(fileadr, adr2)
                text = self.plain_sys_status.toPlainText()
                text += f"[{datetime.datetime.now().strftime('%H:%M')}] Файл \"{file}\" был перемещен в
папку \"{adr2}\".\n"
                self.plain_sys_status.setPlainText(text)
            else:
                text = self.plain_sys_status.toPlainText()
                text += f"[{datetime.datetime.now().strftime('%H:%M')}] Дирректории \"{adr1}\" имеет более
высокий уровень доступа чем \"{adr2}\"!\n"
                self.plain_sys_status.setPlainText(text)
            else:
                text = self.plain_sys_status.toPlainText()
                text += f"[{datetime.datetime.now().strftime('%H:%M')}] Дирректории \"{adr2}\" не
существует!\n"
                self.plain_sys_status.setPlainText(text)
            else:
                text = self.plain_sys_status.toPlainText()
                text += f"[{datetime.datetime.now().strftime('%H:%M')}] Дирректории \"{adr1}\" не существует!\n"
                self.plain_sys_status.setPlainText(text)

def btn3(self):
    adr = QtWidgets.QFileDialog.getExistingDirectory(None, "Select dir")
    self.path_dir_3.setText(adr)
    if adr not in self.list_dir_mand.keys():
        self.type2.setText("None")
    else:
        self.type2.setText(self.list_dir_mand[adr])
    if not check_dir(adr):
        text = self.plain_sys_status.toPlainText()
        text += f"[{datetime.datetime.now().strftime('%H:%M')}] Дирректории \"{adr}\" не существует!\n"
        self.plain_sys_status.setPlainText(text)

def btn2(self):
    adr = QtWidgets.QFileDialog.getExistingDirectory(None, "Select dir")
    self.path_dir_2.setText(adr)
    if adr not in self.list_dir_mand.keys():
        self.type1.setText("None")

```

```

else:
    self.type1.setText(self.list_dir_mand[adr])
self.comboBox_init_2.clear()
if check_dir(adr):
    fails = os.listdir(adr)
    self.comboBox_init_2.addItems(fails)
else:
    text = self.plain_sys_status.toPlainText()
    text += f"[{datetime.datetime.now().strftime('%H:%M')}] Дирректории \"{adr}\" не существует!\n"
    self.plain_sys_status.setPlainText(text)

def btn1(self):

    adr = QtWidgets.QFileDialog.getExistingDirectory(None, "Select dir")
    if check_dir(adr):
        self.path_dir_1.setText(adr)
        if adr not in self.list_dir_mand.keys():
            self.comboBox_1.setCurrentIndex(3)
        else:
            if self.list_dir_mand[adr] == "TopSecret":
                self.comboBox_1.setCurrentIndex(0)
            elif self.list_dir_mand[adr] == "Secret":
                self.comboBox_1.setCurrentIndex(1)
            elif self.list_dir_mand[adr] == "NonSecret":
                self.comboBox_1.setCurrentIndex(2)

        text = "".join(item + "\n" for item in os.listdir(adr))
        self.plain_init.setPlainText(text)
    else:
        text = self.plain_sys_status.toPlainText()
        text += f"[{datetime.datetime.now().strftime('%H:%M')}] Дирректории \"{adr}\" не существует!\n"
        self.plain_sys_status.setPlainText(text)

def pushbtn(self):
    adr = self.path_dir_1.text()
    if check_dir(adr):
        self.list_dir_mand[adr] = self.comboBox_1.currentText()
        file = open("list.txt", "w")
        for key in self.list_dir_mand.keys():
            file.write(key + ";" + self.list_dir_mand[key] + "\n")
        file.close()
        text = self.plain_sys_status.toPlainText()
        text += f"[{datetime.datetime.now().strftime('%H:%M')}] Дирректории \"{adr}\" получила статус  

        \"{self.comboBox_1.currentText()}\"!\n"
        self.plain_sys_status.setPlainText(text)
    else:
        text = self.plain_sys_status.toPlainText()
        text += f"[{datetime.datetime.now().strftime('%H:%M')}] Дирректории \"{adr}\" не существует!\n"
        self.plain_sys_status.setPlainText(text)

```

```

def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow", "Мандатная политика"))
    self.label.setText(_translate("MainWindow", "Уровни доступа:"))
    self.btn_select_dir_1.setText(_translate("MainWindow", "Выбрать папку"))
    self.comboBox_1.setItemText(0, _translate("MainWindow", "TopSecret"))
    self.comboBox_1.setItemText(1, _translate("MainWindow", "Secret"))
    self.comboBox_1.setItemText(2, _translate("MainWindow", "NonSecret"))
    self.comboBox_1.setItemText(3, _translate("MainWindow", "None"))
    self.pushButton_2.setText(_translate("MainWindow", "Изменить уровень доступа"))
    self.label_2.setText(_translate("MainWindow", "Содержимое:"))
    self.label_3.setText(_translate("MainWindow", "Перенос файлов:"))
    self.btn_select_dir_2.setText(_translate("MainWindow", "Выбрать папку"))
    self.btn_select_dir_3.setText(_translate("MainWindow", "Выбрать папку"))
    self.btn_transfer.setText(_translate("MainWindow", "Перенести файл"))
    self.label_4.setText(_translate("MainWindow", "Статус системы:"))

```

```

def main():
    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())

```

```

main()

```