

Лабораторна робота №3

Створення утіліти «DiskInfo»

Хід роботи

У даній лабораторній роботі, 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 системі дисків:

Код:

```
disks = psutil.disk_partitions()
print(f'У системі існують {len(disks)} диски.')
```

Вивід:

```
У системі існують 3 диски.
Диск: C:\, файлова система: NTFS, використаний простір 116.9G Gb, вільний простір 106.7G Gb.
Диск: G:\, файлова система: NTFS, використаний простір 371.5G Gb, вільний простір 560.0G Gb.
Диск: R:\, файлова система: NTFS, використаний простір 120.4G Gb, вільний простір 345.4G Gb.
```

					“Житомирська Політехніка”22.121.17.000 – Лр3		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Ткачук М.А.			Звіт з лабораторної роботи	Літ.	Арк.
Перевір.		Петросян А.Р.					1
Керівник						Аркушів	
Н. контр.						5	
Зав. каф.						ФІКТ Гр. КІ-20-1	

Виводить більшу розгорнуту 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:

```
Диск: C:\\, файлова система: NTFS, використаний простір 116.9G Gb, вільний простір 106.7G Gb.
Диск: G:\\, файлова система: NTFS, використаний простір 371.5G Gb, вільний простір 560.0G Gb.
Диск: R:\\, файлова система: NTFS, використаний простір 120.4G Gb, вільний простір 345.4G Gb.
```

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,
```

		Ткачук М.А			ДУ"Житомирська Політехніка"121.17.000 – ЛрЗ	Арк.
		Петросян А.Р.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

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()

```

		Ткачук М.А			ДУ“Житомирська Політехніка”121.17.000 – ЛрЗ	Арк.
		Петросян А.Р.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

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\ingalipn\Desktop\test folder" # Шлях до дерикторії
    для спостереження
```

		Ткачук М.А			ДУ“Житомирська Політехніка”121.17.000 – Лр3	Арк.
		Петросян А.Р.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

```

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 роботі використовував виклики системних функцій, отримав інформацію про дискову підсистему.

		Ткачук М.А			ДУ“Житомирська Політехніка”121.17.000 – Лр3	Арк.
		Петросян А.Р.				5
Змн.	Арк.	№ докум.	Підпис	Дата		