**Лабораторна робота №3**

**Створення утіліти «DiskInfo»**

**Хід роботи**

У даній лабораторній работі, використовуючи виклики системних функцій, отримати інформацію про дискову підсистему:

1. Список усіх логічних дисків в системі.
2. Отримати тип кожного диску присутнього в системі, та дати пояснення для кожного типу диску.
3. Отримати інформацію про диски в системі та про файлові системи Які Використовують на них.
4. Отримати інформацію про зайнятості та вільне місце на кожному з дисків.
5. Отримати інформацію про системну пам'ять.
6. Отримати інформацію про Назву комп'ютера
7. Отримати Назву поточного користувача
8. Отримати інформацію про поточний системний каталог, Тимчасовий каталог, поточний робочий каталог.
9. Для обраних каталогу на диску, Включити спостереження за змінами, продемонструвати відслідковування більше однієї зміни. Зміни записувати в лог файл.

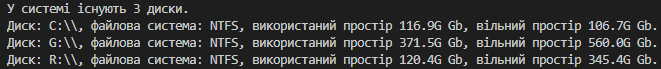
Виводить інформацію про кількість існуючих у системиі дисків:

Код:

disks = psutil.disk\_partitions()

print(f'У системі існують {len(disks)} диски.')

Вивід:



Виводить більшу розгорноту інформацію про диски:

Код:

for i in range(len(disks)):

    disk = str(disks[i]).split('\'')[1]

    fsys = str(disks[i]).split('\'')[5]

    memory = psutil.disk\_usage(disk)

    print(f'Диск: {disk}, файлова система: {fsys}, використаний простір {bytes2human(memory.used)} Gb, '

          f'вільний простір {bytes2human(memory.free)} Gb.')

Вивід:



Виводить іншу необхідну інформацію: про системну пам’ять, назву комп’ютера, назву поточного користувача і інформацію про каталоги.

Код:

systemDrive = os.environ.get('SYSTEMDRIVE')

print(f'Диск {systemDrive}, є системним диском')

print(f'Назва комп\'ютера: {platform.node()}')

print(f'Назва поточного користувача: {getpass.getuser()}')

print(f'Поточний каталог: {os.getcwd()}')

sysCat = os.environ.get('SYSTEMROOT')

tempCat = os.environ.get('TEMP')

print(f'Поточний системний каталог: {sysCat}')

print(f'Тимчасовий каталог: {tempCat}')

print(f'Інформація про системну пам\'ять')

pprint\_ntuple(psutil.virtual\_memory())

Виконується спостереження за вибраним каталог, виводить зміни в консоль і після завершення спостереження записує зміни у файл.

Код:

ACTION = []

def ACTIONSS(done, interval=1):

    ACTIONS = {

        1: "Created",

        2: "Deleted",

        3: "Updated",

        4: "Renamed from something",

        5: "Renamed to something"

    }

    FILE\_LIST\_DIRECTORY = 0x0001

    path\_to\_watch = r"C:\Users\ingaliptn\Desktop\test folder"  # Шлях до дерикторії для спостереження

    hDir = win32file.CreateFile(

        path\_to\_watch,

        FILE\_LIST\_DIRECTORY,

        win32con.FILE\_SHARE\_READ | win32con.FILE\_SHARE\_WRITE | win32con.FILE\_SHARE\_DELETE,

        None,

        win32con.OPEN\_EXISTING,

        win32con.FILE\_FLAG\_BACKUP\_SEMANTICS,

        None

    )

    while not done.wait(interval):

        results = win32file.ReadDirectoryChangesW(

            hDir,

            1024,

            True,

            win32con.FILE\_NOTIFY\_CHANGE\_FILE\_NAME |

            win32con.FILE\_NOTIFY\_CHANGE\_DIR\_NAME |

            win32con.FILE\_NOTIFY\_CHANGE\_ATTRIBUTES |

            win32con.FILE\_NOTIFY\_CHANGE\_SIZE |

            win32con.FILE\_NOTIFY\_CHANGE\_LAST\_WRITE |

            win32con.FILE\_NOTIFY\_CHANGE\_SECURITY,

            None,

            None

        )

        for action, file in results:

            full\_filename = os.path.join(path\_to\_watch, file)

            act = str(f'{full\_filename} {ACTIONS.get(action, "Unknown")}\n')

            ACTION.append(str(f'{full\_filename} {ACTIONS.get(action, "Unknown")}\n'))

            print(act)

done = threading.Event()

threading.Thread(target=ACTIONSS, args=[done], daemon=True).start()

input('Press Enter to exit.\n')

done.set()

for i in range(len(ACTION)):

    f.write(ACTION[i])

f.close()

Повний лістинг програми:

import psutil, os.path, os, time, sys

import win32file

import win32event

import win32con

import threading

import platform, getpass

from psutil.\_common import bytes2human

def pprint\_ntuple(nt):

    for name in nt.\_fields:

        value = getattr(nt, name)

        if name != 'percent':

            value = bytes2human(value)

        print('%-10s : %7s' % (name.capitalize(), value))

disks = psutil.disk\_partitions()

print(f'У системі існують {len(disks)} диски.')

f = open('log.txt', 'r+', encoding='UTF-8')

for i in range(len(disks)):

    disk = str(disks[i]).split('\'')[1]

    fsys = str(disks[i]).split('\'')[5]

    memory = psutil.disk\_usage(disk)

    print(f'Диск: {disk}, файлова система: {fsys}, використаний простір {bytes2human(memory.used)} Gb, '

          f'вільний простір {bytes2human(memory.free)} Gb.')

systemDrive = os.environ.get('SYSTEMDRIVE')

print(f'Диск {systemDrive}, є системним диском')

print(f'Назва комп\'ютера: {platform.node()}')

print(f'Назва поточного користувача: {getpass.getuser()}')

print(f'Поточний каталог: {os.getcwd()}')

sysCat = os.environ.get('SYSTEMROOT')

tempCat = os.environ.get('TEMP')

print(f'Поточний системний каталог: {sysCat}')

print(f'Тимчасовий каталог: {tempCat}')

print(f'Інформація про системну пам\'ять')

pprint\_ntuple(psutil.virtual\_memory())

ACTION = []

def ACTIONSS(done, interval=1):

    ACTIONS = {

        1: "Created",

        2: "Deleted",

        3: "Updated",

        4: "Renamed from something",

        5: "Renamed to something"

    }

    FILE\_LIST\_DIRECTORY = 0x0001

    path\_to\_watch = r"C:\Users\ingaliptn\Desktop\test folder"  # Шлях до дерикторії для спостереження

    hDir = win32file.CreateFile(

        path\_to\_watch,

        FILE\_LIST\_DIRECTORY,

        win32con.FILE\_SHARE\_READ | win32con.FILE\_SHARE\_WRITE | win32con.FILE\_SHARE\_DELETE,

        None,

        win32con.OPEN\_EXISTING,

        win32con.FILE\_FLAG\_BACKUP\_SEMANTICS,

        None

    )

    while not done.wait(interval):

        results = win32file.ReadDirectoryChangesW(

            hDir,

            1024,

            True,

            win32con.FILE\_NOTIFY\_CHANGE\_FILE\_NAME |

            win32con.FILE\_NOTIFY\_CHANGE\_DIR\_NAME |

            win32con.FILE\_NOTIFY\_CHANGE\_ATTRIBUTES |

            win32con.FILE\_NOTIFY\_CHANGE\_SIZE |

            win32con.FILE\_NOTIFY\_CHANGE\_LAST\_WRITE |

            win32con.FILE\_NOTIFY\_CHANGE\_SECURITY,

            None,

            None

        )

        for action, file in results:

            full\_filename = os.path.join(path\_to\_watch, file)

            act = str(f'{full\_filename} {ACTIONS.get(action, "Unknown")}\n')

            ACTION.append(str(f'{full\_filename} {ACTIONS.get(action, "Unknown")}\n'))

            print(act)

done = threading.Event()

threading.Thread(target=ACTIONSS, args=[done], daemon=True).start()

input('Press Enter to exit.\n')

done.set()

for i in range(len(ACTION)):

    f.write(ACTION[i])

f.close()

**Висновки:** у даній лабораторній работі використував виклики системних функцій, отримав інформацію про дискову підсистему.