**Задание для УП ПМ 02.**

**Для выполнения заданий создайте глобальный репозиторий tasks-strings**

**Так же создайте папку на ПК с аналогичным именем. Все задачи необходимо выполнять в данной папке, после его запушить (залить) всё её содержимое в созданную репозиторию.**

**Задача 5.1**

Изучите программный код задач 5.1.

Для задачи 5.1 внесите в словарь данные ещё об одном устройстве.

Создайте новый файл task\_5\_1a, который должен содержать такую-же структуру, как и данная задача, только информация в словаре должна храниться, например, о характеристиках ПК.

Пример: “pc1”:{ “os”:”Windows 10”,

“Processor”:”ADM Phenom II”,

“ram”:”8 Gb”

и т.д. }

решение

# -\*- coding: utf-8 -\*-

"""

Задание 5.1

Вывести информацию о соответствующем параметре, указанного устройства.

Пример выполнения скрипта:

Введите имя устройства: r1

Введите имя параметра: ios

15.4

Ограничение: нельзя изменять словарь london\_co.

Эту задачу можно решить без использования условия if.

"""

london\_co = {

"r1": {

"location": "21 New Globe Walk",

"vendor": "Cisco",

"model": "4451",

"ios": "15.4",

"ip": "10.255.0.1",

},

"r2": {

"location": "21 New Globe Walk",

"vendor": "Cisco",

"model": "4451",

"ios": "15.4",

"ip": "10.255.0.2",

},

"sw1": {

"location": "21 New Globe Walk",

"vendor": "Cisco",

"model": "3850",

"ios": "3.6.XE",

"ip": "10.255.0.101",

"vlans": "10,20,30",

"routing": True,

},

}

device = input("Введите имя устройства: ")

parameter = input("Введите имя параметра: ")

print(london\_co[device][parameter])

пример

**Задача 5.2**

Изучите программный код задачи 5.2. Приведите минимум три примера разных входных данных.

network = input("Введите адрес сети: ")

ip, mask = network.split("/")

ip\_list = ip.split(".")

mask = int(mask)

oct1, oct2, oct3, oct4 = [

int(ip\_list[0]),

int(ip\_list[1]),

int(ip\_list[2]),

int(ip\_list[3]),

]

bin\_mask = "1" \* mask + "0" \* (32 - mask)

m1, m2, m3, m4 = [

int(bin\_mask[0:8], 2),

int(bin\_mask[8:16], 2),

int(bin\_mask[16:24], 2),

int(bin\_mask[24:32], 2),

]

ip\_output = """

Network:

{0:<8} {1:<8} {2:<8} {3:<8}

{0:08b} {1:08b} {2:08b} {3:08b}"""

mask\_output = """

Mask:

/{0}

{1:<8} {2:<8} {3:<8} {4:<8}

{1:08b} {2:08b} {3:08b} {4:08b}

"""

print(ip\_output.format(oct1, oct2, oct3, oct4))

print(mask\_output.format(mask, m1, m2, m3, m4))

пример 1

Введите адрес сети: 10.1.1.0/24

Network:

10 1 1 0

00001010 00000001 00000001 00000000

Mask:

/24

255 255 255 0

11111111 11111111 11111111 00000000

Пример 2

Введите адрес сети: 192.168.10.2/24

Network:

192 168 10 2

11000000 10101000 00001010 00000010

Mask:

/24

255 255 255 0

11111111 11111111 11111111 00000000

Process finished with exit code 0

Process finished with exit code 0

Пример 3

Введите адрес сети: 127.0.0.1/24

Network:

127 0 0 1

01111111 00000000 00000000 00000001

Mask:

/24

255 255 255 0

11111111 11111111 11111111 00000000

Process finished with exit code 0

**Задача 5.3**

Изучите программный код задачи 5.3. На основе данной задачи создайте новый файл 5.3a.

Внесите следующие изменения: в зависимости от того какой тип и номер интерфейса ввел пользователь необходимо выводить информацию в следующем виде:

*Fa0/1 Sector-1 connected trunk a-full a-100 10/100BaseTX  
Fa0/2 Sector-2 connected trunk a-full a-100 10/100BaseTX  
Fa0/3 Sector-3 connected trunk a-full a-100 10/100BaseTX  
Fa0/4 notconnect 1 auto auto 10/100BaseTX  
Fa0/5 port connected 100 a-full a-100 10/100BaseTX  
Fa0/6 connected trunk full 100 10/100BaseTX  
Fa0/7 disabled 100 auto auto 10/100BaseTX*

Т.е. нужно переписать строчку: print(**f"interface** {interface}**"**)

П.с. можно создать словарь с типами представленных интерфейсов и их описанием.

Решение

access\_template = [

"switchport mode access",

"switchport access vlan {}",

"switchport nonegotiate",

"spanning-tree portfast",

"spanning-tree bpduguard enable",

]

trunk\_template = [

"switchport trunk encapsulation dot1q",

"switchport mode trunk",

"switchport trunk allowed vlan {}",

]

interface = {

"Fa0/1": "Sector-1 connected trunk a-full a-100 10/100BaseTX",

"Fa0/2": "Sector-2 connected trunk a-full a-100 10/100BaseTX",

"Fa0/3": "Sector-3 connected trunk a-full a-100 10/100BaseTX",

"Fa0/4": "notconnect 1 auto auto 10/100BaseTX",

"Fa0/5": "port connected 100 a-full a-100 10/100BaseTX",

"Fa0/6": "connected trunk full 100 10/100BaseTX",

"Fa0/7": "disabled 100 auto auto 10/100BaseTX"

}

template = {"access": access\_template, "trunk": trunk\_template}

mode = input("Введите режим работы интерфейса (access/trunk): ")

it = input("Введите тип и номер интерфейса: ")

vlans = input("Введите номер влан(ов): ")

print(f"interface {it}")

print("\n".join(template[mode]).format(vlans))

Пример

Введите режим работы интерфейса (access/trunk): trunk

Введите тип и номер интерфейса: Fa0/1

Введите номер влан(ов): 2

interface Fa0/1

switchport trunk encapsulation dot1q

switchport mode trunk

switchport trunk allowed vlan 2

Process finished with exit code 0

**Задача 5.4**

На основе таблицы (см. ниже) маршрутизаторов напишите скрипт, который бы хранил представленную информацию. При вводе названия модели необходимо выводить её характеристики.

Возьмите за основу: model = {“TE100-S5”: {“product Title”:”S-Port 10/100Mbps Fast Ethernet Switch”,

“10/100Mbps”:”5x”, и т.д.}}

Попробуйте доработать данную задачу что бы можно было вывести данные о всех устройствах у которых MAC = 1k или у кого параметр 10/100Mbps = 5x.

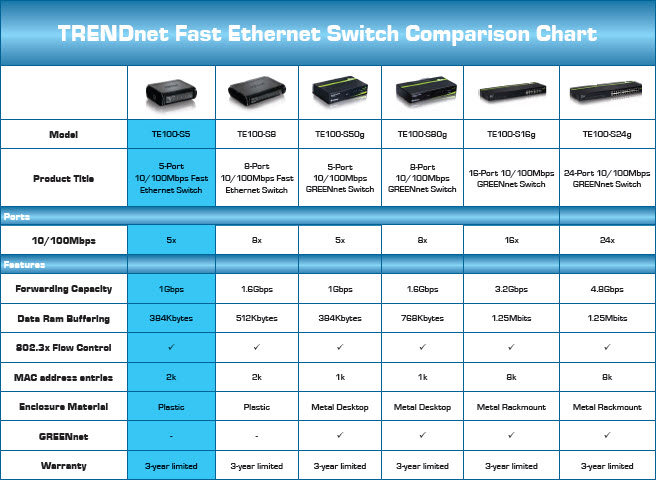


Рис. 1. Таблица моделей маршрутизаторов

**решение**

model = {"TE100-S5":

{"product Title": "5-Port 10/100Mbps Fast Ethernet Switch",

"10/100Mbps": "5x",

"Forearding Capacity": "1 Gbps",

"MAC adress entries": "2 k",

"Enclozure Material": "Plactic"},

"TE100-S8":

{"product Title": "9-Port 10/100Mbps Fast Ethernet Switch",

"10/100Mbps": "8x",

"Forearding Capacity": "1.6 Gbps",

"MAC adress entries": "2 k",

"Enclozure Material": "Plactic"},

}

for switch in model.keys():

if model[switch]['10/100Mbps'] == "5x" and \

model[switch]["MAC adress entries"] == "1k":

print(model[switch])

model = {"TE100-S5":

{"product Title": "5-Port 10/100Mbps Fast Ethernet Switch",

"10/100Mbps": "5x",

"Forearding Capacity": "1 Gbps",

"MAC adress entries": "2 k",

"Enclozure Material": "Plactic"},

"TE100-S8":

{"product Title": "9-Port 10/100Mbps Fast Ethernet Switch",

"10/100Mbps": "8x",

"Forearding Capacity": "1.6 Gbps",

"MAC adress entries": "2 k",

"Enclozure Material": "Plactic"},

"TE100-S50g":

{"product Title": "5-Port 10/100Mbps Fast Ethernet Switch",

"10/100Mbps": "5x",

"Forearding Capacity": "1 Gbps",

"MAC adress entries": "1 k",

"Enclozure Material": "Metal Rackmount"},

"TE100-S80g":

{"product Title": "9-Port 10/100Mbps Fast Ethernet Switch",

"10/100Mbps": "8x",

"Forearding Capacity": "1.6 Gbps",

"MAC adress entries": "1 k",

"Enclozure Material": "Metal Rackmount"},

"TE100-S16":

{"product Title": "5-Port 10/100Mbps Fast Ethernet Switch",

"10/100Mbps": "5x",

"Forearding Capacity": "1 Gbps",

"MAC adress entries": "8 k",

"Enclozure Material": "Metal Rackmount"},

"TE100-S24":

{"product Title": "9-Port 10/100Mbps Fast Ethernet Switch",

"10/100Mbps": "8x",

"Forearding Capacity": "1.6 Gbps",

"MAC adress entries": "8 k",

"Enclozure Material": "Metal Rackmount"},

}

for switch in model.keys():

if model[switch]['10/100Mbps'] == "5x" and \

model[switch]["MAC adress entries"] == "1k":

print(model[switch])

**Задача 5.5**

На основе предыдущей задачи вынесите данные об устройствах во внешний файл json. Напишите программу, которая работает с этим файлом.

**Функционал:**

* Открывает
* Выводит данные
* Выводит данные по параметрам
* Сохраняет изменения

Руководство по работе с json из python:

<https://pyneng.readthedocs.io/ru/latest/book/17_serialization/json.html>