­­МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

­­­

Лабораторна робота №2

ФАЙЛОВА СИСТЕМА ОС. ФАЙЛОВІ МЕНЕДЖЕРИ.

з курсу «Операційні системи»

для студентів базового напрямку 6.08.04 "Комп’ютерні науки"

(заочна форма навчання)

Варіант 14

Виконав студент гр. КНз-2

Чалий Михайло

Львів 2014

## 1. Мета роботи

Мета роботи – ознайомлення з файловою системою FAT-32 та з файловими менеджерами (програмами-оболонками) Nогtоn Соmmander, Windows Commander, FAR.

## 2. Короткі теоретичні відомості

Файлова система ОС (підсистема управління файлами ОС) – призначена для забезпечення організованого збереження даних на зовнішніх запам’ятовуючих пристроях у вигляді файлів та регламентованого доступу до них. Найбільш поширеними файловими системами сьогодні є FAT-32 та NTFS для ОС Windows, а також FFS (BSD UNIX), ext2fs(Linux). Для роботи з різними типами файлових систем в ОС UNIX використовується програмне забезпечення віртуальної файлової системи VFS. Для прозорого доступу до файлів, що знаходяться на різних комп’ютерах в мережі використовується мережева файлова система NFS ОС UNIX.

Файл – інформаційна структура для організованого збереження і використання даних на зовнішніх пристроях. Організація інформації користувачем (представленої даними) у відповідності з вирішуваними задачами визначає логічну організацію файла. Логічна організація файла визначає його як сукупність одиниць інформації, які називаються записами. Записи поділяються на поля, в яких записані конкретні значення даних. Представлення даних файла у зовнішній пам’яті (інформаційні структури, адреси розміщення, кодування, тощо) визначається фізичною організацією файла. Логічна організація файла забезпечує доступ та зручність використання даних програмами користувача (визначає інтерфейс доступу). Фізична організація враховує особливості зовнішніх запам’ятовуючих пристроїв для надійного та ефективного збереження даних та організації доступу до них за допомогою функцій ОС. Зв’язок фізичної та логічної організації файла забезпечує підсистема управління файлами ОС.

Файл є інформаційним об’єктом і позначається іменем. У розширеному імені файла крім самого імені файла вказується його тип. В сучасних ОС файли можуть бути згруповані по певних ознаках в каталогах (директоріях, папках). Каталог є файлом спеціального виду, в якому розміщена інформація про згруповані в ньому файли. Каталоги можуть бути вкладені і утворювати деревовидну іерархічну структуру, яка забезпечує швидке виділення груп і пошук файла в групі. Доступ до необхідного файла забезпечується функціями ОС з використанням шляху доступу. Шлях доступу вказує ім’я носія і послідовність імен вкладених каталогів деревовидної ієрархічної структури, починаючи з верхнього рівня (кореневого каталога) і закінчуючи каталогом із потрібним файлом. Шлях доступу та ім’я файла утворюють повне ім’я файла, яке записується таким чином: [ім’я носія:\ ][каталог\][каталог\][ім’я файла][.тип].

Кожен файл має набір характеристик – атрибутів. Атрибути записуються в дескрипторі файла, який, як і файл, зберігається на магнітному диску. Перед використанням файла його дескриптор з диска зчитується в оперативну пам’ять і використовується файловою системою. Атрибутами файла є: ім’я файла, тип файла, розмір файла, атрибути, що визначають права доступу, час і дата створення файла, час і дата останньої модифікації файла.

Файлова система забезпечує виконання таких основних операцій з файлами: створення файла; видалення файла; відкриття файла; закриття файла; запис у файл; читання з файла; переміщення вказівника. Основними операціями з каталогами є: створити каталог; видалити каталог; змінити каталог; визначити біжучий каталог.

Файлові менеджери (програми-оболонки) – призначені для забезпечення зручного інтерфейсу користувача при виконананні обслуговуючих операцій (маніпуляцій) з файлами, каталогами та дисками, а також забезпечення взаємодії користувача з операційною системою.

## 3. Лабораторне завдання

1. Ознайомтесь з призначенням та можливостями файлового менеджера.
2. Вивчіть та опробуйте на практиці використання службових клавіш для роботи з панелями.
3. Вивчіть та опробуйте на практиці використання стрічки функціональних клавіш.
4. Введіть та відкоректуйте текст за допомогою файлового менеджера. Відмітьте виявлені особливості, переваги та недоліки текстового редактора.
5. Ознайомтесь з командами меню Left і Right та опробуйте їх на практиці.
6. Ознайомтесь з командами меню Files та опробуйте їх на практиці.
7. Ознайомтесь з командами меню Commands та опробуйте їх на практиці.
8. Ознайомтесь з командами меню Options та опробуйте їх на практиці.
9. Напишіть програму файлового менеджера (інтерпретатора команд для роботи з файлами і каталогами), який забезпечує ввід, розпізнавання та виконання заданої команди (згідно варіанту).

|  |  |  |  |
| --- | --- | --- | --- |
| Варіант | Назва команди | Синтаксис і приклад команди | Які функції DOS та BIOS використовуються |
| 14 | Створити каталог і в ньому  створити порожній файл | mk\_df шлях (mk\_df c:\catal\file) | 39h,3Bh,5Bh,3Eh(INT 21h) |

## 4. Рішення

### 

Рис 1 Функцональна клавіша F5, Копіювання файлів

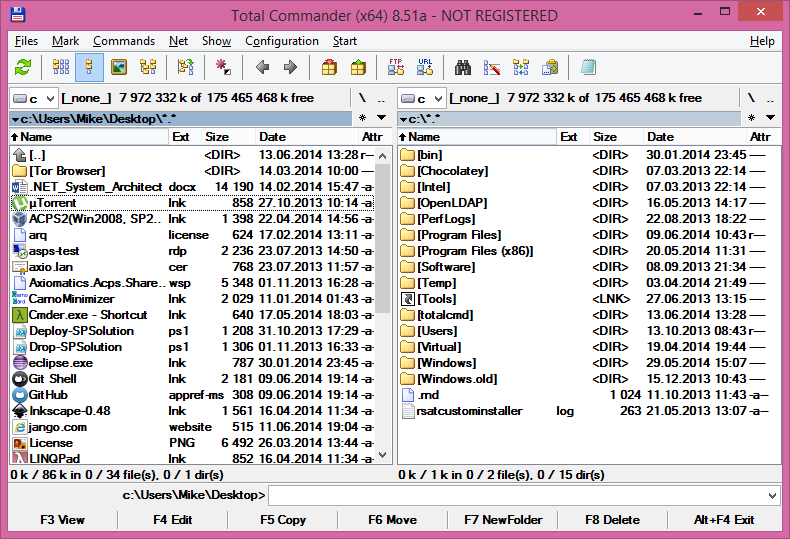


Рис 2 Головне вікно Total Commander

Лістінг програми mk\_df.asm

sseg segment stack

db 256 dup(?)

sseg ends

dseg segment

file\_name db 'EMTPY',0

root\_directory db 'C:\',0

error\_message db 'Unhandled error occured.$'

PSP word ? ;Program segment prefix

dseg ends

cseg segment

assume ss:sseg, cs:cseg, ds:dseg

start:

jmp main

main:

push ds ;Save PSP value

mov ax, seg DSEG ;Point DS and ES at our data segment

mov ds, ax

mov es, ax

pop PSP ;Store PSP value into "PSP"

; Init reading command line

mov es, PSP

mov bx, 82h

; Create new folder

mov dx, bx

mov ah,39h

int 21h

jc erop

; Change directory

mov dx, bx

mov ah,3bh

int 21h

jc erop

; Create new file

mov dx,offset file\_name

mov cx,0 ; No attributes

mov ah,5bh ; Create and open

int 21h

jc erop

mov bx,ax ; Store file handle

; Close file

mov ah,3eh

int 21h

jc erop

exit:

; Exit all good

mov ax,4c00h

int 21h

erop:

; Show error message

mov dx,offset error\_message

mov ah,9

int 21h

jmp exit

cseg ends

end start

## Файл mk\_df.lst

JWasm v2.11, Oct 20 2013

MK\_DF.ASM

00000000 sseg segment stack

00000000 000000000000000000 db 256 dup(?)

00000100 sseg ends

00000000 dseg segment

00000000 454D54505900 file\_name db 'EMTPY',0

00000006 433A5C00 root\_directory db 'C:\',0

0000000A 556E68616E646C6564 error\_message db 'Unhandled error occured.$'

00000023 0000 PSP word ? ;Program segment prefix

00000025 dseg ends

00000000 cseg segment

assume ss:sseg, cs:cseg, ds:dseg

00000000 start:

00000000 EB00 jmp main

00000002 main:

00000002 1E push ds ;Save PSP value

00000003 B80000 mov ax, seg DSEG ;Point DS and ES at our data segment

00000006 8ED8 mov ds, ax

00000008 8EC0 mov es, ax

0000000A 8F060000 pop PSP ;Store PSP value into "PSP"

; Init reading command line

0000000E 8E060000 mov es, PSP

00000012 BB8200 mov bx, 82h

; Create new folder

00000015 8BD3 mov dx, bx

00000017 B439 mov ah,39h

00000019 CD21 int 21h

0000001B 7221 jc erop

; Change directory

0000001D 8BD3 mov dx, bx

0000001F B43B mov ah,3bh

00000021 CD21 int 21h

00000023 7219 jc erop

; Create new file

00000025 BA0000 mov dx,offset file\_name

00000028 B90000 mov cx,0 ; No attributes

0000002B B45B mov ah,5bh ; Create and open

0000002D CD21 int 21h

0000002F 720D jc erop

00000031 8BD8 mov bx,ax ; Store file handle

; Close file

00000033 B43E mov ah,3eh

00000035 CD21 int 21h

00000037 7205 jc erop

00000039 exit:

; Exit all good

00000039 B8004C mov ax,4c00h

0000003C CD21 int 21h

0000003E erop:

; Show error message

0000003E BA0000 mov dx,offset error\_message

00000041 B409 mov ah,9

00000043 CD21 int 21h

00000045 EBF2 jmp exit

00000047 cseg ends

end start

Binary Map:

Segment Pos(file) RVA Size(fil) Size(mem)

---------------------------------------------------------------

<header> 0 0 30 0

sseg 30 0 100 100

dseg 130 100 25 25

cseg 160 130 47 47

---------------------------------------------------------------

1A7 177

Macros:

N a m e Type

@CatStr . . . . . . . . . . . . Func

@Environ . . . . . . . . . . . . Func

@InStr . . . . . . . . . . . . . Func

@SizeStr . . . . . . . . . . . . Func

@SubStr . . . . . . . . . . . . Func

Segments and Groups:

N a m e Size Length Align Combine Class

cseg . . . . . . . . . . . . . . 16 Bit 0047 Para Private ''

dseg . . . . . . . . . . . . . . 16 Bit 0025 Para Private ''

sseg . . . . . . . . . . . . . . 16 Bit 0100 Para Stack ''

Symbols:

N a m e Type Value Attr

PSP . . . . . . . . . . . . . . Word 23h dseg

erop . . . . . . . . . . . . . . L Near 3Eh cseg

error\_message . . . . . . . . . Byte[25] Ah dseg

exit . . . . . . . . . . . . . . L Near 39h cseg

file\_name . . . . . . . . . . . Byte[6] 0h dseg

main . . . . . . . . . . . . . . L Near 2h cseg

root\_directory . . . . . . . . . Byte[4] 6h dseg

start . . . . . . . . . . . . . L Near 0h cseg Public

MK\_DF.ASM: 67 lines, 2 passes, 0 ms, 0 warnings, 0 errors

## 5. Висновок

Отримав практичнi навики використання командної мови та ознайомився з особливостями реалiзацiї операцiйної cистеми MS DOS